Perception of Unstructured Environments for Autonomous Off-Road Vehicles

Zur Erlangung des akademischen Grades einer

DOKTORIN DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Nina Felicitas Heide, M.Sc.

geb. in Bad Urach

Tag der mündl. Prüfung:	06.07.2022
Hauptreferent:	Prof. Dr.-Ing. Michael Heizmann, KIT
Korreferent:	Prof. Dr.-Ing. Hans-Joachim Wünsche, UniBw M

Vorwort

> ***Nicht weil es schwer ist wagen wir es nicht, sondern weil wir es nicht wagen ist es schwer.***

Lucius Annaeus Seneca

> ***Wer fliegen will muss den Mut haben den Boden zu verlassen.***

Walter Ludin

Zuallererst möchte ich meine Dankbarkeit und Liebe für meine Eltern, Gudrun und Horst Heide, ausdrücken. Ihr habt mich stets gefördert, an mich geglaubt und mir alles mitgegeben, was ich für meinen Lebensweg benötigt habe - ich hätte es mir nicht besser wünschen können!

Besonders möchte ich mich auch bei Prof. Dr.-Ing. Michael Heizmann bedanken, welcher mich mit großem Engagement und wertvollen Anregungen durch diese Arbeit geführt hat. Die gemeinsamen Diskussionen während der Entstehung dieser Arbeit haben sehr zu ihrem Wachsen beigetragen. Außerdem möchte ich mich bei Prof. Dr. Hans-Joachim Wünsche für die Bereitschaft zur Übernahme des Korreferats bedanken.

Diese Thesis wurde primär durch die Arbeitsumgebung und meine Kollegen am Fraunhofer IOSB in Karlsruhe ermöglicht, wo ich jeden Tag gefordert und gefördert wurde und die Forschungsarbeiten für diese Thesis mit toller Hardwareausrüstung durchführen konnte. Innerhalb des Fraunhofer IOSB möchte ich mich besonders bei Dr.-Ing. Janko Petereit für die Betreuung dieser Arbeit sowie bei Dr.-Ing. Philipp Woock und Dr.-Ing. Christian Frese für die Diskussion und den Austausch, besonders in der finalen Phase dieser Thesis, bedanken. Ferner möchte ich gerne Alexander Albrecht für die tolle Zusammenarbeit bei gemeinsamen Projekten

und Forschungsfragen sowie allen Kollegen am Fraunhofer IOSB und KIT-IIIT für den inspirierenden Austausch und die vielen fruchtbaren Diskussionen in den letzten Jahren danken.

Besondere Dankbarkeit und Liebe gilt meinem Lebensgefährten Ralf Müller, welcher mich am Boden und in der Luft stets auf wundervolle Weise begleitet und vor allem in der intensiven Phase dieser Thesis mit genau der richtigen Kombination aus Unterstützung und Geduld an meiner Seite war. Abschließend möchte ich Nicole Schriefers und Hptm Mag. (FH) Maximilian Steingassner für einen spannenden und wertvollen Austausch zu den technischen und militärwissenschaftlichen Hintergründen dieser Thesis danken und ein spezielles Dankeschön an Heinz Dietrich richten, mit welchem ich bei schönen Flügen alltägliche Herausforderungen aus der Vogelperspektive betrachten durfte.

Karlsruhe, August 2022 Nina Felicitas Heide

Abstract

Autonomous vehicles require perception capabilities to understand their environment as a necessary prerequisite for controllable and safe interaction. Perception for structured indoor and outdoor environments targets economically lucrative areas such as autonomous passenger transport or industrial robotics, while research on perception for unstructured environments is greatly under-represented in the research field of environment perception. The analyzed unstructured environments pose a particular challenge as the existing, natural and grown geometries mostly do not have a homogeneous structure, and similar textures and difficult-to-separate objects dominate them. This makes capturing these environments and their interpretation difficult, and perception methods must specifically be designed and optimized for this application domain.

This doctoral thesis proposes novel and customized perception methods for unstructured environments and combines them within a holistic, three-level pipeline for autonomous off-road vehicles: low-level, mid-level, and high-level perception. The proposed classic and machine learning (ML) perception methods complement each other. Furthermore, the combination of perception and validation methods for each level facilitates a reliable perception of the possibly unknown environment with loosely coupled and tightly coupled validation methods being combined to ensure a detailed but flexible assessment of the perception methods proposed. All methods were designed as individual modules within the perception and validation pipeline proposed in this thesis, and their flexible combination permits different pipeline designs for a variety of off-road vehicles and use cases according to demand.

Low-level perception contributes a tightly coupled confidence assessment for raw 2D and 3D sensor data to detect sensor failures and ensure sufficient accuracy of the perception sensor data. Furthermore, novel calibration and registration approaches for multi-sensor systems in perception are presented that only use the structure of the surroundings to

register the captured sensor data: a semi-automatic registration approach for multiple 3D Light Detection and Ranging (LiDAR) sensors, and a confidence-based framework combining different registration methods and facilitating the registration of various sensors with differing measurement principles. Hereby, the combination of multiple registration methods validates the registration results in a tightly coupled manner.

Mid-level perception facilitates the 3D reconstruction of unstructured environments with two stereo image disparity estimation methods: a classic, correlation-based method for hyperspectral images, which requires a limited volume of testing and validation data, and a second method that estimates the disparity from grayscale images with convolutional neural networks (CNNs). Novel disparity error metrics and an evaluation toolbox for stereo image 3D reconstruction complement the proposed stereo vision methods and provide loosely coupled validation.

High-level perception focuses on interpreting "single-shot" 3D point clouds for navigability analysis, object detection, and obstacle avoidance. A domain transfer analysis for state-of-the-art semantic 3D segmentation methods provides recommendations for the segmentation performance to be as accurate as possible in new target domains without the generation of new training data. The presented, customized training approach for 3D segmentation methods with CNNs can further reduce the required volume of training data. Pre-modeling and post-modeling explainable artificial intelligence methods provide a loosely coupled validation of the proposed high-level methods with dataset assessment and model-agnostic explanations for CNN predictions.

The decontamination of landfill sites and military logistics constitute the two main use cases in unstructured environments targeted within this thesis. These application scenarios also demonstrate how to bridge the gap between the development of individual methods and their integration in the processing chain for autonomous off-road vehicles with localization, mapping, planning, and control.

Concluding, the proposed perception–validation pipeline provides flexible perception solutions for autonomous off-road vehicles and the accompanying validation ensures accurate and trustworthy perception of unstructured environments.

Zusammenfassung

Autonome Fahrzeuge benötigen die Fähigkeit zur Perzeption als eine notwendige Voraussetzung für eine kontrollierbare und sichere Interaktion, um ihre Umgebung wahrzunehmen und zu verstehen. Perzeption für strukturierte Innen- und Außenumgebungen deckt wirtschaftlich lukrative Bereiche, wie den autonomen Personentransport oder die Industrierobotik ab, während die Perzeption unstrukturierter Umgebungen im Forschungsfeld der Umgebungswahrnehmung stark unterrepräsentiert ist. Die analysierten unstrukturierten Umgebungen stellen eine besondere Herausforderung dar, da die vorhandenen, natürlichen und gewachsenen Geometrien meist keine homogene Struktur aufweisen und ähnliche Texturen sowie schwer zu trennende Objekte dominieren. Dies erschwert die Erfassung dieser Umgebungen und deren Interpretation, sodass Perzeptionsmethoden speziell für diesen Anwendungsbereich konzipiert und optimiert werden müssen.

In dieser Dissertation werden neuartige und optimierte Perzeptionsmethoden für unstrukturierte Umgebungen vorgeschlagen und in einer ganzheitlichen, dreistufigen Pipeline für autonome Geländefahrzeuge kombiniert: Low-Level-, Mid-Level- und High-Level-Perzeption. Die vorgeschlagenen klassischen Methoden und maschinellen Lernmethoden (ML) zur Perzeption bzw. Wahrnehmung ergänzen sich gegenseitig. Darüber hinaus ermöglicht die Kombination von Perzeptions- und Validierungsmethoden für jede Ebene eine zuverlässige Wahrnehmung der möglicherweise unbekannten Umgebung, wobei lose und eng gekoppelte Validierungsmethoden kombiniert werden, um eine ausreichende, aber flexible Bewertung der vorgeschlagenen Perzeptionsmethoden zu gewährleisten. Alle Methoden wurden als einzelne Module innerhalb der in dieser Arbeit vorgeschlagenen Perzeptions- und Validierungspipeline entwickelt, und ihre flexible Kombination ermöglicht verschiedene Pipelinedesigns für eine Vielzahl von Geländefahrzeugen und Anwendungsfällen je nach Bedarf.

Low-Level-Perzeption gewährleistet eine eng gekoppelte Konfidenzbewertung für rohe 2D- und 3D-Sensordaten, um Sensorausfälle zu erkennen und eine ausreichende Genauigkeit der Sensordaten zu gewährleisten. Darüber hinaus werden neuartige Kalibrierungs- und Registrierungsansätze für Multisensorsysteme in der Perzeption vorgestellt, welche lediglich die Struktur der Umgebung nutzen, um die erfassten Sensordaten zu registrieren: ein halbautomatischer Registrierungsansatz zur Registrierung mehrerer 3D Light Detection and Ranging (LiDAR) Sensoren und ein vertrauensbasiertes Framework, welches verschiedene Registrierungsmethoden kombiniert und die Registrierung verschiedener Sensoren mit unterschiedlichen Messprinzipien ermöglicht. Dabei validiert die Kombination mehrerer Registrierungsmethoden die Registrierungsergebnisse in einer eng gekoppelten Weise.

Mid-Level-Perzeption ermöglicht die 3D-Rekonstruktion unstrukturierter Umgebungen mit zwei Verfahren zur Schätzung der Disparität von Stereobildern: ein klassisches, korrelationsbasiertes Verfahren für Hyperspektralbilder, welches eine begrenzte Menge an Test- und Validierungsdaten erfordert, und ein zweites Verfahren, welches die Disparität aus Graustufenbildern mit neuronalen Faltungsnetzen (CNNs) schätzt. Neuartige Disparitätsfehlermetriken und eine Evaluierungs-Toolbox für die 3D-Rekonstruktion von Stereobildern ergänzen die vorgeschlagenen Methoden zur Disparitätsschätzung aus Stereobildern und ermöglichen deren lose gekoppelte Validierung.

High-Level-Perzeption konzentriert sich auf die Interpretation von einzelnen 3D-Punktwolken zur Befahrbarkeitsanalyse, Objekterkennung und Hindernisvermeidung. Eine Domänentransferanalyse für State-of-the-art-Methoden zur semantischen 3D-Segmentierung liefert Empfehlungen für eine möglichst exakte Segmentierung in neuen Zieldomänen ohne eine Generierung neuer Trainingsdaten. Der vorgestellte Trainingsansatz für 3D-Segmentierungsverfahren mit CNNs kann die benötigte Menge an Trainingsdaten weiter reduzieren. Methoden zur Erklärbarkeit künstlicher Intelligenz vor und nach der Modellierung ermöglichen eine lose gekoppelte Validierung der vorgeschlagenen High-Level-Methoden mit Datensatzbewertung und modellunabhängigen Erklärungen für CNN-Vorhersagen.

Altlastensanierung und Militärlogistik sind die beiden Hauptanwendungsfälle in unstrukturierten Umgebungen, welche in dieser Arbeit behandelt werden. Diese Anwendungsszenarien zeigen auch, wie die Lücke zwischen der Entwicklung einzelner Methoden und ihrer Integration in die Verarbeitungskette für autonome Geländefahrzeuge mit Lokalisierung, Kartierung, Planung und Steuerung geschlossen werden kann.

Zusammenfassend lässt sich sagen, dass die vorgeschlagene Pipeline flexible Perzeptionslösungen für autonome Geländefahrzeuge bietet und die begleitende Validierung eine exakte und vertrauenswürdige Perzeption unstrukturierter Umgebungen gewährleistet.

Contents

Nomenclature

General Abbreviations

3PE	Three Pixel Error
A2D2	Audi Autonomous Driving Dataset
AI	Artificial Intelligence
ANN	Artificial Neural Network
BFGS	Broyden–Fletcher–Goldfarb–Shanno optimization algorithm
BIM	Building Information Modeling
CAD	Computer-Aided Design
CBIV	Cross-Based Iterative Voting
CCD	Charge Coupled Devices
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DINO	Differential evolution Initialized Newton-based Optimization
DoB	Degree of Belief
DoG	Difference of Gaussians
DoF	Degrees of Freedom
EDA	European Defense Agency
EDT	Euclidean Distance Transform

ESF	Ensemble of Shape Functions
FAST	Features from Accelerated Segment Test
FGM	Factorized Graph Matching
FLANN	Fast Library for Approximate Nearest Neighbors
FLOPS	Floating Point Operations
FN	False Negative
FoV	Field of View
FP	False Positive
FPFH	Fast Point Feature Histogram
FPGA	Field Programmable Gate Array
GAN	Generative Adversarial Network
GICP	Generalized Iterative Closest Point
GLCM	Gray-Level Co-occurrence Matrix
GPGPU	General Purpose Computation on Graphics Processing Unit
HDR	High Dynamic Ranging
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value color space
ICP	Iterative Closest Point
IFC	Industry Foundation Classes format
IIIT	Institute of Industrial Information Technology
IOSB	Institute of Optronics, System Technologies and Image Exploitation
IoU	Intersection over Union
KIT	Karlsruhe Institute of Technology
kNN	3D kd-tree Nearest Neighbor Search
LiDAR	Light Detection And Ranging

LRC	Left-Right Consistency Check
MCCT	Modified Color Census Transform
MEF	Mertens Exposure Fusion
MI	Mutual Information
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMD	Maximum Mean Discrepancy
MRF	Markov Random Field
MSE	Mean Squared Error
MULE	Multifunctional Utility Logistics Equipment transport
NiN	Network-in-Network
NIR	Near Infrared
NRMSE	Normalized Root Mean Squared Error
PCA	Principle Component Analysis
PFH	Point Feature Histogram
R3PE	Reference-weighted 3PE
RDP	Ramer-Douglas-Peucker
RANSAC	Random Sample Consensus
RGB	Additive color model with Red, Green, and Blue
ROS	Robot Operating System
RMSE	Root Mean Square Error
SAD	Sum of Absolute Differences
SGD	Sum of Gradient Differences
SGBM	Semi-Global Block Matching
SGM	Semi-Global Matching
SIFT	Scale-Invariant Feature Transform

SIMT	Single Instruction and Multiple Thread
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
SVD	Singular Value Decomposition
TAS	Institute for Technology of Autonomous Systems at the Bundeswehr University Munich
THU	Total Horizontal Uncertainty
TN	True Negative
ToF	Time-of-Flight
TP	True Positive
TPU	Total Propagated Uncertainty
TVU	Total Vertical Uncertainty
SURF	Speeded Up Robust Features
UniBw M	"Universität der Bundeswehr München" (Bundeswehr University Munich)
VCCS	Voxel Cloud Connectivity Segmentation
XAI	Explainable Artificial Intelligence

Particular Abbreviations

ACC	Accuracy of input data in *IC-ACC*
CCo	Cost Computation in stereo image disparity estimation
CD	Qualitative Cloud Density criterion in *SET*
CR	Correspondence Randomness in GICP algorithm
cc23	Classic 2D–3D cross-source registration with Contour Cues
cnn23	2D–3D cross-source registration with Convolutional Neural Networks
CoE	Qualitative Consistency on Edges criterion in *SET*

CSGM	Cross-Source Graph Matching
DN21	DarkNet21Seg CNN architecture for semantic 3D segmentation
DN53	DarkNet53Seg CNN architecture for semantic 3D segmentation
DS	Disparity Selection
dsm33	3D–3D cross-source registration with CNN-based Deep Similarity Metric
Geo	Qualitative Geometry criterion in *SET*
GF	Guided Filtering
GOOSE	German Outdoor and Off-road Dataset
graph33	Graph-based, classic 3D–3D cross-source registration
IC	Information Content of input data in *IC-ACC*
IC-ACC	Exploratory dataset analysis method to measure *IC* and *ACC* of input data
ITP	Initial Transformation Preprocessing in enhanced GICP
MDS	Quantitative Mean Distance between Surfaces criterion in *SET*
MF	Median Filter
MoS	Qualitative monochromatic surfaces criterion in *SET*
NNS	Quantitative Nearest Neighbor Search criterion in *SET*
OE	Overexposure
OT	(Sensor) Outage
PD	Prediction Density in stereo image disparity estimation
PI	Precipitation Impairment confidence estimate
PL	Quantitative absolute Path Length criterion for dynamic *SET* evaluation
PSC	Per Sensor Confidence

PPC	Per Point Confidence
RS	Reflecting Surfaces confidence estimate
SA	Source Alignment for domain transfer
SET	Stereo Evaluation Toolbox
SI	Structured Indoor Environments
SO	Structured Outdoor Environments
SOe	Quantitative Surface Orientation criterion in *SET*
SP	Spherical Projection in CNNs for semantic 3D segmentation
SPP	Separate Preprocessing for outlier filtering in enhanced GICP
Squ	SqueezeSeg CNN architecture for semantic 3D segmentation
StS	Shift-to-Source for domain transfer
TC	Temporal Consistency
PUO	Partially Unstructured Outdoor environments
UCSR	Unstructured Cross-Source Registration
UE	Underexposure
UO	Unstructured Outdoor environments
UEM-CNN	Unstructured Environment Matching-CNN
X^{3}Seg	Model-agnostic Explanation of point-wise class predictions in 3D semantic Segmentation

Mathematical and Additional Notations

$\mathbf{1}$	Identity matrix
$\mathbf{a}$	Bold lower-case letters denote vectors.
$\mathbf{A}$	Bold capital letters denote matrices.
$[p_1; p_2; p_3]$	3×1 vector $\mathbf{p}$ in line representation
$[p_1, p_2, p_3]$	1×3 vector $\mathbf{p}$ in line representation

$\mathbf{A}^*$	Transposed matrix $\mathbf{A}$
$\mathbf{p}^*$	Transposed vector for point $\mathbf{p}_i = [x_i; y_i; z_i]^*$
$\#\mathbf{a}$	Number of elements of $\mathbf{a}$
$\sigma(\mathbf{x})$	Standard deviation of a vector $\mathbf{x}$
$\sigma^2(\mathbf{x})$	Variance of a vector $\mathbf{x}$
$\mathcal{N}_{0,1}$	Standard normal distribution with $(\mu = 0, \sigma^2 = 1)$
$\|\mathbf{x}\|_1$	L_1, Manhattan norm of $\mathbf{x}$
$\|\mathbf{x}\|_2$	L_2 norm of $\mathbf{x}$
$\overline{x}$	Empirical mean of numerical variable x
$\overline{\mathbf{x}}$	Empirical mean of vector $\mathbf{x}$
$\mathbf{p}^{2D}$	2D projection or representation of 3D point $\mathbf{p}$ for clarity (e.g. *cc23*)
$\mathbf{p}^{3D}$	3D representation of 3D point $\mathbf{p}$ for clarity (e.g. *cc23*)
$[i, j]$	Pixel in line i (x axis) and column j (y axis) of an image
$\hat{\sum}$	Real, symmetric empirical covariance matrix for SVD
$\mathfrak{so}(3)$	Lie algebra corresponding to $SO(3)$
$\bullet$	Dot product for vectors
$\otimes$	Kronecker product for matrices

Lower-case Letters

b	Offset for range-limit error weighting function to assess disparity estimation results
$\mathcal{B}$	Vehicle body coordinate system for sensor calibration
c^{2D}	$\overline{PSC}$ confidence for 2D images
c_i^{2D}	PSC confidence measure i, $i \in \{\text{OE}, \text{UE}, \text{TC}, \text{H}, \text{GLCM}, \text{T}, \text{S}\}$, for 2D images
c^{3D}	$\overline{PSC}$ confidence measure for 3D point clouds

c_i^{3D}	*PSC*/$\overline{PPC}$ confidence measure i, $i \in \{R_v, R_h, A, S, RS, PI\}$ for 3D LiDAR point clouds
$c_{i,j}^{3D}$	*PPC* confidence measure i, $i \in$ for a point j
$c_{\mathcal{L},j}^{3D}$	*PPC* confidence result for a point j in a 3D LiDAR point cloud
$c_{\mathcal{S},j}^{3D}$	*PPC* confidence result for a point j in a 3D stereo or RGB-D point cloud
c_{OT}^{3D}	*PSC* confidence measure for sensor outage
c_{TC}^{3D}	*PSC* confidence measure for temporal consistency
$\mathcal{C}$	Camera coordinate system
c_X	3D covariance similarity measure in X^{3}Seg
$c_{X,2}$	2D covariance similarity measure in X^{3}Seg
d	Disparity
$d_x(\max)$	Maximum disparity $d_x(\max) = \max_{j=1,k=1}^{M,N}(\mathbf{D}_x[j,k])$, $x \in 1,2$
d_a, d_b	First order derivatives of path elements for waypoint optimization in cost valley approach
$\mathbf{d}$	Vector of singular values in SVD
e_X	Euclidean fitness score similarity measure in X^{3}Seg
e_{fs}	Euclidean Fitness Score for ICP and GICP registration
f	Focal length
f_c	Current sensor publishing frequency
f_d	Desired sensor publishing frequency
$\mathbf{i}$	Vector of concatenated intensity values of a pixel patch in *IC-ACC*
$\mathcal{I}$	Image coordinate system
k	RANSAC-estimated surfaces for static SET evaluation criterion *MDS*

$\mathcal{K}$	Kernel function of kernel Hilbert space for fast ProtoDash algorithm
l_{vox}	Voxel size for VCCS supervoxels in *graph33*
$\mathbf{l}_{\text{vox}}$	Voxel size for input data voxelization in *dsm33*
m_{MED}	Mean Euclidean distance for CNN loss function and testing error in *dsm33*
$\mathbf{m}_{G,m}$	Geometrical center of mass in *cc23*
$\mathbf{m}_{Y,m}$	Center of mass of the luminance values in *cc23*
$\mathbf{n}_i^0$	Normal vector for point i determined with SVD normal estimation in *SET*
n_G	Acceleration factor for calculation on GPGPU in relation to CPU
$\mathbf{n}_k$	Normal vector for RANSAC-estimated surface k
$\mathbf{n}_{i\mathcal{S}}$	SVD-estimated normal vector for an evaluated stereo point in $\mathcal{S}$ to measure the *SOe* score in *SET*
o	Optical center in pinhole camera model $[o_x, o_y]$
$p(i)$	Likelihood of a symbol $i \in I$ to be present inside a message for the calculation of H
p_1, p_2	Tangential distortion in 1D distortion matrix of OpenCV plumb bob model
$\mathbf{p}_s$	3D point of source point cloud in registration
$\mathbf{p}_t$	3D point of target point cloud in registration
$p_{X,1}, p_{X,2}$	First and second principal components similarity measure in X^{3}Seg
$\mathbf{q}$	3D rotation in Quaternion representation
r	Range, radius: Euclidean distance from sensor origin to measured, real-world point
r_C	Euclidean distance of corresponding source and target points in GICP

r_M	Mahalanobis distance in GICP algorithm
$\mathbf{r}$	3D rotation in Euler angle representation
$r_{i,j}$	Distance of two nodes W_i and W_j in supervoxel adjacency graph in *graph33*
r_W	L_2 distance between two nodes in supervoxel adjacency graph in *graph33*
r_X	Relative spatial extent similarity measure in X^{3}Seg
s	Surface variation
s_d	Similarity measure for disparity value d in *UEM-CNN*
$\overline{s}$	Mean surface variation
$\mathfrak{s}$	Sensor coordinate system for sensor calibration
s_X	Surface variation similarity measure in X^{3}Seg
t	Time
$\mathbf{t}$	3D translation vector
$t_{C,\max(d)}$	Calculation time on CPU for maximum disparity $\max(d)$
$t_{G,\max(d)}$	Calculation time on GPGPU for maximum disparity $\max(d)$
t_{in}	2D intensity threshold in *cc23*
t_{gr}	3D gradient threshold in *cc23*
u	Row pixels of range image from spherical projection in semantic segmentation
$\mathbf{u}_L$	Feature vector for left image patch in Siamese network layers of *UEM-CNN*
v	Column pixels of range image from spherical projection in semantic segmentation
v_X	Singular values similarity measure in X^{3}Seg
$\mathbf{v}$	Projection vector to project 3D points onto a 2D pixel grid or reverse in *cc23* and 2D–3D fusion

$\mathbf{v}_R$	Feature vector for right image patch in Siamese network layers of *UEM-CNN*
$\mathbf{t}$	3D translation
w_f	Free width of the cost valley around the optimized track
$w_{\text{gr},m}$	Score for 3D gradient features in *cc23*
w_i	Weights for qualitative and quantitative criteria i in *SET*, for registration results in *UCSR*, and for MEF of multi-spectral images
$w_{\text{in},m}$	Score for 2D intensity features in *cc23*
w_m	Maximum width of the cost valley around the optimized track
w_{PD}	Prototype weights in ProtoDash
$y_{O,c}$	Binary indicator if a class label i is correct or false for the respective observation O in semantic segmentation training with cross-entropy loss
z	Depth, measured via triangulation by passive sensors

Capital Letters

$\mathbf{A}$	Merged affinity matrix of all four descriptors in *graph33*
$\mathbf{A}_E$	Affinity matrix of ESF descriptor in *graph33*
$\mathbf{A}_G$	Global affinity matrix in FGM in *graph33*
$\mathbf{A}_N$	Affinity matrix of voxel centroid normal orientation descriptor in *graph33*
$\mathbf{A}_Q$	Edge affinity matrix in *graph33*
$\mathbf{A}_R$	Affinity matrix of distance to origin descriptor in *graph33*
$\mathbf{A}_W$	Node affinity matrix in *graph33*
$\mathbf{A}_Z$	Affinity matrix of angle to z descriptor in *graph33*
$\mathbf{A}_{\theta_W(\theta,r_{i,j})}$	Edge affinity matrix in *graph33*

B	Baseline of a stereo system
$C(p,d)$	Matching cost of disparity value d for pixel p in *UEM-CNN*
$\mathbf{C}$	Covariance matrix for surface estimation with SVD
$\mathbf{C}_i^{\mathbf{P}_s}$	Covariance matrix of source point cloud $\mathbf{P}_s$
$\mathbf{C}_i^{\mathbf{P}_t}$	Covariance matrix of target point cloud $\mathbf{P}_t$
$\mathbf{C}_M$	Covariance matrix covariance of the Mahalanobis distances r_M in GICP
C_{RDP}	Ordered set of points that represents a curve in Ramer-Douglas-Peucker line simplification
D	Normalized distance measure for VCCS supervoxels
$\overline{D}$	Quantitative average distance from path criterion for dynamic *SET* evaluation
D_c	Euclidean distance in CIELAB color space for over-segmentation with VCCS supervoxels
$\mathbf{D}_{\text{diff}}$	Difference map for two disparity images D_1 and D_2
D_{FPFH}	Distance between the FPFH features for over-segmentation with VCCS supervoxels
D_{max}	Quantitative maximum distance from path criterion for dynamic *SET* evaluation
D_{min}	Quantitative minimum distance from path criterion for dynamic *SET* evaluation
D_s	Spatial distance for over-segmentation with VCCS supervoxels
D_T	Data volume inside each tensor in semantic segmentation training
E	Well-exposedness quality measure in MEF
$\mathbf{E}_p$	Population energy in differential evolution in *dsm33*
F	Frobenius norm

G	Gaussian 1D kernel for fast ProtoDash algorithm in selective X^{3}Seg
G_c	Correlation measure for GLCM analysis
G_e	Energy measure for GLCM analysis
G_h	Homogeneity measure for GLCM analysis
$\mathbf{G}$	Node-edge incidence matrix in *graph33*
$\mathcal{G}$	Graph with n nodes and m directed edges in *graph33*
H	Shannon Entropy
I	Intensity
$\mathbf{I}_k$	k-th sequence image in MEF
$\overline{\text{IoU}}_T$	IoU during training
$\overline{\text{IoU}}_V$	IoU during validation
$I(X;Y)$	Mutual information of X and Y
J_{gm}	Global objective function in FGM in *graph33*
J_{vex}	Convex relaxation in FGM in *graph33*
J_α	Local objective function in FGM in *graph33*
J_{cav}	Concave relaxation in FGM in *graph33*
J_{smooth}	Regulation term in FGM in *graph33*
$\mathbf{K}$	Camera calibration matrix
$\mathcal{K}$	Reproducing kernel Hilbert space that belongs to the kernel function $\mathcal{k}$
L	Training loss of neural networks
L_{CE}	Cross-entropy loss for semantic 3D segmentation ANNs
L_f^{3D}	Line from origin to $\mathbf{p}_f$ in 3D space
L_{ef}^{3D}	Line from $\mathbf{p}_e$ to $\mathbf{p}_f$ in 3D space
$\mathbf{L}_{\text{MEF}}$	Laplacian pyramid in MEF

$L_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$	L_2 norm of ground truth decalibration applied for *dsm33* training
$\hat{L}_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$	L_2 norm of CNN-estimated decalibration in *dsm33*
$\mathcal{L}$	Short notation of a LiDAR point cloud
$\mathcal{M}$	Manhattan metric
$\mathbf{M}$	N-dimensional manifold embedded in $\mathbf{R}^n$ with $n \leq N$
M_{DE}	Predefined threshold for convergence in truncated differential evolution in *dsm33*
N_S	Number of scenes in semantic segmentation of 3D point clouds
$\mathcal{N}(i)$	Neighborhood of a pixel i
O	Contrast quality measure in MEF
$P_{i,k}^{j}$	Penalty score for criterion i in hyperspectral disparity estimation with configuration k on image j
$\mathbf{P}$	Projection matrix in stereo camera calibration
$\mathbf{P}_s$	Source point cloud in 3D–3D registration
$\mathbf{P}_t$	Target point cloud in 3D–3D registration
$\mathcal{P}$	Gaussian pyramid in MEF
P_{RDP}	Polygon in Ramer-Douglas-Peucker line simplification
$\boldsymbol{PL}_{\text{GT}}$	Ground truth for PL criterion in dynamic *SET* evaluation
$\boldsymbol{PL}_{\text{SLAM}}$	Path length in visual SLAM for PL criterion in dynamic *SET* evaluation
$P(O, i)$	Predicted probability that observation O is of class i in semantic segmentation
$\mathbf{Q}$	Edges of the adjacency graph in *graph33*
$\mathbf{Q}_p$	Perspective transformation matrix in stereo camera calibration
$\mathbf{R}$	3×3 rotation matrix

$\mathbf{R}_{v_i}$	Rotation matrix for point $\mathbf{p}_i$ with normal vector $\boldsymbol{v}_i$ in SVD
R_{MEF}	Final fused 2D image in MEF
$R_{m,n}$	Cross-correlation of the intensities of two image patches n and m in *IC-ACC*
R_{seed}	Seed resolution of VCCS supervoxels
$\mathcal{S}$	Short notation of stereo point cloud for registration and SET
S	Saturation quality measure in MEF
S_{dyn}	Dynamic *SET* score
S_{qual}	Qualitative, static *SET* score
S_{quan}	Quantitative, static *SET* score
T_L	Lower threshold for range-limit error weighting function to assess disparity estimation results
T_U	Upper threshold for range-limit error weighting function to assess disparity estimation results
$\mathbf{T}$	Transformation matrix
$\hat{\mathbf{T}}$	Estimated transformation as registration result
$\mathbf{T}_{\text{d}}$	Artificially decalibrated transformation
$\mathbf{T}_{\text{GT}}$	Ground truth transformation
$\mathbf{T}_{\text{ref}}$	Reference transformation, registration ground truth
U	Width of range image in spherical projection for 3D segmentation
$\mathbf{U}$	Matrix of left singular vectors in SVD
V	Height of range image in spherical projection for 3D segmentation
$\mathbf{V}$	Matrix of right singular vectors in SVD
$\mathbf{W}$	Nodes of the adjacency graph in *graph33*
$\mathcal{W}$	Weight map to calculate disparity error metrics related to camera distance

$\hat{\mathbf{W}}_{[i,j],k}$	Normalized weight map in MEF for image k with pixels $[i, j]$
$\mathcal{X}$	Function space that contains $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$
$\mathcal{X}^{\infty}$	Large dataset containing all samples in ProtoDash (database of encompassing X^{3}Seg)
$\mathcal{X}^{\in}$	Small subset with representative samples in ProtoDash (database of selective X^{3}Seg)
Y	Luminance
$\mathbf{Y}$	Optimization representation of quadratic assignment problem in graph matching

Greek Letters

α_r	Metric 3D measurement accuracy of LiDAR sensors
α_v	Metric, vertical resolution of LiDAR sensors
$\Delta(d)$	Difference in first order derivative of two path elements in cost valley approach for constrained planning
Δ_{TV}	Ratio between $\overline{\text{IoU}}_V$ and $\overline{\text{IoU}}_T$
$\Delta(\overline{\text{IoU}}_V)$	Alternation of $\overline{\text{IoU}}_V$ prior to and after current iterative training step
ε_C	Covariance weighting parameter for surface estimation in GICP algorithm
ε_d	Disparity estimation error (in pixels)
ε_{EF}	Euclidean fitness epsilon in GICP algorithm
ε_R	Maximum difference of two consecutive rotations for convergence of BFGS in GICP
ε_T	Maximum difference of two consecutive translation for convergence of BFGS in GICP
ε_z	Depth estimation error (in meters)
ε_{RDP}	Constant threshold parameter, respectively constant size of the path hull, in Ramer-Douglas-Peucker line simplification

ζ	Normalization constant for D_c in over-segmentation with VCCS supervoxels in *graph33*
θ	Vector of Euler angles between two graph nodes in *graph33*
$\theta_{\mathrm{W}}(\theta, r_{i,j})$	Descriptor of distance and angle between two graph nodes W_i and W_j with distance $r_{i,j}$ in *graph33*
κ	Over-segmentation weight for spatial distribution in VCCS supervoxels in *graph33*
λ_c	Over-segmentation weight for color information in VCCS supervoxels in *graph33*
$\lambda_{\mathbf{a}}$	Corresponding eigenvalue of covariance matrix for eigenvector $\mathbf{a}$ in PCA
λ_{LS}	Regularization parameter for weighted least squares filtering during SGBM post-processing
μ	Empirical mean/Expectation value for $\mathcal{N}_{0,1}$ defining data filtering thresholds
$\mu(\ln F)$	Empirical mean of registration result in terms of F norm for clarity
$\mu(L_2)$	Empirical mean of registration result in terms of L_2 norm for clarity
ν	Over-segmentation weight for normal direction in VCCS supervoxels in *graph33*
$\boldsymbol{\nu}_i$	Surface normal for point $\mathbf{p}_i$ in SVD
ξ_i	Relative point density after binning to estimate the *IC* of point clouds
σ	Standard deviation
σ^2	Variance
σ_G	Width of Gaussian 1D kernel G
σ_{LS}	`SigmaColor` parameter for SGBM post-processing defining the filtering sensitivity on source image edges

τ	Sensor FoV
ϕ	Homogenization coordinate in *graph33*
χ	Power factor in range-limit error weighting function to assess disparity estimation results
ψ	Homogenization coordinate in *graph33*
ω_E	Weighting exponent for well-exposedness quality measure in MEF
ω_O	Weighting exponent for contrast quality measure in MEF
ω_S	Weighting exponent for saturation quality measure in MEF

1 Introduction

Robotic systems such as autonomous off-road vehicles require perception capabilities to "see" and understand their environment. To this end, perception senses the environment and builds a reliable, detailed representation as a key requirement for controllable and safe interaction of autonomous off-road vehicles with the physical world. Off-road vehicles belong to the class of mobile robotic systems and some off-road vehicles can also manipulate their environment, such as excavators.

Perception is derived from the Latin *perceptio*, which means gathering or receiving [238], and describes the capture and understanding of the environment by organizing, identifying, and interpreting the acquired sensory information. Heizmann et al. [115] state that perception is one of the greatest challenges for an autonomous operation of mobile robots in unstructured environments. To this end, this doctoral thesis proposes novel and customized, classic and machine learning (ML) perception methods for off-road vehicles in unstructured environments, and combines them within a three-level perception pipeline composed of low-level, mid-level, and high-level perception. Accompanying validation methods for each level facilitate an in-depth assessment of the perception methods and ensure an accurate and valid perception of unstructured environments.

1.1 Scope and Objectives

The perception solutions proposed in this thesis primarily target the perception of unstructured environments in cross-country, off-road scenarios and mostly away from public roads. Target applications are autonomous off-road vehicles: heavy construction machinery for the decontamination of hostile environments, search and rescue robotics, as well as agricultural systems and unmanned ground systems in defense. Typical use

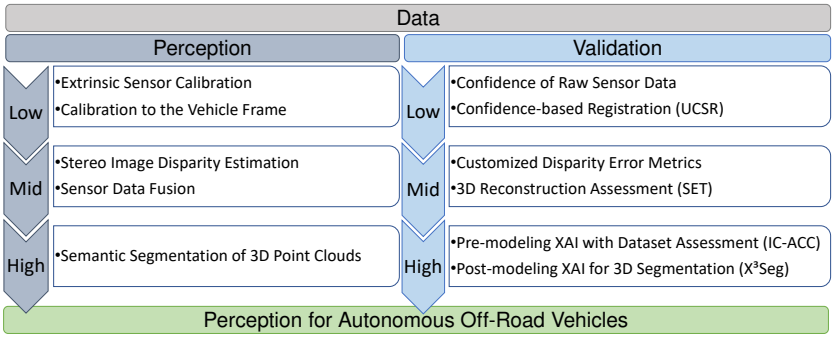


Figure 1.1 Low-level, mid-level, and high-level perception with corresponding validation methods for off-road vehicles. Tight coupling of perception–validation is integrated for confidence of raw sensor data, *UCSR*, and *IC-ACC*. Disparity error metrics, 3D reconstruction assessment, and X^3Seg are loosely coupled with the analyzed perception methods.

cases are the remediation of landfills [216], where a high concentration of unknown and possibly harmful substances occurs, and autonomous off-road transport. In both cases the use of autonomous platforms can lead to a notable reduction of potential risks for humans.

The targeted unstructured environments are primarily dominated by similar textures and characterized by the absence (and non-observance) of controlled, clearly separable, and recognizable topological structures. They often consist of natural and grown structures, such as trees, bushes, and rocks, along with unknown structures encountered in decontamination or defense scenarios. Cooperative and advanced driving behaviors are unnecessary and structures with known geometry, e.g., parking lots in urban environments, are not relevant for autonomous off-road navigation [156]. The distinction between passable and non-passable terrain, obstacle avoidance, and reaction to unknown scenes have priority instead.

The proposed methods are structured according to the low-level, mid-level, and high-level structure proposed in Khan et al. [148] and represent the natural information flow in perception, as illustrated in Figure 1.1 and further discussed in Section 2.2. Low-level perception comprises the confidence assessment of raw sensor data and the registration of sensor data from multi-sensor systems for calibration and registration

purposes. According to Khan et al. [148], mid-level perception generates 3D information from 2D images and comprises stereo image disparity estimation as well as sensor data fusion. Concluding, the proposed low-level and mid-level methods process 2D image data from passive camera systems as well as 3D point cloud data from Light Detection And Ranging (LiDAR) sensors and stereo image disparity estimation. Mapping, planning, and control for off-road vehicles with manipulation capabilities such as autonomous excavators require a geometric 3D reconstruction of the environment to facilitate autonomous exploration and potential manipulation tasks. Hence, the presented high-level perception methods interpret the perceived 3D data as 3D point clouds of single scenes for object detection and obstacle avoidance. These 3D point clouds can be direct 3D sensor outputs or 3D processing results from low- and mid-level perception. Furthermore, the interpretation and understanding as well as the accuracy and trustworthiness assessment of perception results forms a highly relevant part of perception for autonomous systems. This thesis extends the concept of Khan et al. [148] for high-level perception and includes the understanding and explanation of the perception results in the sense of explainable artificial intelligence (XAI) which tries to assign human-understandable explications for neural network's decisions.

Customized perception and validation solutions for each level facilitate a trustworthy perception of the – possibly unknown – environment. The proposed low-level, mid-level, and high-level methods are designed as individual modules within the perception and validation pipeline. Their flexible combination allows different pipeline designs for a variety of off-road vehicles and use cases according to demand while providing the basis for the subsequent navigability analysis, localization, mapping, planning and control for autonomous navigation or manipulation of the environment. Section 1.1.3 presents two exemplary pipeline designs for off-road vehicles.

In the following, the term multi-sensor system should refer to a visual/optical sensor system for 2D and 3D perception. The proposed methods are demonstrated on multi-sensor systems consisting of LiDAR sensors and camera systems, as they are commonly encountered on off-road vehicles for unstructured environments. This demonstrates the applicability of the proposed methods for the targeted use cases. It also

points out that the proposed pipeline concept can overcome the gap between developing isolated methods and their integration into a processing chain required for autonomous off-road vehicles.

The integration of other types of perception sensors such as time-of-flight (ToF) cameras or radar sensors is possible due to the flexible and generic character of the proposed methods. The perception of structured environments and potential subsequent processing steps such as mapping, localization, and planning are partly covered for comparison purposes. This highlights the generalization potential of the proposed methods to other environments and considers the compatibility to these subsequent steps and to concrete, practical applications that exceed the proof-of-concept demonstrations in this thesis.

1.1.1 Perception and Validation

For safe and controllable interaction with the environment, perception for autonomous off-road vehicles does not only require perception methods but also complementing validation methods. To this end, this thesis combines loosely and tightly coupled validation methods for the proposed perception solutions and the utilized data therein.

This thesis combines classic and ML methods with accompanying classic validation methods for each level facilitating a thorough and holistic examination of the proposed perception methods for unstructured environments. Hereby, both classic perception and validation methods contribute to the validation and understanding of ML perception results and represent a first step towards developing trustworthy artificial intelligence (AI) systems meeting the ethics guidelines specified by the high-level expert group on AI (AI HLEG)[1]. Furthermore, the proposed accompanying validation methods address the decision explanation for

[1] Ethics Guidelines for Trustworthy AI: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, access on 24.01.2022.

artificial systems[2] to meet legal provisions and laws, such as in the General Data Protection Regulation of the European Union[3].

The present thesis combines the proposed perception and validation methods into a perception pipeline for off-road vehicles in unstructured environments. Nevertheless, this thesis cannot address all facets of perception required in any particular cases. The primary focus lies on the contribution of perception and accompanying validation methods for the discussed low-, mid-, and high-level perception steps. The perceived sensor data is interpreted as static, "single-shot" scenes captured at the same time and with a sufficiently accurate sensor synchronization unless described otherwise. Dynamic objects, such as cars in urban traffic or humans, are not considered in particular over time as these do hardly occur in the unstructured and hazardous environments analyzed.

1.1.2 Classic Methods and Machine Learning

The definition of AI is complex and often unclear due to a high number of definitions that partially contradict one another [61, 63, 153]. Hence, a clear separation into AI and non-AI methods is hardly possible. This thesis separates into classic methods, with a determination of model parameters by a human expert, and ML methods, which are subject to data-driven modeling. Both classic and ML methods are interpreted as part of AI, and classic perception and validation methods and ML methods form a more complex artificial system for perception within this thesis, as detailed in Section 2.1.

Classic methods are model-based. Their explicit modeling in development and optimization requires expert knowledge, and implies a top-down specification process on the basis of the determined requirements. Furthermore, classic methods provide deterministic, inherently explainable, transparent, and trustworthy decision-making with logical reasoning. ML methods are data-driven, and their modeling process is driven by the features of the examined data in a bottom-up manner without explicit

[2] The Alan Turing Institute: Impact story: A right to explanation, https://www.turing.ac.uk/research/impact-stories/a-right-to-explanation, access on 12.01.2021.

[3] European Parliament and the Council of the European Union: General Data Protection Regulation, http://data.europa.eu/eli/reg/2016/679/oj, access on 24.01.2022.

Figure 1.2 Off-road vehicles operate in unstructured environments primarily dominated by similar textures, naturally occurring structures, and difficult-to-separate objects. The primary applications in the present thesis target the remediation of landfill sites and transport applications in defense.

programming. This thesis focuses on combining classic and ML methods to provide a broad and comprehensive perception for unstructured environments.

1.1.3 Application Environments and Use Cases

The decontamination of hazardous environments [216] and autonomous off-road navigation for defense applications have been selected as the two application scenarios for off-road vehicles in this thesis.

In general, application environments for autonomous vehicles can be divided into structured and unstructured environments [50, 156, 270]. Perception for structured indoor and outdoor environments targets highly interesting areas from an economic perspective: autonomous passenger transport, logistics, or assistance and rehabilitation robotics.

However, research on perception for unstructured environments is greatly under-represented in the research field of environment percep-

tion. The analyzed unstructured environments pose a particular challenge because the existing, naturally grown geometries mostly do not have a homogeneous structure, making capturing these environments and their interpretation difficult. Therefore, perception methods must specifically be designed and optimized for this application domain. Figure 1.2 shows exemplary scenes in unstructured environments.

The proof-of-concept demonstrations in decontamination were conducted in the competence center "ROBDEKON"[4]. ROBDEKON is dedicated to the research on robotic systems for the decontamination of hazardous environments and funded by the Federal Ministry of Education and Research within the scope of the German Federal Government's "Research for Civil Security" program. The competence center is coordinated by the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)[5].

The defense use cases presented were realized in close cooperation with the Institute for Autonomous Systems Technology (TAS) of the Bundeswehr University Munich[6]. The financial support from BAAINBw U6.2 and the "Wehrtechnische Dienststelle" 41 (WTD) of the Bundeswehr for the presented defense applications is gratefully acknowledged.

The proposed methods and approaches are demonstrated on the different technology demonstrators for decontamination and defense applications IOSB.BoB, IOSB.amp Q1, "Technologieträger Unbemanntes Landfahrzeug" (TULF), and IOSB.Alice depicted in Figure 1.3. Technical details for the technology demonstrators are shown in Section 7.1, and exemplary perception and validation pipelines for the regarded technology demonstrators are described in Section 7.2.

[4] Competence Center ROBDEKON: https://robdekon.de/, access on 17.01.2022.
[5] https://www.iosb.fraunhofer.de/, access on 17.01.2022.
[6] https://www.unibw.de/tas-en/main, access on 23.12.2021.

(a) IOSB.BoB.

(b) IOSB.BoB, IOSB.amp Q1 and Q2.

(c) TULF.

(d) IOSB.Alice.

Figure 1.3 Off-road vehicles for unstructured environments that are all equipped with multi-sensor systems for perception and localization. Section 7.1 describes the algorithmic basis for perception, localization, mapping, planning, and control. Images (a), (b), (d) © Fraunhofer IOSB, (c) courtesy of WTD41, Bundeswehr.

1.2 Scientific Contributions

This thesis' main scope is placed on perception methods for unstructured environments with loosely and tightly coupled validation methods combined in a perception pipeline for autonomous off-road vehicles. The main contributions of this thesis are:

- Tightly coupled confidence assessment for raw 2D and 3D sensor data (Section 4.1) and confidence-based data fusion for cross-source 3D point clouds generated from differing measurement principles (Section 5.3).

- A semi-automatic registration approach to register multiple 3D LiDAR sensors [323] (Section 4.2).
- A confidence-based registration framework [329] that combines multiple 2D–3D and 3D–3D registration methods for cross-source sensor data to extrinsically register multi-sensor systems with tightly coupled validation using only the structure of the surroundings (*UCSR*, Section 4.3).
- Disparity estimation from stereo camera images for unstructured environments with a classic method for hyperspectral images [324] and the *UEM-CNN* architecture [327] (Section 5.1).
- The *SET* evaluation toolbox for 3D reconstruction results from stereo image disparity estimation [325] (Section 5.2.2).
- A domain transfer analysis for state-of-the-art methods in the semantic segmentation of 3D point clouds with recommendations for enhanced domain transfer performance and a customized training approach to reduce the required volume of training data (Section 6.1).
- A first step towards validating and understanding high-level perception results by pre-modeling XAI with dataset assessment and recommendations for the generation of optimized training data reducing the data volume required to train neural networks [326] (*IC-ACC*, Section 6.2.1).
- The X^3Seg approach to facilitate post-modeling, model-agnostic XAI for the semantic 3D segmentation in unstructured environments [330] (see Section 6.2.2).
- A novel planning constraint optimizing the driving performance of autonomous off-road vehicles in unstructured environments [328] (Section 7.6).

1.3 Thesis Structure

The thesis consists of eight consecutive chapters. Chapter 2 provides an overview of relevant state-of-the-art perception and validation methods, and Chapter 3 elaborates theoretical foundations for the presented contributions to low-level, mid-level, and high-level perception. Chapter 4 and 5 discuss the proposed contributions to low-level and mid-level per-

ception, and the high-level perception methods proposed are described in Chapter 6. Chapter 7 bridges the gap between the development of individual methods, their integration in the required processing chain for autonomous off-road vehicles for concrete use cases, and the subsequent utilization of their results, e.g., in mapping and planning. The experimental results and proof-of-concept demonstrations of all proposed perception and validation methods follow the methodical discussion of each method. A summary of the proposed methods and an outlook on future work in Chapter 8 conclude the present thesis.

2 State of the Art

The state of the art presented here does not claim completeness. Its purpose is to provide an overview of state-of-the-art perception and validation methods and primarily to focus on research that is related to the methods proposed in this thesis.

2.1 Artificial Intelligence and Machine Learning

ML comprises a wide field of methods from early methods, such as support vector machines, to artificial neural networks (ANNs). Support vector machines are subject to a rather discriminative modeling but with a determination of model parameters on the basis of the selected data. Current ML research concentrates on the subdomain of ANNs, and perception with ML methods typically requires deep ANNs with a higher number of layers due to the complexity of the perceived sensor data.

ANNs are derived from biological neural networks and were already subject to research in the 1960s and earlier [132, 211, 273, 274]. The first ANN was proposed in 1951 by Minsky and Edmonds [197] and their success and popularity notably increased when the computation power reached a sufficient level to analyze deep ANN architectures with a high number of layers. Therefore, ANNs have revolutionized perception starting from the ImageNet competition [233] in 2012. Neural networks, also denoted nature analogous methods in Klüver and Klüver [153], take a unique position in image processing due to their huge success in recent years. They exhibit tremendous potential and highly contributed to increased accuracy and speed in 2D and 3D image exploitation. The deep learning review of Le Cun et al. [162] provides an overview on the state of the art until its publication in 2015 and illustrates research areas that especially benefit from ML with neural networks, such as image

processing, speech recognition, and object detection. Le Cun et al. [162] furthermore summarize the benefits of ANNs in image processing that contribute to their recent success: "local connections, shared weights, pooling, and the use of many layers" [162, p. 439].

Goodfellow et al. [92] state that ML algorithms can be classified according to their problem statement: classification as a discrete problem predicts one class for the input data, e.g., class labels are assigned to all 3D points in the semantic segmentation of point clouds (see Section 2.5.1), while regression aims to estimate one or more numerical values on the basis of the input data, such as the registration of two 3D points clouds (see Section 2.3.3).

LeCun et al. [162] state that perception mostly relies on feed-forward networks with multi-layer perceptrons composed of multiple processing layers. Furthermore, the authors [162] describe that ANNs in 2D image and 3D point cloud processing typically have input layers with a large receptive field. Therein, convolutional layers combine local neighborhoods and contain a set of filters (kernels) convoluted with the receptive field of the input layer, as further discussed in [148]. They reduce the size of the receptive field for the next layer for kernels of at least size 2×2, while upsampling convolutions increase the receptive field size of the next layer with dilated convolutions [148]. Activation function layers and fully connected layers facilitate the mapping of non-linear relations, and pooling layers provide a downsampling, e.g., with maximum pooling that only transfers the maximum of the local neighborhood to the next layer. Batch normalization layers, often combined with dropout, reduce the internal covariate shift of a network and provide regularization [130]. This helps to prevent over-fitting and benefits the network's generalization ability.

ANNs form an important part of ML in perception research and learn from the externally supplied or self-generated data as stated by the European Defense Agency (EDA) that defines AI as "…the capability provided by algorithms of selecting, optimal or sub-optimal choices from a wide possibility space to achieve specific goals by applying different strategies including adaptivity to the surrounding dynamical conditions and learning from own experience, externally supplied or self-generated data" [61]. As previously stated, a clear separation into AI and non-AI methods is hardly possible, and this thesis separates classic and ML

methods. Following the definition of the EDA, this thesis classifies ML methods as primarily data-driven. Fully automatic design processes for ML architectures that are subject to recent research in AutoML [171, 190] even expand this and deduce the model architecture from the data [171]. However, automated network design is rather detrimental for perception methods aiming at transparency and validation and is hence not discussed further within this thesis.

The analogies between recognition and intuition in psychology highlight the importance of examining and combining classic and ML methods in perception. Recognition in psychology is similar to the operation of classic methods reaching decisions in a calculating and logical way, while intuitive decision making resembles the black box operations inside ANNs with fast, rather stereotypic and learned behaviors. In the area of human thinking, the psychologist Kahneman [144] introduces a metaphor of two systems closely related to ANNs and classic methods: system one is "fast, automatic, frequent, emotional, stereotypic, unconscious" [63] and hence similar to decision making in ANNs, while system two is "slow, effortful, infrequent, logical, calculating, conscious" [63], alike classic methods. Nobel Prize winner Herbert Simon furthermore states that "…intuition is nothing more and nothing less than recognition" [258], while Falchi [63] elaborates that neither humans nor intelligent AI systems should solely rely on intuition for decision-making [63, 145]. From this, Falchi [63] derives the two most important conditions in ML decision-making: a sufficiently familiar environment to facilitate its predictability, and the availability of sufficient and valid training data. Falchi [63] hence recommends the use of ML methods as a part of more complex artificial systems involving classic, non-intuitive processes in addition to intuitive ML processes relying on the recognition of learned patterns and correlations. Hodler et al. [121] furthermore propose a subdivision into narrow and general AI that interprets classic and ML approaches as AI in a broad sense: narrow AI is centered on one task, such as image classification for 2D images, while general AI denotes multiple, more general abilities, such as planning, object detection and obstacle avoidance, learning, or problem-solving.

Naujoks et al. [204] demonstrate that the combination of classic and ML methods can improve the performance of perception methods and help

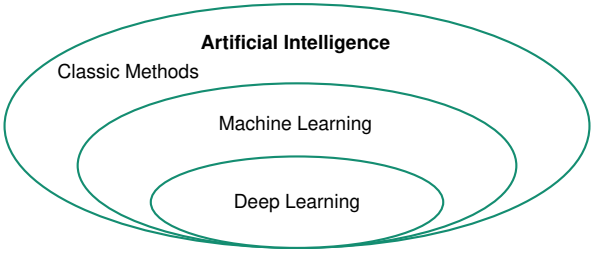


Figure 2.1 Overview of the AI definition for this thesis.

to overcome the need for large datasets, as discussed in Section 2.5.1. The classic methods in this thesis process environment data in a deterministic and predictable manner on the basis of a mathematical modeling of the respective system. Classic methods are strong if a mathematical and analytical description of the underlying model is possible, such as in the case of physical laws of nature, where they perfectly complement ML methods. To conclude, Figure 2.1 illustrates the AI definition chosen within this thesis.

The transfer of the thoughts of Simon [258], Kahneman and Klein [145], Kahneman [144], Falchi [63], and Hodler et al. [121] into perception demonstrates that ML methods do not require the domain knowledge of an expert in development as it is required for classic methods. However, it also becomes clear that ML methods do not incorporate domain knowledge in the way classic methods do. Consequently, the aim to understand perception results and the development of trustworthy systems leads to a combination of classic and ML methods within a general AI system, according to Hodler et al. [121], facilitating a holistic analysis and application of perception methods in this thesis.

2.2 Pipelines and Abstraction Levels

Autonomous navigation in unstructured environments requires similar primary processing steps as autonomous driving in structured environments. Liu et al. [173] describe the processing pipeline for structured environments as an interaction of perception, localization, prediction, routing, decision, planning, and control. They define sensing, percep-

tion, and decision as the three cornerstones of a system architecture for autonomous systems. Here, perception comprises the sensing and perception for autonomous robotic systems with 2D and 3D sensor information prior to the mapping step. Furthermore, Reinoso and Paya [226] propose the subdivision of navigation for mobile robotic systems into mapping, localization, planning, and control. The perception and validation methods in this thesis address the sensing and perception steps, according to Liu et al. [173].

Khan et al. [148] and Beyerer et al. [16] propose processing chains for machine vision being closely related. Khan et al. [148] subdivide machine vision into the discussed low-, mid-, and high-level steps, while Beyerer et al. [16] separate into image acquisition, digitization, preprocessing, information compression and extraction, and decision. Preprocessing in Beyerer et al. [16] corresponds to low-level vision in Khan et al. [148], information compression and extraction [16] is equivalent to mid-level vision [148]. High-level vision [148] corresponds to the decision processing step [16]. Digitization in [16] succeeds image acquisition and provides raw, digital images that constitute this work's starting point for low-level perception methods. To conclude, the term perception in this thesis refers to 2D machine vision and 3D perception.

2.3 Low-Level Perception

2.3.1 Confidence Measures for Raw Sensor Data

Sensor confidence analysis estimates the reliability and accuracy of raw sensor data to decide if the captured sensor information is an accurate and reliable representation of the environment. Hence, confidence measures indicate the probability of a measurement to be correct. Two types of confidence and reliability estimation methods exist. Error detection and recovery aim to modify the sensors and their performance to make the sensor data more believable. However, perception for autonomous off-road vehicles requires the second group: confidence and reliability assessment methods determining which sensors perform reliably. In the context of confidence assessment, the terms confidence and reliability are used as synonym within this thesis. In general, two groups of sen-

sors can be distinguished: one group provides single measurements, such as temperature sensors [76, 127, 178], where the sensor confidence is equivalent to the measurement confidence, while the second group yields numerous measurements from one "single-shot" capture and each 2D image pixel and 3D point provides a single measurement as it is the case for all perception sensors.

Frolik et al. [76] discuss the self-validation, fusion, and reconstruction of the acquired sensor data for the first group of sensors. They exploited the fact that multiple sensors measure similar quantities, such as depth estimation in 3D space with 3D LiDAR sensors or stereo camera systems. Usually, the measured parameters may be correlated, the sensors are not truly redundant but quasi-redundant, and one single confidence measure from multiple sensors is derived by the measurements of these similar quantities [76]. While Frolik et al. [76] deduce confident measurements by a combination of multiple, quasi-redundant sensor measurements, Hughes [127, 128] estimates the reliability of each sensor. The theoretical background of [127, 128] is inspired by the psychological research of Lawrence Marks on the human sensory system [183]. According to Marks [183], certain properties, such as intensity or duration, exist for all sensors. These analogous attributes and qualities may be in different forms but nevertheless similar for all senses – or sensors. The systematic assessment of sensor confidence in [127, 128] can indicate the trust to be placed into an individual sensor on the basis of its estimated reliability in the sensor model.

Broten and Wood [25] examine the confidence levels of sensor outputs for multi-sensor arrays on the basis of ANNs. They propose a combination of sensor fusion and ANNs inspired by the central idea of standard addition in analytical chemistry: it is assumed that an ANN can learn the relationships between the outputs of simulated sensor arrays and the individual analyte concentrations in a mixture of analytes. The learned relationships determine the confidence level of the sensor outputs applied in data fusion from the examined sensors. An ANN with three layers is trained to estimate a numeric value for the confidence level of a sensor output in the range from 0 % to 100 %. However, the approach of Broten and Wood [25] requires an approximately linear relation of the individual analyte concentration as well as the accurate dosage of the

analyte [85]. This can only be guaranteed in simulated data, as examined in [25], and the ANN-based estimates would require an explanation of the confidence assessment with XAI methods (see Section 2.5.3). Hence, this approach is hardly applicable for sensor data from unstructured, potentially unknown environments in safety-critical applications.

The theoretical accuracy of LiDAR systems is analyzed in [7] for airborne laser scanning. The authors [7] state that LiDAR accuracy depends on the signal-to-noise ratio of the reflected signals and on LiDAR beam resolution. For robotic applications, Thrun et al. [268] define four types of measurement errors for beam models of range finders, such as LiDAR sensors: correct measurements with rather low and local noise, unexpected objects, failures, and random measurements. These measurement errors are modeled in a probabilistic manner and integrated into the state vector in the subsequent mapping and localization step. For instance, the authors [268] recommend modeling noise for correct measurements by a narrow Gaussian distribution within the limited measurement range of the respective sensor. Failures denote the missing of obstacles, e.g., for reflecting surfaces or objects absorbing the emitted light of a LiDAR sensor, while random measurements refer to phantom measurements mostly caused by multi-path scattering or sensor crosstalk [268]. As proposed in [268], the probabilistic modeling requires a tight and computationally expensive coupling of perception and mapping that can also limit the generic application of confidence measures.

In contrast to [268], Wolf and Berns [296] propose a generic sensor-fusion approach that conducts a separated uncertainty analysis for sensor data to increase the robustness in the subsequent classification and mapping steps. This uncertainty analysis is integrated as quality assessment inside a layered perception framework similar to the level structure for perception in this thesis. Environment modeling and confidence measures are conducted for volume pixels, so-called voxels. The authors [296] derive the measurement quality for LiDAR sensors from the beam expansion and propose three beam models: "beam distribution, constant accuracy, dynamic accuracy"[296, p. 3]. The beam distribution model requires the availability of the relevant modeling parameters from the sensor manufacturer. However, the authors state that these parameters are not available for some LiDAR types, such as Velodyne LiDAR sen-

sors, and constant and dynamic accuracy modeling is exploited for these LiDAR sensors. The stereo image disparity estimation quality mainly defines the quality of stereo camera systems. Here, Wolf and Berns [296] propose exponential and quadratic error models for the quality assessment of stereo cameras and a customized filtering approach that eliminates low-quality 3D points from stereo image disparity estimation.

Motten and Claesen [199] state that textureless image regions have a higher probability of inducing incorrect depth estimates. Hence, texture analysis can provide a confidence estimate for stereo image disparity estimation. Beyerer et al. [16] define texture as "a two-dimensional structure with a certain deterministic or statistical regularity"[16, p. 651] and distinguish structural, structural-statistical, and statistical textures. Julesz and Bergen [143] propose a texture measure that utilizes a discrete Markov Random Field model to describe the relationship between a pixel and its respective neighbors, while Hu and Ensor [125] propose the analysis of image textures via their Fourier spectrum. The authors [125] describe texture using a collection of properties, such as pattern size or directionality, within a texture descriptor. Beyerer et al. [16] state that gray-level co-occurrence matrices (GLCM) analyze spatial dependencies of image pixels to other, neighboring image pixels. They are often used in medical imaging [18, 312] and landscape classification [103]. GLCM are mostly described with second order statistics, such as correlation and homogeneity, as detailed in [317], and also facilitate texture analysis, as described in [103].

Confidence assessment is also subject to research in other scientific domains, such as wireless sensor networks or subsea sensor spreads. Scheffel and Fröhlich [244] propose a confidence attribution scheme to increase sensor reliability in wireless sensor networks. Each value, respectively sensor measurement, is supplemented with a confidence level. This contrasts the approaches of Hughes [127, 128], who assigns a confidence estimate to the sensors, and of Frolik et al. [76], who deduce confident measurement from quasi-redundant, multiple sensors. The authors [244] state that their approach leads to increased resilience in case of sensor faults and data injection by intruders. In the subsea domain, the theoretical uncertainty of sensor spreads is modeled with the Total

Propagated Uncertainty (TPU)[1] that is formed by combining the Total Vertical Uncertainty (TVU) and the Total Horizontal Uncertainty (THU) for sonar sensor systems in digital terrain modeling.

2.3.2 2D and 3D Features

2D features provide the basis for feature-based 2D image registration. Feature-based visual simultaneous localization and mapping (SLAM) tracks features in subsequent keyframes to estimate the camera poses for a localization inside a map. The majority of 2D features in feature-based visual SLAM rely on edge detection, and a high contrast in images benefits 2D feature detection. Well-known examples for 2D features are scale-invariant feature transform (SIFT) [175], speeded up robust features (SURF) [10], ORB (Oriented FAST [284] and rotated BRIEF [30]) [232], and KAZE [2]. ORB-SLAM relies on ORB features which are based on rotation invariant noise-resistant BRIEF descriptors and achieve an equivalent performance as SIFT with a lower computational effort, according to [232]. Wang et al. [287] propose tracking robust features in multiple layers of contrasted images for visual SLAM. Here, multi-layered representations of images as different contrast-enhanced versions of the original images ensured a constant brightness in subsequently taken images [287]. The approach of [287] is demonstrated on SIFT, SURF, and ORB features and achieves an improved robustness, especially in changing lighting conditions similar to the visual SLAM with high-dynamic ranging (HDR) images discussed in Section 4.4.

In 3D space, Point Feature Histograms (PFH) [235] and Fast Point Feature Histograms (FPFH) [234] can be utilized to describe the local geometric structure of 3D points with underlying surface and point neighborhood on a detailed level. Rusu et al. [234, 235] state that both are invariant to sampling densities and noise levels of neighbors, while FPFH features are optimized for fast calculation and preserve the majority of the discriminative power of PFH features.

Voxel Cloud Connectivity Segmentation (VCCS) [209] supervoxels operate on a voxel representation and describe the local details of a set

[1] Kristensen, Ole: https://www.eiva.com/about/eiva-log/how-navimodel-handles-theoretical-uncertainty-of-subsea-sensor-spreads, access on 26.11.2021.

of voxels with $\mathbf{f} = [x, y, z, L, a, b, \text{FPFH}_{1...33}]$, whereby L, a, and b denote the CIELAB color information.

Ensemble of shape functions (ESF640) with 640 degrees of freedom (DoF) divide point sets in ten shape function histograms [295]. Shape functions capture point distance, area shape, and angle, while the underlying, real surface is approximated with a voxel grid and separates the shape functions into 64 bins for each function. The authors [295] state that ESF descriptors are insensitive to outliers, holes in the data, coarse object boundaries, and noise. The cross-source graph matching method (CSGM) of Huang et al. [126] utilizes ESF640 descriptors to preserve the local structure of the point set inside each extracted VCCS supervoxel, as discussed in Section 2.3.2 and 4.3.4.

2.3.3 Calibration and Registration

Registration is subject to extensive research in medical imaging [109], calibration of multi-sensor systems [54, 74, 158, 311], and SLAM for autonomous platforms [60]. The calibration of multi-sensor systems requires the intrinsic calibration of each sensor as well as the calibration of each sensor to the vehicle frame. The intrinsic calibration of 3D perception sensors, such as rotating 3D LiDAR sensors, is usually provided by the sensor manufacturer, while the intrinsic camera calibration is typically performed after mounting the sensor on the platform. The libraries of OpenCV and the Robot Operating System (ROS)[2] provide well-established software for the intrinsic calibration of individual 2D cameras[3] and camera systems in horizontal or vertical stereo-setup[4]. Stereo camera systems require an extrinsic calibration, as discussed in Section 3.3.

In general, the registration of multiple sensors to the vehicle frame can also be achieved via the extrinsic calibration and registration of all

[2] https://www.ros.org/, access on 18.01.2022.

[3] https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, access on 28.10.2021.

[4] Camera Calibration, 3D Reconstruction: http://wiki.ros.org/camera_calibration, https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, access on 18.01.2022.

sensors relative to each other and their joint registration to the vehicle frame, as described in Section 4.2.

Registration methods can be separated into online and offline methods, local and global methods, as well as direct and transformative methods [126]. Maye et al. [186] state that offline methods search for the minimum of a cost function, while online methods mostly optimize the state vector of a Bayes filter. Offline methods facilitate the validation of the registration results prior to their utilization on the platform. Consequently, sensor calibration and sensor data registration for critical applications typically relies on offline methods. The local or global character of a registration method is defined by its optimization method [234]. Local approaches, such as the extensively researched and evolved ICP algorithm [38, 250, 311], guarantee local optimality. Global methods overcome the problem of convergence in local minima due to faulty initialization, but with the major drawback of extensive computational effort. Appropriate direct and transformative as well as local and global methods are discussed subsequently.

Registration requires a common representation of the input data. Direct methods minimize the distance between aligned points or features [15, 188, 269] and require less computational effort but are mostly unsuitable for registering data from different types of sensors. Transformative methods convert the registration into a model correspondence problem by transforming 3D points from the Euclidean space to other representations [48, 202]. Consequently, transformative registration methods include all 2D–3D registration methods as well as feature-based methods extracting 3D features from point clouds and transforming them into the feature space for registration [234, 295]. Here, abstract data representation can increase the robustness but risks information loss and a higher complexity of the registration method.

2D–3D Registration. Pandey et al. [207], Dhall et al. [49], and Geiger et al. [82] present registration approaches of cameras to a single LiDAR sensor with calibration targets. Dhall et al. [49] extract correspondences from 2D images and 3D point clouds with ArUco markers to determine accurate rigid-body transformations between a LiDAR and a single camera, while Geiger et al. [82] present a similar approach with checkerboards. Kümmerle et al. [158] propose an automatic calibration approach for

multiple cameras and depth sensors using a spherical calibration target. The 2D images are projected onto a spherical screen that projects a sphere onto a circle. It is assumed that the depth sensors provide range data in an ordered structure with rows and columns; hence the presented method is not applicable for unordered point clouds. Furthermore, the authors of [158] provide a detailed overview on calibration errors and their impact on the fused data. They state that a high robustness of a 2D–3D sensor calibration with checkerboards requires a suitable board distribution with different orientations as normals close to the viewing ray prevent a correct detection. Park et al. [212] demonstrate that active depth sensors are sensitive to the reflectivity of the surface, which influences the measured depth and can lead to a notable depth offset between black and white areas. For stereo cameras setups and RGB-D cameras, the camera's resolution is notably higher than the resolution of the depth sensor. Hence, the precision of the vertex detection with 3D edges in the point cloud is the major limiting factor in the 2D–3D calibration with boards. Planar targets with holes as utilized in [74, 280] are subject to similar depth estimation errors due to inaccurate feature point detections [158].

However, the utilization of calibration targets in unstructured environments is deliberately avoided in this thesis to keep humans out of potentially hazardous environments. Without calibration targets, the 2D–3D calibration of passive camera systems and active depth sensors is also possible [32, 95, 165]. Gräter et al. [95] measure LiDAR reflections in a customized darkroom and later minimize reprojection errors. This calibration setting is not applicable for off-road vehicles, especially not for heavy construction machinery. Castorena et al. [32], Pujol-Miro et al. [219], and Levinson and Thrun [165] match intensity features from camera images and depth edges from range sensors. Inaccuracies can occur if different features for intensity and depth are extracted and wrongly associated. Alternatively, the mutual information (MI) between surface intensities can be maximized as presented in Pandey et al. [206]. However, this can be subject to wrong associations for large initial decalibrations. Transformative, cross-source registration methods without calibration targets often rely on a combination of a coarse and a fine registration step [54, 219]. Dutschk et al. [55] compare Gaussian processes

and weighted least squares methods to fuse 2.5D sensor data in surface inspection and present a proof of concept with confocal microscopy and white light interferometry on real and simulated data using root-mean-square error (RMSE), correlation, Structural Similarity Index Measure (SSIM), and multi-scale SSIM metrics. They found that weighted least squares methods perform better with less computational effort in fusing optical cross-source measurement data. The surface is approximated using a robust, iterative moving least squares method which fuses implicit surfaces depending on the uncertainty σ^2 of the model with a weighting factor $w_i = 1/\sigma^2$ [55] similar to the confidence-based weighting introduced in *UCSR* (see Section 4.3). Furthermore, Dutschk et al. [54] present a registration routine for multimodal data in surface inspection. Coarse and fine registration are combined to increase the registration result's reliability, accuracy, and success rate [54]. The coarse alignment step extracts contours, while the fine registration of [54] step relies on area-based mutual information. Pujol-Miro et al. [219] introduce a classic registration of 2D images to unorganized 3D point clouds by extracting and matching relevant features, so-called contour cues. Similar to [54], coarse registration with contour extraction and a subsequent fine registration are combined [219], which provides a promising approach for detecting contours in unstructured environments that is further discussed in Section 4.3.2.

RegNet of Schneider et al. [246] is the first convolutional neural network (CNN) proposing an extrinsic calibration of multimodal sensors with six DoF and compares favorably to classic approaches on the KITTI 2012 dataset [83]. RegNet combines a joint initial estimate and online correction of the extrinsic calibration parameters for 3D LiDAR and 2D RGB data. The typical steps in registration – feature extraction, feature matching, and global optimization – are combined into one CNN. The 3D point cloud is projected onto a 2D depth image using the intrinsic camera matrix and the initial transformation estimate. A mean calibration error of 0.06 m in translation and 0.28° in rotation is achieved on KITTI [83] from a maximum decalibration of 1.5 m and 20°.

Liu et al. [172] propose an online calibration method based on RegNet [246] with the additional integration of a stereo camera system. The projected LiDAR depth map and the stereo depth map are registered and

fused. The fused LiDAR–stereo depth map is subject to subsequent registration inside a RegNet-like architecture. The evaluation is conducted on data from structured indoor environments with clearly separated objects. Unstructured environments are not considered.

CalibNet [131] estimates the rigid six DoF transformation between a 2D camera and a 3D LiDAR in real-time [131]. Similar to Schneider et al. [246], calibration targets are not required, and image feature extraction is conducted with a pre-trained ResNet-18 network [113]. Inputs to CalibNet are the intrinsic camera matrix, an RGB image, and a LiDAR point cloud. The network is trained to maximize the geometric and photometric consistency of LiDAR clouds and RGB images by applying 3D Spatial Transformer Networks [105]. Both photometric and point cloud distance loss within the 3D spatial transformer layer are selected as training losses in [131]. CalibNet is evaluated on the KITTI 2012 dataset [83] and can correct decalibrations up to $\pm 20°$ and 0.2 m with a mean accuracy of 0.004 m in translation and 0.41° in rotation [131].

CMRNet [33] regards registration in the SLAM context and estimates image localization inside a pre-existing 3D LiDAR map with a mean accuracy of up to 0.27 m in translation and 1.07° in rotation on the KITTI odometry dataset. The network architecture of CMRNet is inspired by PWC-Net [262] for optical flow predictions. On the basis of a rough initial estimate for the camera pose, the PWC-Net architecture is used without weight sharing and upsampling layers. A fully connected layer prior to the first layer for optical flow estimation facilitates regression. A smooth L_1 norm of the translation is used as training loss, as proposed in Girshick [89].

In medical imaging, the intensity-based registration of pre-operative 3D data to intra-operative 2D data presents a key requirement in medical imaging and image-guided intervention. Pre-operative 3D data originates from computed tomography, cone-beam computed tomography, magnetic resonance imaging, and CAD models of medical devices, while intra-operative 2D data mainly consists of X-ray images. Miao et al. [195] present a CNN regression approach for real-time registration of a 3D X-ray attenuation map from computer tomography to a 2D X-ray image. The complex CNN regression task is separated into multiple, simple

sub-tasks by a hierarchical application of the regressor and training on local zones.

3D–3D Registration of Similar-Source Data. 3D–3D registration minimizes an error metric between two point clouds. The local Iterative Closest Point (ICP) approach [15, 311] minimizes the distances between single corresponding points of two point clouds. Point-to-Plane-ICP minimizes the distance of corresponding points on estimated surfaces to estimated surfaces orientations in the other cloud. Generalized-ICP (GICP) [250] minimizes the distance between approximated local surfaces and calculates the orientation of estimated surfaces from a singular value decomposition (SVD) of the covariance matrix. It notably outperforms ICP and Point-to-Plane-ICP in terms of root mean square error (RMSE) of the registration result [250].

In order to overcome the problem of convergence in local minima, Fitzgibbon [69] proposes an alternative registration approach for the registration of 2D LiDAR sensors, where the Levenberg-Marquardt algorithm is used for error minimization, and registration of 3D clouds is only treated theoretically. Therefore, it remains open whether this approach is suitable in the more complex registration of 3D clouds.

Gao and Spletzer [78] and Schneider et al. [247] propose online calibration approaches with additional requirements that are hard to realize for tracked off-road vehicles in unstructured environments. Gao and Spletzer [78] propose an extrinsic online calibration approach of multiple LiDAR sensors on a mobile platform with calibration targets. A priori information can be integrated, and optimization constraints, such as calibration tolerances, can be taken into account. Furthermore, global optimality can be achieved according to [78]. In their findings, the mean absolute error lies in between 13.48 cm and 21.35 cm between the target reprojection residuals [78]. However, retro-reflective tape has to be mounted to poles inside the sensor FoV as landmarks in pairs of two with a known size and initial sensor pose estimates with at least $\pm 5°$ rotational accuracy are required. Schneider et al. [247] propose an odometry-based extrinsic online sensor calibration, where relative orientation and translation of two sensors are calculated using their time-synchronized pose changes. However, the sensor poses within the odometry coordinate

system have to be known, and the reliability of odometry measurements for tracked vehicles is limited in case of slippage.

Frese et al. [75] present a registration approach for 3D sensors to the vehicle frame exploiting the 3D model of the vehicle. The manipulator arm of the platform lies inside the sensors FoV, and its inclusion in the 3D model allows registration of the LiDAR point cloud to the 3D model. Here, the alignment result of the 3D model to the 3D sensor cloud yields the calibration to the vehicle frame.

3D–3D Registration of Cross-Source Data. Direct 3D–3D registration methods are mostly unable to register cross-source point clouds because common features of both modalities are hard to find due to different cross-source characteristics.

To register highly dense cross-source point clouds, Mellado et al. [189] combine Random Sample Consensus (RANSAC) [68] and downsampling. However, it remains unclear if their approach can register less dense point clouds. The registration of 3D point clouds with strong cross-source characteristics requires transformative registration methods to achieve a sufficiently similar representation. Jian and Vemuri [136] transform 3D clouds to Gaussian Mixture models, which implies the minimization of a statistical discrepancy measure. Deng et al. [48] suggest mapping 3D point clouds into a shape representation. The Schrödinger distance transform minimizes the geodesic distance of the clouds described on the unit Hilbert sphere.

CSGM [126] registers cross-source point clouds with a combination of local details and global structure. Over-segmentation with VCCS supervoxels extracts global characteristics according to Papon et al. [209]. The voxel centers form the nodes of the adjacency graph that represents the global cloud structures. Local details are captured in ESF640 descriptors. The registration problem is solved by factorized graph matching [314]. The combination of local and global information increases the robustness in cross-source registration similar to [54]. Finally, ICP refines the graph matching results and extracts the rigid six DoF transformation. CSGM is applied to register different types of structured point clouds, such as multi-view stereo clouds from a KinectFusion camera to synthetic clouds [126]. 3D clouds from single view stereo, point clouds from un-

structured environments, or clouds with outliers, artifacts, and notably different densities were not evaluated.

Park et al. [210] propose a CNN architecture to calibrate and fuse 3D point clouds from 3D LiDAR and stereo image depth reconstruction. Their architecture consists of a calibration and a depth fusion network with the calibration network projecting LiDAR clouds onto 2D disparity maps and registering them to the stereo image disparity map. The depth fusion network fuses both disparity images on the basis of the extrinsic calibration and generates high-precision disparity maps. Both networks are trained on the KITTI dataset [83] with pseudo ground truth labels [210]. An experimental evaluation of the proposed architecture is conducted for different, structured scenes and across different sensor settings. However, the projection of both 3D clouds into 2D space introduces information loss.

Haskins et al. [109] present a method for the 3D–3D registration of medical cross-source data that combines a deep similarity metric with a composite optimization strategy. The 3D data from ultrasound and magnetic resonance constitutes the input into a volumetric CNN. The two imaging modalities present a notable difference in their appearance similar to cross-source 3D data from LiDAR sensors and stereo image disparity estimation. The composite optimization determines a suitable initialization with differential evolution that subsequently initializes the Newton-based Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization of the similarity metric. The BFGS optimization provides an iterative solution for the unconstrained optimization problem. Linear interpolation is used to compare the resulting two pixel sets on the basis of the learned similarity metric. The authors [109] demonstrate that their method outperforms classic MI and state-of-the-art, feature-based methods on the basis of a target registration error that measures the geometric distance of manually selected points.

2.3.4 HDR Fusion, RGB–NIR Fusion

Mertens et al. [192] present the implicit Mertens Exposure Fusion (MEF) method to fuse multiple exposure sequences of the same image into one HDR image. The implicit fusion in 8-bit low dynamic range facilitates a simple acquisition pipeline without the need to integrate the camera cal-

ibration, compute a camera response function, or tone-mapping. Simple quality measures are used to generate the scalar-valued weight map for blending the input images: saturation, contrast, and well-exposedness. Furthermore, Li and Lei [166] compare multi-exposure fusion results using the SSIM score and state that CNN features can improve the quality of fused images compared to most classic approaches, but they achieved a lower SSIM score than MEF.

The fusion of multi-spectral image data, e.g., RGB and NIR images, is typically conducted with PCA, Wavelet, or Curvelet transformation or via a transformation of the color space. PCA does not fuse information from different images, but it identifies and preserves only those channels that exhibit the highest variance, as discussed in Section 3.4. Sappa et al. [237] state that Wavelet transformation enables the time-frequency-representation of signals with a frequency-dependent resolution and compare different image fusion methods with the discrete wavelet transform in NIR and Long Wave Infra-Red. Curvelet transformation is the higher dimensional generalization of the Wavelet transform [179]. Due to this generalization, images captured from different angles can be fused. However, this generalization is not required with prism camera systems providing an RGB and an NIR image through the same optic. RGB and NIR images can also be fused in the Hue, Saturation, Value (HSV) color space. The RGB image is transformed into HSV, and the NIR image is exchanged with the Value image for images from the same optic [70]. Contrasting the RGB–NIR fusion approaches discussed, this thesis proposes MEF for RGB–NIR fusion, as detailed in Section 4.4.

2.4 Mid-Level Perception

2.4.1 Disparity Estimation from Stereo Images

Wheatstone [293] presents the first known investigation of human binocular vision and proposes the first detailed plans to view image pairs with a stereoscope in apparent 3D. The first instruments to perform correlation on digital images were discussed in the 1960s [279]. The development of Charge Coupled Devices (CCD) [24] and Complementary Metal Oxide Semiconductor (CMOS) imaging sensors [73] mainly led to the stereo im-

agery becoming a central focus in computer vision research. Stereo image disparity estimation is still relevant despite many years of research [118, 177, 302]. Repetitive patterns, occluded areas, as well as uniform image parts are not solved yet. Numerous local [139, 266], semi-global [81, 118], global [23, 155, 223], and seed-growing [34, 35] approaches to the stereo correspondence problem exist.

Classic Disparity Estimation from Stereo Images. Pears et al. [214] divide classic stereo image disparity estimation algorithms into correlation-based and feature-based stereo matching methods. Correlation-based methods produce dense disparity maps, while feature-based methods match detected keypoints and yield sparse point clouds of the keypoints. Local methods consider small image areas, so-called patches, whereas global methods consider the whole image during optimization. Local, correlation-based stereo matching can roughly be divided into four steps: matching cost computation, cost aggregation, optimization, and disparity refinement. They provide fast results with the drawback of less accurate disparity maps [139]. Global approaches like graph cuts [23, 155] or sub-pixel accuracy methods provide very accurate results but the high computation effort makes them inappropriate for real-time systems.

Semi-global matching (SGM) [118, 119] ranges among the most popular, classic stereo image disparity estimation methods. It facilitates an accurate and yet efficient local matching by comparing the mutual information (MI) between two images and integrating it into a global smoothness constraint. Kallwies et al. [146] combine horizontal and vertical disparity images for enhanced depth estimation. Kallwies et al. [147] further present a stereo processing approach that fuses the SGM-estimated horizontal and vertical costs prior to the initial disparity selection to process image triplets. The initial disparity image is generated by choosing the disparity value corresponding to the lowest costs (winner-takes-it-all). The early fusion approach in [147] demonstrates that SGM on image triplets outperforms individual horizontal and vertical stereo image disparity estimation as well as the late disparity fusion approach of [146].

The local, correlation-based CCRADAR approach [139] compares the similarity of pixel intensities from the image pair for standard RGB images to calculate an initial disparity value for each pixel. Different cost functions are defined and evaluated: Modified Color Census Transform

(MCCT), the Sum of Absolute Differences of the intensity values (SAD), as well as the Sum of Gradient Differences (SGD) along the horizontal (SGD_x) and vertical image axis (SGD_y). The cost values of each pixel are combined with regard to the selected weight of the cost functions. Aggregation of the cost values is performed with Guided Filtering (*GF*) [112], and initial disparity images are generated with a winner-takes-it-all strategy. Post-processing includes a left-right consistency check (LRC), cross-region-based voting, and detection of remaining artifacts and refinement. CCRADAR ranked 9 of 167 in the Middlebury Stereo Evaluation for RGB images at the time of writing this thesis[5].

In contrast to RGB images, additional spectral channels can provide additional information for correlation-based similarity measures. Hyperspectral images are commonly used in photometric stereo imaging [203, 205], remote sensing [79, 169], food analysis [56, 264], or victim detection with rescue robots [271]. Local approaches, such as CCRADAR [139], facilitate a separate processing of each color channel for multi- or hyperspectral images and are suitable for parallelization on General Purpose Computation on Graphics Processing Units (GPGPU). This can provide an enhanced stereo image disparity estimation in unstructured environments, as discussed in Section 5.1.1.

Disparity Estimation from Stereo Images with CNNs. Stereo image disparity estimation with CNNs typically represents the stereo correspondence problem as a classification problem [162, 307, 309], and cost computation and aggregation are learned by CNNs [139, 266, 310]. Common CNN architectures consist of two Siamese networks [309, 310].

Žbontar and LeCun [309, 310] propose the Matching Cost-CNN (MC-CNN) to predict the matching accuracy of two image patches. Here, the learned similarity measure between two image patches is used to initialize the matching costs and to deduce initial disparity estimates. These initial disparity estimates can be post-processed by classic methods, such as cross-based cost aggregation, LRC, or a median filter [309]. The Siamese networks are trained in a supervised manner with stochastic gradient descent on mini batches. The initial MC-CNN approach [309]

[5] Middlebury Stereo Evaluation, v2: https://vision.middlebury.edu/stereo/eval/, access on 09.12.2021.

achieved an error rate of 2.61 % on the KITTI 2012 dataset that ranged among the top methods at the time of evaluation in August 2014 [310]. Furthermore, Žbontar and LeCun [310] proposed two architectures, a fast and an accurate architecture whose major difference lies in the combination of their convolutional layers' outputs. Here, the fast architecture compares the similarity score by extracting a vector from each of the two input patches and computes the cosine similarity by normalization and a dot product. Both the accurate and fast architecture were evaluated on the KITTI 2012, KITTI 2015, and the Middlebury datasets and ranged among the most successful methods in all benchmarks in October 2015 [309]. Currently, MC-CNN (fast) ranks 121/211 in KITTI[6] and 32/162 in Middlebury[7]. Training of both MC-CNN architectures was conducted on the synthetic Middlebury dataset, and the test of MC-CNN on the real-world KITTI 2012 dataset [83] showed promising results that indicate a favorable domain transfer performance, especially for fast MC-CNN.

Luo et al. [177] extend MC-CNN [309, 310] and treat the correspondence problem as a multi-class classification problem that represents all possible disparity values as classes. The Siamese networks with shared parameters consist of four layers with spatial convolutions with small filters, such as 5×5 or 3×3, and a rectified linear unit. The last layer of the Siamese networks only consists of the spatial convolution and spatial batch normalization to preserve the information in negative values. The receptive field is 9×9 if four layers with 3×3 filters are used [177]. Finally, a product layer with a dot-product fuses the processing results of the two Siamese networks receiving patches from the left and right image [177]. The approach of Luo et al. [177] notably outperforms both fast and accurate MC-CNN [310] on KITTI 2015 in terms of runtime and all error metrics on the validation set. The important findings include the observation that subpixel enhancement and refinement do not always improve the resulting disparity map [177]. The authors also state that simple cost aggregation, as in block matching [266] or CCRADAR [139], improves local smoothness and benefit the results.

[6] Stereo Evaluation 2012: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo, access on 28.10.2021.

[7] https://vision.middlebury.edu/stereo/eval3/, access on 28.10.2021.

Other disparity estimation approaches with CNNs are presented with the DispNet architecture [187], and the cascade residual learning approach of Pang et al. [208] inside a two-stage CNN. The first stage of [208] evolves from DispNet [187], the second stage explicitly rectifies the disparity images initialized in the first stage and generates residuals across multiple scales [208]. Similar to MC-CNN, the similarity between image patches was learned in a supervised manner on KITTI 2015. Evaluation was conducted on FlyingThings3D [187], Middlebury 2014 [243], and KITTI 2015 [191]. The method of Pang et al. [208] ranked 173/307[8], while accurate MC-CNN ranked 214/307[9]. However, the authors [310] show that MC-CNN yields less disparity estimation errors than cascade residual learning [208] on test images with a higher number of unstructured elements.

Combined solutions of mid-level stereo image disparity estimation and high-level semantic segmentation within one network are, for instance, presented in Bleyer et al. [19–21], Hane et al. [106], and Yamaguchi et al. [301]. However, their integration in the modular perception–validation pipeline proposed would not allow a flexible combination of the perception methods proposed in this thesis according to demand.

For post-processing, Drouyer et al. [52] propose the densification of sparse disparity maps and demonstrate it on MC-CNN results [309]. Here, the captured scene is assumed as a collection of different objects, and the surface of each object is assumed as a composition of multiple simple shapes that can be modeled as planes using RANSAC. The detected planes are filled with points to generate a dense point cloud. The approach of Drouyer et al. [52] ranked number 12 in version 3 of the Middlebury Benchmark in January 2022[10]. However, scenes in unstructured environments are not composed of objects with partially smooth surfaces, and the top-down segmented regression segmentation of [52] is hence not applicable.

[8] Stereo Evaluation 2015: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo, access on 17.01.2022.

[9] Stereo Evaluation 2015, access on 17.01.2022.

[10] Middlebury Stereo Evaluation - Version 3: https://vision.middlebury.edu/stereo/eval3/, access on 26.01.2022.

2.4.2 Confidence Measures for Disparity Maps

Several model-specific approaches were published to assess the confidence of stereo image disparity estimation methods within the last years [110, 111, 124, 218, 281]. Here, most model-specific stereo confidence measures evaluate the cost volume processed by the stereo image disparity estimation algorithm to determine the most likely disparity value [218]. For instance, the peak ratio measure [124] is a popular, model-specific confidence measure for single disparity estimates and measures the distance between the minimum cost value and the second-lowest cost value to derive a reliability measure: the higher the distance, the higher the reliability of the estimated disparity. Another pixel-wise confidence assessment of disparity estimates is proposed by Veld et al. [281] and demonstrated on MC-CNN [309]: a pixel-wise analysis of the cost functions especially targets the confidence assessment of disparity estimates for difficult image areas, such as periodic structures, and eliminates wrongly estimated disparities. The proposed confidence measure assumes that a truly matching disparity value does not differ from the given ground truth more than one pixel, and that the minimal matching cost for each pixel is clearly identifiable by a single, distinguishable minimum of the cost function [281], similar to the peak ratio measure.

Other confidence metrics target a simple and fast calculation on small, low-cost devices, such as field programmable gate arrays (FPGA). They utilize a decision tree in combination with an additional disparity refinement, as proposed in Motten and Claesen [199]. The authors of [199] use texture and depth differences between the center pixel and neighboring pixels of the same color as confidence metrics to train a binary decision tree, and each internal node represents a decision on a feature. Texture is measured using a fixed window of intensity values around the examined center pixel within a pixel window similar to the range-filtering method discussed in Section 4.1.2.

Seki et al. [251] analyze discriminate features to predict the reliability of correspondence estimates and a confidence fusion method for dense disparity estimations. A classifier with two input features determines the model-specific confidence estimate. Neighboring pixels with consistent disparity values – and hence a small gradient of the disparity values – are more likely to be a correct estimate. Hence, the approach of Seki et

al. [251] assigns high confidence to a clearly distinguishable minimum of the cost function similar to [124, 281].

Häusler et al. [110, 111] analyze model-specific disparity estimation confidence measures on synthetic and real-world disparity estimation results to predict potentially erroneous areas in disparity maps. They evaluate curvature, perturbation, peak ratio, and LRC, and show that the performance of confidence measures can vary notably between synthetic and real-world data. The minimum of accumulated costs, disparity variance, maximum likelihood metric, and shape of the cost function provide decent results for cost function dependent measures. LRC, nowadays a common post-processing method for disparity maps, yields fast and satisfying results evaluating the cost-independent measures, according to Häusler et al. [110, 111], and was hence selected for the post-processing of both proposed stereo image disparity estimation algorithms in this thesis (see Section 5.1).

2.4.3 Sensor Data and Information Fusion

Multiple sensors with different measuring modalities provide a wider breadth of information for the environment perception of autonomous systems. According to Heizmann et al. [115], information fusion tries to determine the best combination of available sensor systems and processing methods for optimal exploitation of sensor resources and processing capabilities. Consequently, information fusion is also referred to as (sensor) data fusion in this thesis as it combines substantial information from multiple sensors, and creates a composite 2D image or 3D point cloud with a higher information content and a more useful representation for perception tasks [259]. Information fusion can be conducted on different abstraction layers from raw sensor data up to the fusion of processed sensor data with various abstraction layers and a semantic format, e.g., the fusion of object detection results from two different sensors [58].

Many works pursue object-oriented approaches [87, 88, 115] that target the results of the environment interpretation within high-level perception. Gheta et al. [88] propose an object-oriented information architecture that combines prior knowledge and real-world sensory information within a central object-oriented environment model. Each information is characterized by its uncertainty in a Degree-of-Belief (DoB) distribution [87, 88,

115]. Heizmann et al. [115] propose a method to determine an optimal selection of the input sensor data for environment perception and an object-oriented environment model. Different sensor systems and data processing methods for information extraction from the environment are summarized within the concept of information channels, and the authors identify the optimal combination of all available information channels as the primary challenge in information retrieval. For the optimal input data selection, Heizmann et al. [115] assume a sufficient number of sensors and processing methods, and deduce their optimal combination with Bayesian statistics and an objective DoB interpretation.

As proposed in Dürr et al. [53], iterative fusion approaches combine semantic segmentation and sensor data fusion. High-level perception results, such as a semantic segmentation of camera and LiDAR data, are required for the fusion process. This inhibits a flexible combination of different low-level, mid-level, and high-level methods required for the perception of autonomous off-road vehicles.

To conclude, perception for unstructured environments requires the fusion of raw sensor data to prevent information loss: 2D pixels with intensity measures, 3D measurements with optional intensity information, as well as 3D point clouds with optional confidence information. Particularly 3D–3D fusion of multiple 3D point clouds benefits from confidence assessment and filtering of inaccurate 3D information, especially when stereo camera point clouds are involved. Consequently, object-oriented and iterative fusion approaches are not utilized for the sensor data from unstructured environments analyzed in this thesis. Here, Bayesian networks provide a well-established methodology to consider probabilistic and, hence, confidence information from different hierarchical levels in fusion processes. The Bayesian image fusion approach proposed by Beyerer et al. [17] is an efficient option for the fusion of real-world sensor data. It requires one single measure to describe a DoB in contrast to other methodologies, such as Fuzzy theory [76] or the Dempster–Shafer theory [152]. Briefly summarized, Bayesian statistics exploit a special interpretation of the probability theory axioms of Kolmogorov [167]. Contrasting classical statistics, a probability can be interpreted as DoB permitting the integration of uncertainty and confidence measures. The authors [17] state that the Bayesian approach performs notably better

than other methods for fusing real-world image data, and it was hence selected for the confidence-based 3D–3D fusion in this thesis (see Section 4.1).

2.5 High-Level Perception

Classification assigns one class to a 2D image or a 3D point cloud, while object detection can recognize multiple objects with bounding boxes. The semantic segmentation of 2D images and 3D point clouds yields pixel-wise and point-wise classifications [93].

Naujoks et al. [204] suggest the combination of ML and classic, model-based methods for the semantic detection of landmarks in autonomous off-road driving. RGB images, light detection, and LiDAR clouds are input to the proposed method. It outperforms state-of-the-art ML classification methods and highlights the relevance of classic, model-based methods to complement ML methods. This also demonstrates that especially domains with limited data availability, as the perception of unstructured environments, can benefit from this combination to overcome the need for substantial amounts of data [204].

2.5.1 Semantic Segmentation

Navigation and especially manipulation in unstructured environments require an accurate, pixel-wise or point-wise semantic segmentation and bounding box approaches are unsuitable. The central challenge in semantic segmentation with CNNs is to interpret global information while local information has to be preserved. This requires deep feature architectures to map local-to-global, such as in non-linear pyramid structures. Since 2010, classic methods are mostly outperformed by ML approaches and pixel-to-pixel, end-to-end trained CNNs exceeded the state-of-the-art in semantic image segmentation on the benchmark datasets PASCAL VOC [62] and NYUDv2 [257] according to Long et al. [174]. Hybrid methods for semantic segmentation apply a classic feature extractor such as SpinImages [140] together with an ML classifier such as support vector machines [1]. Typically, semantic segmentation starts with the extraction of features that are summarized by the downsampling character of convo-

lutional layers. The subsequent upsampling assigns a class label to every single pixel or point. Features for 2D image segmentation are extracted in 2D, whereas the feature extraction in 3D segmentation can be conducted in 2D or 3D space. The feature extraction and interpretation with CNNs in 3D space require the application of 3D convolutions. A discretization using a voxel structure as also applied in 3D–3D registration [109] can help to cope with the huge amount of 3D point cloud data.

PointNet [221] proposes a classification method for 3D point clouds. Possible other 3D data representations include a spatial subdivision differing from cubic voxelization, such as OctNet [228] or VoxSegNet [289]. Rendered 2D image views perform feature extraction in 2D, such as SnapNet [22] or SqueezeSeg with spherical projections [298–300]. Superpoint Graph [159] or splats [261] are other possible representations to segment 3D point clouds with increased efficiency. Generally, 3D feature extraction with 3D convolutions requires a notably higher processing time than feature extraction and interpretation using 2D convolutions, as detailed in Section C.1.2. This is confirmed by Milioto et al. [196] wherein the authors state that the 2D segmentation of spherical projections is notably less computationally expensive than class predictions in 3D space. Hence, segmentation in 3D space by now hardly achieves the real-time capability that is required for autonomous off-road vehicles and this thesis focuses on the segmentation of spherical projections in 2D space.

The SqueezeSeg [298] and SqueezeSegV2 [299] architectures estimate class labels during the semantic segmentation process on the basis of a 2D spherical projection. SqueezeSeg evolves from the SqueezeNet architecture [129] and the authors [298] combine real-world and synthetic data from a LiDAR simulator included in Grand Theft Auto V, a video game, to increase the amount of available training data.

RangeNet++ [196] is inspired by SqueezeSeg and achieved a notably better performance in terms of the Intersection over Union (IoU) metric according to Equation 6.3. RangeNet++ designates the basic network architecture using a spherical 2D projection according to [196] to transform 3D image data into 2D space. In contrast to other segmentation approaches, RangeNet++ was specifically designed to work with arbitrary CNN backbones for the segmentation of 2D range images. Behley et al. [11] and Milioto et al. [196] evaluate state-of-the-art segmentation archi-

tectures on SemanticKITTI [11], such as PointNet++, SPGraph, SqueezeSeg [298], SqueezeSegV2 [299], and the RangeNet++ architectures DarkNet21Seg and DarkNet53Seg [196] with DarkNet backbones [65, 225]. DarkNet53Seg yielded an $\overline{\text{IoU}}$ of 49.9 % for the 19 static classes on the test sequences 11 to 21 of SemanticKITTI [11], while SqueezeSeg only achieved an $\overline{\text{IoU}}$ of 29.5 % for the test sequences. Behley et al. [11] assume that the low segmentation performance of SqueezeSeg is caused by the fact that the size and complexity of SemanticKITTI cannot be mapped properly with the number of network parameters in SqueezeSeg. The DarkNet architectures permit a notably higher number of parameters, and the results in [11] substantiate this hypothesis. The authors [11] further show that the sparsity of LiDAR clouds becomes challenging for large distances as the $\overline{\text{IoU}}$ of DarkNet53Seg reduced to less than 25 % in 50 m distance to the sensor. Furthermore, Milioto et al. [196] evaluate three different horizontal pixel resolutions of 2D range images (512, 1024, and 2048) and show that the most accurate segmentation results on SemanticKITTI were achieved with DarkNet53Seg on a 2048×64 spherical projection with kd-tree Nearest Neighbor Search (kNN) post-processing.

Dürr et al. [53] propose an iterative ML approach for the semantic segmentation of 3D LiDAR clouds where a range view representation of 3D clouds is used, similar to Milioto et al. [196]. Camera features are additionally integrated iteratively to increase the robustness and accuracy of the semantic segmentation method. Features are extracted from camera and LiDAR data with a fusion module that transforms the resulting feature maps into a common space. Fusion is conducted iteratively by applying the fusion module for LiDAR and cameras feature maps on different scales in 2D space. Here, semantic segmentation of the 2D camera images with ResNet50 building blocks yielded better results than the deep layer aggregation technique [303]. The approach of Dürr et al. [53] based on camera images and LiDAR point clouds outperformed other state-of-the-art methods on SemanticKITTI. However, a flexible combination of different low-level, mid-level, and high-level methods as required within the perception pipeline proposed in this thesis is not possible with iterative fusion approaches.

2.5.2 Domain Transfer in 3D Segmentation

Domain transfer and domain adaption aim to transfer CNN architectures trained on a specific domain in supervised manner to other domains [122, 133, 160], such as different types of LiDAR sensors or application environments. Their central motivation is to circumvent the generation of labeled training data in a new target domain. Domain adaption includes CNN retraining, while domain transfer designates the transfer of a neural network architecture to another domain without additional data or retraining. Adversarial approaches utilize domain-invariant representations. However, adversarial approaches are difficult in terms of transparency and debugging and tend to fail for pixel-level domain shifts, according to Hoffman et al. [122]. Some works rely on generative adversarial networks (GAN) or simulation to adapt the labeled images or point clouds to the target domain [122, 163].

Fernando et al. [67] deduce a mapping function for domain adaption that aligns source and target domain. Here, the eigenvectors of source and target domain are estimated and represent both domains in subspaces. Jaritz et al. [133] propose a method for unsupervised, cross-modal domain adaption for domain shifts such as day-to-night or dataset-to-dataset and demonstrate it on SemanticKITTI [11], nuScenes [29], and the Audi Autonomous Driving Dataset (A2D2) [86]. In contrast to most other works on unsupervised domain adaption, Jaritz et al. [133] aim at the domain shift of multimodal datasets and require 2D images and 3D images for semantic 3D segmentation. The proposed architecture combines segmentation loss on the source domain and cross-modal loss on the source and target domain, and the two modalities – 2D images and 3D point clouds – learn from each other by mutual mimicking. Langer et al. [160] propose a domain adaption approach for semantic 3D segmentation CNNs onto different LiDAR sensors with different FoV and resolution. They aggregate 3D point cloud data from the source domain, a Velodyne HDL-64E, with 3D SLAM, and a dense 3D model extracts semi-synthetic data to retrain the model for the target domain. Geodesic correlation alignment [299] with cross-entropy loss for the source domain and with geodesic loss for source and target domain is proposed to minimize the domain shift. However, domain adaption requires a retraining for each notable domain change.

A good domain transfer performance of segmentation architectures can overcome the need for an additional generation of training data in the new domain by capturing and labeling new data or generating synthetic data, as discussed in [160]. Position-invariant features for the semantic segmentation of the spherical projection presents one possibility to increase the domain invariance of CNNs in semantic segmentation: Burkhardt et al. [28] demonstrate the existence and extraction of 2D features that allow a position-invariant description of planar contours and grayscale images to facilitate a position-invariant 2D pattern recognition, while Schulz-Mirbach [248] discusses the existence of complete invariant features spaces and derives criteria that ensure the existence of this feature space. Schulz-Mirbach [249] extends the concepts of [28, 248] and proposes a more generic algorithmic solution to construct invariant features for certain changes of the input data. The discussed methods require feature representations that are invariant to integration and represent joint properties for equivalent patterns, as discussed in [249]. They are applicable for finite groups and compact topological Lie groups [117] and imply the possibility of attributing the group with a group average.

2.5.3 Explainable AI

Explainable AI (XAI) tries to assign human-understandable explications for neural networks' decisions as this does not become clear from the algorithms themselves. The explanation of the predictions and decisions of ML systems can be achieved on three levels: pre-modeling explainability, explainable modeling, and post-modeling explainability [9]. The goal of pre-modeling explainability is to examine and understand data used to develop models prior to training. It includes exploratory data analysis, dataset description standardization, dataset summarization, and explainable feature engineering. So far, most methods for exploratory data analysis summarize their main characteristics and focus on statistic parameters within numeric and categorical features [114]. Explainable modeling aims at developing technically transparent models inherently understandable for human operators. Post-modeling XAI, or post-hoc explainability, targets the model-specific or model-agnostic, post-modeling explanation of predictions from AI systems with an inherent black box character. Well-proven post-modeling XAI methods include the analysis

of model predictions via backward propagation and the creation of proxy models. Model-specific approaches are customized for a specific model, while model-agnostic approaches are independent of the underlying AI model. Post-modeling XAI can be subdivided further into visualization, explanation via simplification, textual explanations, and example-based explanations [9][11].

Exploratory Dataset Analysis. Exploratory dataset analysis provides a well-established strategy to determine statistical patterns and correlations inside the assessed data [12]. It exhibits central heuristics and computational tools as a part of exploratory statistics. Exploratory data analysis and exploratory statistics look for patterns in the data [84]. They complement statistical paradigms, such as complex statistical modeling with Bayesian inference. Often, exploratory data analysis is applied in the model formulation process in Bayesian inference. Data visualization reaches beyond the common estimation and testing approaches, as they are typically conducted in the description standardization and summarization of datasets [272]. Gelman [84] introduces model checking to compare original data to the data that the analyzed model reproduces as a combination of exploratory and confirmatory data analysis. Heiler and Michels [114] combine descriptive and exploratory data analysis to, inter alia, analyze frequency distribution, measures of position and dispersion, and the representativeness of samples.

Most research focuses on one target application, such as classification [181, 286]: Mani et al. [181] propose an in-depth method to test the coverage of deep neural network models by examining the dataset quality with statistical measures. Wang and Liu [286] detect class structure ambiguities in classification and propose a reorganization strategy in case of decreasing accuracy. The open-source AI Explainability 360 Toolkit [5][12] presents one of the first toolkits with different explanation methods, such as data explanation or local and global post-modeling.

[11] B. Khaleghi: The How of Explainable AI: Pre-modeling Explainability, https://towardsdatascience.com/the-how-of-explainable-ai-pre-modelling-explainability-699150495fe4, access on 25.01.2022.

[12] IBM Research Trusted AI: AI Explainability 360, http://aix360.mybluemix.net/, access on 17.01.2022.

Prototype Methods. Prototype methods such as MMD-critic [149] and ProtoDash [101] apply a metric, e.g., the maximum mean discrepancy metric (MMD) [97], to compare two data distributions $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$. In this thesis, prototype methods are applied to identify prototypes and criticism for the post-modeling explanation of ML methods by determining a small set of samples for example-based explanations (data distribution $\mathcal{X}^{\in}$) that optimally represent another, notably larger dataset ($\mathcal{X}^{\infty}$). Gurumoorthy et al. [101] introduce ProtoGreedy, a slow and greedy prototype selection algorithm with an objective function that satisfies a key property of weak submodularity [46], and ProtoDash, a notably faster prototype selection algorithm. In contrast to MMD-critic, ProtoDash can operate with any positive definite kernel and Gurumoorthy et al. [101] derive approximation guarantees for the fast ProtoDash algorithm using the proof of weak submodularity. The MMD metric is applied to solve the two-sample problem according to [98]: it tests if two data distributions $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$ are different by selecting samples from each set and comparing them with a well-behaved function that yields large values on the samples taken from $\mathcal{X}^{\infty}$ and small values on the points from $\mathcal{X}^{\in}$, with small being as negative as possible. The test statistic to derive this well-behaved function and to compare the two values is the MMD which depends on the class of smooth functions selected to compare two samples. Here, Gretton et al. [98] evaluated different function classes and selected the unit balls in the characteristic reproducing kernel Hilbert spaces [77] as they converge towards zero if the data distributions of $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$ are equal. However, they are also constrained enough for the empirical MMD estimate to converge to its expectation value for an increasing sample size and hence proved useful for the MMD metric in ProtoDash according to [101].

2.6 Application Scenarios

This thesis focuses on scarcely examined and challenging unstructured environments. The combination of LiDAR and camera systems represents the state-of-the-art visual perception system for autonomous off-road vehicles in these environments. Hence, the proposed methods are pri-

marily demonstrated in the perception of unstructured environments with regard to their application on autonomous off-road vehicles.

2.6.1 AI for Defense

The Fraunhofer IOSB chairs the Fraunhofer Group for Defense and Security (Fraunhofer VVS). Fraunhofer VVS proposes seven Grand Defense-Technological Challenges for the post-2020 defense research, starting with AI and Autonomy that is "expected to become crucial military force enablers in the mid-2020s"[13]. The major impact of ML in defense applications is seen in equipping defense systems with autonomous navigation capabilities. Hence, the perception methods proposed in this thesis can target the dwindling soldier numbers and allows humans to remain outside of dangerous environments.

2.6.2 Application Environments

Kolski et al. [156] describe structured environments as "depending entirely on such structure being present in their surroundings" [156, p. 1]. Structured environments contain controlled, clearly separable topological objects and many smooth surfaces. Contrasting this, unstructured environments "ignore any structure that exists" [156, p.1]. The DARPA Urban Challenge [27] characterizes unstructured environments as "free-navigation zones" where no restrictions for the planned path are given except obstacle avoidance. Touati et al. [270] separate structured from hostile environments, which are highly susceptible to system failures and restrict the possibility of human intervention.

2.6.3 Robotics, Autonomous Systems, and Planning

The work of Thrun et al. [268] on probabilistic robotics undoubtedly ranges among the most popular works in robotics research. Sensor modeling for robotic applications according to [268] is discussed in Section 2.3.1.

[13] Fraunhofer VVS: Grand defense-technological challenges for Europe post-2020, http://publica.fraunhofer.de/documents/N-521471.html, access on 03.12.2021.

Furthermore, state estimation, robot motion, localization and mapping, as well as decision processes are detailed in [268].

Different types of information architectures are possible in perception. The object-oriented information architecture for perception proposed in [88] is laid out for autonomous system applications and generates a world model similar to the localization and mapping element of the algorithm toolbox (ATB) [60]. Differing from the ATB system architecture [60] and the fusion methods for raw sensor data discussed in Section 2.4.3, Gheta et al. [88] propose an object-oriented high-level modeling of the environment. However, this is rather unsuitable for workspace monitoring in unstructured environments requiring a point-wise 3D reconstruction of the environment [60].

Forkel et al. [71] propose a probabilistic terrain estimation approach and demonstrate that a clear separation of the passable terrain and obstacles optimize autonomous driving in unstructured off-road environments. 3D point clouds from a Velodyne Alpha Prime LiDAR with 128 diodes, a horizontal FoV of 360°, a vertical FoV of 40°, and a semantic segmentation with two classes (terrain and obstacles) constitute the input data to the proposed method. The authors [71] state that temporal accumulation and spatial smoothing with a maximum posterior estimation of the terrain surface facilitate a superior separation into passable terrain and obstacles, and demonstrate that an additional obstacle mapping on the basis of an occupancy grid leads to notably increased performance in terrain estimation.

Typical requirements for the driving behavior of autonomous vehicles in structured, urban environments include stopping by pedestrian crossings and lane keeping [173]. Planning is often split into longitudinal and lateral planning. However, this cannot be transferred into planning for off-road driving in unstructured environments as it focuses on passable terrain, obstacle avoidance, and deviations being as small as possible from the shortest path.

Meyer and Filliat [194] divide planning in discretized space into discretization of the search space, path and universal plan computation, and the final learning of the universal plan. Discretization can be integrated into the mapping step, and an already discretized map may constitute the input for planning [215]. Here, a metric map is well suited for the

following search-based planning step [194, 215]. Zafar and Mohanta [306] state that planning with potential fields presents the possible risk of local minima in gradient descent, and cost functions in search-based planning can achieve globally optimal solutions. Petereit [215] demonstrates that search-based planning is more favorable in unstructured environments. Benchmark approaches for path planning compare the time required for plan computation, the resulting path length, the smoothness of the plans, and the clearance and success rate of planning. Cohen et al. [41] show a comparison of different motion planning methods in indoor environments on the basis of these criteria.

Multifunctional utility logistics equipment transport (MULE) or a human driver that leads a group of multiple vehicles (Convoying) provide the opportunity to send fewer humans into dangerous zones. Here, the avoidance of static obstacles and dynamic objects as well as the passing of difficult passages, such as sharp turns and roadblocks, is required. In contrast to classic waypoint navigation that typically requires a human operator to specify the waypoints, MULE and Convoying avoid this manual intervention and present the two main application scenarios in autonomous transport within the scope of the European Land Robot Trial (ELROB)[14].

2.6.4 Datasets

Perception for unstructured environment struggles with the limited availability of data for test and verification that is needed for both classic and ML approaches. Datasets for autonomous driving in structured environments and in agriculture can provide a base for the pre-training of neural networks and the testing of classic approaches. Nevertheless, data from the targeted unstructured environments is needed and in particular deep ANNs require huge data volumes to reach their outstanding performance.

Many datasets exist for structured environments, e.g., in cities or on motorways. Geiger et al. [83] present the KITTI 2012 dataset with color

[14] ELROB 2018: Transport–MULE, Transport–Convoying: https://www.elrob.org/files/elrob2018/Transport_Mule_V3.pdf, https://www.elrob.org/files/elrob2018/Transport_Convoy_V3.pdf, access on 06.12.2021.

and grayscale stereo images, Velodyne HDL-64E 3D LiDAR point clouds, a GPS/IMU inertial navigation system, and extrinsic sensor calibration information. The baseline for the grayscale and RGB stereo camera systems is 0.54 m, and disparity images with 1241×376 px and $90° \times 35°$ opening are provided. The HDL-64E yields an accurate 3D reconstruction that also serves as ground truth for disparity images and is annotated with 3D bounding box tracklets for object detection. Generally, the KITTI Benchmark demonstrated that state-of-the-art algorithms that perform very well in controlled laboratory conditions often yield below-average results in real-world scenarios [83]. KITTI 2015 [191] extended KITTI 2012 with 400 additional scenes and focuses on the scene flow estimation in street scenes. The SemanticKITTI dataset [11] provides point-wise labels for the 3D LiDAR clouds of the KITTI Vision Odometry Benchmark [82] that segment spherical 2D projections of 3D clouds, as discussed in Section 6.1. It contains 2D label maps with 28 classes to train and evaluate CNNs for the semantic segmentation of 3D point clouds, where 19 classes are static and well-defined.

Cityscapes [43] contains video data, stereo images, GPS, ego-motion odometry data, and temperature from 50 cities. Different seasons, lighting conditions, and daytimes are covered. Eight semantic classes are provided for the 2D images on pixel- and instance-level. Berkeley DeepDrive [305] comprises video sequences from more than 1100 driving hours and includes inertial measurement data, GPS, and timestamps. Pixel- and instance-level annotations, drivable area information, and 2D bounding boxes are available for typical road objects, such as buses or cars. Like Cityscapes [43], DeepDrive neither includes 3D point clouds from LiDAR nor ground truth data for stereo image disparity estimation. Geyer et al. [86] present the A2D2 dataset with 2D images and 3D point clouds from 360° LiDAR sensors captured in structured, urban environments. Annotations include 3D bounding boxes and labels for semantic and instance segmentation. The nuScenes dataset [29] also provides training and test data for urban driving scenarios with 3D bounding box labels for 23 object classes. The Waymo Open Dataset [263] contains 1150 scenes with synchronized 3D LiDAR and camera data from urban and suburban environments. 2D bounding box annotations are provided for the images and the LiDAR points clouds are annotated with 3D bounding boxes.

The geospatial H3D dataset by Kölle et al. [154] provides perception data from aerial laser scanning and textured 3D meshes from multi-view stereo image disparity estimation captured with unmanned aerial vehicles. However, the notably different bird's eye perspective of the captured data compared to the common perspective and FoV in the perception for off-road vehicles impedes its application here.

In contrast to structured environments, datasets captured in unstructured environments are very rare. Metzger et al. [193] introduce the TAS500 dataset that provides training and test data for the semantic segmentation of 2D images from unstructured environments. Fine-grained vegetation and terrain class distributions facilitate the training of CNNs to differentiate drivable surfaces and natural obstacles. The authors [193] demonstrate that a subdivision can increase the prediction accuracy in semantic 2D segmentation into fine-grained semantic classes. The DeepScene dataset [278] was captured in forest environments with a small mobile robotic platform. It provides multi-spectral image data and depth ground truth from a BumbleBee2 stereo camera subject to the accuracy limitations discussed in Section 3.8. Pixel-wise ground truth labels are available for obstacle, void, sky, trail, grass, and vegetation.

The FieldSAFE dataset [157] provides data from LiDAR, multi-spectral cameras, radar, and localization, and contains data from one agricultural field and annotations in 2D bird's eye view for obstacle detection in agriculture. The Sugar Beets 2016 dataset [36] includes a multi-spectral camera, LiDAR, RGB-D camera, and localization data from agricultural environments. The FoV of the LiDAR sensor and the cameras does not correspond, and a depth ground truth for stereo image disparity estimation is available from the RGB-D camera with accuracy limitations, according to Section 3.8. The SemanticUSL [137] and the RELLIS-3D dataset [138] were recorded in structured and unstructured off-road environments at the Texas A&M University with an Ouster OS1-64 LiDAR with a vertical FoV of 45° on a Clearpath Robotics Warthog platform. SemanticUSL consists of 16578 unlabeled scans and 1200 scans labeled according to the 19 static SemanticKITTI classes, while the class structure of RELLIS-3D and SemanticKITTI differ. RELLIS-3D contains 3D LiDAR point clouds from Ouster OS1-64 and Velodyne Ultra Puck 32, 3D point clouds from two stereo cameras, and RGB images from two cameras.

The Middlebury datasets [119, 239–241, 243] ranged among the first synthetic datasets for stereo image disparity estimation with a dense ground truth. The Middlebury Stereo Evaluation provides a benchmark on selected synthetic indoor images with favorable textures and close objects. Mayer et al. [187] propose the Blender-rendered FlyingThings3D, Monkaa, and Driving datasets with accurate and dense ground truth to train CNNs in disparity, optical flow, and scene flow estimation. The authors [187] find that the fine-tuning of DispNet on KITTI 2015 with small disparity values optimizes the performance on this dataset but degrades performance on other datasets with a higher disparity range. It is assumed that training with a small disparity range decreases the ability to correctly predict large disparity values on other datasets [187].

3 Theoretical Foundations

This chapter elaborates on the theoretical foundations such as the targeted sensors, data representations, accuracy constraints, and basic data processing concepts for 2D and 3D data. Unless described otherwise, the perceived sensor data in this thesis is interpreted as "single-shot" scenes that are captured at the same time and with a sufficiently accurate sensor synchronization. Sequences of multiple scenes with course-of-time information are not explicitly considered. Exceptions are, e.g., the temporal consistency analysis of sensor data. Furthermore, the availability of a sufficiently accurate localization of the off-road vehicle is assumed for all perception methods where required.

3.1 Sensor Systems and Data Representation

LiDAR sensors yield accurate but sparse 3D point clouds, while camera images and stereo camera 3D reconstruction provide a dense, colored representation of the environment with limited geometrical accuracy.

3D data provides geometric, shape, and scale information [99], and the most primitive representation of 3D data is a point in Euclidean space [234]. 3D point clouds are the most common representation of 3D data and summarize a large number of 3D data points, while other options are depth images, meshes, or volumetric grids [99]. The spatial relations to other points are not naturally represented by their order in the data structure. In general, point clouds preserve the original 3D information without any discretization that mostly comes with information loss. Consequently, 3D data is represented, saved, and processed in the form of 3D point clouds in this thesis.

Active sensors systems, such as LiDAR, radar, or ToF cameras, illuminate the environment and are mostly independent to daytime, respec-

tively artificial light. Distance measures r are obtained via the time t an actively emitted signal requires from emittance to reception

$$r = \frac{c \cdot t}{2}. \tag{3.1}$$

LiDAR sensors are the most important active perception sensors for off-road vehicles in unstructured environments, so that all finalists in the DARPA Urban Challenge used LiDAR sensors as primary sensors for 3D information[1] [27]. They provide highly accurate distance measures due to the controlled emittance of laser pulses and an accurate detection of the reflected signals. The combination of several beams inside one LiDAR can retrieve an accurate, sparse 3D reconstruction of the environment. Radar sensors are often present in autonomous vehicles in structured environments. However, radar is seldomly used for off-road vehicles as the little presence of metallic structures in unstructured environments is not favorable for its measurement principle. Velodyne LiDAR sensors with 16, 32, 64, or 128 beams, for instance, achieve up to $\pm$ 2 cm accuracy between 1 m and 100 m distance from the respective sensor as well as a minimum angular resolution of 0.1° and 0.4°[2]. Rotating 3D LiDAR sensors capture non-dense 3D point clouds with a horizontal FoV of 360° and yield unordered clouds, as they do not correspond to the matrix structure of images in contrast to ordered, dense stereo camera clouds. Rotating 3D LiDAR sensors feature different vertical FoVs and scan patterns due to their number of diodes, while solid-state LiDAR sensors capture a limited FoV similar to stereo camera systems. However, they cannot provide a horizontal FoV of 360° with one sensor, and are currently unable to replace rotating 3D LiDAR sensors. The particular scan patterns of solid-state LiDAR sensors most probably require the application of cross-source registration methods for their registration to rotating LiDAR sensors. Subsequently, the term LiDAR will refer to rotating 3D LiDAR sensors for clarity. The technical performance of

[1] https://velodynelidar.com/blog/hdl-64e-lidar-sensor-retires/, access on 24.11.2021.

[2] Velodyne LiDAR: Puck: https://velodynelidar.com/products/puck/, access on 06.12.2021; HDL-64E: https://web.archive.org/web/20210811205736/https://velodynelidar.com/products/hdl-64e/, access on 06.12.2021.

LiDAR sensors is defined by different parameters[3]: detection range, FoV, scan pattern, immunity to crosstalk, detection rate, multiple returns, and range precision/accuracy, allowing a first, quantitative estimate of the global sensor performance without a prior analysis of the captured data. The detection range is influenced by the internal sensor properties, by external influences, such as snow or sunlight, and the captured objects. The immunity to crosstalk can be optimized by synchronizing LiDAR rotations for multiple LiDAR sensors. The reception of more than one return is possible due to the active nature of LiDAR sensors. The capability to receive multiple returns leads to a higher resolution of the 3D point cloud, and partially obstructed objects can be reconstructed in some cases. Finally, range precision and accuracy provide a good initial estimate of the measurement quality of a LiDAR sensor similar to the depth estimation accuracy ε_z in stereo camera systems (see Equation 3.17).

Passive sensors comprise all types of passive cameras such as RGB, grayscale, multi- and hyperspectral cameras and depend on sufficient ambient – natural or artificial – light. Camera systems deliver color information within a pixel-wise, dense representation of the perceived environment. 2D images contain the respective luminous intensity values in the cells of an ordered matrix structure, and images with multiple spectral channels are typically saved with one matrix per channel. Stereo image disparity estimation yields depth information for each 2D pixel if the distance of the two optical centers of the stereo camera setup is known, whereby disparity measures the position offset between the representation of a specific 3D point of the surroundings in two different image planes. Stereo 3D reconstruction yields ordered, dense 3D point clouds that can also be referred to as 3D images due to their structural similarity to 2D images. However, their depth estimation accuracy decreases quadratically with increasing distance, as discussed in Section 3.8. Light field imaging captures a 4D representation with two spatial and two angular coordinates, or the space of non-occluded rays within a scene [134]. Uhlig et al. [275] interpret light field cameras as camera arrays and propose surface reconstruction and disparity estimation sim-

[3] M. Müller: Understanding LiDAR Parameters and Technical Specifications, https://www.blickfeld.com/blog/understanding-lidar-performance/, access on 25.01.2022.

ilar to stereo image disparity estimation. Advantages in the approach of [275] are the possibility of digital post-capture focusing and synthesizing new viewpoints. Typical applications of light field cameras include close-range, indoor applications, such as automated optical inspection or face and gesture recognition. Uhlig et al. [276] state that the calibration of light field cameras is typically model-dependent, and present a model-independent calibration approach to overcome this. Nevertheless, the complex processing of the geometric structure of light fields limits their application potential for the perception in unstructured, off-road environments.

Active camera systems are independent of the captured scene's illumination compared to stereo camera systems [107, 308]. The group of active camera systems includes cameras for structured illumination, ToF cameras, and the extension of light field techniques to ToF cameras, as discussed in [134]. Depth estimates of active cameras are currently less accurate than 3D measurements of (rotating) LiDAR sensors. Jayasuriya et al. [134] state that ToF cameras measure the length of the optical path traveled by amplitude-modulated light. The amplitude modulated signal is analyzed by various devices, including photogates and photonic mixer devices, and advantages are a high density of the depth measurements compared to LiDAR sensors. The operation with a single frequency is disadvantageous and often subject to measurement inaccuracies from phase wrapping ambiguity and multi-path interferences due to scattering or translucent objects [134]. ToF cameras conduct active measurements in the NIR spectral range. They are hence subject to measurement errors in outdoor environments due to the NIR spectrum of daylight. Furthermore, their comparatively low resolution leads to depth estimation errors close to the edges due to the mixed pixel effect [218].

The complementary sensor characteristics of active LiDAR sensors and passive camera systems can detect and intercept failures of individual sensors or systematic measurement errors in difficult environmental conditions, such as rain or snowfall. Furthermore, the data fusion of multiple, complementary sensors can notably improve the measurement data accuracy.

3.2 Sensor Poses and Transformations

Sensor calibration determines the relative transformation between two coordinate systems. The extrinsic calibration of multi-sensor systems relates each sensor to the vehicle body frame $\mathcal{B}$. The mathematical representation of transformations in this thesis is oriented towards the recommendations of [292]: a matrix $\mathbf{T}$ that transforms a point $\mathbf{p}^{\mathcal{S}}$ in the sensor frame $\mathcal{S}$ to the vehicle body frame $\mathcal{B}$ is denoted as $\mathbf{T}_{\mathcal{S}}^{\mathcal{B}}$. The affine transformation matrix $\mathbf{T}$ transforms each point $\mathbf{p}$ from $\mathcal{S}$ to $\mathcal{B}$ with

$$\mathbf{p}^{\mathcal{B}} = \mathbf{T}_{\mathcal{S}}^{\mathcal{B}} \cdot \mathbf{p}^{\mathcal{S}}. \tag{3.2}$$

This thesis only uses right-handed coordinate systems, and sensor calibration consists of rigid transformations to map a vector $\mathbf{x}'$ into a new coordinate system that is represented by a combination of rotation $\mathbf{R}$, and translation $\mathbf{t}$ with:

$$\mathbf{x}' = \mathbf{R} \cdot \mathbf{x} + \mathbf{t}, \tag{3.3}$$

where $\mathbf{t}$ is represented as a 3D vector in Euclidean space. The representation of rotations in Euler angles is error-prone as they are neither unique nor continuous. Euler angles do not adhere to the constraints of the Euclidean space, and the gimbal lock problem can introduce singularities. Singularity-free representations of rotations are 3×3 rotation matrices $\mathbf{R}$ or quaternions $\mathbf{q}$, and only an over-parameterization of the rotation leads to a singularity-free representation. However, an over-parameterization leads to the loss of one determinable parameter during the optimization step. The application of Lie groups provides a well-established strategy to circumvent the problem of over-parameterization by transferring the rotations $\mathbf{R} \in SO(3)$ into a local Euclidean space according to [102]. Lie groups are smooth manifolds, and each N-dimensional manifold $\mathbf{M}$ embedded in $\mathbb{R}^n$ with $n \leq N$ has an n-dimensional tangential space for each point $\mathbf{p} \in \mathbf{M}$. Hence, each $\mathbf{p}$ on a $\mathbf{M}$ has a local Euclidean space [117]. A rigid transformation is a pose $\mathbf{T} \in SE(3)$ and $\mathbf{T}$ can be split into a rotation $\mathbf{R} \in SO(3)$ and a translation $\mathbf{t}$:

$$\begin{pmatrix} \mathbb{R}^{3\times3} & \mathbf{t} \\ 0_{1\times3} & 1 \end{pmatrix}. \tag{3.4}$$

Consulting the theorem of von Neumann and Cartan [117], $SO(3)$ is a subgroup of $GL(3, \mathbf{R})$ and a Lie group. Furthermore, $\mathbf{T} \in SE(3)$ can be

represented as a 4×4 matrix and is isomorph to a subset of $GL(4, \mathbf{R})$. Hence, $SE(3)$ is also a smooth manifold and a Lie group. Intuitively, a manifold facilitates considering the infinitesimal small neighborhood of a point $\mathbf{p}$ as a flat, Euclidean space within its corresponding tangential space. In conclusion, $\mathbf{T}$ globally lies within $SE(3)$, a 12-dimensional manifold and a semi-direct product of the groups $SO(3)$ and $\mathbb{R}^3$. Locally, the corresponding Lie algebra $\mathfrak{so}(3)$ maps the local structure of an $SE(3)$ pose onto a Euclidean space. This concept, in-depth discussed in [117], allows a singularity-free optimization of transformations $\mathbf{T}$ during the estimation of an optimal registration result by transferring the optimization process onto a locally Euclidean structure using the Lie group concept.

The registration approaches discussed in Section 4.2 and 4.3 assume a singularity-free transformation representation and optimization. This assumption holds in all evaluated cases as a rough initial estimate is always provided for the optimization within the presented registration methods. The optimizations, which determine the relative orientations, operate on a small and delimited range around the identity, guaranteeing a sufficient distance to the non-unique and non-continuous operating range if Euler angles are used for rotation representation.

3.3 Camera Calibration and Stereo Vision

The width of 2D images is represented by x, the height by y, and the depth z is consequently negative in right-handed coordinate systems. The 2D image frame – the sensor frame for all cameras – is specifically marked as 2D for images with $x^{2D} \times y^{2D}$ pixels, and 3D points are denoted as $\mathbf{p}^{3D}$ to distinguish between the sensor coordinate systems where useful.

The intrinsic camera calibration relates natural camera units – pixels – to metrical units. Both, intrinsically calibrating a single camera and calibrating stereo camera systems, rely on well-proven tools, such as OpenCV[4] in combination with chessboards. Intrinsic calibration with OpenCV assumes a pinhole camera model with plumb bob distortion that

[4] Camera calibration With OpenCV: https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, access on 28.12.2021.

jointly models radial and tangential distortion. It yields a 1D distortion matrix and a camera calibration matrix $\mathbf{K}$ according to [26] with

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.5}$$

where f_x, f_y are the focal lengths in pixel coordinates, while o_x, o_y describe the optical center for lens cameras.

Stereo camera calibration[5] yields a rectification matrix and a projection matrix for each camera as well as the perspective transformation matrix. Image rectification ensures the parallel alignment of the optical axes for stereo image disparity estimation. Here, $\mathbf{P}$ specifies the camera matrix of the processed, rectified images. The 2D projection of a 3D point $\mathbf{p}^{3D}$ in the camera frame onto the rectified image pixel $[j, k]$ is calculated with

$$\begin{pmatrix} j \\ k \\ w \end{pmatrix} = \mathbf{P} \cdot \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} f'_x & 0 & o'_x & \mathbf{T}x \\ 0 & f'_y & o'_y & \mathbf{T}y \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}. \tag{3.6}$$

where $x_c = j/w$ and $y_c = k/w$ describe the projections for both images of the stereo camera pair. The fourth column of $\mathbf{P}$ relates the position of the optical center of the second camera to the optical center of the first camera with $\mathbf{T} = \mathbf{1}$ for the reference camera and $\mathbf{T}y = 0$, $\mathbf{T}x = -f'_x \cdot B$ for the second camera of a horizontal stereo camera setup with baseline B. Summarizing, $\mathbf{K}$ contains the intrinsic camera parameters for distorted, raw images, and $\mathbf{P}$ is applied to project 3D points in the camera coordinate system to rectified 2D pixel coordinates.

The triangulation of 2D images in stereo vision estimates two matching points x_1 (reference image) and x_2 (second image) by solving the correspondence problem. The distance r between the image points x_1 and x_2 corresponding to an equivalent real-world point is denoted as disparity $d = \|x_1 - x_2\|$ for rectified images. The corresponding depth z is calculated with

$$z = \frac{f \cdot B}{d}. \tag{3.7}$$

[5] Robot Operating System, StereoCalibration: http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration, access on 28.12.2021.

Stereo camera systems rely on the photometric content of a scene, and their accuracy depends on specific contents of the captured scene. Here, poorly textured regions and repetitive patterns often lead to less accurate stereo image disparity estimates [218]. Furthermore, correspondences can only be estimated up to a certain accuracy due to repetitive and ambiguous texture, blur, and other unfavorable image characteristics. The image resolution defines the area corresponding to the estimated z. With increasing z, the area covered by one pixel increases quadratically, as discussed in Section 3.8.

A single-channel disparity image with pixels $[j,k]$ can be projected into 3D space with W, X, Y, and Z from the perspective transformation matrix $\mathbf{Q}_p$ (see Section B.1.1):

$$\begin{pmatrix} x([j,k]) \\ y([j,k]) \\ z([j,k]) \end{pmatrix} = \begin{pmatrix} \frac{X}{W} \\ \frac{Y}{W} \\ \frac{Z}{W} \end{pmatrix}. \tag{3.8}$$

Knowing B in a horizontal stereo camera setup, both clouds are transformed into the same coordinate system. Outlier filtering with a nearest neighbor search, similar to point cloud preprocessing in Section 4.2.1, and LRC in stereo post-processing can remove streaks created by disparity estimation errors.

3.4 Principle Component Analysis

Principle Component Analysis (PCA) summarizes a dataset of n dimensions with p data points into an $m \times m$ data matrix $\mathcal{X}$ with $m < n$ dimensions. For this, PCA determines the linear combination of the columns of $\mathcal{X}$ to represent a dataset in m dimensions with maximum variance using the linear combinations

$$\sum_{j=1}^{m} a_j \mathbf{x_j} = \mathcal{X}\mathbf{a}. \tag{3.9}$$

The variance of the linear combination is calculated as $var(\mathcal{X}\mathbf{a}) = \mathbf{a}^*\mathbf{S}\mathbf{a}$ for $\mathbf{a} = \{a_1, \ldots, a_m\}$ with the constants $a_1, \ldots, a_m$. $\mathbf{S}$ is the $m \times m$ sample covariance matrix of the dataset. This reduces PCA to the solution of an eigenvector/eigenvalue problem $\mathbf{S}\mathbf{a} - \lambda\mathbf{a} = 0$, identical to $\mathbf{S}\mathbf{a} = \lambda\mathbf{a}$,

with eigenvectors $\mathbf{a}$ and $\lambda_{\mathbf{a}}$ describing the corresponding eigenvalue of the covariance matrix $\mathbf{S}$. The eigenvalues are the variances of the linear combinations $var(\mathcal{X}\mathbf{a}) = \lambda$ [141]. Covariance matrices of size $m \times m$ are real symmetric with m real eigenvalues $\lambda_{\mathbf{k}}, k \in 1, \ldots, m$. The corresponding eigenvectors can be defined as an orthonormal set of vectors to form a centered m-dimensional coordinate system for the representation of the dataset. Applying a Lagrange multiplier approach as proposed by Jolliffe et al. [141] guarantees the uncorrelatedness of the principal components $\mathcal{X}\mathbf{a}$.

PCA allows dimension reduction while keeping a preferably high volume of information [141] by maximizing the variance during the selection of the uncorrelated principal components. It constitutes a basic, exploratory tool for data analysis and can be obtained from the singular value decomposition (SVD) of a data matrix [91, 141]. The principal components are linear combinations of the original variables. This thesis applies PCA to analyze values with identical measurement units and scale; problems due to different units in variance calculation do not arise [141]. Applications for PCA in this thesis include surface orientation estimation in 3D point clouds and dimension reduction from 3D to 2D.

A dimension reduction of higher dimensional data such as 3D point clouds into 2D or 1D facilitates a faster analysis. For dimension reduction from 3D to 2D with PCA, the first and the second principal component compose the axes of the new 2D coordinate system and the analyzed 3D data is transformed into 2D by mapping onto the two new axes.

3.5 Analysis of 2D Image Data

Numerous measures are known to analyze the quality and information content of 2D image data. Different quality measures to measure the information content (IC) for 2D images or image patches are presented and analyzed in this thesis:

- Image contrast: Shannon entropy,
- Difference of Gaussians (DoG),
- Histogram of Oriented Gradients (HOG), and
- Scale-Invariant Feature Transform (SIFT), Speeded up Robust Features (SURF), Features from Accelerated Segment Test (FAST).

The *IC* can be applied to filter image patches that are detrimental in the training process of neural networks instead of helpful, such as image patches with very low texture information in stereo image depth estimation.

Shannon [254] proposes an entropy measure H to assess the *IC* of a message in the context of signal processing that is based on the probability $p(i)$ of a symbol $i \in I$ to be present inside a message with $N_I = \#I$ the number of possible symbols. The information content of 2D and 3D data can be analyzed similarly to the *IC* of a message. This thesis transfers the Shannon entropy in the image domain to assess the *IC* captured in 2D images [326, 327]. The *IC* of 2D image data (IC_{r2D}) is contained in the luminous intensity values of the captured spectral channels. Thus, the *IC* of a defined 2D pixel grid can be measured by its Shannon entropy. In 8-bit images, I contains all possible, discretized intensities $I \in \{0, 1, \dots, 255\}$. Hence, the Shannon entropy $H(m)$ of an image patch m with $N \times N$ pixels that fulfill $[i, j] \in m$ assesses the information in m:

$$H(m) = -\Big(\sum_{N_I} p(I_{[i,j]}) \cdot \log_2(p(I_{[i,j]})) \Big). \tag{3.10}$$

Here, m is represented by a set of intensity values $I[i, j]$ that yields one $I[i, j]$ for each pixel $[i, j]$ in grayscale images. The probability p of the intensity value $I[i, j]$ to be contained in the set $\{I[i, j]\}_{i \in \{1,\dots,N\}, j \in \{1,\dots,N\}}$ is $p(I[i, j])$ and counts the relative frequency of each $I[i, j]$. N_I denotes the number of different intensity values in m. A high Shannon entropy indicates a high *IC* and thus higher differences in between the pixel's intensity values. In the case of 9×9 image patches ($N = 9$), the maximum entropy is $H(m) = 6.340$. $H(m)$ proved useful to measure the contrast of the image patch, as demonstrated in Section 6.2.1.

Marr et al. [184] utilize DoG, respectively Gaussian blurring with two valid blurring radii and a subtraction of the two blurred images, to visualize edges inside images. DoG extracts edges on locally-connected pixels with notable changes of the intensity values compared to their neighboring pixels, and measures the intensity differences inside the image. It can hence also indicate a high *IC* as a high difference in intensity represents rich textures. HOG features proposed by Dalal et al. [45] also detect edges on the basis of their respective gradients saving their orientation for applications such as object detection by comparing the

HOG representations of the assessed image and its HOG model. The authors [45] state that HOG features are well-suited to visual object recognition and outperform existing feature sets in the human detection test case. An implementation is included in the SciKit-Image library[6]. SIFT [176], SURF [10], and FAST [231] are gradient-based descriptors which often occur in highly textured image areas and a high number of detected SIFT, SURF, and FAST features can indicate a high *IC* of an image patch similar to the Shannon entropy.

3.6 Analysis of 3D Point Cloud Data

Geometric 3D information can be analyzed in a 3D representation [326] or after a dimension reduction. While various different metrics exist, geometric structure and point density present two measures of major importance and applicability [323, 329].

Points in 3D point clouds represent point sets on real surfaces, and their structure can be described using their surface variation. This thesis opts for the surface variation calculation described in [234]. The surface variation around each point $\mathbf{p}_i$ inside a point set $\mathbf{P}^k$ is determined using the eigenvalues of the covariance matrix $\mathbf{C}$. Hence, the surface variation s_i for a point $\mathbf{p}_i$ is equal to its curvature. The surface variation for $\mathbf{p}_i$ can be formulated as

$$s_i = \frac{\lambda_3}{(\lambda_1 + \lambda_2 + \lambda_3)}, \quad \lambda_3 = \min(\lambda_j), \quad j \in \{1, 2, 3\}. \tag{3.11}$$

The surface variation s_i is calculated for each point $\mathbf{p}_i$ and indicates the structured or unstructured character of a point cloud. To combine the values s_i of a point set $\mathbf{P}^k$, the empirical mean of s_i for all points $\mathbf{p}_i \in \mathbf{P}^k$, denoted $\overline{s}$, is calculated:

$$\overline{s} = \frac{1}{k} \sum_{i=1}^{k} s_i. \tag{3.12}$$

The surface variation is scale-invariant. A small $\overline{s}$ indicates that all points in the neighborhood $\mathbf{P}^k$ are close to the plane tangent to the surface. Struc-

[6] SciKit-Image library: https://scikit-image.org/docs/dev/api/skimage.measure.html, access on 04.11.2021.

tured environments contain a high number of smooth surfaces. Thus, a low $\overline{s}$ indicates a rather structured character of the perceived environment, while unstructured environments are typically characterized by a high $\overline{s}$.

Surface estimation can directly derive the surface normals from the point cloud data [234]: an SVD of the covariance matrix of a point set converts the normal estimation problem to a least-squares plane fitting estimation problem and determines a normal vector estimate for each point as a plane tangent to the estimated surface [14, 253]. The covariance matrix of a 3D point set P^k, $P^k = \{\mathbf{p}_1, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_k\}$, is represented as real, positive semi-definite, symmetric matrix $\mathbf{C} \in \mathbb{R}^{3\times3}$ according to [234]:

$$\mathbf{C} = \frac{1}{k}\sum_{i=1}^{k}(\mathbf{p}_i - \overline{\mathbf{p}}_i)(\mathbf{p}_i - \overline{\mathbf{p}}_i)^T. \tag{3.13}$$

3D planes can be described uniquely by a point $\mathbf{x}$ lying inside the plane and a normal vector $\mathbf{n}$ that indicates the surface orientation. Least-squares plane fitting aims at minimizing the distance $r = (\mathbf{p}_i - \mathbf{x})\mathbf{n}$ of a point $\mathbf{p}_i \in \mathbf{P}^k$ to a plane via $\overline{\mathbf{p}} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{p}_i$. An Eigenvalue decomposition of C is possible as covariance matrices are naturally quadratic and $\overline{\mathbf{p}}$ can be solved by an eigenvalue decomposition of $\mathbf{C}$:

$$\mathbf{C}\mathbf{v}_j = \lambda_j\mathbf{v}_j, j \in \{1,2,3\}, \lambda_j \in \mathbb{R}. \tag{3.14}$$

The eigenvectors $\mathbf{v}_j$ are orthogonal and correspond to the principal components of $\mathbf{P}^k$. The eigenvector corresponding to the smallest, non-zero eigenvalue approximates the normal vector $\mathbf{n}$ and solves the first order 3D plane fitting problem.

In addition to the surface variation, the geometric structure of 3D point clouds can be analyzed after a homogenization process that transforms 3D points back into the cylindrical projected coordinates ϕ, r, and z – their coordinate system of origin in the capture process. The proposed homogenization process yields a more uniform point distribution for active, rotating sensors, and, hence, constitutes the most promising representation for cross-source registration [329]:

$$\phi = \arcsin(y/\sqrt{x^2+y^2}), \quad r = \sqrt{x^2+y^2}, \quad \text{and} \quad z = z. \tag{3.15}$$

Furthermore, the point distributions and the total number of points can be applied to analyze the *IC* for point clouds and compare 3D point clouds. The relative point distribution $\overline{\xi}$ within the horizontal FoV of 360° can be determined by a binning of the homogenized points according to ϕ. The ξ_i for bin i analyzes the relative amount of points N_i in bin i in relation to the total number of points N inside a cloud: $\xi_i = N_i/N$.

A high number of bins, e.g., $i = 36$ and 10° per bin, ensures a fine-grained analysis and is especially beneficial if notably different areas are captured in a cloud. The variance $\sigma^2(\xi)$ and the empirical mean $\overline{\xi}$ (see Equation 3.12) can be regarded additionally. A high $\sigma^2(\xi)$ is not favorable as it indicates an inhomogeneous point distribution inside the point cloud. Uniform point distributions and a high number of 3D points illustrate a proper representation of all cloud sectors inside a point cloud (see Section 6.2.1.1). This can indicate a high *IC* favorable for perception methods of all levels, such as cross-source registration or semantic 3D segmentation.

3.7 2D–3D Fusion of Color and Geometry

2D–3D fusion assigns color intensity values to 3D LiDAR points within the camera FoV. It projects the color information from the camera into 3D space by assigning a color intensity value to each 3D point. It requires the availability of an intrinsically and extrinsically calibrated sensor setup with at least one camera and one 3D LiDAR sensor that provides the coordinate transformation between the pixel space and the 3D cloud space. The 3D LiDAR ($\mathcal{L}$) points $\mathbf{p}_{\mathcal{s},\mathcal{L}}$ are transformed into the sensor coordinate system of the camera $\mathcal{c}$ with $\mathbf{p}_{\mathcal{c}} = \mathbf{T}^{\mathcal{c}}_{\mathcal{s},\mathcal{L}} \cdot \mathbf{p}_{\mathcal{s},\mathcal{L}}$, equivalent to the *cc23* registration approach discussed in Section 4.3.2. The projection vectors $\mathbf{v} = [v_i; v_j; 1]^*$ are related to the image coordinate system $\mathcal{i}$. If the image coordinate system is not equivalent to the sensor coordinate system $\mathcal{s}$, a second transformation with $\mathbf{T}^{\mathcal{i}}_{\mathcal{s}}$ is required: $\mathbf{p}_{\mathcal{i}} = \mathbf{T}^{\mathcal{i}}_{\mathcal{s}} \cdot \mathbf{p}_{\mathcal{s}}$. The

projection of the 3D point into the image plane is obtained by reformulating Equation 4.30 for each pixel $[i, j]$ as

$$\begin{pmatrix} v_i \\ v_j \\ 1 \end{pmatrix} = z_{\vartheta,\mathcal{L}}^{-1} \cdot \mathbf{K} \cdot \mathbf{T}_{\vartheta,\mathcal{L}}^{\mathcal{C}} \cdot \begin{pmatrix} x_{\vartheta,\mathcal{L}} \\ y_{\vartheta,\mathcal{L}} \\ z_{\vartheta,\mathcal{L}} \\ 1 \end{pmatrix} = z_{\mathcal{C}}^{-1} \cdot \mathbf{K} \cdot \begin{pmatrix} x_{\mathcal{C}} \\ y_{\mathcal{C}} \\ z_{\mathcal{C}} \end{pmatrix}. \tag{3.16}$$

The color intensity of the pixel that corresponds to the projection of the respective 3D point inside the 2D pixel grid determines its color.

3.8 Accuracy in 2D and 3D Imaging

The accuracy in 2D images and the 2D–3D registration of image and LiDAR data is limited by the area covered by one pixel. This area increases quadratically with linearly increasing distance z, and the real-world resolution in x (width) and y (height) depends on z. For an exemplary JAI AD-130 GE camera with 1296×966 px, one pixel (px) covers 0.8 cm of horizontal length for a horizontal FoV and width of 10 m, and 1.6 cm for a horizontal FoV of 20 m.

Rotating 3D LiDAR sensors have a horizontal FoV of $\tau_h = 360°$. The angular beam resolution τ_v' can be estimated from the vertical sensor FoV τ_v, the number of diodes that form the typical ring structure, and the beam distribution. Furthermore, τ_v', the horizontal angular resolution τ_h', and the maximum measurement range of a LiDAR sensor $\max(r)$ are given in the datasheet. The angular resolutions τ_v' and τ_h' can be converted into metric resolutions $\alpha_v' = r \cdot \sin(\tau_v')$ and $\alpha_h' = r \cdot \sin(\tau_h')$. As a reference, the Velodyne HDL-64E has an average measurement accuracy of ± 2 cm.

Accuracy limits for the 3D reconstruction with stereo camera systems result from the disparity matching process, as discussed in Section 2.3.1. The inherent depth estimation error ε_z in stereo disparity estimation amounts to

$$\varepsilon_z = \frac{z^2 \cdot \varepsilon_d}{B \cdot f}, \tag{3.17}$$

with the disparity estimation error ε_d. The quadratic estimation error ε_z constitutes the most influential degradation in the 3D estimation accuracy of stereo camera systems. The offset and the linear bias in the quadratic

quality assessment modeling proposed by Wolf and Berns [296] have a minor influence on the 3D measurement accuracy, especially for 3D measurements with large depth values, which are hence discarded for the accuracy modeling within this thesis. Disparity estimation errors up to a maximum of three pixels from the reference data (3PE) are accepted as a correct estimation for autonomous vehicles as proposed by Geiger et al. [82]. Consequently, the depth estimation error for a pixel with $z = -2\,\mathrm{m}$ is notably smaller than for $z = -10\,\mathrm{m}$. The stereo camera setup of the IOSB.amp Q1 has two JAI AD-130GE cameras with $1296 \times 966\,\mathrm{px}$, $B = 0.62\,\mathrm{m}$, and $f = 686.85\,\mathrm{px}$. The depth estimation accuracy with $\varepsilon_d = 3\,\mathrm{px}$ equates to $\varepsilon_z = 0.704\,\mathrm{m}$ at $z = -10\,\mathrm{m}$. Hence, the achievable accuracy in depth estimation from stereo disparity estimation is determined by the disparity range and the resolution of the input images, and stereo camera point clouds can contain plate-shaped structures originating from the quantization within per-pixel or per-subpixel disparity estimates. Furthermore, the achievable accuracy in the 3D–3D registration of stereo camera and LiDAR data mainly depends on the depth estimation accuracy from stereo images, and the attainable x^{3D} and y^{3D} accuracy is identical to 2D–3D registration.

3.9 Registration and Multi-Sensor Calibration

Data fusion from multiple sensors into one coordinate system requires their extrinsic calibration – the relative translation and orientation of all sensor coordinate systems $\mathcal{S}_i$. Utilization of the captured sensor data for the environment perception of a robotic system additionally requires the calibration of each sensor to a common vehicle coordinate system $\mathcal{C}$. Hereby, registration designates sensor calibration and the subsequent application of the calibration results to transform, and hence register, the captured sensor data inside one common coordinate system. Accurately calibrating perception sensors is crucial for all further processing steps as an inaccurate calibration notably degrades the data quality and can even result in misleading or dangerous perception data.

An affine transformation $\mathbf{T}_{\mathcal{S}}^{\mathcal{C}}$ transforms each point $\mathbf{p}$ from the sensor frame $\mathcal{S}$ to $\mathcal{C}$: $\mathbf{p}_{\mathcal{C}} = \mathbf{T} \cdot \mathbf{p}_{\mathcal{S}}$. The calibration to $\mathcal{C}$ can be achieved by an

extrinsic calibration (and registration) of all sensors and the subsequent calibration of one sensor to the platform frame.

Two approaches for extrinsic calibration are possible. As a first option, the relative sensor poses can be measured manually. However, this is especially difficult if relative rotations have to be determined or the platform coordinate system lies inside the platform. Alternatively, the captured sensor data can be related with registration methods that yield the relative sensor poses in a semi-automatic or automatic manner. Registration of multi-sensor systems can be formulated as an optimization problem with six DoF limiting the search space for potential solutions to rigid and affine 3D transformations, and with a special focus on a singularity-free representation. Many methods register sensor data within a two-stage process [54, 126, 195, 323]: a coarse initialization ensures that $\mathbf{T}$ is located in the vicinity of the globally optimal solution. A second, fine registration step determines the locally optimal solution, often with Newton-based optimization methods. Visual overlays can generally verify a singularity-free transformation and always provide a qualitative assessment of the registration accuracy (see Figure 4.20). For 2D–3D registration, the projection of a depth image onto an RGB image validates the determined ground truth transformation if their accurate overlay is clearly visible, such as in Figure 4.12. The following assumes that a ground truth reference transformation is available for all use cases. However, visual overlay allows a qualitative evaluation for use cases without a reference transformation. Preprocessing or also a qualitative accuracy assessment for registered 3D point clouds is facilitated by kNN filtering, as proposed in Section 4.2.1.

Similar-source data denotes data captured with sensors using the same measurement principle, such as 3D LiDAR sensors. The direct registration of similar-source point clouds can be applied to extrinsically calibrate sensor systems with multiple similar-source sensors. Cross-source sensor data is obtained from different types of sensors, e.g., 2D images from a passive camera and 3D point clouds from a LiDAR sensor, often provides complementary information due to multimodal sensor data. They exhibit inherently different characteristics due to their different measurement principles discussed in Section 3.1, such as outliers, artifacts, and notably different densities. Hence, 2D–3D registration requires cross-

source registration methods. The registration of a 3D LiDAR point cloud and a 3D point cloud generated with the depth information from stereo disparity estimation also belongs to cross-source registration.

2D–3D registration matches images and point clouds, while 3D–3D registration determines the relative poses of 3D point clouds. The registration of 2D LiDAR sensors as proposed in [69] is not regarded as it is no longer required in off-road applications with the availability of 3D LiDAR sensors.

Data for cross-source registration can also be viewed as data from two different domains. Processing methods for cross-source data want to accomplish domain transfer in a wider sense and aim at a possible domain invariant representation. Cross-source data is characterized by inherently different structures: differences in scale, measurement density, accuracy, noise, and outlier characteristics. Consequently, cross-source 3D data is considerably more difficult to register than similar-source data. A common data representation is required before extracting and matching common features for correspondence matching. For 2D–3D registration, a common representation can be achieved by projecting a 2D image into 3D space, mapping a 3D point cloud onto a 2D depth image, or generating a 3D point cloud from 2D stereo camera images. Naturally, the 3D–3D registration of point clouds is also possible without prior transformation. Transformative methods facilitate a common data representation if data is given in two different dimensions, e.g., the registration of a 2D image to a 3D point cloud. Even for a common data representation, direct registration methods such as ICP [15] are mostly unable to detect sufficiently similar, common features for correspondence matching in cross-source data, and specially crafted and optimized, transformative registration methods are required. Unstructured environments with natural and grown topological structures introduce additional difficulties into the registration process, as highlighted in exemplary images for unstructured environments (see Figure 1.2 and Figure 4.18).

The registration of accurate but sparse 3D LiDAR point clouds to dense but less accurate 3D stereo camera point clouds is an especially challenging registration scenario. For this purpose, the estimated disparity images must be projected into 3D space to generate 3D stereo camera point clouds, as described in Section 3.3. Stereo image depth estimation

inaccuracy can lead to plate-like structures in larger distances, while clouds from rotating 3D LiDAR sensors mostly show a ring structure due to the LiDAR measurement principle. The GICP algorithm likely matches the plate-like structures in point clouds from stereo 3D reconstruction to the estimated smooth, locally planar surfaces in the LiDAR cloud. An ICP registration is likely to match the plate-like structures onto the ring-shaped point sets contained in the sparse LiDAR cloud. Here, the application of transformative methods can provide a more abstract and thus more related representation of the cross-source data that enables an accurate registration process.

3.10 Generalized ICP

Segal et al. [250] derive Generalized ICP (GICP) from ICP as a special case of point-to-plane-ICP. GICP assumes a Gaussian distribution of the points $\mathbf{p}_{s,i}$ and $\mathbf{p}_{t,j}$ in both, source cloud $\mathbf{P}_s$ and target cloud $\mathbf{P}_t$:

$$\mathbf{p}_{s,i} \sim \mathcal{N}(\hat{\mathbf{p}}_{s,i}, \mathbf{C}_i^{P_s}), \tag{3.18}$$

$$\mathbf{p}_{t,j} \sim \mathcal{N}(\hat{\mathbf{p}}_{t,j}, \mathbf{C}_j^{P_t}). \tag{3.19}$$

$\mathbf{C}_i, \mathbf{C}_j$ are the estimated covariance matrices. Consequently, the Euclidean distance r_{ij} between a corresponding set of points $(\mathbf{p}_{s,i}, \mathbf{p}_{t,j})$ is also subject to a Gaussian distribution according to Segal et al. [250] and $\mathbf{T}_{P_s}^{P_t}$ transforms $\mathbf{p}_{s,i}$ to the target frame using the current GICP estimate:

$$r_{ij} \sim \mathcal{N}\left(\mu = 0, \mathbf{C}_i^{\mathbf{P}_t} + (\mathbf{T}_{P_t}^{P_s})\mathbf{C}_i^{P_s}(\mathbf{T}_{P_t}^{P_s})^*\right). \tag{3.20}$$

The current transformation estimate from source to target is calculated with a maximum likelihood estimation

$$\mathbf{T} = \text{argmax}_{\mathbf{T}} \prod_i p\left(r_{ij}\right) = \text{argmax}_{\mathbf{T}} \sum \log\left(p(r_{ij})\right), \tag{3.21}$$

that encodes the probability p of the Euclidean distance r_{ij} and is simplified to

$$\mathbf{T} = \text{argmin}_{\mathbf{T}} \sum_i \left(r_{ij}^{(\mathbf{T})}\right)^* \mathbf{C}_M{}^{-1} \left(r_{ij}^{(\mathbf{T})}\right). \tag{3.22}$$

For optimization, GICP uses the quadratic Mahalanobis distance [180] r_M for N_C corresponding point pairs:

$$r_M = \frac{1}{N_C} \cdot \sum_{i=0}^{N_C-1} \left(r_{ij}\right)^* \mathbf{C}_M{}^{-1} \left(r_{ij}\right). \tag{3.23}$$

The covariance of the Mahalanobis distances r_M is obtained from

$$\mathbf{C}_M = \mathbf{C}_i^{P_t} + \mathbf{T}_{P_t}^{P_s} \mathbf{C}_i^{P_s} (\mathbf{T}_{P_t}^{P_s})^* \tag{3.24}$$

and adds a weighting to r_M in Equation 3.23.

The assumption of a Gaussian distribution of the points inside the clouds integrates the central idea of point-to-plane ICP that all point clouds have an inherent structure. Their surface information is integrated into the registration estimate and improves the registration performance compared to ICP. From the assumption of structured clouds, Segal et al. [250] deduce that real-world surfaces are two-manifolds inside a 3D Euclidean space due to at least piece-wise differentiability, and assume that all surfaces inside the structured clouds are locally planar. In proceeding, the two point clouds are treated as two surfaces being isomorph to each other. Slightly different points are sampled from two different perspectives, and perfect correspondences cannot be reached. However, points provide constraints along their surface normals that are represented inside the respective covariance matrix $\mathbf{C}_i$. In order to model the surface, a high covariance along the estimated, locally planar surface is set in Equation 3.25. Furthermore, a low covariance value ε_C along the direction of the estimated surface normal is assigned to each sampled point. The covariance matrix $\mathbf{C}_M$ is weighted in an inverse manner inside the minimization term given in Equation 3.22, and errors along the surface normal notably increase the error metric, while errors inside the estimated, locally planar surfaces hardly increase the error for a small ε_C.

The surface normals $\boldsymbol{\nu}_i$ and the rotation matrices $\mathbf{R}_{\boldsymbol{\nu}_i}$ correspond to the points $\mathbf{p}_i$ in source or target cloud. The covariance matrices $\mathbf{C}_i$ for the points $\mathbf{p}_i$ are calculated from

$$\mathbf{C}_i = (\mathbf{R}_{\boldsymbol{\nu}_i}) \begin{pmatrix} \varepsilon_C & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}_{\boldsymbol{\nu}_i}{}^*, \quad i \in \{s, t\}. \tag{3.25}$$

The surface normal for a point set is obtained by an SVD of its covariance matrix $\mathbf{C}_i$. Segal et al. [250] compute $\mathbf{C}_i$ as empirical covariance $\hat{\sum}$ of the 20 closest points and independent of the density of the point clouds. The singular vector corresponding to the smallest singular value indicates the direction of the surface normal. The SVD of $\hat{\sum}$ yields the singular vectors inside $\mathbf{U}$ with the corresponding singular values $\mathbf{d}$. As $\hat{\sum}$ is real and symmetric, $\mathbf{U} = \mathbf{V}^*$ holds for the left and right singular vectors of $\mathbf{U}$ and $\mathbf{V}$:

$$\hat{\sum} = \mathbf{UdV} = \mathbf{UdU}^*. \tag{3.26}$$

The covariance is reconstructed to

$$\hat{\sum} = \sum_{i=0}^{2} \left(d_i \mathbf{u}(i) \mathbf{u}(i)^* \right), \tag{3.27}$$

using the column vectors of $\mathbf{U}$, the left singular vectors $\mathbf{u}(i)$, and the corresponding weighting in d_i with $d_0 = \varepsilon_C$, $d_1 = 1$, and $d_2 = 1$ by replacing $\mathbf{d}$ with $\mathrm{diag}(\varepsilon_C, 1, 1)$ according to Equation 3.25.

3.11 Registration Error Metrics and Decalibration

The registration accuracy of the GICP algorithm is commonly measured with the Euclidean fitness score e_{fs} [123]. It represents a benchmarking value calculated from the Euclidean distances $r_{j,i} = \mathbf{p}_{t,(j,i)} - \mathbf{p}_{s,(j,i)}$ of target points ($\mathbf{p}_t$) and transformed source points ($\mathbf{p}_s$) weighted with the number of correspondences N_C:

$$e_{\mathrm{fs}} = \frac{\sum_{i=i}^{N_C} \left({r_{x,i}}^2 + {r_{y,i}}^2 + {r_{z,i}}^2 \right)}{N_C}. \tag{3.28}$$

This thesis applies two different error metrics to assess the registration results for cross-source sensor data: the Frobenius norm F providing an exclusive assessment of the transformation, and the L_2 norm evaluating the registration accuracy of the data. L_2 can yield potentially large errors for corresponding point pairs in major distances depending on the absolute distance values from the sensor origin, while F directly rates the transformation as a registration result. The combination of both metrics provides a comprehensive assessment of the transformation and

the registered data for the subsequent application. Furthermore, F and L_2 prove useful for comparing the performance of different registration methods and combining the results using a confidence-based metric as proposed in the *UCSR* method elaborated in Section 4.3.

F measures the deviation of the estimated registration result $\hat{\mathbf{T}}$ from the reference transformation $\mathbf{T}_{\text{ref}}$ directly [126]:

$$F = \left\| \mathbf{T}_{\text{ref}} - \hat{\mathbf{T}} \right\|_F = \sqrt{\sum_{i=1}^{4} \sum_{j=1}^{3} |\mathbf{T}_{\text{ref},(i,j)} - \hat{T}_{(i,j)}|^2}. \tag{3.29}$$

The last row of $\mathbf{T}$ is omitted due to equivalence in 4×4 transformation matrices. F is vulnerable to the scale of rotation versus translation. For F, the consideration of the translational component in meters primarily leads to F providing an assessment of the rotational components of the registration results as translation errors are relatively small in comparison. In addition, the L_2 norm of corresponding point pairs is used as a second error metric. It is equivalent to the e_{fs} used to assess the registration accuracy in similar-source registration. However, the source point cloud in cross-source registration is transformed with $\mathbf{T}_{\text{ref}}$ to only consider valid corresponding pairs for L_2. The source cloud transformed with $\mathbf{T}_{\text{ref}}$ is then compared to the transformation of the source cloud with $\hat{\mathbf{T}}$. The L_2 norm evaluates the corresponding point pairs after applying the registration result. In this thesis, the analysis of L_2 in meters proved useful for a more intuitive error interpretation which is why the L_2 is defined as follows and in accordance with Eq. (6.7):

$$L_2(\mathbf{T}_{\text{ref}}, \hat{\mathbf{T}}) = \sqrt{\frac{1}{N_C} \sum_{i=1}^{N_C} \left\| \mathbf{T}_{\text{ref}} \mathbf{p}_{s,i} - \hat{\mathbf{T}} \mathbf{p}_{s,i} \right\|_2}. \tag{3.30}$$

As a reference, the L_2 results in the registration of unstructured data can be compared to the L_2 results of CSGM [126] that is the basis for the proposed *graph33* method: the registration of two clouds of the "Stanford Bunny"[7] with artificial noise and 20 % outliers by Huang et al. [126] achieved $\mu(L_2(\text{CSGM})) = 1.792\,\text{m}$.

[7] The Stanford 3D Scanning Repository: http://graphics.stanford.edu/data/3Dscanrep/, access on 03.11.2021.

L_2 and F, respectively $\mu(\ln F)$, measure the registration accuracy, while standard deviation σ and variance σ^2 indicate the stability of the method on several runs in this thesis similar to [55]:

$$\mu(\ln F) = \frac{1}{n}\sum_{i=1}^{n} \ln F_i \qquad \sigma^2(\ln F) = \frac{1}{n}\sum_{i=1}^{n} \big(\ln F_i - \mu(\ln F)\big)^2. \tag{3.31}$$

The L_2 measures $\mu(L_2)$ and $\sigma^2(L_2)$ are calculated accordingly. Low values for $\sigma^2(\ln F)$ and $\sigma^2(L_2)$ indicate a small variation of the registration results, indicating a high stability of the registration method to generate comparable results on several runs of the same scenes for registration.

The training data generation for the CNN-based registration of cross-source data with *cnn23* and *dsm33* utilizes three decalibration levels to describe the applied artificial translations and rotations as demonstrated subsequently. Level S decalibrations transform the input data with up to ±0.25 m in translation and up to ±4.3° in rotation; level M applies up to ±0.50 m and ±8.6°. Level L decalibrations alter the input data with up to ±1.0 m in translation and up to ±17.2° in rotation.

3.12 Filtering Thresholds in Data Analysis

The threshold levels in data analysis and filtering within this thesis were chosen according to the standard normal distribution with $(\mu = 0, \sigma^2 = 1)$ $\mathcal{N}_{0,1}$ on the basis of a normalized score between 0.0 and 1.0. Thereby, the maximum score indicates the best and most beneficial result for the conducted data analysis process. The experimentally justified weak threshold eliminates all data samples that achieve less than 68.27 % ($\mu \pm 1.0\sigma$) of the possible maximum of 1.0/100%, while a medium threshold excludes all samples from further processing that score less than 86.64 % ($\mu \pm 1.5\sigma$) and hence less than 0.8664. The strong filtering threshold proposed in this thesis only keeps samples that achieve an analysis result higher than 95.45 % ($\mu \pm 2.0\sigma$). Comparing all analyzed data samples against the weak, medium, or strong threshold eliminates detrimental samples, as demonstrated in Section 5.3 and 6.2.1.

4 Low-level Perception

Section 4.1 presents a confidence and accuracy assessment of raw 2D and 3D sensor data that facilitates a tightly coupled validation of the captured sensor data independent of its further processing in low-, mid-, or high-level perception. The proposed confidence assessment can be conducted prior to all perception methods proposed in this thesis to detect sensor failures and ensure sufficient accuracy of the perception sensor data.

Section 4.2 and 4.3 propose novel registration methods for the extrinsic calibration of multi-sensor perception systems. The utilization of calibration targets, such as chessboard or retro-reflective tape detailed in Section 2.3.3, were avoided in the unstructured, hazardous environments analyzed in this thesis as they require human intervention to manipulate the environment. Hence, all registration methods in this thesis facilitate sensor data registration without calibration targets and use only the structure of the surroundings in contrast to most state-of-the-art methods (see Section 2.3.3). Section 4.2 presents a semi-automatic registration approach for similar-source sensor data from multiple LiDAR sensors to determine their extrinsic calibration and their calibration to a robotic platform by registration of the corresponding sensor data. The unstructured cross-source registration framework *UCSR* described in Section 4.3 [329] facilitates a confidence-based fusion of registration results from multiple cross-source registration methods. It implies a tightly coupled validation of the individual registration methods optimized for unstructured environments via the confidence-based fusion of the registration results. Herein, *cc23* and *cnn23* present two registration methods for the cross-source registration of 2D and 3D data specially optimized for unstructured environments, as described in Section 4.3.2 and 4.3.3. Furthermore, two customized registration methods for the cross-source registration of 3D data from unstructured environments, *graph33* and *dsm33*, are presented in Section 4.3.4 and 4.3.5.

Finally, Section 4.4 discusses a novel 2D fusion method for intensity information from different spectral bands into one 2D image that can provide a basis for the mid-level perception methods proposed in this thesis, such as disparity estimation from stereo images.

4.1 Sensor Data Confidence

The confidence assessment of raw perception sensor data increases the resilience to measurement inaccuracies and sensor faults and leads to a more accurate understanding of the environment. Filtering of potentially inaccurate sensor data can mitigate a detrimental influence on perception methods, which is especially important for critical applications such as perception for autonomous vehicles, where the sensor data is input to object detection and obstacle avoidance. Each perception sensor yields numerous measurements from one "single-shot" capture, as discussed in Section 2.3.1, and each 2D pixel or 3D point is interpreted as an individual measurement. Consequently, perception for autonomous off-road vehicles requires a point-by-point confidence assessment to lay the foundation for a trustworthy workspace monitoring and driving behavior in unstructured environments.

The confidence assessment approach proposed in this thesis combines local and global confidence measures: local confidence analyzes the probability of measurement errors for individual pixels or points with their per pixel/point confidence (*PPC*), while global measures evaluate full 2D images or 3D point clouds for their global, per-sensor and scene confidence (*PSC*).

The presented confidence measures are generic and independent of the subsequent data processing. They supplement each individual measurement with a confidence estimate similar to the confidence attribution scheme of [244]. Like Wolf and Berns [296], the focus is placed on the confidence analysis for 3D measurements from LiDAR and stereo camera systems. RGB-D cameras were considered as they also rely on disparity matching for depth estimation and are subject to similar accuracy limitations than stereo camera systems. Contrasting Wolf and Berns [296], this thesis analyzes the individual, local and global confidence of 2D pixels and 3D points instead of a voxel model of the environment. A 2D confi-

dence analysis for the stereo images that are input to disparity estimation provides a tightly coupled, pre-modeling validation by deciding upon the execution or omission of disparity estimation. The presented approach can be regarded as a three-step approach:

1. Analysis of 2D image data providing the input for stereo vision,
2. Examine raw 3D point clouds from LiDAR, RGB-D, and stereo camera systems,
3. Generate confidence estimate for each 3D measurement.

In order to preserve the generic and model-agnostic character of the proposed confidence assessment, 3D point clouds from stereo and RGB-D camera systems are regarded as raw sensor data hereinafter, and confidence of disparity images is not analyzed here but in Section 2.4.2. Consequently, raw sensor data comprises individual 2D images input to stereo image disparity estimation and from RGB-D cameras, 3D point clouds from active, rotating 3D LiDAR sensors, as well as 3D point clouds from stereo and RGB-D cameras.

Numerous local and global confidence measures are possible. This thesis analyzes a combination of state-of-the-art and novel, experimentally justified confidence measures on their contribution to an expressive confidence estimate. The confidence measures proposed were designed for a low computational effort facilitating a confidence assessment for each "single-shot" capture of the environment where required. More complex, global confidence measures can be calculated prior to sensing and task execution if applicable or during run time with a lower frequency.

Cameras are sensitive to difficult lighting conditions. The depth estimation accuracy of stereo cameras is inherently limited and depends on the environmental conditions and the captured scene, while LiDAR sensors yield a sparse 3D representation of the environment but reflecting surfaces, such as smooth metallic surfaces or puddles, do not provide measurements. The measurement principle of rotating LiDAR sensors is related to the measurement principle of solid-state LiDAR sensors and ToF cameras, and the presented methods are partially applicable for those active sensor systems. However, confidence for solid-state LiDAR sensors and ToF cameras is not analyzed here as they do not belong to the state-of-the-art sensor setup for the perception of unstructured environments.

4.1.1 Sensor Outage and Temporal Consistency

Sensor outage is analyzed for all previously specified sensors with c_{OT}, a confidence estimate for sensor outages. For measuring the outage in ROS, comparing the desired frequency (f_d) and the current publishing frequency (f_c) of sensor messages with `rostopic hz` yields c_{OT} as $c_{OT} = f_c/f_d$, while a sensor outage leads to $c_{OT} = 0$.

The proposed temporal consistency measure evaluates the equivalence of multiple, "single-shot" captures within a static environment by comparing subsequently captured images and point clouds. Notable differences in single pixel intensity values for 2D images and geometric 3D measurements in point clouds can indicate a limited temporal consistency due to noise and thus limited sensor reliability. A high temporal consistency can indicate low noise and hence high confidence. However, the consistency analysis requires a static scene that is typically available prior to the navigation or manipulation task of the autonomous off-road vehicle. The temporal consistency for 2D images (c_{TC}^{2D}) exploits the SSIM and normalized root mean squared error (NRMSE) measures to compare multiple 2D images of a static scene, as detailed in Section 4.1.4. Alternatively, similarity measures from correlation-based disparity estimation from stereo images methods such as SAD [139] can be utilized to determine the temporal consistency. The temporal consistency of 3D point clouds (c_{TC}^{3D}) is evaluated with a 3D registration approach such as ICP [38]. The relative position and orientation are equivalent to the identity (**1**) as environment and platform are static. A low result for the e_{fs} dissimilarity score and a high number of correspondences indicates a high temporal consistency of two subsequent point clouds. Furthermore, an outlier detection with kNN filtering, as described in Section 4.2.1, can be used to evaluate the temporal consistency for 3D point clouds. The number of filtered outliers provides a reliability estimate for the 3D measurements inside the 3D LiDAR, stereo, and RGB-D clouds. Here, c_{TC}^{2D} and c_{TC}^{3D} are calculated as *PPC* and integrated into Equation 4.7 to Equation 4.9.

4.1.2 Confidence for 2D Images

Six confidence measures are proposed for the 2D confidence in addition to c_{OT} and c_{TC}^{2D}:

- c_{OE}^{2D}, c_{UE}^{2D} (*PPC*): overexposure and underexposure (see also exposure and saturation for MEF in Section 4.4),
- c_{H}^{2D} (*PSC*): contrast and information content (*IC*),
- c_{S}^{2D} (*PSC*): similarity of stereo image pair,
- c_{T}^{2D} (*PSC*): texture analysis and range filtering,
- c_{GLCM}^{2D} (*PSC*): correlation (G_c), homogeneity (G_h), energy (G_e) of GLCM matrix.

Here, c_{S}^{2D}, c_{T}^{2D}, and c_{GLCM}^{2D} are designed to assess the probability of correct disparity estimations in the next processing step. A confidence estimate close or equal to 1.0 represents the highest confidence, while a confidence estimate close to 0.0 indicates a very low confidence.

The difference to the minimum ($\min(\boldsymbol{I}) = 0$) and maximum intensity ($\max(\boldsymbol{I})$) for each spectral channel and each pixel $[i, j]$ of an RGB image is analyzed with c_{OE}^{2D} and c_{UE}^{2D}:

$$c_{OE}^{2D}[i,j] = 1.0 - \frac{(I_b)^{\chi} + (I_g)^{\chi} + (I_r)^{\chi}}{3.0 \cdot (\max(\boldsymbol{I}))^{(\chi)}}. \tag{4.1}$$

The discrete, maximum luminous intensity $\max(\boldsymbol{I})$ is derived from the bit depth of the image, e.g., $\max(\boldsymbol{I}) = 255$ for 8-bit images and different weightings are tested via the power factor χ. Multiplying with $\max(\boldsymbol{I})$ allows us to derive the color information for the visual analysis with intensities $[0, \max(\boldsymbol{I})\boldsymbol{I}]$. The normalization with $\max(\boldsymbol{I})$ yields confidence values in $[0, 1]$. Underexposure is analyzed with c_{UE}^{2D}:

$$c_{UE}^{2D}[i,j] = \frac{(I_b)^{\chi} + (I_g)^{\chi} + (I_r)^{\chi}}{3.0 \cdot (\max(\boldsymbol{I}))^{(\chi)}}. \tag{4.2}$$

The global, per-sensor confidences *PSC* naturally equals $\overline{c_{OE}^{2D}}$, $\overline{c_{UE}^{2D}}$, $\overline{c_{T}^{2D}}$, etc., according to Equation 3.31.

Dataset assessment measures the *IC* of a 2D image based on its Shannon entropy (c_{H}^{2D}) and compares the similarity of two image patches with SSIM, as discussed in Sections 3.5 and 6.2.1.1. Hence, c_{H}^{2D} for an $N \times M$ image or image patch m is calculated with $c_{H}^{2D} = H_m / \max(H)$. The similarity confidence measure c_{S}^{2D} for two $N \times M$ images or patches m and n is compared with SSIM and NRMSE, as detailed in Section 6.2.1.1:

$$c_{S}^{2D} = \mathrm{SSIM}(m,n) - \mathrm{NRMSE}(m,n) / \max(\mathrm{NRMSE}(m,n)). \tag{4.3}$$

Here, NRMSE = 0.0 and SSIM = 1.0 describe the equivalence of m and n. Naturally, a high *IC* of single images and a high similarity of the input image pairs for stereo disparity estimation algorithms indicate a high confidence.

Regular, repetitive color and intensity patterns, and the complete absence of texture complicate an accurate disparity estimation from stereo images for local, correlation-based methods. Horizontal stereo camera setups require preferably different color/intensity patterns in the horizontal image direction. If a high amount of horizontally repetitive patterns or texture elements are present within an image, a high confidence of the 3D stereo camera point cloud is improbable. Hence, $c_{\mathrm{T}}^{2\mathrm{D}}$ for horizontal stereo camera setups evaluates the image's texture for repetitive patterns in the horizontal axis.

Beyerer et al. [16] state that texture analysis does not require an explicit texture model. Thus, $c_{\mathrm{T}}^{2\mathrm{D}}$ relies on a feature-based texture analysis of the grayscale conversion of the analyzed RGB images with statistical texture properties. Range filtering in an $n \times n$ neighborhood is conducted to derive $c_{\mathrm{T}}^{2\mathrm{D}}$. Range filtering derives a range image in which each output pixel contains the intensity difference within the $n \times n$ neighborhood around the analyzed pixel of the input image. The selected sizes of the $n \times n$ neighborhood are derived from the size of the input image patches for the subsequent local, correlation-based stereo matching method and $n = 9$ and $n = 19$ are compared for *UEM-CNN* proposed in Section 5.1.2.

For Fourier analysis, the input image is transferred to the frequency domain, where higher frequencies indicate abrupt intensity transitions. Peaks at certain frequencies can indicate repetitive image patterns in the centralized magnitude of the Fourier spectrum facilitating qualitative texture analysis similar to range filtering. To conclude, range filtering and Fourier analysis support a qualitative user assessment of repetitive, periodic textures on the input confidence for the subsequent disparity estimation from stereo images algorithm prior to the sensing task.

Furthermore, $c_{\mathrm{GLCM}}^{2\mathrm{D}}$ was designed to analyze the spatial dependencies of image pixels to other image pixels with the second-order statistics correlation, homogeneity, and energy. Correlation measures the dependency of a pixel intensity on the intensity of its neighbors with a value between $[-1, 1]$. A high correlation indicates a high predictability of pixel

relationships, according to Hall-Beyer [103], and a high number of repetitive structures inside an image. Homogeneity analyzes the closeness of the element distribution to the diagonal of the GLCM matrix, where monotonous images exhibit high values in the diagonal. In contrast, images with a notable texture exhibit low diagonal values and high values in the upper right and the lower left corners of the GLCM matrix. Energy is also designated uniformity, and calculates the sum of squared elements of the GLCM matrix in a range of [0, 1]. Here, high energy indicates a constant image with a little amount of repetitive texture. Evaluation showed that neither correlation nor homogeneity facilitates a statement on an image's suitability for disparity estimation from stereo images, as also demonstrated in Table 4.1, and G_c and G_h are not considered here for confidence analysis. An energy $G_e \approx 1$ indicates a constant image with little repetitive texture, and a low G_e is preferable for disparity estimation from stereo images. Thus, c^{2D}_{GLCM} combines G_e for different offsets b with $c^{2D}_{GLCM} = \overline{(1.0 - G_e(b))}$.

4.1.3 Confidence for 3D LiDAR Point Clouds

Thrun et al. [268] analyze noise, measurement failures, random measurements, and unexpected objects for range finders such as LiDAR sensors from a more mapping-oriented perspective as discussed in Section 2.3.1. In contrast to [268], this thesis analyzes the confidence of 3D LiDAR measurements independent of the subsequent processing module with the following confidence measures:

- c^{3D}_{A} (PSC): constant accuracy beam modeling (from datasheet),
- $c^{3D}_{R_v}$, $c^{3D}_{R_h}$ (PPC): dynamic accuracy beam modeling,
- c^{3D}_{S} (PSC): surface variation accord. to Equation 3.12,
- c^{3D}_{RS} (PSC): occurrence of reflecting surfaces,
- c^{3D}_{PI} (PSC): probability of precipitation impairment.

Nevertheless, measurement noise as proposed in [268] is considered in c^{3D}_{A}, $c^{3D}_{R_h}$, and $c^{3D}_{R_v}$. A probability estimate for measurement failures, random measurements, and unexpected objects is included in c^{3D}_{RS} and c^{3D}_{PI} with qualitative user estimates. The constant 3D measurement accuracy α_r of rotating 3D LiDAR sensors can be extracted from the datasheet

as inspired by [296], and $c_{\mathrm{A}}^{3\mathrm{D}}$ is determined in relation to the desired 3D measurement accuracy $\alpha_{r,\mathrm{req}}$:

$$c_{\mathrm{A}}^{3\mathrm{D}} = \begin{cases} 1.0 - \frac{\alpha_r}{\alpha_{r,\mathrm{req}}} & \alpha_r \leq \alpha_{r,\mathrm{req}} \\ 1.0 & \alpha_r > \alpha_{r,\mathrm{req}}. \end{cases} \tag{4.4}$$

The dynamically modeled beam resolution measure $c_{\mathrm{R_v}}^{3\mathrm{D}}$ is derived from the number of diodes and the vertical FoV τ_v that yield the angular resolution τ_v' and the metric resolution $\alpha_v' = r \cdot \sin(\tau_v')$. The $c_{\mathrm{R_h}}^{3\mathrm{D}}$ measure for rotating 3D LiDAR sensors with $\tau_h = 360°$ is derived from the metric resolution $\alpha_h' = r \cdot \sin(\tau_h')$:

$$c_{\mathrm{R_v}}^{3\mathrm{D}} = 1.0 - \frac{\alpha_v'}{\max(r) \cdot \sin(\tau_v)} \quad , \quad c_{\mathrm{R_h}}^{3\mathrm{D}} = 1.0 - \frac{\alpha_h'}{\max(r)}. \tag{4.5}$$

This naturally results in a high $c_{\mathrm{R_h}}^{3\mathrm{D}}$ for a small and favorable τ_h'. If a minimum measurement range is given for the respective LiDAR sensor, $c_{\mathrm{R_h}}^{3\mathrm{D}}$ and $c_{\mathrm{R_v}}^{3\mathrm{D}}$ can be set to zero for too close 3D measurements.

The surface variation measure $c_{\mathrm{S}}^{3\mathrm{D}}$ is determined, as detailed in Section 3.6 and demonstrated in Section 6.2.1. A high surface variation $\overline{s}$ indicates a rather unstructured character of the environment. As a result, the sparse LiDAR measurement may not be able to reconstruct all surface geometries properly. A low $\overline{s}$ shows a high number of smooth surfaces in the captured environment. Thus, sparse LiDAR measurements have a higher probability for a more accurate surface reconstruction. An evaluation of $\overline{s}$ is conducted in pre-modeling XAI with *IC-ACC* (see Section 6.2.1). The maximum $\max(\overline{s})$ for normalization in $c_{\mathrm{S}}^{3\mathrm{D}}$ can be derived from Equation 3.11: the numerator is necessarily smaller than the denominator, thus $\max(\overline{s}) \leq 1.0$ always holds, and $c_{\mathrm{S}}^{3\mathrm{D}}$ is calculated as $c_{\mathrm{S}}^{3\mathrm{D}} = 1.0 - \overline{s}$.

The environmental conditions for $c_{RS}^{3\mathrm{D}}$ and $c_{PI}^{3\mathrm{D}}$ can hardly be measured and are hence integrated into the confidence assessment as qualitative user estimates with $c_{RS}^{3\mathrm{D}}, c_{PI}^{3\mathrm{D}} \in \{0; 0.25; 0.50; 0.75; 1.0\}$, if sufficient knowledge on the environmental conditions is available.

4.1.4 Confidence for 3D Point Clouds from Stereo and RGB-D Cameras

Four confidence measures are proposed for 3D point clouds from stereo and RGB-D camera systems:

- $c^{3D}_{\varepsilon_z}$ (*PPC*): theoretical depth accuracy ε_z,
- c^{3D}_{S} (*PSC*): surface variation accord. to Equation 3.12,
- $c^{3D}_{\varepsilon_x}$ (*PPC*): theoretical resolution in x axis,
- $c^{3D}_{\varepsilon_y}$ (*PPC*): theoretical resolution in y axis.

The theoretical accuracies $c^{3D}_{\varepsilon_x}$, $c^{3D}_{\varepsilon_y}$, and $c^{3D}_{\varepsilon_z}$ are analyzed separately, similar to the TVU and THU concept in the subsea domain discussed in Section 2.3.1. Wolf and Berns [296] propose a quadratic error modeling for disparity estimation from stereo images. The quadratic nature of ε_z in $c^{3D}_{\varepsilon_z}$ takes up this idea. It is designed as anti-proportional to the stereo camera depth estimation accuracy ε_z in Equation 3.17 with a predefined maximum tolerable depth inaccuracy $\max \varepsilon_z$ and already includes an estimate on the expected accuracy of the disparity estimation with ε_d:

$$c^{3D}_{\varepsilon_z} = 1.0 - \frac{\varepsilon_z}{\max(\varepsilon_z)} \quad \text{for} \quad \varepsilon_z \leq \max(\varepsilon_z), \quad \text{else}: \quad c^{3D}_{\varepsilon_z} = 0.0. \tag{4.6}$$

The surface variation indicates a scene's structured or unstructured character and is measured on the LiDAR cloud, as detailed for c^{3D}_{S}. Here, it is assumed that the inherent discretization in disparity estimation from stereo images implies a less accurate 3D reconstruction of environments with a high surface variation.

The theoretical resolutions $c^{3D}_{\varepsilon_x}$ and $c^{3D}_{\varepsilon_y}$ are derived from $\varepsilon_x = \Delta x / N$ and $\varepsilon_y = \Delta y / M$ for an image with $N \times M$ pixels that covers an area of $\Delta x\,\mathrm{m} \times \Delta y\,\mathrm{m}$ and calculated according to Equation 4.6 with $\max(\varepsilon_x) = \max(\varepsilon_y) = \max(\varepsilon_z)$. However, the 3D reconstruction inaccuracies introduced by ε_x and ε_y are notably smaller than the depth estimation inaccuracy ε_z and were mostly negligible in the confidence analysis of 3D stereo and RGB-D camera clouds.

4.1.5 2D and 3D per Sensor Confidence

The empirical mean of all *PPC* values yields the *PSC* for each 2D confidence measure. The *PSC* estimates, designated c_i^{2D}, for applicable confidence measures i are combined in c^{2D}:

$$c_i^{2D} = \frac{\sum_{j=1}^{N} \sum_{k=1}^{M} c_i^{2D}[j,k]}{N \cdot M}, i \in \{OE, UE, TC\}, \tag{4.7}$$

$$c^{2D} = c_{OT}^{2D} \cdot c_{TC}^{2D} \cdot \overline{c_i^{2D}}, i \in \{OE, UE, \mathrm{H}, \mathrm{GLCM}, \mathrm{T}, \mathrm{S}\}, \tag{4.8}$$

with $\overline{c_i^{2D}}$ the empirical mean of all confidence estimates c_i^{2D}. The estimated c^{2D} values are input to the confidence assessment of the 3D stereo camera point cloud and provide an a priori confidence for disparity estimation from stereo images. The 2D confidence estimates are designed to assess the potential benefit of generating and integrating a 3D cloud from the disparity estimation from stereo image results. Hence, generating disparity estimation from stereo images for sufficiently confident images implies a tight coupling of perception and validation.

The c^{3D} estimates for LiDAR, stereo camera, and RGB-D camera clouds are determined with conditional probabilities similar to a Bayesian fusion approach, as proposed in [245] and discussed in Section 2.4.3:

$$c_{\mathcal{S},j}^{3D} = c_{\mathcal{S},OT}^{3D} \cdot c_{\mathcal{S},TC}^{3D} \cdot c^{2D} \cdot \overline{c_{i,j}^{3D}}, i \in \{\varepsilon_z, \mathrm{S}, \varepsilon_x, \varepsilon_y\}, \tag{4.9}$$

$$c_{\mathcal{L},j}^{3D} = c_{\mathcal{L},OT}^{3D} \cdot c_{\mathcal{L},TC}^{3D} \cdot (\overline{c_{i,j}^{3D}} + \overline{c_k^{3D}}), i \in \{\mathrm{R_v}, \mathrm{R_h}\}, k \in \{\mathrm{A}, \mathrm{S}, RS, PI\}. \tag{4.10}$$

If c_{TC} estimates are not available, $c_{TC} = 1.0$ is set.

The proposed confidence estimation process yields one confidence measure $c_{\mathcal{L},j}^{3D}$ or $c_{\mathcal{S},j}^{3D}$ for each individual 3D measurement j. Confidence estimates can be applied to filter raw perception data without a subsequent fusion, or they can provide an input to the confidence-based 3D–3D fusion process detailed in Section 5.3. Furthermore, the estimated confidences are also applicable for other subsequent processing steps, such as a confidence-aware and adaptive mapping, planning, and control discussed in Section 7.2.

Image	m_{GLCM}	G_c	G_e	G_h	$c^{\text{2D}}_{\text{GLCM}}$	$c^{\text{2D}}_{UE}(\chi)$	$c^{\text{2D}}_{OE}(\chi)$	c^{2D}_{H}	c^{2D}
Figure 4.1(a)	1	0.996	0.197	0.969	0.833	0.678	0.659	0.897[*1]	**0.813**[*2]
Figure 4.1(a)	9	0.963	0.160	0.879	0.833	0.678	0.659	0.897[*1]	**0.813**[*2]
Figure 4.1(a)	19	0.937	0.145	0.833	0.833	0.678	0.659	0.897[*1]	**0.813**[*2]
White image	–	undef.	1.0	1.0	0.0	1.0	0.0	0.0	0.375[*2]
Black image	–	undef.	1.0	1.0	0.0	0.0	1.0	0.0	0.375[*2]

[*1] H(Fig. 4.1(a)) = 7.174 with max(H) = 8.0 for a 1296×964 px 8-bit image.
[*2] $c^{\text{2D}}_{OT} = c^{\text{2D}}_{TC} = c^{\text{2D}}_{\text{T}} = 1.0$.

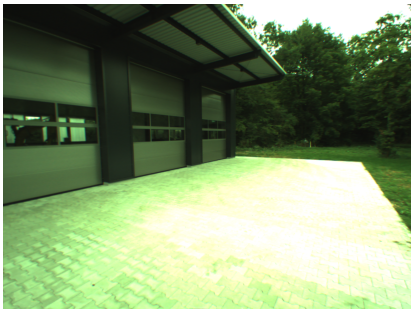
Table 4.1 2D confidence analysis with $m_{\text{GLCM}} \in \{1, 9, 19\}$ and $\chi = 3.0$.

4.1.6 Proof of Concept: Sensor Data Confidence

Sensor Outage, Temporal Consistency, and Outlier Filtering. The detection of sensor outages with `rostopic hz` is trivial. Shannon entropy and surface variation for c^{2D}_{H} and c^{2D}_{S} are illustrated in Section 6.2.1.1 and 6.2.1.4, while kNN outlier filtering for 3D point clouds is demonstrated in Section 4.2.4.

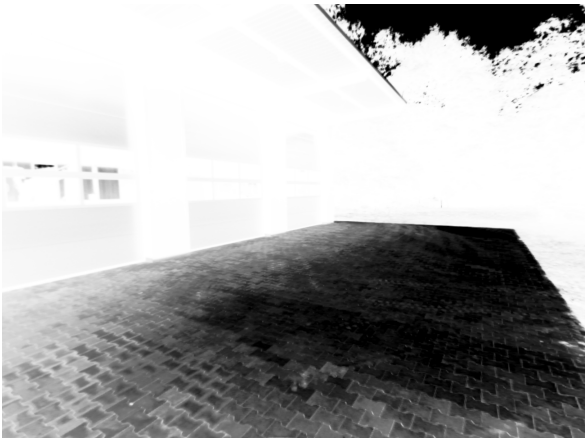
Confidence of 2D Images. Figure 4.1 demonstrates the pixel-wise *OE* and *UE* confidence measures for 2D images. A power factor $\chi \geq 1.0$ for *OE* and *UE* yields a lower weighting of the intensity differences in contrast to $\chi = 1.0$ where c^{2D}_{UE} equals a grayscale image and c^{2D}_{OE} an inverted grayscale image of the RGB image. The *OE* and *UE* measures facilitate a scalable, quantitative consideration of the sensitivity of a subsequent disparity estimation method for overexposure and underexposure. A higher χ weights low probabilities for overexposure and underexposure less than $\chi = 1.0$ and $\chi = 3.0$ or higher has proved useful for disparity estimation methods that cope well with overexposure and underexposure, such as *UEM-CNN* (see Section 5.1.2).

Figure 4.1(a) depicts the exemplary *IOSB-Reg* image chosen to demonstrate range filtering facilitating a fast visual analysis of intensity changes on edges and possibly repetitive patterns. As expected, range filtering of Figure 4.1(a) did not indicate a high amount of periodic, repetitive texture elements and $c^{\text{2D}}_{\text{T}} = 1.0$ is set. Table 4.1 presents the results of the confidence analysis of Figure 4.1(a).



(a) Original 2D RGB image.

(b) c_{UE}^{2D} for $\chi = 3.0$, $c_{UE}^{2D} = 0.678$.



(c) c_{OE}^{2D} for $\chi = 3.0$, $c_{OE}^{2D} = 0.659$.

(d) Range filtering 9×9.

(e) Range filtering 19×19.

Figure 4.1 2D confidence measures for an exemplary *IOSB-Reg* image. White with $I = \max(I)$ highlights high confidence, black color with $I \to 0$ indicates low confidence. Images (d) and (e) show range filtering results for (a).

Measure	Value	Justification
$c_{\mathrm{A}}^{3\mathrm{D}}$	1.0	$\alpha_r \pm 2\,\mathrm{cm}$, min. $\alpha_{r,\mathrm{req}} = 5\,\mathrm{cm}$
$c_{\mathrm{R_v}}^{3\mathrm{D}}(r = 10\,\mathrm{m})$	0.999	$\alpha'_v = r \cdot \sin(0.4°) = 0.07\,\mathrm{m}$
$c_{\mathrm{R_v}}^{3\mathrm{D}}(r = 100\,\mathrm{m})$	0.985	$r = \max(r) = 100\,\mathrm{m}$, $\alpha'_v = 0.698\,\mathrm{m}$
$c_{\mathrm{R_h}}^{3\mathrm{D}}(r = 100\,\mathrm{m})$	0.999	$\alpha'_h = 0.140\,\mathrm{m}$

Table 4.2 3D confidence analysis for Velodyne HDL-64E point clouds with $\tau_h = 360°$, $\tau_v = 26.9°$, $\tau'_h = 0.08°$, $\tau'_v = 0.4°$, and max. measurement range $\max(r)$.

Confidence of 3D LiDAR Point Clouds. Table 4.2 demonstrates that comparatively high confidence is estimated for 3D LiDAR point clouds if no sensor outage occurs. This corresponds with the high importance of rotating 3D LiDAR sensors in the perception sensor setup for off-road vehicles in unstructured environments. Equidistant beam distribution for the HDL-64E sensor with 64 diodes confirms $\tau'_v = 26.9°/64 = 0.420° \approx 0.4°$. Qualitative user estimates are required for $c_{RS}^{3\mathrm{D}}$ and $c_{PI}^{3\mathrm{D}}$. For instance, $c_{RS}^{3\mathrm{D}} = 0.50$ proved useful for a rainy day with small puddles and $c_{RS}^{3\mathrm{D}} = 0.0$ for heavy snowfall. For a medium or high presence of reflecting surface in the close range of the vehicle, $c_{PI}^{3\mathrm{D}} = 0.50$ or $c_{PI}^{3\mathrm{D}} = 0.25$ provide satisfying results, while $c_{PI}^{3\mathrm{D}} = 1.0$ is recommended in their absence.

Confidence of 3D Stereo and RGB-D Clouds. An exemplary $c_{\varepsilon_z}^{3\mathrm{D}}$ value for a JAI AD-130GE camera, as discussed in Section 3.8 is derived as $\varepsilon_z = 0.704\,\mathrm{m}$, $\varepsilon_d = 3\,\mathrm{px}$, and an exemplary $\max(\varepsilon_z) = 1.0\,\mathrm{m}$ yield $c_{\varepsilon_z}^{3\mathrm{D}}(r = 10\,\mathrm{m}) = 0.296$. The surface variation measure $c_{\mathcal{S},\mathrm{S}}^{3\mathrm{D}}$ is equivalent to the LiDAR cloud $\mathcal{L}$. The theoretical accuracies $c_{\varepsilon_\mathrm{x}}^{3\mathrm{D}}$ and $c_{\varepsilon_\mathrm{y}}^{3\mathrm{D}}$ can be demonstrated on an exemplary, captured area of $13.0\,\mathrm{m} \times 9.67\,\mathrm{m}$ with $\varepsilon_x = \varepsilon_y = 0.01\,\mathrm{m}$ and yields $c_{\varepsilon_\mathrm{x}}^{3\mathrm{D}} = c_{\varepsilon_\mathrm{y}}^{3\mathrm{D}} = 1.0$.

4.2 3D–3D Registration of Similar-Source Data

Direct registration approaches facilitate the registration of similar-source LiDAR sensor data [323] and are thus preferred to register multiple LiDAR sensors to a common vehicle coordinate system as they yield accurate, stable results with comparatively low computation effort. As discussed Section 2.3.3, more complex cross-source registration methods

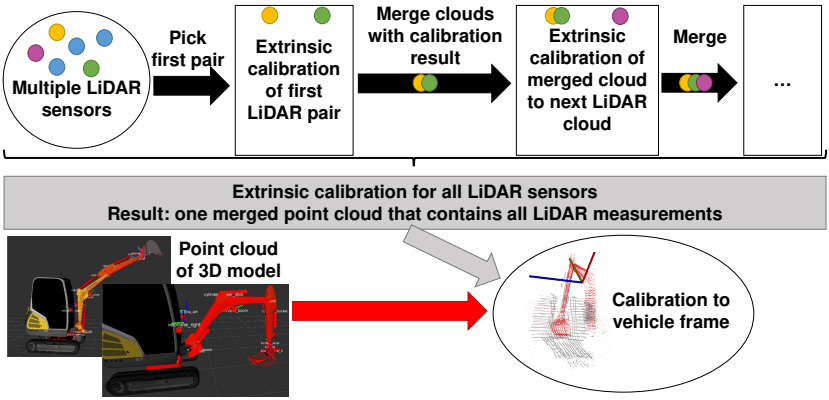


Figure 4.2 Registration approach according to [323]: the LiDAR clouds colored yellow and green are the first pair for extrinsic calibration, subsequently the merged LiDAR cloud (yellow and green) is registered to the LiDAR represented by the purple circle, etc. After extrinsic calibration and sensor data registration, the merged LiDAR cloud is registered to the vehicle frame.

are not required for similar-source data. This thesis avoids methods with calibration targets [49, 78, 158, 207] or additional sensor data as proposed in [247] to facilitate a generically applicable sensor calibration for different sensor setups and platforms and without human intervention, especially in hazardous environments.

This thesis proposes an enhanced GICP: a combination of GICP [250] registration, as detailed in Section 3.10, with customized preprocessing and enhanced surface estimation for registration to determine the extrinsic sensor calibration of multiple 3D LiDAR sensors to an off-road vehicle. The registration of the LiDAR sensors to the vehicle frame exploits a synthetically generated point cloud from the vehicle's 3D model via the platform's computer-aided design (CAD) data. Figure 4.2 provides an overview of the subsequently discussed registration approach [323]. Rough estimates of the relative sensor poses and the rough estimate of one sensor relative to the vehicle coordinate system present the single user-generated step in the presented registration procedure. The captured

LiDAR data is visualized with rviz[1], bound to so-called 3D interactive markers, and the user moves and rotates the markers until the clouds are roughly visually aligned. These initial pose estimates were utilized during preprocessing and also achieved a useful initial cloud alignment for the locally operating GICP algorithm that provided an optimization constraint to limit faulty local convergence without the need for on-site measurements by hand. This way, the sole manual step in calibration can also be performed on recorded sensor data and thus decentralized after data capture. Further on, registration is also designated as merge highlighting the merge process of point clouds.

4.2.1 Preprocessing

Three preprocessing steps are proposed to optimize the registration process for similar-source 3D–3D data. Most LiDAR sensors require a minimum distance r_{min} for valid measurements. If so, only points $\mathbf{p}_i = [x_i, y_i, z_i]^*$ with

$$\left\|\mathbf{p}_i\right\| = \sqrt[2]{(x_i^2 + y_i^2 + z_i^2)} \geq r_{\text{min}} \tag{4.11}$$

are kept for further processing. Outliers do not benefit registration accuracy, and their elimination improves the registration accuracy, which yields more robust registration results. To this end, separate preprocessing (*SPP*) treats each cloud independently of the other clouds [323] and is conducted for all LiDAR clouds during extrinsic calibration: outliers are filtered using a kNN search for each $\mathbf{p}_i$ in a search sphere of ratio r. A minimum number of neighbors inside this sphere is required so that the analyzed point is not eliminated as outlier.

A third preprocessing step removes areas only captured in one of the two point clouds and thus non-applicable for registration. This preprocessing is denoted as initial transformation preprocessing (*ITP*) as it exploits the user-generated initial transformation estimates to align the clouds roughly. Similar to *SPP*, different sphere ratios for the kNN search and minimum numbers of neighbors were evaluated. *ITP* is conducted for all pairs in the first calibration step, and only points fulfilling the

[1] wiki.ros.org/rviz, access on 30.12.2021.

constraint specified in Equation 4.11 are kept for the GICP registration step.

Furthermore, a maximum distance of the points in the merged LiDAR cloud from the origin can be derived from the vehicle's geometry for the registration to the vehicle frame. A maximum of $r = 2.5\,\mathrm{m}$ was applied for the IOSB.BoB platform because it is a small platform, and no sensor can ever be further away. In addition, filtering 3D points that cannot be detected by the LiDAR sensors from the 3D model cloud with backface culling, a method to determine whether a polygon is visible from a certain viewpoint, further improved the calibration result. The normal information for backface culling is extracted from a COLLADA file, and contains the direction and sign of the normal vector.

One-to-many correspondences assign one point of the target cloud to multiple points in the source cloud. They lead to a high e_{fs} despite a proper registration result and can be analyzed for additional verification of the selected GICP parameters, as elaborated in Section 4.2.4. One-to-many correspondences especially occur in point clouds with notably different point densities or numbers of points.

4.2.2 Extrinsic Calibration with Enhanced GICP

The preprocessed 3D point clouds are input to the registration process outlined in Figure 4.2, and registration using the extrinsic sensor calibration results facilitates the fusion of all individual point clouds into one point cloud. This fused point cloud is subsequently referenced to the coordinate system of the platform by registering it to the 3D cloud extracted from its 3D CAD model, as detailed in Section 4.2.3. At first, the obtained point clouds are registered in pairs to determine the extrinsic calibration of the respective sensors using the GICP algorithm [250]. The sensor order is crucial, and the GICP registration accuracy is measured with the Euclidean fitness score given in Equation 3.28. A sufficiently large FoV has to be shared for proper registration; otherwise, inaccurate registration results or too little correspondences for convergence of GICP are found. Using an appropriate selection and merging instead of single point clouds for registration, the e_{fs} can be reduced by more than 60 %, as demonstrated in [323]. The LiDAR that provides the largest, most dense point cloud is selected as the target LiDAR sensor and determines the

coordinate system for the subsequent merge processes. The respective source cloud is merged in the coordinate system of the target LiDAR using the GICP registration result. The merged source-target cloud is registered to the next point cloud that shares a maximally large common FoV. Again, the registered point clouds are merged in the frame of the target LiDAR. This procedure is repeated until all LiDAR clouds are merged into one cloud. An additional visual assessment is applied within this thesis to check for faulty convergence in local minima.

Furthermore, this thesis proposes an enhanced GICP parameterization optimized for the registration of 3D LiDAR point clouds from unstructured environments. A special focus is on the point set considered for the normal estimation with SVD. Consequently, the following parameterizations were evaluated on real-world data, as detailed in Section 4.2.4:

1. Correspondence randomness (CR),
2. The maximum of the correspondence distance (r_C),
3. GICP convergence criterion $\varepsilon_{\mathrm{EF}}$: Euclidean fitness epsilon,
4. GICP convergence criterion δ: maximum difference of two consecutively estimated transformations,
5. GICP convergence criteria ε_R,ε_T: maximum difference of two consecutive rotations (R) or translations (T) in δ.

Here, CR defines the number of nearest neighbor points considered in calculating the empirical covariance $\hat{\sum}$. The maximum distance between a point set $\mathbf{p}_s$ and $\mathbf{p}_t$ to be accepted as correspondences is designated $\max(r_C)$, while $\varepsilon_{\mathrm{EF}}$ defines the maximum L_2 error of consecutive transformation estimates weighted with the number of correspondences, as detailed in Equation 6.7. The convergence criterion δ is the maximally allowed difference of consecutively estimated transformations, weighted with $1/\varepsilon_R$ and $1/\varepsilon_T$. The weighting parameters ε_R and ε_T are input to the convergence criterion δ.

4.2.3 Registration to the Vehicle Frame

The calibration to the vehicle coordinate system exploits a registration of the synthetically generated point cloud from CAD data and the LiDAR measurements in the merged cloud from extrinsic calibration. Technically, the registration of the merged cloud to the 3D model cloud is a cross-source 3D–3D registration. However, the 3D model and the LiDAR cloud

are both accurate and structurally similar, and the point density of the 3D model cloud can be adjusted to the cloud density of the LiDAR cloud. The merged LiDAR cloud is generated from multiple, rotating 3D LiDAR sensors in different orientations, and the LiDAR ring structure is lost. Hence, customized preprocessing for both 3D point clouds facilitates the direct registration of a 3D cloud from multiple LiDAR sensors to a 3D cloud extracted from CAD data with the proposed enhanced GICP algorithm.

The merged LiDAR cloud contains all LiDAR measurements in the coordinate system of one selected LiDAR sensor and constitutes the registration source. The source cloud primarily contains points representing vehicle elements within the FoV of the merged LiDAR cloud, such as the boom, dipper stick, and excavator bucket. *ITP* preprocessing removes all LiDAR points that do not benefit the registration step using the initial transformation estimate. The preprocessed source cloud is then registered to the 3D model cloud with ground truth normal information. The optimal alignment of these clouds estimates the pose of the main LiDAR sensor in relation to the vehicle frame. From here, the extrinsic calibration results facilitate the determination of all sensor poses relative to the common vehicle frame.

The 3D model of the platform provides accurate geometric information for the target cloud, as proposed in [75]. The SVD of the covariance matrix estimates the normal information for the merged LiDAR cloud, as discussed in Section 3.6. A fixed number of neighboring points for calculating the covariance matrix is compared to the consideration of all neighboring points inside a sphere with ratio r around a point $\mathbf{p}_i$ according to Equation 4.11. Contrasting extrinsic calibration, the normal information from the CAD model replaces the normal estimation via SVD.

CAD models commonly contain geometric information as triangle structures, and 3D vertices specify the locations of the triangle corners. Depending on the structure of the CAD model, each element of the platform, such as dipper stick, boom, or undercarriage for an excavator, is given separately or already integrated into one complete model. In either case, the generation of a point cloud from the vertices is not sufficiently dense to register another point cloud onto it. Densification is required

to fill the areas inside the vertex triangles with additional points. The parameter form of the plane equation specifies a plane P_{3D} in 3D space that contains the points $\mathbf{p}_a$, $\mathbf{p}_b$, and $\mathbf{p}_c$ as

$$P_{3D} = \mathbf{p}_a + r(\mathbf{p}_b - \mathbf{p}_a) + s(\mathbf{p}_c - \mathbf{p}_a), \tag{4.12}$$

with the scalar values r and s. The straight line L^{3D} through $\mathbf{p}_b$ and $\mathbf{p}_c$ is defined by $L^{3D} = \mathbf{p}_b + t(\mathbf{p}_c - \mathbf{p}_b)$. For $t \in [0, 1]$, the points lie between $\mathbf{p}_b$ and $\mathbf{p}_c$. The distance of an arbitrary point $\mathbf{p}_e$ from a straight line L_f^{3D} through $\mathbf{p}_f$ is calculated from $(\mathbf{p}_c - \mathbf{p}_b)(\mathbf{p}_f - \mathbf{p}_e) = 0$ using the perpendicular line L_{ef}^{3D} with $L_{ef}^{3D} \perp L_f^{3D}$. For a fixed t, $\mathbf{p}_f$ can also be calculated from $\mathbf{p}_f = \mathbf{p}_b + t \cdot (\mathbf{p}_c - \mathbf{p}_b)$. Furthermore, $\mathbf{p}_e$ is also described by

$$\mathbf{p}_e = \mathbf{p}_a + r \cdot (\mathbf{p}_b - \mathbf{p}_a) + s \cdot (\mathbf{p}_c - \mathbf{p}_a). \tag{4.13}$$

This can be transformed to

$$t = \frac{\Big(\mathbf{p}_b - \mathbf{p}_a - r(\mathbf{p}_b - \mathbf{p}_a) - s(\mathbf{p}_c - \mathbf{p}_a)\Big)\Big(\mathbf{p}_c - \mathbf{p}_b\Big)}{\big\|\mathbf{p}_c - \mathbf{p}_b\big\|^2} \tag{4.14}$$

with fixed values r and s. The centroid of all triangles $\mathbf{p}_a$, $\mathbf{p}_b$, and $\mathbf{p}_c$, as well as points with $t \in [-1, 1]$, lie inside the triangle area. The point $\mathbf{p}_e$ with $t \in [-1, 1]$ lies in the vertex triangle defined by the vertex points $\mathbf{p}_a$, $\mathbf{p}_b$, and $\mathbf{p}_c$ for

$$|\mathbf{p}_e L_{ef}^{3D}| \geq \sqrt{(L_{af}^{3D})^2 + (L_{ae}^{3D})^2}. \tag{4.15}$$

For $t \in [-1, 1]$ and the constraint in Equation 4.15, the vertex triangles can be filled with points, and a dense point cloud is generated as a registration target. If single platform elements are given, dense clouds are generated for all individual platform elements and subsequently fused into one coordinate system to form a 3D model cloud for registration.

Two vectors, which are orthonormal to the given surface normal vector $\mathbf{n}$, are calculated. The first two elements of one orthonormal vector $\mathbf{a}$ ($a(0)$,

$a(1)$) as well as the first element of the second orthonormal vector $\mathbf{b}$ ($b(0)$) can be chosen randomly. The other elements are calculated with

$$a(2) = \frac{-(n(0)a(0) + n(1)a(1))}{n(2)} \tag{4.16}$$

$$b(1) = \frac{-(b(0)a(0) + b(2)a(2))}{a(1)} \tag{4.17}$$

$$b(2) = \frac{b(0)(n(1)a(0) - n(0)a(1)}{a(1)n(2) - a(2)n(1)}, \tag{4.18}$$

and $\mathbf{a}$ and $\mathbf{b}$ are normalized and form an orthonormal basis with $\mathbf{n}$. Furthermore, an arbitrary point $\mathbf{p}_s$ cannot lie on a platform element and is eliminated if its surface normal $\mathbf{n}_s$ does not fulfill

$$\frac{\pi}{2} \leq \arccos \frac{\left\|\mathbf{n}_s \bullet (\mathbf{p}_s - \mathbf{p}_o)\right\|}{\left\|\mathbf{n}_s\right\|\left\|(\mathbf{p}_s - \mathbf{p}_o)\right\|}, \tag{4.19}$$

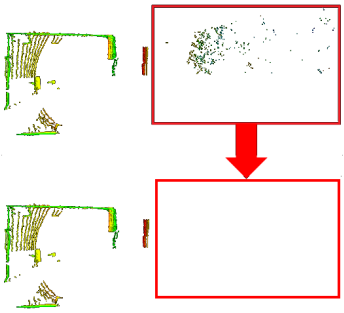
with $\bullet$ the dot product and $\mathbf{p}_o$ the origin of the sensor coordinate system.

The ground surface can be used as an additional registration element besides the platform parts captured by the LiDAR sensors, and this can increase the accuracy and robustness of the registration. It requires the assumption of a flat ground surface and the integration of an artificial ground plane in the 3D model cloud. The orientation of the undercarriage in the 3D model determines the orientation of the artificial ground. Alternatively, the ground plane can be eliminated from the source point cloud using the RANSAC algorithm with a 2D plane model [323].

4.2.4 Proof of Concept: Extrinsic Calibration

The proposed 3D–3D registration method with extrinsic sensor calibration is demonstrated for three Velodyne VLP-16 LiDAR sensors on the IOSB.BoB platform, as shown in Figure 1.3. The LiDAR clouds are preprocessed prior to extrinsic calibration with *SPP*, *ITP*, and an experimentally justified minimum distance $r_{\text{min}} = 0.85\,\text{m}$ as a trade-off between close range measurements and the minimum distance for valid measurements indicated in the datasheet[2]. Figure 4.3 shows the outlier filtering re-

[2] Velodyne LiDAR: https://velodynelidar.com/products/puck/, access on 24.04.2022.



SPP result in bird's-eye view.

Figure 4.3 Outlier filtering with *SPP* conducted with a min. of 10 neighbors in $r = 0.15\,\mathrm{m}$ and points inside the red box were removed as outliers.

sults with *SPP*, and Table 4.4 shows that *SPP* notably reduces the e_{fs} of the GICP-registered point cloud. A minimum of 10 neighbors within a sphere of $r = 0.15\,\mathrm{m}$ achieved the lowest e_{fs} with manually verified alignment and a sufficient number of correspondences. *ITP* yielded the lowest e_{fs} in combination with a visually verified transformation result for a minimum of 400 neighboring target points in $r = 1.5\,\mathrm{m}$ and reduced the e_{fs} to less than 2 % compared to the e_{fs} without preprocessing [323].

The point clouds from the left and right LiDAR sensor were registered at first as they shared the largest common FoV. Subsequently, the clouds of the left and right LiDAR were fused into one cloud. This fused cloud provided the registration target for the point cloud from the LiDAR sensor mounted to the boom of IOSB.BoB. Table 4.3 summarizes the proposed enhanced GICP parameterization to register multiple 3D LiDAR sensors without calibration targets in unstructured environments and compares them to the parameter recommendations of [250].

The influence of one-to-many correspondences was examined for registering the left LiDAR cloud to the right LiDAR cloud and showed that eliminating the target points with more than 50 one-to-many correspondences reduces the e_{fs} from $2.167\,\mathrm{m}^2$ to $0.060\,\mathrm{m}^2$ (see also Table A.1). Allowing up to 100 one-to-many correspondences per target point did not further decrease the e_{fs}. This filtering did neither influence the convergence nor the registration result, so it is not used in the final 3D–3D en-

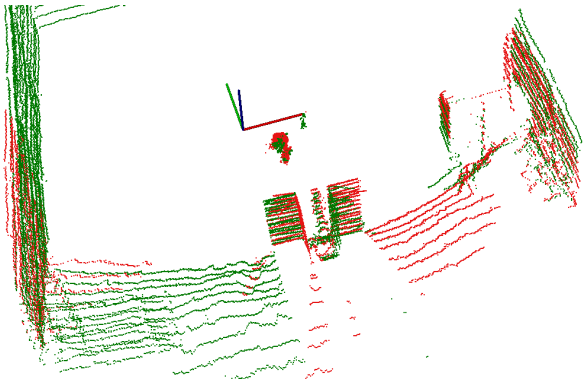


Figure 4.4 Registration using the extrinsic calibration result of first LiDAR pair in bird's-eye view; side walls and polystyrene blocks next to the bucket illustrate the registration accuracy. The x axis is colored red, y green, and z blue.

hanced GICP registration for similar-source clouds. However, it showed the limits of the e_{fs} metric to assess the registration accuracy and verified the parameter selection for the presented, enhanced GICP.

Table 4.4 shows the achieved calibration accuracy in terms of e_{fs}. Partially unstructured outdoor environments (*PUO*) and structured outdoor environments (*SO*) were evaluated separately. The selection of the boom and left LiDAR sensors as the first calibration pair did not yield satisfactory results, and a large common FoV for the left and right LiDAR clouds notably lowered the e_{fs}. Both proposed preprocessing methods lowered the e_{fs} and increased the registration accuracy. The lower registration accuracy for the *PUO* scene compared to the *SO* scene highlights the difficulty of registering sensor data from unstructured environments.

The measurement accuracy of Velodyne VLP-16 LiDAR sensors[3] is ± 0.03 m. For the left and right LiDAR sensor, the relative translation could be measured by hand and amounted to 1.07 m. The registration accuracy achieved with the proposed enhanced GICP equaled the sensor

[3] Velodyne LiDAR: https://velodynelidar.com/products/puck/, access on 24.04.2022.

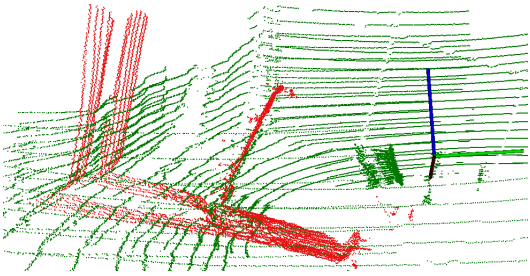


Figure 4.5 Registration result exploiting the determined extrinsic calibration of the LiDAR mounted below the excavator boom (red, source) to the merged point cloud from first LiDAR pair (green, target) with three side walls [323].

measurement noise with a registration result of 1.07 m ± 0.03 m for the relative translation on different datasets.

Figure 4.4 shows the registration results exploiting the extrinsic calibration for the first (left) and second (right) LiDAR sensor mounted to the sides of the excavator cabin, and Figure 4.5 depicts the registration results of the third LiDAR sensor mounted on the boom in a different orientation to the left and right LiDAR sensors.

To conclude, a valid extrinsic calibration of multiple LiDAR sensors with a correct and robust estimation of translation and rotation could be achieved with the proposed preprocessing and GICP enhancements in unstructured environments. At least one surface not being aligned with the roll or yaw axis of the vehicle had to be captured by all LiDAR sensors in addition to the ground plane. With these requirements, accurate extrinsic calibration of LiDAR sensors without additional calibration objects and manual measurements in a partially unstructured environment could be provided.

4.2.5 Proof of Concept: Registration to the Vehicle Frame

The CAD model point cloud was extracted from the COLLADA data of the IOSB.BoB platform, and a dense point cloud with normal information was generated as described in Section 4.2.3. *SPP* of the merged LiDAR cloud with a minimum of 1000 neighboring points in $r = 0.15$ m preserved the LiDAR measurement points lying on the excavator boom, dipper

Parameter	Segal et al. [250]	Extrinsic	Vehicle
r, *SPP*	–	0.15 m	0.15 m
min(NN), *SPP*	–	10	1000
r, *ITP*	–	400	–
CR	20	100	–[1]
r, SVD[1]	–	–	0.40 m
ϵ_{EF}	1 m^2	1.0 m^2	1.0 m^2
max(r_C)	5 m	1.5 m	1.0 m
δ	1	1	1
ϵ_R, ϵ_T	0.001	10^{-4}	10^{-4}

[1] SVD normal estimation in sphere of r.

Table 4.3 Enhanced GICP parameterization for the 3D–3D registration of similar-source data from unstructured environments to extrinsically calibrate (and register) multiple Velodyne VLP-16 3D LiDAR sensors without additional calibration targets: extrinsic LiDAR calibration and registration (Extrinsic) and registration to the excavator vehicle frame (Vehicle).

stick, and bucket and eliminated most (around 78 %) of the LiDAR points lying outside the region of interest for registration. The normal estimation for the merged source LiDAR cloud inside a specified sphere with ratio r lowered the e_{fs} and also increased the number of correctly estimated DoF for all evaluated diameters $r \in \{0.10\,\text{m}; 0.25\,\text{m}; 0.40\,\text{m}\}$ in contrast to the normal estimation proposed by [250]. The lowest e_{fs} of 0.022 m^2 and highest number of correctly estimated DoF was achieved with $r = 0.25$ m and a lowered excavator arm. Only the assumption of a flat ground surface and the addition of an artificial ground plane inside the CAD model cloud achieved a stable and valid registration with a low e_{fs} score for all evaluated diameters r. The lowest mean e_{fs} with a virtual ground surface was 0.014 m^2 with $r = 0.40$ m, while $r = 0.10$ m and $r = 0.25$ m yielded a mean e_{fs} of 0.015 m^2. Figure 4.6 illustrates the registration result exploiting the determined calibration to the vehicle frame of the IOSB.BoB platform.

For the presented application on the IOSB.BoB excavator, at least one LiDAR sensor had to capture parts of the boom, the whole bucket, and a sufficiently large part of the dipper stick. It was necessary to perform

Source	Target	e_{fs}	Preprocessing	Data
Left	Right	4.340 m^2	–	*SO*
Left	Right	2.169 m^2	*SPP*	*SO*
Left	Right	2.566 m^2	*SPP*	*PUO*
Left	Right	0.064 m^2	*SPP*, *ITP*	*PUO*
Left	Boom	20.016 m^2	–	*SO*
Boom	Left	0.319 m^2	–	*SO*
Boom	Merged Left–Right	0.294 m^2	–	*SO*
Boom	Merged Left–Right	4.661 m^2	–	*PUO*
Boom	Merged Left–Right	0.292 m^2	*SPP*	*SO*
Boom	Merged Left–Right	0.186 m^2	*SPP*	*PUO*
Boom	Merged Left–Right	0.137 m^2	*SPP*, *ITP*	*PUO*

Table 4.4 Similar-source 3D–3D registration accuracy exploiting extrinsic calibration with enhanced GICP according to Table 4.3 for different source and target clouds captured in outdoor environments. The e_{fs} values are empirical mean values from different *SO* and *PUO* scenes. *SPP* with a min. of 10 neighbors in r = 0.15 m, *ITP* with a min. 400 neighbors in r = 1.5 m.

the registration on flat ground with an artificial, flat ground plane inside the model cloud to achieve valid and accurate registration to the vehicle frame, and a sufficiently lowered arm is required to capture the dipper stick. As the position of the excavator's arm can also be adjusted in autonomous or teleoperation from a distance, this requirement could always be met.

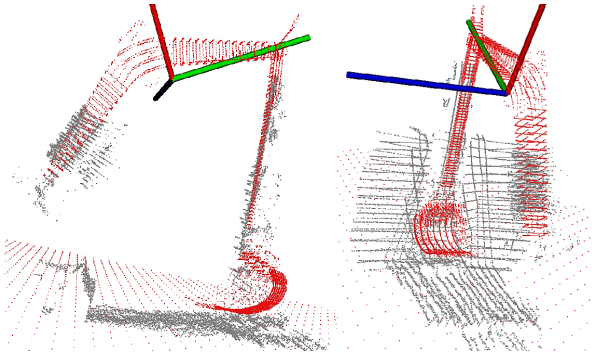


Figure 4.6 Registration to the vehicle frame [323]: the preprocessed LiDAR source cloud (gray) was registered to the 3D model target cloud assuming an approximately flat ground (red).

4.3 *UCSR*: Confidence-Based Registration Framework for Cross-Source Sensor Data

The unstructured cross-source registration framework *UCSR* facilitates a confidence-based fusion of registration results from different registration methods. This implies the tightly coupled validation of the registration results and yields higher robustness in case of errors or inaccuracies, such as noise or difficult environments, compared to the registration with individual methods as the results of more accurate methods are considered with a higher weighting.

UCSR includes the presented *cc23*, *cnn23*, and *graph33* methods as summarized in Figure 4.7. The *graph33* method is integrated as one 3D–3D registration method that requires the availability of a stereo camera system and comes with the inherent depth estimation inaccuracies discussed in Section 3.8, while *dsm33* is intended as a first proof of concept for the successful 3D–3D registration of data from unstructured environments with neural networks. It was not integrated in *UCSR* as it currently implies similar depth estimation inaccuracies as *graph33* limiting its contribution to a higher registration accuracy. Section 4.3.1 details the confidence-based fusion of individual registration results. The individual registration methods developed *cc23*, *graph33*, *cnn23*, and *dsm33* are

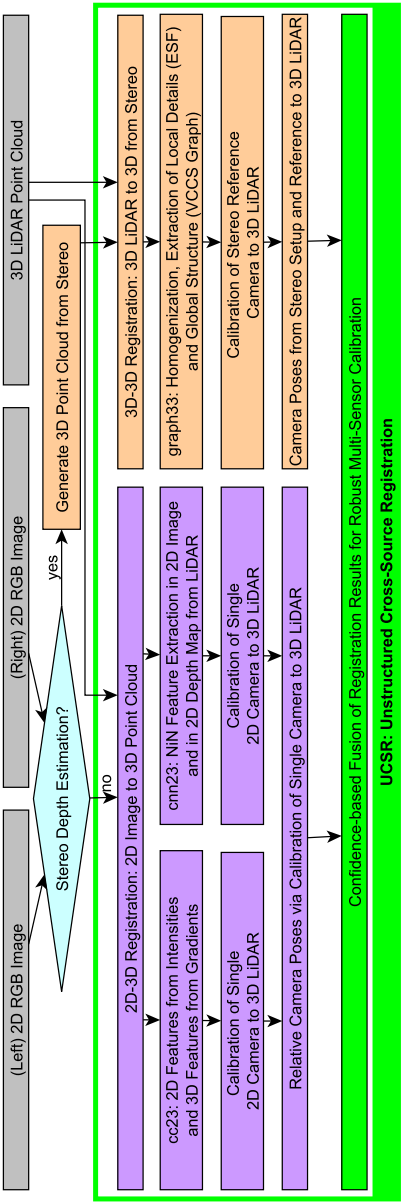


Figure 4.7 Cross-source data input, preprocessing, registration, and fusion for the proposed 2D–3D and 3D–3D registration methods fused into *UCSR*.

introduced and demonstrated in Section 4.3.2 to 4.3.5. Section 4.3.6.1 compares the two proposed, classic registration methods *cc23* and *graph33*, while Section 4.3.6.2 contrasts the *cnn23* and *dsm33* methods that rely on neural networks. Finally, Section 4.3.7 demonstrates the proposed, confidence-based *UCSR* registration framework [329].

4.3.1 Tight Coupling to Validate Registration Results

UCSR analyzes the strengths and weaknesses of individual registration methods and their correlations to the input data characteristics. It combines the registration results for equivalent sensor setups in multi-sensor systems estimated from different registration methods into one validated extrinsic sensor calibration by sensor data registration. The estimated accuracy, reliability, and tolerance to noise measure the confidence of the registration results determining the weights in the fusion process and improves the registration of cross-source data qualitatively and quantitatively.

Each registration method included in the *UCSR* framework yields one independent registration solution, a relative pose from camera to LiDAR. The discussed visual overlay can confirm the validity of each registration result, and only valid registration results are considered in *UCSR*. The best transformations in terms of the Frobenius norm F are used for the confidence-based fusion in *UCSR* as F directly evaluates the difference between the different transformations achieved by the multiple registration methods in *UCSR*. Prior empirical evaluations showed that accuracies in the range of single-digit centimeters could provide a functional reconstruction of the environment for autonomous off-road vehicles [216, 323, 324]. Consequently an accuracy, as illustrated in Figure 4.12, proved sufficient for most navigation and manipulation tasks.

Two empirically justified weighting options $w_{i,1}$ and $w_{i,2}$ were compared for the confidence-based fusion of the registration results. Both weightings are developed to indicate the importance of the respective registration result i within the confidence-based *UCSR* framework with a high weighting for high registration accuracy. Therefore, Equation 4.20 ensures that registration methods with lower accuracy of transformation estimates are considered less than registration results from highly confident methods. The relative size of the non-logarithmic values of F

yielded a very low relative influence of F for the weight calculation, while $\ln F$ achieved a balanced weighting of F and L_2. The standard deviations $\sigma(\ln F)$ and $\sigma(L_2)$ in $w_{i,1}$ were compared to $\sigma^2(F)$ and $\sigma^2(L_2)$ in $w_{i,2}$, as also proposed for uncertainty modeling in [55] to fuse implicit surfaces in surface inspection sensor data:

$$w_{i,k} = \frac{1}{\mu(L_{2,i})} + \frac{1}{\sigma^k(L_{2,i})} + \frac{1}{\mu((\ln F)_i)} + \frac{1}{\sigma^k((\ln F)_i)}, \quad k \in [1,2]. \tag{4.20}$$

The *cc23* method exhibited the highest difference between the weighting options in Equation 4.20 due to low values for $\mu(L_2)$ and $\mu(\ln F)$ with a higher variation in terms of σ, σ^2, and especially in terms of L_2. Low mean values are of greater importance than low variances or standard deviations as the primary focus is accurate registration. Furthermore, the inclusion of the inverse quadratic σ^2 in *UCSR* yielded a lower weighting in relation to the mean values along with low distances between corresponding points or pixels (L_2). In addition, low deviations from the ground truth transformation (F) are weighted higher than a low variation of the results. Concluding, the confidence-based weights w_i for each individual and valid registration result i in *UCSR* are calculated with the empirically justified $w_{i,2}$ inspired by [55].

In order to avoid the singularity problems discussed in Section 3.2, only singularity-free representations are considered in *UCSR*. The transformations are represented as 3×1 translation vector $\mathbf{t}$ and a quaternion $\mathbf{q}$ that contains the rotational information. Translation $\mathbf{t}$ and rotation $\mathbf{q}$ are weighted according to $w_i = w_{i,2}$ to fuse the calibration results. The fusion is conducted for each element of $\mathbf{t}$ and $\mathbf{q}$ separately, and the weights w_i of m included registration methods are normalized with $1/\sum_{i=1}^{m} w_i$ to

$$\mathbf{t}_{j,\text{conf}} = \frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} \mathbf{t}_{j,i} \cdot w_i, j \in \{x, y, z\}. \tag{4.21}$$

The rotational information described by $\mathbf{q}_{j,\text{conf}}, j \in \{w, x, y, z\}$ cannot be calculated equivalently to the translation due to the mathematical characteristics of quaternions [182]. It is assumed that the rotation $\mathbf{q}$ is only changes slightly in a locally planar manner due to a sufficiently high registration accuracy of the considered registration methods, and the representation of $\mathbf{q}$ in Euler angles is assumed as singularity-free. The

weighted fusion of $\mathbf{q}$ is performed in Euler angles according to Equation 4.21 and converted back to a quaternion $\mathbf{q}$ afterwards.

Concluding, *UCSR* presents a flexible approach for the accurate registration of 2D and 3D multi-sensor systems in unstructured environments. Other approaches for 2D–3D or 3D–3D registration can be integrated flexibly due to the generic design of the *UCSR* framework. With *cnn23*, a CNN-based 2D–3D registration approach is combined with a classic 2D–3D registration approach (*cc23*) and a classic 3D–3D registration method for cross-source point clouds (*graph33*) [329]. Combining the proposed registration methods in *UCSR* can achieve improved, accurate, and stable registration results in challenging, unstructured, and also in manufactured, structured environments.

4.3.2 *cc23*: Classic 2D–3D Cross-Source Registration

The presented, classic 2D–3D registration approach *cc23* is inspired by the contour cues approach of Pujol-Miro et al. [219] for structured scenes and was optimized for the registration of 2D and 3D cross-source data from unstructured outdoor environments in this thesis. The feature detections of *cc23* are based on the assumption that contours can be detected by intensity changes in 2D images and changes in the estimated surface orientations in 3D point clouds similar to [219]. Features from the 2D image and the 3D point cloud, so-called contour cues, are extracted for the cross-source registration inside a common feature space.

4.3.2.1 Projection and Contour Cues

The camera is modeled as an ideal pinhole camera, and the 2D RGB images are undistorted using the camera calibration matrix $\mathbf{K}$. The projection vectors $\mathbf{v}_m$ describe the ray of possible 3D locations for each pixel $m = [i, j]$ and relate the original 2D image to the 3D scene. Each vector represents the 3D line connecting the optical center of the camera at the origin $\mathbf{p}_{\text{or}}^{\text{3D}} = [0\ 0\ 0]^*$ to the corresponding 2D point $\mathbf{p}_m^{\text{2D}} = [x_m y_m]^*$ of the pixel m inside the optical plane of the camera. Hence, the respective $\mathbf{v}_m$

describes all possible locations of $\mathbf{p}_m^{2D}$ in 3D space. With $\mathbf{p}_{or}^{3D} = [0\ 0\ 0]^*$, and $z_m^{3D} = 1$, $\mathbf{p}_m^{3D}$ is calculated with:

$$\begin{pmatrix} x_m \\ y_m \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{i-o_x}{f_x} \\ \frac{j-o_y}{f_y} \\ 1 \end{pmatrix}. \tag{4.22}$$

A gradient threshold t_{gr} and an intensity threshold t_{in} are defined, and only features fulfilling $w_{gr,m} \geq t_{gr}$ and $w_{in,m} \geq t_{in}$ are selected as contour cues. This yields a set of 3D gradient features from the 3D cloud and a set of 2D intensity features from the 2D image as an input to image-to-cloud ICP.

2D contour cues. Canny edge detection analyzes the intensity distribution of neighboring pixels to detect 2D image features. A score $w_{in,m}$ is estimated for each pixel $m = [i, j]$ that corresponds to the projected 3D point $\mathbf{p}_m^{3D}$ depending on the intensity differences of the pixel and its neighboring points. The 3D correspondences for 2D image contours are denoted as $\mathbf{p}_{in,m}^{3D}$. A strong change in intensity is detected by analyzing the luminance values Y_m of each 3D representation $\mathbf{p}_{in,m}^{3D}$ and its neighbors in 3D space. The luminance Y is calculated according to the BT709 luminance definition of the International Telecommunication Union[4] to compare intensity values I_R, I_G, and I_B of each channel: $Y = 0.2126 \cdot I_R + 0.7152 \cdot I_G + 0.0722 \cdot I_B$.

The distance of the center of mass of the luminance values $\mathbf{m}_{Y,m}$ and the geometrical center $\mathbf{m}_{G,N,m}$ of the observed point neighborhood with N_{in} points is measured with $N_{in} = N$ for clarity

$$\mathbf{m}_{Y,N,m} = \frac{1}{\sum_{m=1}^{N} Y_m} \sum_{m=1}^{N} Y_m \cdot \mathbf{p}_{in,m}^{3D} \tag{4.23}$$

$$\mathbf{m}_{G,N,m} = \frac{1}{N} \sum_{m=1}^{N} \mathbf{p}_{in,m}^{3D} \tag{4.24}$$

$$w_{in,N,m} = \left\| \mathbf{m}_{G,N,m} - \mathbf{m}_{Y,N,m} \right\|. \tag{4.25}$$

[4] Recommendation ITU-R BT.709-6: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.709-6-201506-I!!PDF-E.pdf, access on 07.11.2021.

If a sharp contour – a steep change in the intensity level – is located close to the analyzed $\mathbf{p}^{3D}_{in,m}$, a shift between the center of mass and the geometrical center occurs. An additional center of mass is calculated that considers the relative and not the absolute changes in intensity, and distinguishes between thin and thick contours according to [219] with

$$\mathbf{m}_{(1-Y),N,m} = \frac{1}{\sum_{m=1}^{N}(1-Y_m)} \sum_{m=1}^{N}(1-Y_m) \cdot \mathbf{p}^{3D}_{in,m}. \tag{4.26}$$

The final score for each 2D feature is calculated with $N = N_{in} =\in \{4, \dots, 8\}$ using

$$w_{in,N,m} = \min\left(\left\|\mathbf{m}_{G,N,m} - \mathbf{m}_{Y,N,n}\right\|, \left\|\mathbf{m}_{G,N,m} - \mathbf{m}_{(1-Y),N,m}\right\| \right) \tag{4.27}$$

to determine the weight $w_{in,N,m}$ for each $\mathbf{p}^{3D}_{N,m}$ within the connectivity neighborhood N. The average distance of the N neighbors to the analyzed $\mathbf{p}^{3D}_{in,m}$ is calculated with $\sum_{o=1}^{N} r_N$, $N_{min} = 4$, and $N_{max} = 8$, and a normalized intensity score $w_{in,m}$ combines the scores to achieve comparable results:

$$w_{in,m} = \frac{1}{(N_{min} - N_{max})} \sum_{N=N_{min}}^{N_{max}} w_{in,N,m} \cdot \frac{1}{\sum_{o=1}^{N} r_N}. \tag{4.28}$$

Pujol-Miro et al. [219] propose an optional extraction of intensity features from LiDAR reflectance measurements that can be integrated similar to 2D intensity features. This is not included in *cc23* to preserve its generic applicability for point clouds generated from the fusion of multiple and potentially different LiDAR sensors.

3D contour cues. 3D gradient features in the point cloud are identified by a multiscale analysis, as proposed in Pauly et al. [213]. Points with high variation in the surface normals' direction are considered as points of major importance, such as object boundaries. The local neighborhood of $\mathbf{p}^{3D}_{gr,m}$ is mapped inside a scatter matrix that is interpreted as covariance matrix $\mathbf{C}_{gr}$ using its N_{gr} neighboring points. The estimation of surface orientations from the covariance matrix is conducted, as described in Section 4.2.2. An SVD of $\mathbf{C}_{gr}$ yields the estimated surface orientation for each point $\mathbf{p}^{3D}_{gr,m}$. The surface variation $s_N(\mathbf{p}^{3D}_{gr,m})$ is calculated according to Equation 3.11. The stepwise increase of the number of neighbors N

introduces the multi-scale character of the method described in [213]. Each point N with a threshold $t_{\mathrm{gr}} \leq s_N(\mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},m})$ is counted by the persistence of the surface variation s_N. Hence, the persistence $w_{\mathrm{gr},m}$ maps the dependency of the surface variation for a point $\mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},m}$ of the number of included neighboring points N. It also constitutes the selection criteria for a 3D point $\mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},m}$ of the cloud as a 3D feature by requiring a persistence that exceeds the threshold t_{gr}.

Image-to-Cloud ICP. The *cc23* method utilizes image-to-cloud ICP for registration as proposed in [219]. Contrary to the classic ICP approach of [15], image-to-cloud ICP minimizes the distance r between 2D image pixels using the 3D points lying on the projection vectors $\mathbf{v}_m$ and the 3D points $\mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},n}$ of the point cloud for each correspondence with

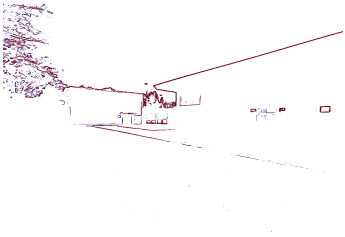
$$r = \frac{\left\| \mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},m} \bullet (\mathbf{p}^{\mathrm{3D}}_{\mathrm{gr},m} - \mathbf{v}_m) \right\|}{\|\mathbf{v}_m\|}. \tag{4.29}$$

4.3.2.2 Proof of Concept: Unstructured Environments

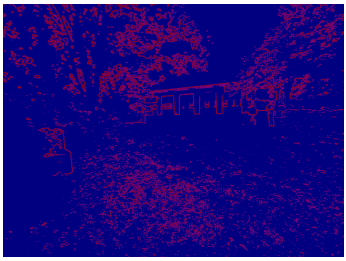
The *cc23* method is demonstrated on 2D RGB images and 3D LiDAR point clouds ($\mathcal{L}$) of the *IOSB-Reg* dataset. Table 4.5 specifies the *cc23* parameterization that presented the most promising results for registering 2D RGB images from a JAI AD-130GE camera and 3D point clouds of a Velodyne HDL-64E ($\mathcal{L}$) in unstructured outdoor environments. For the Canny edge detection of 2D intensity features, a kernel size of 3, a low threshold of 100, and an upper threshold of 300 proved useful for the analyzed data from unstructured environments. With a step size $\Delta N = 1$ for intensity and gradient features, N is increased by one for $N_{\mathrm{in}} \in \{N_{\mathrm{min,in}}, \ldots, N_{\mathrm{max,in}}\}$ and $N_{\mathrm{gr}} \in \{N_{\mathrm{min,gr}}, \ldots, N_{\mathrm{max,gr}}\}$. Figure 4.8 shows the *cc23* feature extraction results from the intensity values of two exemplary 2D RGB images and the gradients of the corresponding Velodyne HDL-64E 3D point clouds. Table 4.6 and Table 4.8 demonstrate that *cc23* clearly outperforms other classic registration approaches on data from unstructured environments: ICP, CSGM ($l_{\mathrm{vox}} = 0.24\,\mathrm{m}$ [126]), and the subsequently discussed *graph33*.

(a) Original 2D RGB image.

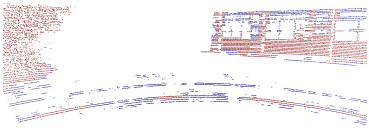
(b) Original 2D RGB image.



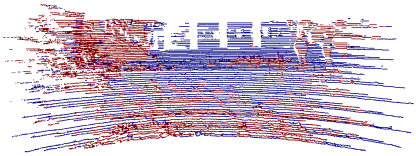
(c) 2D int. features image for (a).



(d) 2D int. features image for (b).



(e) 3D grad. features $\mathcal{L}$ for (a).



(f) 3D grad. features $\mathcal{L}$ for (b).

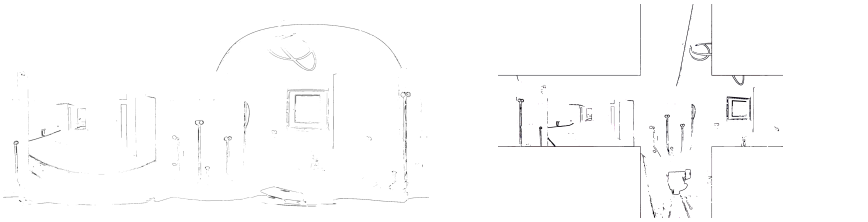
Figure 4.8 Contour cue extraction and registration in *cc23*. Blue color indicates a low and red a high feature relevance ($w_{\text{in},m}$, $w_{\text{gr},m}$) for registration. $\mathcal{L}$ is projected onto the 2D image plane and colored according to the associated depth. The background contrast in (d) is chosen to highlight the difficulty of the 2D contour cue extraction in images from unstructured environments. Images (a)–(f) © Fraunhofer IOSB.

Parameter	*UO*	*SI*	Description
$N_{\text{min,in}}$	3	5	Min. neighbors for 2D intensity features
$N_{\text{max,in}}$	10	25	Max. neighbors for intensity features
$N_{\text{min,gr}}$	20	5	Min. neighbors for 3D gradient features
$N_{\text{max,gr}}$	100	75	Max. neighbors for gradient features
$t_{\text{min,in}}$	0.25	0.30	Lower threshold for intensity features
$t_{\text{max,in}}$	0.60	0.80	Upper threshold for intensity features
$t_{\text{w,in}}$	0.04	0.01	Weight threshold for intensity features
$t_{\text{min,gr}}$	0.30	0.30	Lower threshold for gradient features
$t_{\text{max,in}}$	0.70	0.80	Upper threshold for gradient features
$t_{\text{w,gr}}$	0.04	0.05	Weight threshold for gradient features

Table 4.5 Experimentally justified parameterization for *cc23* feature extraction to register 2D and 3D cross-source data from structured indoor (*SI*) and unstructured outdoor (*UO*) environments.

4.3.2.3 Proof of Concept: Structured Environments

The *cc23* method is also applicable for structured indoor environments, e.g., for registering 360° images to 3D models of a building. Digital representations of buildings are typically modeled in the Building Information modeling (BIM) format saved in the Industry Foundation Classes format (IFC). The feature extraction from a BIM model can be conducted error-free as it directly contains the ground truth features, and a dense 3D point cloud can be generated from the BIM model, as detailed in Section 4.2.3. In contrast to the intensity feature extraction from unstructured environments, a preprocessing that filters short and strongly curved contours benefited the feature extraction step. Subsequently, the contour cues were passed on to the point-to-line ICP step for registration. Figure 4.9 shows an example of the contour cue extraction from non-rectified 360° images of the same indoor scenery. In the special case of non-rectified 360° images, the representation as a cube can limit the influence of distortion effects in feature extraction. Both results showed satisfactory intensity features with proper feature extraction for structured indoor environments. This experimentally justifies the *cc23* parameters for structured indoor environments given in Table 4.5.



(a) 2D int. feature extraction. (b) 2D int. features for cube representation.

Figure 4.9 *cc23* feature extraction in structured indoor environments with two different representations: Image (a) shows the extracted intensity features from the original, non-rectified 360° image, while (b) shows the intensity feature extraction for the alternative cube representation. Images © Fraunhofer IOSB.

4.3.3 *cnn23*: 2D–3D Cross-Source Registration with Neural Networks

The 2D–3D *cnn23* method relies on neural networks to register 2D–3D cross-source data and is inspired by RegNet proposed in [246]. RegNet was the first CNN providing an extrinsic calibration of multimodal sensors with six DoF. It achieves accurate and promising results and outperforms classic 2D–3D approaches in terms of accuracy on data from structured outdoor environments. Schneider et al. [246] train different RegNet-type CNNs for different magnitudes of artificial decalibration. Contrasting the iterative refinement approach of Schneider et al. [246], *cnn23* only trains one network for a maximum decalibration of 20° and 1.5 m. This provides a flexible, generic, and robust approach for multi-sensor registration in the application on off-road vehicles as the initial decalibration is not always known in sufficient detail. The presented *cnn23* method is optimized for unstructured environments with an in-depth data augmentation as well as transfer training on data from unstructured environments.

4.3.3.1 Preprocessing and Network Architecture

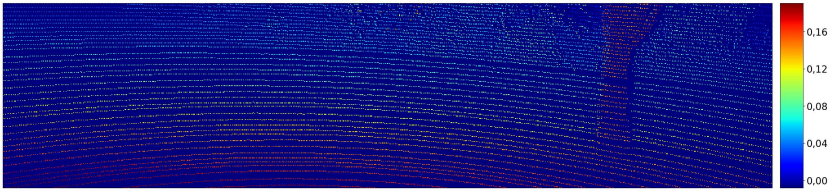
At first, the *cnn23* method projects the 3D LiDAR point cloud on the sensor image plane as depth image with pixels $[i, j]$ for the registration of a 2D image to a 3D point cloud in 2D space:

$$z_c \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{T}_i \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}. \tag{4.30}$$

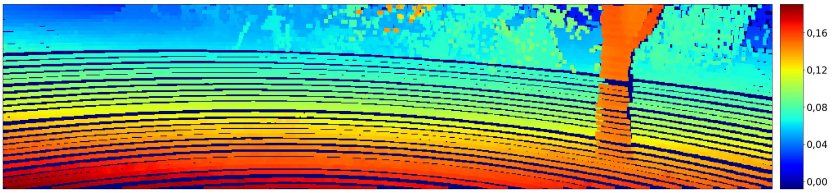
Each 3D point in the camera frame c relates the corresponding pixel $m = [i, j]$ to the sensor origin with the initial transformation estimate $\mathbf{T}_i$ and keeps the measured LiDAR distance as inverse depth in z_c. Consequently, the projected LiDAR depth map contains z_c from LiDAR data or $z_c = 0$ for pixels without a corresponding 3D LiDAR point.

The depth image can be reconstructed using the corresponding inverse depth z_c for each pixel m. With non-inverse depth values, the maximum pooling layer at the input to the *cnn23* architecture uses the highest depth value of a neighborhood as a reference pixel. This leads to the assignment of greater depth values to neighboring pixels with an originally smaller depth value. Inverse depth values z_c avoid occlusion errors and reduce artifacts in the maximum pooling stage. Figure 4.10 shows the projected depth image prior to and after two maximum pooling layers that generated a notably denser image.

Furthermore, the input data – camera image and LiDAR depth image – are centered to achieve a similar scaling in the training data. Schneider et al. [246] state that Network-in-Network (NiN) blocks have more favorable convergence characteristics than standard convolutional layers. Weight initialization for LiDAR depth image feature extraction, feature matching, and regression is conducted with the Xavier initialization method [90]. Furthermore, feature extraction and matching in *cnn23* are conducted in NiN blocks, as proposed in [170]. Here, three convolutional layers form a NiN block with ReLU activation functions followed by a maximum pooling layer. The first convolutional layer in the NiN block defines its filter size such that a NiN_3 block has a 3×3 convolutional layer as a first layer. The second and third convolutional layers inside a NiN block are



(a) Sparse, raw depth image.



(b) Densified depth image after two max poolings.

Figure 4.10 Projected LiDAR depth input images. Images © Fraunhofer IOSB

fully connected (with filter sizes 1×1), and the number of filters inside each of these three layers is identical.

Feature Extraction. The LiDAR depth image is subject to two maximum poolings before feature extraction, which notably increases the density of the LiDAR feature map and approximates it to the density of the pixel-wise dense RGB image. Feature extraction for RGB image and LiDAR depth map uses NiN_{11} blocks followed by NiN_5 and NiN_3 blocks. Two NiN blocks match the features after concatenating the results from RGB and depth image extraction, and a last NiN block with a depth of 512 concludes the feature extraction of both RGB and depth images. The initial weights for the feature extraction from RGB images were chosen as proposed by Lin et al. [170] for their participation in the ImageNet challenge [47, 233] with NUS-BST. As ImageNet aims at image classification, the last NiN block of [170] for classification was omitted. Furthermore, feature extraction weights for RGB and depth features were not shared as a different number of filters were used.

Feature Matching. Feature matching combines the extracted features from the RGB and the interpolated LiDAR depth image and finally determines matching pairs of RGB–depth features for the global regression

step. It concatenates the feature maps from the NiN_3 block inside a NiN_5 and NiN_3 block, which yields 512 feature channels for regression succeeding the second NiN block in feature matching.

Regression. The global regression step pools the information from both sensors inside two fully connected layers succeeded by an Euclidean loss function. Schneider et al. [246] state that quaternion representations clearly outperform the representation in Euler angles. Hence, *cnn23* represents the translation as three element vector $\mathbf{t}$ and the rotation as a four element quaternion $\mathbf{q}$. Regression is built with a fully connected layer of depth 512 and two separated paths for translation and rotation with two fully connected layers for both translation and rotation. The first NiN block of depth 512 uses 5×5 filters, and the two separated paths consist of NiN blocks with 3×3 filters. Regression estimates the translation in two fully connected layers of depth 256 and 3, while rotation is estimated with layers of 256 and 4 depth.

4.3.3.2 Training, Validation, and Testing

Pre-training for *cnn23* was conducted on the KITTI 2012 dataset [80]. Fine-tuning was performed on the *IOSB-Reg* dataset described in Section 7.4 to adapt *cnn23* to unstructured environments. KITTI contains 13,084 2D–3D pairs for training, and *IOSB-Reg* provides 146 2D–3D pairs for the fine-tuning of *cnn23*. A validation split of $15/146 = 0.103$ proved useful for *IOSB-Reg* in domain adaption, and 115 image–cloud pairs from KITTI were excluded from training. Testing was only conducted on *IOSB-Reg* data as the focus is the registration of 2D and 3D data from unstructured environments.

The *cnn23* method was trained in a supervised manner, and training data was generated by the artificial rotation and translation of 2D images and 3D point clouds. The decalibration range $[-\max(\mathbf{T}_d), \max(\mathbf{T}_d)]$ defines the interval for the extraction of randomly decalibrated training data. For a proper training of *cnn23*, decalibration with a maximum translation of 50 px and a maximum rotation of 15° proved useful. The translation shift in pixels is applied inside the image plane of the 2D RGB and depth images along the x and y axes, while the rotation is applied around the z axis that captures the depth in the corresponding intensity value of the depth image pixel. The input size of the images was restricted

to 255 × 942 px as indicated by the green box in Figure 4.12 to facilitate a proper data augmentation.

The ground truth transformation of the *IOSB-Reg* dataset $\mathbf{T}_{\mathrm{GT}}$ was measured by hand and validated by the visual assessment (see Section 3.9). The known decalibration $\mathbf{T}_d$ was generated by adding the randomly generated decalibration $\mathbf{T}_d \in [-\max(\mathbf{T}_d), \max(\mathbf{T}_d)]$ to the ground truth $\mathbf{T}_{\mathrm{GT}}$ and set as initial transformation $\mathbf{T}_i$. The desired registration output is equivalent to the decalibration $\mathbf{T}_d$ described with the translational elements $\mathbf{t}$ and rotational elements $\mathbf{q}$. Both datasets KITTI and *IOSB-Reg* were augmented using rotation and shifting. Here, the augmentation according to [246] tripled the size of the datasets. The L_2 and F norms measure the registration performance of *cnn23* on the test splits of the KITTI and the *IOSB-Reg* datasets, as described in Section 3.11.

The ADAM optimizer [151] was chosen for training as the adaptive determination of the learning rate for each parameter depending on floating mean value and squared gradient proved favorable. The state-of-the-art parameters in [151] proved useful and were selected to train *cnn23*: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\eta = 10^{-8}$. Euclidean loss is applied for translation $\mathbf{t}$ and rotation $\mathbf{q}$. Hence, the Euclidean distance between the network-estimated decalibration vectors $\hat{\mathbf{t}}_d$ and $\hat{\mathbf{q}}_d$ and the randomly applied decalibrations $\mathbf{t}_d$ and $\mathbf{q}_d$ are combined into a 7D vector $\mathbf{l}_d$ to calculate the loss L_{cnn23}, according to Schneider et al. [246]:

$$L_{cnn23} = \sqrt[2]{\sum_{i=1}^{7} (\hat{\mathbf{l}}_d - \mathbf{l}_d)^2}, \tag{4.31}$$

$$\text{with} \quad \mathbf{l}_d = [t_1, t_2, t_3, q_1, q_2, q_3, q_4]. \tag{4.32}$$

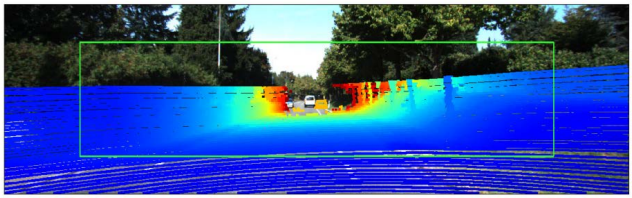
Training, activations, iterative registration, and augmentation were analyzed on the KITTI 2012 and the *IOSB-Reg* dataset on the basis of a right-handed coordinate system[5]: depth is measured positively along the z axis, x points to the right and y downwards.

The training of the *cnn23* network on the KITTI dataset with the augmentation methods proposed in [246] (*cnn23-K*) was evaluated initially

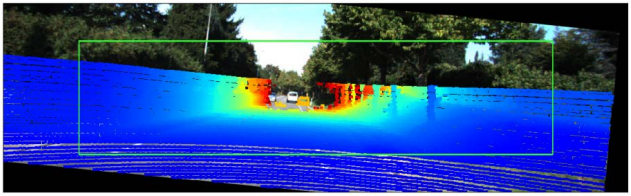
[5] Sensor setup: http://www.cvlibs.net/datasets/kitti/images/setup_top_view.png, access on 24.12.2021.

with a learning rate of 10^{-5} and a validation split of $115/13,084 = 0.0089$. Here, the training and validation losses showed signs of over-fitting for large initial decalibrations, and in some cases the global regression result did not properly correct the decalibration of the input data. Thus, it is assumed that with the training according to [246] the network tends to learn the ground truth rather than a proper analysis of the input data. Furthermore, it is assumed that these erroneous characteristics are related to the data structure of the KITTI dataset. KITTI was captured with one vehicle in urban and suburban environments as well as on highways, and the distribution of structured and unstructured areas in relation to the vehicle orientation is often identical: the front and back areas mainly have a structured character, and the road with a rather smooth surface and uniform color mostly dominates the center of the image, while the sides of the captured area can mostly be characterized as unstructured environments, as further discussed in Section 6.2.1. Hence, the structure of all RGB and depth input images is very alike and probably too similar to allow correct learning of the feature extraction and matching process required for proper registration. The domain transfer discussed in Section 6.1.4 justifies the critical influence of too similar image and point cloud structure on CNN training. Figure 4.11 compares the inference results obtained with data augmentation and training, as described in [246] (*cnn23-K*), to the proposed enhanced augmentation [335] (*cnn23-K-e*) for large decalibrations.

In order to prevent over-fitting here, the enhanced data augmentation *cnn23-K-e* is proposed instead of *cnn23-K* optimizing *cnn23* for the registration of cross-source data from unstructured environments [335]. 2D images and 3D point cloud data after conversion to depth images are normalized with the standard deviation of the complete image or by scaling the pixel intensity values within $[0, 1]$ or $[-1, 1]$ to become independent of different exposure conditions and depth ranges. Furthermore, the input data – LiDAR depth image and camera image – is normalized by subtracting the mean value of all pixel intensities from the value of each single pixel. Each pairing of a projected depth image and an RGB image is augmented with additional and suitable translational shifts and rotations to achieve a uniform distribution in $[-\max(\mathbf{T}_d), \max(\mathbf{T}_d)]$. The occurrence of invalid image areas due to missing pixel information from



(a) Registration with augmentation according to [246] (*cnn23-K*).



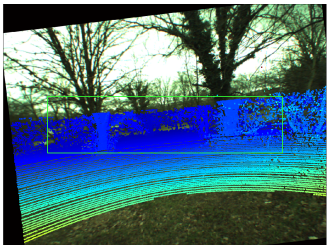
(b) Registration with proposed enhanced augmentation for *cnn23-K-e*.

Figure 4.11 Comparison of the registration results with the two analyzed data augmentation strategies to train *cnn23*: Image (a) depicts the registration result with augmentation as proposed by Schneider et al. [246], while (b) shows the *cnn23* registration results for a training with the enhanced augmentation proposed in this thesis (*cnn23-K-e*). Only *cnn23-K-e* achieved a valid registration result. The initial decalibration was -4 px along x, -32 px along y, and a rotation with –5.59° around z (clockwise).

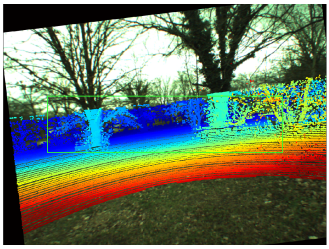
the RGB or the depth image limits the maximum decalibration range of the input data, and the image area is reduced as visualized with the green box in Figure 4.12.

Experimental evaluation showed that fine-tuning on *IOSB-Reg* (*cnn23-I*) and the proposed enhanced augmentation of the training data could prevent over-fitting of *cnn23-I* and *cnn-K-e* such that regularization and dropout are not required [335]. The number of epochs was not fixed due to the generation of new training data during training by random decalibration. The most effective iterations were selected using a batch size of one to also facilitate network training with limited computation capacity. The number of iterations varied from one to three million.

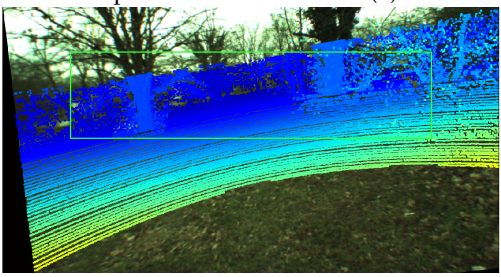
To conclude, the *cnn23* model was pre-trained on KITTI with the proposed enhanced augmentation (*cnn23-K-e*) and subsequently trained on



(a) Decalibrated input.



(b) Ground truth.



(c) *cnn23* registration result.

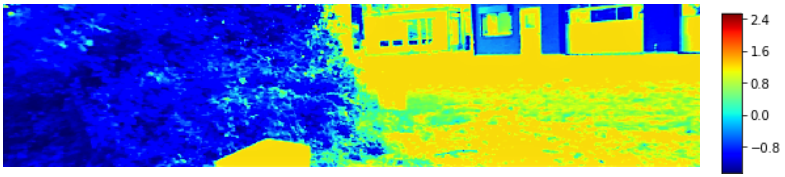
Figure 4.12 Decalibrated input, ground truth, and 2D–3D *cnn23* (*cnn23-K-e*) registration results on *IOSB-Reg* data after enhanced, uniform augmentation training on KITTI 2012 and fine-tuning on *IOSB-Reg*. Images © Fraunhofer IOSB.

146 unstructured scenes of the *IOSB-Reg* dataset for a domain adaption to unstructured environments (*cnn23-I*, see Figure 4.14). The reduction of the learning rate on the *IOSB-Reg* data to 10^{-6} prevented the loss of the trained feature filters, and *cnn23* with pre-training on KITTI showed a promising generalization performance, as demonstrated subsequently.

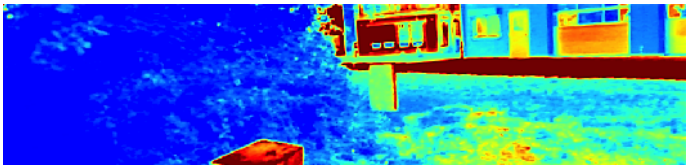
4.3.3.3 Proof of Concept: *cnn23*

The *cnn23* training was conducted on a cluster of eight NVIDIA Tesla V100 GPUs for 5,000,000 iterations (around 79 hours) and on an NVIDIA RTX 2080S GPU for 321,200 iterations. The *cnn23* method was tested on different decalibrations with up to 2.0 m translation along each axis and 20° around each axis. Even if only a small overlap of the 2D image and the respective 3D cloud was available, as is the case for decalibration

(a) 2D RGB input to *cnn23*.



(b) 1st filter (i=1) of 1st *cnn23* layer for RGB image feature extraction.



(c) 2nd filter (i=2) of 1st *cnn23* layer for RGB image feature extraction.

Figure 4.13 Image (b) and (c) visualize selected filter activations of the first *cnn23* layers (96 filters with 11 × 11 kernels) extracting 2D features from the image depicted in (a). The activations show a preference for structured elements. Images © Fraunhofer IOSB.

with 2.0 m and 20° on an image with 942 × 225 px, a registration accuracy of ln F = -0.463 and L_2 = 2.566 m was achieved. Figure 4.12 depicts the registration result with *cnn23* on exemplary 2D image and 3D point cloud data from the *IOSB-Reg* dataset. Enhanced augmentation was performed with a maximum of 15° rotation and with up to 50 pixels translation in the horizontal and vertical direction of the 2D data, and *cnn23* achieved a mean translation error of 4.9 cm in x, 3.0 cm in y, and 7.6 cm in z, as well as a mean rotational accuracy of 1.24° around x, 0.43° around y, and 1.47° around z.

An analysis of the layer activations in 2D RGB and LiDAR depth image feature extraction underlines the difficulty of registering data from

unstructured environments compared to structured environments. Figure 4.13 shows the first layer activations in 2D RGB feature extraction. Here, the artificial, structured test objects provide the expected insight that structured elements and clearly separated objects with defined borders are preferred for registration, while bush elements with similar texture and close to the sensors only contributed very little to the registration. In contrast, higher weights were assigned to the slightly overexposed grassland part of the image rather counter-intuitive for human vision. For LiDAR depth images, the characteristic ring structure of rotating 3D LiDAR sensors was still dominant in the activations of the initial layers but lost its influence in subsequent layers with a higher focus on global characteristics. Naturally, the preference for structured elements adds to the challenge of registering cross-source data. However, the proposed customizations within the analyzed methods facilitated the registration without calibration targets and human intervention in hazardous environments.

Figure 4.14 compares the performance of *cnn23* with training according to [246] on both the KITTI dataset (denoted *cnn23-K*) and the KITTI dataset with fine-tuning on *IOSB-Reg* (*cnn23-I*). The initial decalibration is given as a reference. As expected, the domain adaption training on the *IOSB-Reg* dataset improved the validation performance on the *IOSB-Reg* dataset with data from unstructured environments and slightly degraded the performance on the KITTI validation data. Furthermore, Figure 4.14 analyzes the influence of the proposed enhanced augmentation by comparing the validation results on the KITTI dataset's augmented validation data with non-augmented validation data. It shows that the enhanced augmentation of the training data improved the performance of *cnn23* in both cases. Hence, the notably higher registration accuracy of *cnn23* with an enhanced augmentation of the training data justifies the enhanced augmentation. Furthermore, a more stable performance for different decalibration scenarios of the input data is indicated. Table 4.8 compares the resulting error metrics of the proposed *cnn23* method to *cc23* and *graph33* and shows that *cnn23* clearly outperforms the two classic approaches.

Furthermore, Schneider et al. [246] applied an iterative registration approach and executed RegNet several times with increasing registration accuracy. This approach was evaluated for *cnn23* with a special focus

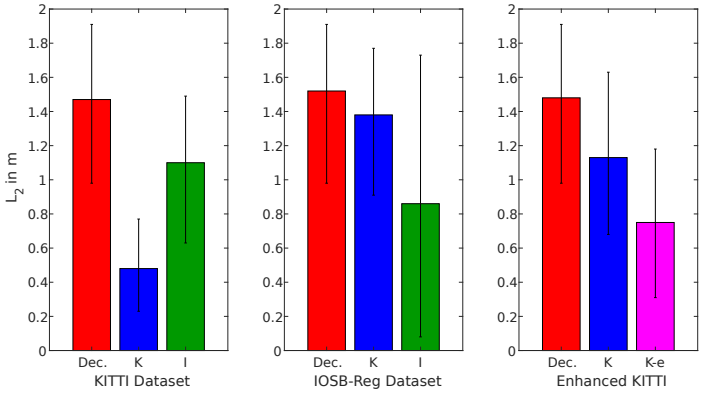


Figure 4.14 Registration accuracy of *cnn23* in terms of L_2 on the chosen validation data: image (a) compares the *cnn23* accuracy on KITTI (*cnn23-K*, K) and *IOSB-Reg* (*cnn23-I*, I) data with standard augmentation according to [246] on the KITTI dataset, while image (b) compares the accuracy on K and I for training on KITTI according to [246] with fine-tuning on *IOSB-Reg* data, and image (c) shows that the *cnn23* registration accuracy with the proposed enhanced augmentation (K-e) notably increases the registration accuracy of *cnn23* in comparison to the augmentation according to [246] (K).

on registering multimodal data from unstructured environments for up to five iterations of *cnn23* and after fine-tuning on the *IOSB-Reg* dataset. The iterative approach did not improve the registration accuracy for level S and M decalibrations (see Section 3.11). However, a second run of *cnn23* that builds on the registration result of the first *cnn23* run slightly improved the registration accuracy for level L decalibrations.

CalibNet [105] achieved a mean accuracy of 0.043 m and 0.41° for a rather small maximum decalibration of ±20° and 0.2 m (see Section 2.3.3), while Schneider et al. [246] evaluated a maximum translation of 1.5 m and a rotation of 20° and achieved a mean calibration error of 0.06 m and 0.28° on the KITTI 2012 data from structured environments [83]. However, the proposed *cnn23* method outperformed RegNet [246] on data from unstructured environments, as depicted in Figure 4.14. CalibNet of Handa et al. [105] achieved a slightly more accurate registration on KITTI 2012 than *cnn23* but for notably lower initial decalibrations. Concluding,

cnn23 can provide valid and robust results for 2D and 3D cross-source data from unstructured environments. A mean registration accuracy of 0.052 m in translation, 1.24° around x, 0.43° around y, and 1.47° around z was achieved on the *IOSB-Reg* test data with level L decalibrations (±1.0 m, ±17.2°). Hence, *cnn23* achieved sufficient accuracy to register cross-source sensor data from unstructured environments for off-road vehicles, as detailed in [323] and [324].

4.3.4 *graph33*: Classic 3D–3D Cross-Source Registration

Deng et al. [48] state that Gaussian Mixture model approaches fail in registering clouds with different densities, thus making them unsuitable for cross-source registration. The authors [48] state that their approach is less sensitive to noise and outliers than Gaussian Mixture models but it also indicated high sensitivity to those problems on the evaluated, structured, similar-source data. The approach of [189] with RANSAC and downsampling achieved promising results for highly dense point clouds but proved difficult for less dense point clouds due to missing data and their different noise characteristics.

Huang et al. [126] propose the transformative CSGM registration approach with a combination of local and global characteristics that provides a promising approach to reduce the influence of cross-source data characteristics, such as notably different point densities, artifacts, and outliers. Hence, CSGM is chosen as a basic method for *graph33* – a classic 3D–3D registration method for cross-source point clouds from unstructured environments prevailing in off-road robotics. The authors [126] demonstrate CSGM on different types of 3D point clouds from structured environments, and the multitude of challenging conditions encountered in registering cross-source data from unstructured environments in *graph33* require an adaption and extension of the basis CSGM method proposed in [126]. Depth estimation inaccuracies, such as in stereo image 3D reconstruction, further complicate the registration process. Hence, *graph33* extends and optimizes CSGM for the registration of point clouds from rotating 3D LiDAR sensors ($\mathcal{L}$) and point clouds from stereo image disparity estimation ($\mathcal{S}$) from unstructured environments. The *graph33* method works as follows:

1. Over-segmentation with VCCS supervoxels,

2. Encode local details in ESF640 descriptors,
3. Extract global graph structure,
4. Factorized graph matching (FGM).

Contrasting CSGM [126], *graph33* includes multiple optimizations for cross-source data from unstructured environments:

- Condensation of $\mathcal{L}$ (prior to over-seg.),
- Transformation to homogenization coordinates (prior to over-seg.),
- Additional descriptors including prior knowledge (Section 4.3.4.2),
- And customized correspondence rejection following FGM.

4.3.4.1 Feature Extraction and Homogenization

The supervoxel adjacency graph represents the global structure with the local voxel centroid points as nodes. ESF640 descriptors [295] describe the local structure of the point set inside each extracted VCCS supervoxel for further processing. The authors [295] state that ESF640 surface approximation increases the robustness to noise, outliers, and different densities for cross-source data. Hence, they were chosen to register sensor data from unstructured environments.

At first, $\mathcal{L}$ is condensed by a fusion of subsequently captured "single-shot" clouds to obtain a higher point density for the subsequent registration process. Here, eleven point clouds were captured in a static environment to inhibit inaccuracies due to motion blur or moving objects. Their condensation, as detailed in Section 4.2.2, proved useful, and additional registration was not required in contrast to [82] (see Figure A.6).

In order to overcome the problem of notably different densities, both point clouds are transformed into homogenized coordinates prior to the supervoxel clustering [329]. Contrasting Huang et al. [126] working with a Cartesian point representation, this thesis proposes homogenization with

$$r = \frac{z}{10}, \quad \phi = \arctan\frac{y}{x}, \quad \psi = \arctan\frac{\sqrt{x^2 + y^2}}{z} \tag{4.33}$$

which achieved a rather uniform distribution of the points, especially inside 3D LiDAR point clouds. This homogenization procedure is similar to the geometric 3D analysis described in Section 3.6 with an additional reduction of the scale of z and a dependency of ψ on z. This dependency

leads to an increasing cluster size of the voxels linearly dependent on their distance to the sensor origin.

After densification and homogenization, the clouds are subdivided into VCCS supervoxels via the normalized distance measure D of two supervoxels:

$$D = \sqrt{\frac{\lambda_c D_c^2}{\zeta^2} + \frac{\nu D_s^2}{3R_{\text{seed}}^2} + \kappa D_{\text{FPFH}}^2}. \tag{4.34}$$

R_{seed} is the initial seed resolution of the voxels and normalizes the spatial component. D_c denotes the Euclidean distance of the color information in CIELAB color space and is normalized by a fixed constant ζ [209], while D_s is the spatial distance. FPFH features are only utilized for supervoxel extraction as proposed in [126], and D_{FPFH} denotes the distance between the FPFH features calculated according to [8].

Different parameters for the size as well as for the weights of the voxel criteria were evaluated. For Cartesian coordinates, appropriate results were achieved by setting a voxel size of $l_{\text{vox}} = 0.10\,\text{m}$. Over-segmentation weights for VCCS supervoxels extraction are color information (λ_c), normal direction (ν), and spatial distribution (κ). The best results were obtained with $\lambda_c = 0.6$, $\nu = 1.0$, and $\kappa = 0.6$ for $\mathcal{S}$ and with $\nu = 1.0$ and $\kappa = 0.6$ for $\mathcal{L}$ without color information. An $l_{\text{vox}}/R_{\text{seed}}$ ratio of 2/25 follows the suggestions of [126, 209] and yields $R_{\text{seed}} = 1.25\,\text{m}$ for Cartesian coordinates. For homogenized coordinates, $l_{\text{vox}} = 0.04\,\text{rad}$ achieved a rather uniform point distribution.

The proposed homogenization prior to over-segmentation increased the robustness of *graph33* to artifacts, noise, and the limited estimation accuracy itself. Hence, both input clouds are transformed into homogenized coordinates prior to over-segmentation in *graph33*. Clusters close to the origin were small, whereas large distances lead to the formation of big clusters. With the registration of $\mathcal{L}$ to $\mathcal{S}$ clouds, the over-segmentation in homogenized coordinates also increased the robustness of *graph33* as it coped well with the quadratically increasing inaccuracy in stereo image disparity estimation.

4.3.4.2 Descriptors, Graph Representation, and Correspondence Rejection

Descriptors. CSGM of [126] relies on an automated scaling of the point clouds via bounding boxes. However, this bounding box approach requires clearly separated objects with well-defined boundaries that are not present in unstructured environments. Hence, the bounding box method of [126] is not applicable in *graph33*, and a priori knowledge is integrated to substitute the automated scaling.

It is known that the sensors are mounted onto the same platform, which ensures that they approximately observe the same part of the scene. Three additional descriptors are proposed in this thesis complementing the ESF640 descriptor ($\mathbf{A}_E$) to represent this knowledge in the graph matching problem:

- $\mathbf{A}_N$: estimated normal orientation of the voxel centroid,
- $\mathbf{A}_R$: distance of respective voxel centroid point to sensor origin,
- $\mathbf{A}_Z$: angle of the vector that connects the voxel centroid to the sensor origin in relation to the z axis of the camera frame c (negative depth).

An affinity matrix is generated for each descriptor: ESF640 ($\mathbf{A}_E$), distance to origin ($\mathbf{A}_R$), angle to z ($\mathbf{A}_Z$), and normal ($\mathbf{A}_N$). Normalization and merging of the affinity matrices into one affinity matrix $\mathbf{A}$ by an element-wise summation of the matrices proved useful to combine the proposed descriptors in *graph33*. Each merged descriptor $\mathbf{A}$ is again normalized for further processing and represented with its affinity matrix $\mathbf{A}$ as a node in the subsequent graph. As a results, registration is turned into a graph matching problem in feature space.

Graph Matching. The global graph structure consists of graph nodes and edges, and the graph matching problem is solved with FGM, as proposed in [126, 314]. The merged descriptors $\mathbf{A}$ constitute the nodes W and are derived from the voxel centroid points. The edges Q represent the adjacent relations between the supervoxels. The normalized distance D between the descriptors is computed according to Equation 4.34. The distance $r_{i,j}$ of two nodes W_i and W_j is given by

$$r_{i,j} = \frac{r_W}{R_{\text{seed}}}, \text{with} \quad L_2 \text{ distance} \quad r_W = \left\| \mathbf{W}_i - \mathbf{W}_j \right\|_2 . \tag{4.35}$$

The Euclidean distance $r_{i,j}$ and the Euler angles $\boldsymbol{\theta}$ between W_i and W_j are combined into a descriptor $\boldsymbol{\theta}_{\mathbf{W}}(\boldsymbol{\theta}, r_{i,j})$.

A graph is described by $\mathcal{G} = \{\mathbf{W}, \mathbf{Q}, \mathbf{G}\}$ with n nodes, m directed edges, and the feature matrices

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n] \in \mathbf{R}^{d_\mathbf{w} \times n} \quad \text{and} \quad \mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_m] \in \mathbf{R}^{d_\mathbf{q} \times m} \tag{4.36}$$

with $\mathbf{d}_w = \dim(\mathbf{w})$ and $\mathbf{d}_q = \dim(\mathbf{q})$.

A node-edge incidence matrix $\mathbf{G} \in [0,1]^{n \times m}$ specifies the graph's topology. The node affinity matrix $\mathbf{A}_\mathbf{W} \in \mathbf{R}^{n_1 \times n_2}$ measures the similarity of each possible node pair of $\mathcal{G}_1$ ($\mathbf{w}_i$) and $\mathcal{G}_2$ ($\mathbf{w}_j$) in feature space with

$$\mathbf{A}_{W(i,j)} = \frac{r_{i,j}}{\max_{i,j} r_{i,j}}. \tag{4.37}$$

The edge affinity matrix $\mathbf{A}_Q \in \mathbf{R}^{m_1 \times m_2}$ quantifies the similarity between all possible edge pairs of $\mathcal{G}_1$ and $\mathcal{G}_2$. The graph edges are represented with the descriptor $\boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})$ to ensure a correct representation of the graph structure. The relative orientation of the two merged descriptors for $\mathcal{G}_1$ and $\mathcal{G}_2$ with affinity matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ in the feature space is represented with

$$\boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j}) = \begin{pmatrix} \phi_x \\ \phi_y \\ \phi_z \\ r_{i,j} \end{pmatrix} = \begin{pmatrix} \arccos(\frac{x}{r_{i,j} \cdot \sin(\phi_z)}) \\ \arccos(\frac{y}{r_{i,j} \cdot \sin(\phi_z)}) \\ \arccos(\frac{z}{r_{i,j}}) \\ \left\| \mathbf{W}_i - \mathbf{W}_j \right\|_2 \end{pmatrix}, \tag{4.38}$$

and $\boldsymbol{\theta}$ provides the input for the edge affinity matrix

$$\mathbf{A}_{\boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})} = \frac{\left\| \boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})_i - \boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})_j \right\|_2}{\max_{i,j} (\left\| \boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})_i - \boldsymbol{\theta}_\mathbf{W}(\boldsymbol{\theta}, r_{i,j})_j \right\|_2)}. \tag{4.39}$$

Graph matching aims at finding correspondences between the nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$ maximizing the global consistency score J_{gm}, as elaborated in [314]. $J_{gm}(\mathbf{X})$ is represented in the quadratic form as non-convex, global objective function

$$J_{gm}(\mathbf{X}) = \mathbf{x}^* \mathbf{A}_G \, \mathbf{x} \tag{4.40}$$

and $\mathbf{x}$ is the vectorization of $\mathbf{X} \in \{0,1\}^{n_1 \times n_2}$. $\mathbf{X}$ contains the correspondence information of the nodes from $\mathcal{G}_1$ and $\mathcal{G}_2$ with $x_{i_1 i_2} = 1$ if the i_1-th node of $\mathcal{G}_1$ corresponds to the i_2-th node of $\mathcal{G}_2$. The global affinity matrix $\mathbf{A_G} \in \mathbf{R}^{n_1 n_2 \times n_1 n_2}$ represents the pair-wise similarity of nodes $\mathbf{W}$ and edges $\mathbf{Q}$. FGM presented in [314] does not require an explicit computation of the affinity matrix due to its factorization.

With $\mathbf{A} \in \mathbf{R}^{n_1 n_2 \times n_1 n_2}$ the quadratic assignment problem can be optimized with

$$\max_{\mathbf{x}} \mathbf{x}^* \mathbf{A_G} \mathbf{x}, \quad \text{s.t.} \quad \mathbf{Yx} \leq \mathbf{b}, \mathbf{x}^* \mathbf{Yx} \in \{0,1\}^{n_1 n_2} \quad \text{and} \quad \mathbf{Y} = \begin{pmatrix} \mathbf{1}_{n_2}^* \otimes \mathbf{I}_{n_1} \\ \mathbf{1}_{n_2} \otimes \mathbf{I}_{n_1}^* \end{pmatrix} \tag{4.41}$$

with $\mathbf{1}_{n_2}$ a vector of ones, $\mathbf{I}_{n_1} \in \mathbf{R}^{n_1 \times n_1}$ an identity matrix, and $\mathbf{b}$ equal to a vector of ones with $\mathbf{b} = \mathbf{1}_{n_1+n2}$.

The factorization proposed in [314] optimizes graph matching methods and divides $\mathbf{A_G}$ into smaller matrices such that the optimization can be conducted iteratively using

$$\max_{\mathbf{X}} J_\alpha(\mathbf{X}) = (1-\alpha) J_{\text{vex}}(\mathbf{X}) + \alpha J_{\text{cav}}(\mathbf{X}). \tag{4.42}$$

$J_{\text{vex}}(\mathbf{X})$ denotes the convex relaxation, while $J_{\text{cav}}(\mathbf{X})$ is the concave relaxation introduced in [314]. The energy function in *graph33* is adapted to

$$\max_{\mathbf{X}} J_\alpha(\mathbf{X}) = (1-\alpha) J_{\text{vex}}(\mathbf{X}) + \alpha J_{\text{cav}}(\mathbf{X}) + J_{\text{smooth}}(\mathbf{X}). \tag{4.43}$$

$J_{\text{smooth}}(\mathbf{X})$ is an additional regulation term that considers the rigid nature of rotation and translation in registration with the projection difference of neighboring correspondence points:

$$J_{\text{smooth}}(\mathbf{X}) = -\sum_{i \in \mathbf{X}} \sum_{j \in \mathbf{D}} \frac{\left\| \mathbf{w}_i - \mathbf{w}_j \right\|_2 - \left\| \mathbf{w}_{i,c} - \mathbf{w}_{j,c} \right\|_2}{(n_1 \cdot n_2)}. \tag{4.44}$$

$\mathbf{D}$ contains the neighbors of point $\mathbf{w}_i$ and the correspondences to $\mathbf{w}_i$ and $\mathbf{w}_j$ are denoted $\mathbf{w}_{i,c}$ and $\mathbf{w}_{j,c}$. As a result, FGM yields a correspondence matrix, as depicted in Figure 4.17, where correspondence or non-correspondence are encoded binarily.

Zhou and de la Torre [314] also discuss the problem of local and global optimality. Local maximization of J_α in convex space does not guarantee

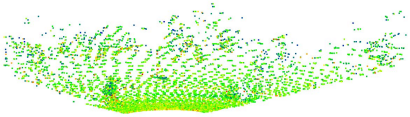
a globally optimal solution of the non-convex, global objective function $J_{gm}(\mathbf{X})$. The authors of [314] cope with this problem by discarding temporary solutions with a bad score for $J_{gm}(\mathbf{X})$. This achieved a notable optimization of the convergence of the quadratic assignment problem, according to [314]. Hence, FGM in *graph33* can effectively deal with local minima and can lead to a higher registration accuracy similar to the utilization of other graph matching techniques evaluated in [314].

Correspondence Rejection. Huang et al. [126] utilize RANSAC for correspondence rejection but this is not applicable for data from unstructured environments due to the absence of a clear geometric structure. Consequently, a novel correspondence rejection method was designed for *graph33*: all possible correspondences between source and target points are considered and mapped into a Euclidean 3D space as hypotheses. The highest density of hypotheses inside this 3D space is determined, and all correspondences lying outside a preliminarily defined sphere are rejected. This correspondence rejection is performed for each source point and achieved a notable reduction of the set of possible correspondences. Different radii for the rejection sphere were examined, and a radius of 2.0 m yielded very promising results with a proper but not too strict, rejection of false correspondences in *graph33*.

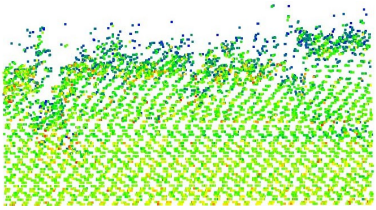
4.3.4.3 Proof of Concept: *graph33*

The proposed *graph33* method is demonstrated on the *IOSB-Reg* dataset (Section 7.4). 3D point clouds of a Velodyne HDL-64E LiDAR ($\mathcal{L}$) were registered to 3D point clouds from stereo image disparity estimation with SGBM ($\mathcal{S}$) parameterized, as described in Section B.1.1. $\mathcal{L}$ is selected as source cloud due to its sparsity and higher accuracy, while $\mathcal{S}$ is the registration target. Figure 4.15 compares the LiDAR cloud in Cartesian coordinates to the LiDAR cloud after the proposed homogenization that achieved an approximately uniform distribution of the LiDAR points. The OpenCV implementation[6] of SGBM [118] was applied to estimate disparity in stereo camera images, and 3D point clouds were generated

[6] OpenCV: cv::stereo::StereoBinarySGBM Class Referencecv::stereo::StereoBinarySGBM Class Reference, https://docs.opencv.org/3.4/d1/d9f/classcv_1_1stereo_1_1StereoBinarySGBM.html, access on 07.11.2021.



(a) $\mathcal{L}$ Cartesian.

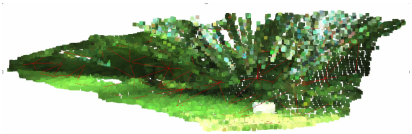


(b) $\mathcal{L}$ homogenized.

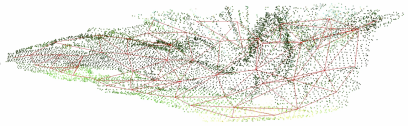
(c) $\mathcal{S}$ Cartesian.

(d) $\mathcal{S}$ homogenized.

Figure 4.15 Front views of the source and target clouds in *graph33*: the LiDAR clouds $\mathcal{L}$ are colored according to the reflectance measured by the LiDAR sensors. Images © Fraunhofer IOSB.

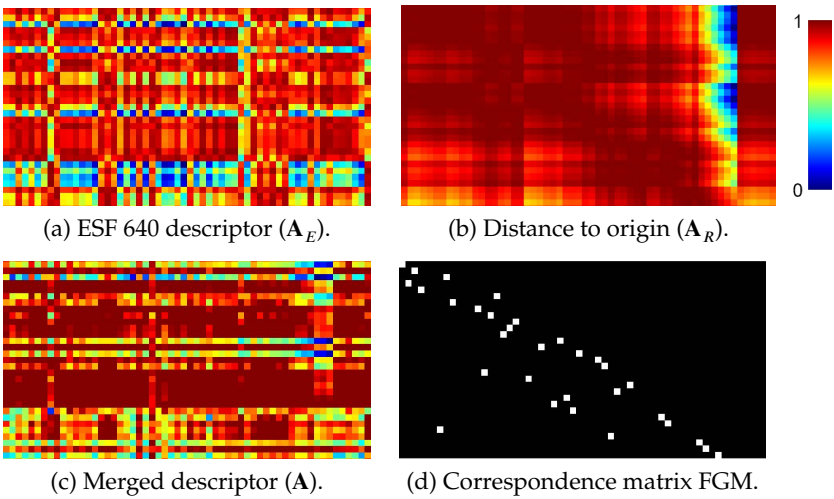


SGBM $\mathcal{S}$.



Supervoxel graph.

Figure 4.16 Over-segmentation in *graph33*: $\mathcal{S}$ from SGBM is registered to the $\mathcal{L}$. The supervoxel graph is extracted from the stereo camera cloud with the voxel centroid points as nodes and their connections as edges for the scene of Figure D.3. Images © Fraunhofer IOSB.



(a) ESF 640 descriptor ($\mathbf{A}_E$). (b) Distance to origin ($\mathbf{A}_R$).

(c) Merged descriptor ($\mathbf{A}$). (d) Correspondence matrix FGM.

Figure 4.17 Exemplary descriptor affinity matrices $\mathbf{A}_E$, $\mathbf{A}_Z$, $\mathbf{A}_R$, and $\mathbf{A}_N$ combined into one merged, normalized descriptor ((c), $\mathbf{A}$). The corresponding affinity matrices $\mathbf{A}_Z$ and $\mathbf{A}_N$ are depicted in Figure A.7. Vertical axes represent source nodes, horizontal axes target nodes. The color scaling in (a)–(c) indicates the affinity values between the source and target nodes, and a value of one describes complete affinity between two nodes. Image (d) depicts exemplary FGM estimated correspondences in *graph33*, and white matrix elements in (d) indicate correspondence between the respective source and target nodes. Images © Fraunhofer IOSB.

with SGBM parameterization according to Section B.1.1. Camera calibration was also performed with OpenCV. The disparity estimation errors of SGBM produced streaks in the point cloud, which additionally complicated proper 3D–3D registration. Figure 4.16 illustrates the over-segmentation of the stereo camera point cloud with the resulting nodes and edges of the graph. $\mathbf{A}$ combines the affinity matrices of the four descriptors $\mathbf{A}_E$, $\mathbf{A}_Z$, $\mathbf{A}_R$, and $\mathbf{A}_N$, as illustrated in Figure 4.17.

ICP [15] or GICP [250] hardly achieved valid registration results for cross-source data with a large initial decalibration. In order to compare ICP and *graph33*, both methods were tested in the registration of similar-source LiDAR clouds captured in unstructured environments. Here, *graph33* clearly outperformed ICP in the similar-source registration on

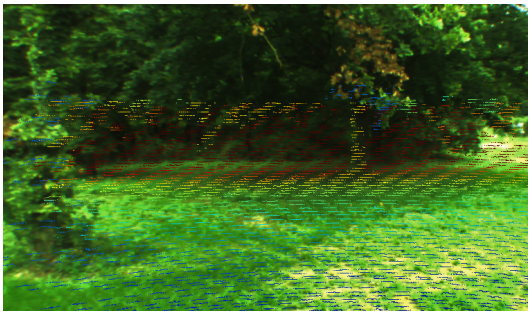


Figure 4.18 3D–3D registration result with *graph33* on exemplary *IOSB-Reg* scene. The visual overlay shows the projection of $\mathcal{L}$ onto the RGB image reduced to the region of interest for clarity. Blue color for $\mathcal{L}$ indicates small depth, while red shows high depth. The contour alignment on the trunk structures qualitatively validates the *graph33* registration accuracy.

two artificially decalibrated LiDAR clouds of an *IOSB-Reg* scene: ICP achieved $\ln F = 0.225$ and $L_2 = 2.26$ m, while *graph33* yielded $\ln F \leq$ -7 and $L_2 = 3$ cm. An in-depth evaluation of *graph33* and the enhanced GICP algorithm presented in Section 4.2.2 on cross-source data from unstructured environments was not conducted as the plate-shaped discretization errors in SGBM point clouds led to inaccurate registration results with GICP in prior experimental evaluations. Consequently, *graph33* lends itself well as a method for registering 3D cross-source data from unstructured environments.

The presented, classic 2D–3D registration method *cc23* outperformed *graph33* and yielded lower errors for the F and L_2 metrics. The *graph33* method achieved the least accurate result in a scene with more than 25 m of distance to the elements contributing to the registration in the scene. It is thus assumed that the major cause for this was the limited depth accuracy of the stereo camera point cloud. As a reference, $z = 25$ m with an accepted three pixel error limit in disparity estimation corresponds to $\varepsilon_z = 4.4$ m. Nevertheless, *graph33* yielded a higher registration accuracy in a few cases where *cc23* experienced difficulties achieving valid registration results.

Further comparison of *cc23*, *graph33*, and CSGM is presented in Section 4.3.6.1. Figure 4.18 shows an exemplary registration result of *graph33*

on two images of the *IOSB-Reg* dataset. Concluding, *graph33* with the optimizations proposed in this thesis clearly outperformed CSGM as proposed in [126] on data from unstructured environments. The *graph33* method can achieve successful registration results under the different exposure and light conditions as well as in the different seasons captured in the *IOSB-Reg* dataset.

4.3.5 *dsm33*: 3D–3D Cross-Source Registration with Neural Networks

The *dsm33* method facilitates the 3D–3D registration of cross-source data with CNNs and is inspired by the work of Haskins et al. [109] in medical imaging. It was methodically elaborated in collaboration with Leitritz [335] who primarily conducted the proof of concept detailed in Section A.3.4. The *dsm33* method is intended as a first step towards the 3D–3D registration with neural networks and as an impulse for further research in this direction. Figure 4.19 illustrates the workflow of the proposed *dsm33* method that registers 3D cross-source data by combining a learned deep similarity metric (*dsm*) and a classic, evolutionary optimization algorithm. It does not use correspondences in contrast to other 3D–3D registration methods on the basis of CNNs [57, 126]. The *dsm33* method registers 3D LiDAR clouds ($\mathcal{L}$) to SGBM stereo camera point clouds ($\mathcal{S}$) and is designed and optimized to register 3D cross-source point clouds captured in unstructured environments similar to *graph33*. $\mathcal{L}$ is selected as the registration target due to its higher measurement accuracy. Consequently, the source $\mathcal{S}$ is subject to augmentation to train the similarity metric. The similarity between the considered 3D cross-source clouds is represented as a CNN-based regression problem, and the transformation between source and target point cloud is estimated on the basis of the learned similarity metric.

However, *dsm33* comes with the inherent accuracy limitation of stereo camera depth estimation like *graph33*. At this point in time, the 2D resolution of images and the available computing power cannot provide a sufficiently high level of accuracy in stereo image disparity estimation to replace the 3D LiDAR sensors in unstructured environments, and sub-pixel stereo methods are no alternative for increased depth estima-

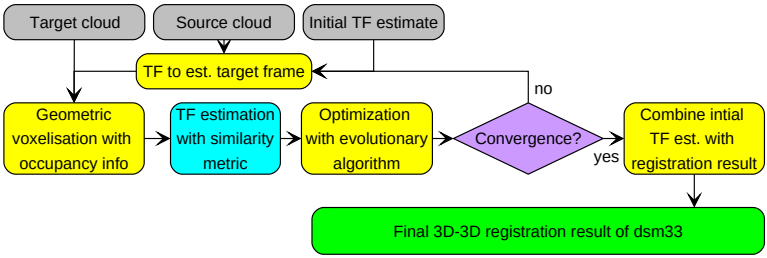


Figure 4.19 *dsm33* workflow: the CNN highlighted in blue contains the 3D similarity metric, classic processing steps are colored in yellow, input data in gray.

tion accuracy. The *dsm33* method is presented as a 3D–3D registration method relying on neural networks within this thesis but not integrated into *UCSR* by now. The increase of computing power and cameras with higher resolution in the future will decrease the depth inaccuracies in stereo camera disparity estimation and notably improve the registration accuracy in 3D–3D cross-source registration.

4.3.5.1 Preprocessing and Network Architecture

The 3D clouds are mapped onto a voxel grid to enable the similarity metric estimation inside a CNN architecture. The voxel size depends on the geometric expansion of the input data, and a too coarse voxelization leads to a loss of accuracy and information content, while a too fine voxelization leads to an exploding consumption of memory.

The $\mathcal{S}$ and $\mathcal{L}$ input clouds are represented as 3D images with two channels and cropped into a cuboid of $20\,\mathrm{m} \times 20\,\mathrm{m} \times 10\,\mathrm{m}$ prior to voxelization. This cuboid is centered around the origin of the respective clouds to fit the camera FoV. This considers the different nature of the *dsm33* data in contrast to [109] and generates a consistent data representation despite the cross-source point sets subject to registration. An additional bounding box is laid around the extracted cuboid of $\mathcal{S}$ to determine the voxel size. The bounding box limits are multiplied by a factor of 1.1 to preserve the input data in case of decalibrations. Hence, the maximum size of the

bounding box is $l_{bx} \times l_{by} \times l_{bz} = 22\,\text{m} \times 22\,\text{m} \times 11\,\text{m}$. The 3D voxel size $\mathbf{l}_{\text{vox}}$ is determined by

$$\mathbf{l}_{\text{vox},i} = \max\left(\frac{l_{bi}}{l_{vi}}\right), \quad i \in \{x, y, z\}, \tag{4.45}$$

with $l_{vi}, i \in \{x, y, z\}$ the maximum dimension of the respective 3D cloud. 3D clouds with binary information are utilized in *dsm33* instead of 3D voxels with grayscale information in [109], and the encoded binary information specifies the voxel occupancy.

In addition to the $512 \times 512 \times 32$ voxelization proposed in [109], a voxelization of the input point clouds with 256×256×32 and 128×128×32 was proposed and evaluated in *dsm33*. The RGB intensities of $\mathcal{S}$ and the intensity measurements of $\mathcal{L}$ were not considered in *dsm33* to preserve the generalization of the method to other kinds of input data. After preprocessing, the voxelized, cuboid-structured 3D volume data is input into the CNN in *dsm33*.

The input layer of *dsm33* processes the 3D volumes, and the nine subsequent volumetric convolutional layers perform the feature extraction step. The 3D cloud pair forms one cloud with two channels as proposed in [109], and the convolutional layers compute the inner product between all filters and the corresponding cloud patches with a stride of 1 in each direction for feature extraction. A ReLU activation function layer follows each convolutional layer and suppresses negative values in the output feature maps of the convolutional layer. Maximum pooling layers are inserted for downsampling: the first pooling layer with $2 \times 2 \times 2$ is inserted after the first convolutional layer with a subsequent ReLU activation. A second pooling layer with $1 \times 2 \times 2$ succeeds the third convolutional–ReLU layer. A skip connection concatenates the second maximum pooling layer's result with the sixth convolutional–ReLU layer combination. A final fully-connected layer outputs the estimated deep similarity metric as a scalar value. The estimated similarity indicates the quality of the registration. The weights are not shared between $\mathcal{S}$ and $\mathcal{L}$ as different numbers of filters are used.

In order to choose a suitable similarity metric for binary 3D images, MI [291] was compared to the learned similarity metric in *dsm33* for

the analyzed cross-source data, as described in [109]. MI measures the mutual dependency of two variables X and Y as

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right). \tag{4.46}$$

Here, H is the Shannon entropy that analyzes the information content, according to Equation 3.10, while $p(x, y)$, $p(x)$, and $p(y)$ are the respective probabilities. Each voxelized 3D image is considered as a random variable of a sample consisting of several 3D images to calculate the MI. Each voxel is a binary, random experiment as a voxel can be full or empty in *dsm33*. The probabilities can be determined by counting the number of occurrences of $X = 0$ with $Y = 0$ or $Y = 1$ and $X = 1$ with $Y = 0$ or $Y = 1$ in the corresponding voxels of two images to be registered. The proof of concept detailed in Section A.3.4 demonstrates that the learned similarity metric outperformed MI on 2D and 3D data from unstructured environments (see Figure A.8).

4.3.5.2 Training and Augmentation

Haskins et al. [109] use the target registration error as a loss function for training (see Section 2.3.3). In contrast, *dsm33* utilizes the L_2 norm between the ground truth and the registration result as the approach of [109] is neither comfortable nor suitable for cross-source sensor data registration requiring manual user intervention. The L_2 norm proved additionally favorable as it models translation and rotation errors within one value that considers the effective distance of the points after applying the registration result.

At first, the LiDAR cloud is transformed into the coordinate system of $\mathcal{S}$ by applying the ground truth reference transformation. Secondly, the predefined training decalibration is applied to $\mathcal{L}$ in stereo coordinates. Before the cuboid extraction, the input clouds are transformed with the ground truth to train the CNN for similarity estimation. Level S, level M, and level L decalibrations are applied subsequently. The L_2 norm is purely measured on $\mathcal{L}$ to ensure the selection of corresponding points

that map the same 3D point within the L_2 calculation. It provides a scalar output:

$$L_{dsm33} = \frac{1}{N} \sum^{N} (\hat{L}_2(\mathcal{L}, \mathcal{L}_{\text{dec}}) - L_2(\mathcal{L}, \mathcal{L}_{\text{dec}}))^2. \tag{4.47}$$

The desired value – the ground truth L_2 norm between equivalent points in $\mathcal{L}$ prior to decalibration and $\mathcal{L}_{\text{dec}}$ after decalibration – is encoded in $L_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$. Hence, $L_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$ is defined by the input data decalibration. $\hat{L}_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$ denotes the CNN-estimated L_2 between $\mathcal{L}$ and $\mathcal{L}_{\text{dec}}$.

Furthermore, the impact of different augmentation strategies was evaluated in *dsm33*. Here, a customized augmentation strategy that yielded a uniform distribution of decalibrations (*dsm33-U*) in the training, testing, and validation data proposed in [335] was compared to an augmentation strategy with an approximately Gaussian distribution of the decalibrations. At first, the source clouds are transformed to target coordinates. Secondly, different decalibrations between the source and the target cloud were generated on the basis of the ground truth information from step one. The standard augmentation strategy proposed in [109] (*dsm33*-N) transforms both source and target with a random decalibration to obtain different point distributions in 3D space. The uniform augmentation *dsm33-U* proposed in this thesis outperforms the augmentation proposed by Haskins et al. [109]. Furthermore, only *dsm33-U* correctly estimates translational decalibrations, as depicted in Figure A.8(d).

Haskins et al. [109] evaluated different optimizers: SGD, SGD with Nesterov momentum, RMSprop, Adagrad, Adam, and Adadelta. The Adam optimizer yielded a superior performance with a learning rate of 10^{-5} and was therefore selected to train the deep similarity metric CNN in *dsm33*.

4.3.5.3 Differential Evolution Optimization

A suitable optimization strategy must be selected as the learned similarity metric is neither convex nor smooth. Truncated differential evolution is selected as recommended by Haskins et al. [109] as it does not require the optimization problem to be differentiable [260]. Furthermore, evolutionary optimization algorithms are especially well-suited for problems with a higher dimensionality, such as registration. With the learned similarity metric in *dsm33* as target function, the exploration of the search

space was achieved in a rapid and holistic manner, and a solution close to the optimum was mostly found. Haskins et al. [109] applied a differential evolution initialized Newton-based optimization (DINO) where the differential evolution results are subject to further BFGS optimization.

Truncated differential evolution generates multiple parameter vectors for the wanted transformations in each iteration and transforms the source clouds to estimate the registration error. Parameter vectors with favorable results are selected as a parent for the reproduction in the next population. This yielded a convergence of the algorithm within a small number of iterations. The population energies $\mathbf{E}_p$ contain the $L_2(\mathcal{L}, \mathcal{L}_{\text{dec}})$ values of the individual transformations making up the population as potential solutions inside an array structure. Convergence is examined after each generation and terminates the differential evolution optimization if

$$M_{\text{DE}} = \sigma(\mathbf{E}_p) \leq 0.01 \cdot \left\| \overline{\mathbf{E}_p} \right\|_1 \tag{4.48}$$

is fulfilled or the maximum number of iterations are reached.

The experimental evaluation of *dsm33* on data from unstructured environments showed that in contrast to [109], a local, Newton-based optimization such as BFGS did not improve the differential evolution optimization results. This was probably caused by the different characteristics of the medical data of [109] and the *IOSB-Reg* data from unstructured environments. $\mathcal{S}$ suffered from notably higher inaccuracies due to the limited depth estimation accuracy. 3D data from stereo image disparity estimation has a high density, whereas LiDAR data is accurate but sparse. With these training samples, the model is non-convex and non-smooth in many areas, and the local BFGS optimization was probably unable to escape from this local optima towards a global optimum. However, neither the meta-heuristic nature of differential evolution nor the local, Newton-based BFGS optimization can guarantee a globally optimal solution, and *dsm33* utilizes a truncated differential evolution without an additional BFGS optimization as it proved useful for the registration of 3D–3D cross-source data from unstructured environments. Naturally, increased estimation accuracy of the CNN on the basis of the learned similarity metric increased the probability for the optimization to converge into a global optimum.

Experimental evaluation in [335] showed that the registration results of *dsm33* proved to be very likely to be close to the global optimum using

the truncated differential evolution of SciPy with different parameter configurations for each iteration. The multi-pass approach proposed in [109] conducts each inference multiple times and achieves a higher registration accuracy for the medical imaging data of [109] but also requires a notably higher computation effort prior to optimization. A thorough experimental evaluation of the multi-pass approach on the cross-source data regarded in this thesis did not improve the registration accuracy. Hence, multiple iterations of the inference step are not conducted in *dsm33*, and the deep similarity metric is only estimated once prior to optimization.

Furthermore, differential evolution optimization requires a similarity metric that can also achieve a high validation error as long as the validation error does not rise again after a certain number of epochs, and the CNN-based similarity metric estimation in *dsm33* proved useful for the subsequent processing steps.

The *dsm33-N* similarity metric still yielded a higher registration accuracy than MI. The notably lower standard deviation for both *dsm33* configurations underlines the higher robustness of the proposed CNN-based similarity metric.

4.3.6 Comparison of Individual Cross-Source Registration Methods

A ground truth reference transformation was available for all use cases within this thesis. A visual overlay can provide a qualitative assessment of the registration accuracy for use cases without a reference transformation. As proposed in Section 4.2.1, kNN filtering facilitates preprocessing or a qualitative accuracy assessment for registered 3D point clouds.

4.3.6.1 Classic Registration Methods

Table 4.6 compares ICP and CSGM according to [126] with the proposed classic *graph33* and *cc23* methods on the *IOSB-Reg* dataset. The *cc23* methods yielded lower and better F and L_2 results than *graph33*, and it is assumed that the major cause for this is the already limited depth estimation accuracy in stereo camera clouds. The proposed *cc23* and *graph33* methods were also compared to CSGM as proposed in [126]

Error Metric	*ICP*	*CSGM*	*graph33*	*cc23*
Best result on single image				
$\ln F$	**-0.424**	1.852	1.215	0.400
L_2	3.695	4.316	**1.626**	2.858
Average registration accuracy and robustness				
$\mu(\ln F)$	2.635	4.410	3.755	**1.520**
$\mu(L_2)$	5.760	8.716	9.248	**5.672**
$\sigma^2(\ln F)$	1.221	1.362	2.312	**0.296**
$\sigma^2(L_2)$	**0.927**	5.049	4.888	16.607
Weights w_i for *UCSR* fusion (according to Equation 4.20)				
$w_{i,1}$	2.477	1.244	1.232	**2.913**
$w_{i,2}$	2.451	0.885	0.849	**4.274**

L_2 is given in m. ln is the natural logarithm in m.

Table 4.6 Registration accuracy of *cc23*, *graph33*, CSGM, and ICP on 27 *IOSB-Reg* validation and test images. CSGM is parameterized with $l_{\text{vox}} = 0.30\,\text{m}$.

and the *graph33* method optimized for unstructured environments. Here, both methods clearly outperformed CSGM of [126] on data from these environments. ICP was included as a reference for the accuracy of the proposed methods and achieved a lower $\sigma(L_2)$ and a higher $\sigma^2(F)$. L_2 is similar to the e_{fs} metric optimized by the BFGS algorithm in the ICP algorithm's inner loop, which explains its low σ and the relatively low L_2 error compared to the F results with other methods. Furthermore, the low σ and σ^2 values of the ICP algorithm with a high L_2 error can indicate a faulty convergence within a local minimum.

Table 4.6 shows that *graph33* outperformed CSGM according to [126] on data from unstructured environments. Hence, the adaptions and optimizations proposed in Section 4.3.4 provide the desired improvement for unstructured environments. Nevertheless, the registration accuracy of *graph33* alone is not sufficient to register multi-sensor systems on off-road vehicles. The *graph33* method achieved the third-best result in terms of F and L_2 ($\ln F = 1.215$, $L_2 = 1.656\,\text{m}$) on an $\mathcal{S}$–$\mathcal{L}$ cloud pair that captured many planar surfaces from industrial buildings and cobblestone pavement with a similar texture. However, *cc23* achieved the most accurate registration result on mainly unstructured images that mainly captured grass, trees, and bushes, while it achieved the

worst results on the above mentioned $\mathcal{S}$-$\mathcal{L}$ cloud pair with industrial buildings and cobblestone pavement. Concluding, *cc23* achieved the most promising results to register 2D cameras to 3D LiDAR sensors from unstructured environments in a classic manner and outperformed ICP, CSGM, and *graph33* in terms of F and L_2. However, *cc23* and *graph33* showed different strengths that mutually complement each other and facilitate a higher registration accuracy than the individual application of only one method.

4.3.6.2 Registration Methods with Neural Networks

Table 4.7 compares the registration performance of the *dsm33* configuration with the highest accuacy (*dsm33-U*) with *cnn23* and classic ICP for level S decalibrations according to Section 3.11. In terms of F and L_2, *cnn23* achieved the highest registration accuracy, while *dsm33* yielded the most accurate estimation of the rotation $\mathbf{r}$. The achieved $\overline{\Delta \mathbf{t}} = 6.9\,\text{cm}$ for level S and a $\overline{\Delta \mathbf{t}} = 5.2\,\text{cm}$ for level L decalibrations were slightly higher than the registration accuracy achieved in similar-source registration that equals the noise of rotating 3D LiDAR sensors ($\pm 3\,\text{cm}$). However, a mean registration error in the range of single-digit centimeters proved useful for the environment perception of off-road vehicles in previous works [323, 324]. Figure 4.20 visualizes the registration results for *cnn23*, *dsm33-U*, and classic ICP on an exemplary scene of the *IOSB-Reg* test dataset as projections in 2D space and further justifies the applicability of the achieved registration results to register multi-sensor systems for the perception of unstructured environments. 3D stereo camera point clouds generated with SGBM are registered to 3D LiDAR point clouds with *dsm33* and *ICP*, while *cnn23* matches 2D RGB images to 3D LiDAR point clouds. ICP could not improve the registration accuracy, especially not for small decalibrations. The *cnn23* method yielded the most accurate translation estimates for level S and L decalibrations. The analyzed *cnn23* architecture has more than 17 M network parameters, while *dsm33* has more than 22 M parameters. The runtime for one registration with *cnn23* on a PC with 32 GB RAM CPU and an RTX 2080S GPU amounted to 20 ms, while the registration of two cross-source clouds with *dsm33* could require more than ten minutes. This does not exclude the applicability of *dsm33* in the registration of cross-source sensor data in the field, as

Error Metric	*cnn23*	*dsm33-U*	ICP	Decalib.
Average registration accuracy and robustness ($n = 15$).				
$\mu(\ln F)$	**-1.187**	-1.171	-0.041	-1.259
$\mu(L_2)$	**0.491**	0.556	1.197	0.724
$\sigma(\ln F)$	**-2.244**	-2.129	-0.860	-2.659
$\sigma(L_2)$	0.461	**0.259**	0.454	0.150
$\min(L_2)$	**0.120**	0.184	0.736	0.492
$\min(\ln F)$	**-3.147**	-2.453	-0.553	-1.864
Registration accuracy for selected DoFs.				
$\overline{\Delta \mathbf{t}}$ along z axis	**0.037**	0.097	0.597	0.148
$\overline{\Delta \mathbf{r}}$ around x axis	0.826	**0.770**	3.682	2.414
$\overline{\Delta \mathbf{r}}$ around z axis	1.783	**1.731**	1.811	1.887
$\overline{\Delta \mathbf{t}}$	**0.069**	0.141	0.455	0.138
$\overline{\Delta \mathbf{r}}$	1.241	**1.007**	2.525	2.138

L_2 is given in m. Rotations are given in degrees.

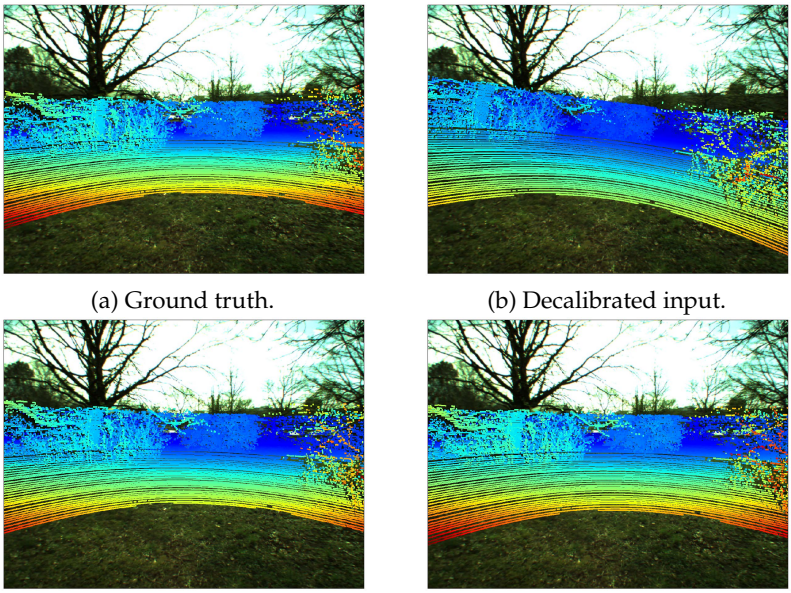
Table 4.7 Registration results of *cnn23*, *dsm33-U*, and classic ICP for level S decalibrations on 14 *IOSB-Reg* test images. Registration on each test scene is performed $n = 15$ times to estimate accuracy and robustness.

sensor calibration is not required in real-time. However, it becomes clear that the suitability of *dsm33* in the practical application for the presented use cases is still limited.

Concluding, *cnn23* estimated relative translations with the highest accuracy, while *dsm33* determines relative rotations with higher accuracy than *cnn23*. Equivalent to *graph33*, *dsm33* requires the availability of two cameras and an a priori generation of a 3D stereo camera point cloud.

4.3.7 Proof of Concept: *UCSR*

The applicability of *UCSR* to register 2D RGB cameras and a 3D LiDAR is demonstrated on *IOSB-Reg* data from unstructured environments without calibration targets. Table 4.8 compares the individual registration methods in *UCSR* as well as the confidence-based fusion of *cc23*, *cnn23*, and *graph33* inside the *UCSR* framework on the basis of the F and L_2 error



(a) Ground truth. (b) Decalibrated input.

(c) *cnn23* registration result. (d) *dsm33*-U registration result.

Figure 4.20 Registration results for an *IOSB-Reg* scene with *cnn23-I* and *dsm33-U* for a level M decalibration ([-0.347 m; -0.082 m; -0.142 m]; [8.03°; 5.97°; 8.03°]). Images © Fraunhofer IOSB.

metrics. It is shown that *UCSR* achieves a considerably lower mean accuracy in terms of $L_2(\text{UCSR}) = 0.868$ m in unstructured environments than CSGM in the registration of the "Stanford Bunny" (see Section 3.11 [126]).

The combination of *cc23*, *cnn23*, and *graph33* increased the accuracy and robustness of the registration as well as the range of possible input data for valid registration scenarios. Out of the unstructured images of the *IOSB-Reg* dataset, 15 image-LiDAR pairs were selected for validation and 15 for testing. Each individual registration method was run for 15 times ($n = 15$) on each scene to determine the accuracy and robustness results for the respective error metrics as some of the evaluated registration methods rely on local, Newton-based optimization (*cc23*) or neural networks that require a more in-depth analysis than classic methods due to their black box character.

Error Metric	Individual methods in *UCSR*			*UCSR*
	cc23	*cnn23*	*graph33*	
Best result on validation data.				
$\ln F$	0.306	-1.514	0.901	**-1.616**
L_2	2.126	**0.342**	3.495	0.709
Accuracy and robustness on all test scenes with $n = 15$.				
$\mu(\ln F)$	1.224	-1.016	3.189	**-1.446**
$\mu(L_2)$	3.636	**0.695**	6.287	0.868
$\sigma^2(\ln F)$	0.525	0.245	1.331	**0.028**
$\sigma^2(L_2)$	3.448	0.256	3.859	**0.059**
$w_{i,2}$ for *UCSR* fusion according to Equation 4.20 and Equation 4.21				
$w_{i,2}$	1.451	8.890	**0.724**	–

L_2 is given in m. ln is the natural logarithm in m.

Table 4.8 Registration accuracy with individual methods and with *UCSR* for level L decalibrations according to Section 3.11.

The *cnn23* method outperformed classic *cc23* and *graph33* in terms of accuracy and robustness. However, it has to be considered that CNNs require a huge amount of training data that is often hard to find, especially in unstructured environments. *UCSR* performed better than the best registration result of its individual components *cc23*, *cnn23*, and *graph33*. The combination of classic and CNN-based registration methods in *UCSR* provided valid, verifiable, and accurate registration results and can concurrently validate CNN methods in critical applications such as autonomous off-road vehicles.

Schneider et al. [246] achieved a mean calibration error of 6 cm in translation and 0.28° in rotation on the KITTI 2012 dataset [83]. However, structured environments with smooth surfaces are notably more favorable for cross-source registration without calibration targets than the unstructured environments regarded in this thesis. With an empirical mean of 5.10 cm in translation, the *UCSR* translation registration accuracy was higher for unstructured environments than the accuracy achieved by Schneider et al. [246] in the less-challenging, structured environments

captured in KITTI. The *UCSR*-achieved mean rotational error of 0.956° for unstructured environments was higher than the rotation error of [246] and CalibNet [131] in registering KITTI data from structured environments. This higher rotational error of *UCSR* is assumed to be caused by the notably smaller number of smooth surfaces complicating sensor data registration in unstructured environments. The *UCSR* framework yielded a lower variance and thus an increased robustness of the registration compared to the evaluated, individual registration methods. Hence, it can provide valid registration results in a wide range of different environments, especially for autonomous off-road vehicles in unstructured environments.

4.4 2D Image Fusion

The fusion of different spectral channels or differently exposed images into one image leads to an increased information density in one image for subsequent processing steps, such as visual SLAM, with similar calculation time. This indicates that visual SLAM with multi-spectral and HDR images can lead to a more accurate localization and mapping, and it paves the way for their application in more challenging, unstructured outdoor environments. Figure 4.21(a) depicts the processing pipeline for images from multi-spectral prism cameras.

4.4.1 2D Fusion of Multi-Spectral Images

Multi-spectral prism cameras capture intensity information from multiple spectral channels in different images but with identical characteristics in terms of lens, resolution, and FoV. RGB images in the visible spectrum (400 nm to 650 nm) and in NIR spectrum covering the wavelengths from 750 nm to more than 1000 nm are provided. Different fusion approaches for multi-spectral images are known, as discussed in Section 2.3.4. PCA only produces the channels with the highest variance, as detailed in Section 3.4, and is hence not suitable if the information from all channels shall be preserved.

HDR image fusion fuses multiple images of identical spectral information that depict different image areas in different qualities into one image.

The fusion of RGB and NIR images is similar as different image areas are captured with different image qualities. Thus, the idea of selecting favorably depicted image areas for the final image is similar in HDR and multi-spectral image fusion. Albrecht and Heide [320] compare different HDR fusion methods for feature-based visual SLAM for person indoor navigation and use MEF [192] to merge differently exposed images on the basis of saturation, well-exposedness, and contrast quality measures. Here, the contrast quality measure provided the most promising results on grayscale HDR images for feature-based visual SLAM with gradient-based feature extraction [320].

This thesis proposes the application of MEF to fuse RGB and NIR image information due to the discussed similarities to HDR fusion, where MEF is well-established and provides accurate, satisfactory results. To the best of the author's knowledge, MEF was not used previously to provide fused RGB–NIR images. MEF uses saturation S, contrast O, and well-exposedness E quality measures to generate the scalar-valued weight map to blend the input images. For each pixel $[i, j]$ in the k-th image, the weight is calculated:

$$w_{ij,k} = (O_{ij,k})^{\omega_O} (S_{ij,k})^{\omega_S} (E_{ij,k})^{\omega_E}, \tag{4.49}$$

with ω_O, ω_S, and ω_E the corresponding weighting exponents defined according to the importance of the respective quality measure. Contrast is measured with a Laplacian filter on the grayscale conversion of each image. The standard deviation within each channel is computed at each pixel for the saturation measure. The well-exposedness measure weights each intensity based on its closeness to 0.5 using a Gauss curve with $\sigma = 0.2$:

$$\exp\Big(- \frac{(i - 0.5)^2}{2\sigma^2} \Big). \tag{4.50}$$

Two RGB and two NIR images are captured by the stereo camera setup, and the RGB and NIR images from each camera are fused using MEF. Well-exposedness is used as a second quality measure in addition to contrast to take different imaging characteristics of RGB and NIR into account during the fusion process. NIR images contain only one channel, while the RGB images are converted to grayscale images. Subsequently, the RGB and the NIR image from one camera are fused using the experimentally justified combination of the well-exposedness E and contrast O

quality measures with equivalent weighting $\omega_O = \omega_E = 0.5$ according to Equation 4.49 for each pixel $[i, j]$ in image $\mathbf{I}_k$.

The image is decomposed using a Laplacian pyramid $\mathbf{L}_{\text{MEF}}$ as discussed in Mertens et al. [192]. Pyramid coefficients are not processed directly but blended depending on the value inside the scalar weight map saved as a Gaussian pyramid. Each level l of the Laplacian pyramid is the weighted mean value of the Laplacian compositions of sequence k, the k-th input image of the sequence that is subject to exposure fusion in [192] ($\mathbf{I}_k$), and l from the normalized weight map $\hat{\mathbf{W}}_{ij,k}$:

$$\hat{\mathbf{W}}_{[i,j],k} = \left[W_{[i,j],1} + W_{[i,j],2} \right]^{-1} W_{[i,j],k}. \tag{4.51}$$

The resulting image $\mathbf{R}_{\text{MEF}}$ is calculated from the NIR and grayscale input images using the weighting $\hat{\mathbf{W}}_{[i,j],k}$:

$$R_{\text{MEF}}[i, j] = \sum_{k=1}^{N} \hat{\mathbf{W}}_{[i,j],k} I_{[i,j],k}. \tag{4.52}$$

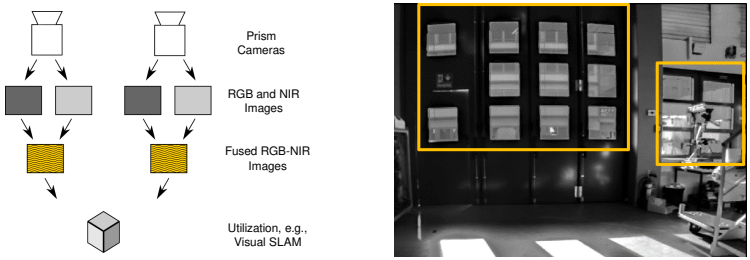
The RGB–NIR fusion utilizes $N = 2$ with $\mathbf{I}_1$ the grayscale and $\mathbf{I}_2$ the NIR image. This fuses the images by blending image features instead of intensities, which addresses the seam problem.

MEF for images from two multi-spectral cameras provides two fused RGB–NIR images for visual SLAM or disparity estimation from stereo images. The benefits of the utilization of fused RGB–NIR images in feature-based visual SLAM are described in Section A.4.

4.4.2 Proof of Concept: RGB–NIR Fusion

The application of MEF on RGB and NIR images is independent of the captured environment, and the proposed MEF 2D fusion approach was demonstrated on structured indoor environments with a large window facade to include outdoor elements. The RGB and NIR images for the presented image fusion were captured with the sensor setup of the *IOSB-Reg* dataset and showed promising results in the fusion of RGB and NIR images, as depicted in Figure 4.21(b). The outdoor elements in Figure 4.21(b) illustrate the straightforward application of the presented 2D fusion method for arbitrary structured and unstructured application scenarios. The fusion of the intensity information of the three RGB channels

and the monocular NIR images facilitates the use of these fused images for fast processing in visual SLAM or disparity estimation from stereo camera images. Section A.4 discusses the benefits of fused multi-spectral and HDR images for visual SLAM for primarily structured indoor and outdoor environments.



(a) Setup for RGB–NIR fusion. (b) Fused RGB–NIR image.

Figure 4.21 Hardware setup and 2D image fusion result with MEF for RGB–NIR images from a JAI AD-130GE camera with $\omega_E = \omega_O = 0.5$. Benefits from RGB–NIR fusion included that the captured outdoor elements are clearly visible through the window structure in the fused image, as highlighted in yellow.

5 Mid-Level Perception

This thesis proposes two stereo image disparity estimation methods for unstructured environments: a classic stereo image disparity estimation method for hyperspectral images in Section 5.1.1 [324], and the *UEM-CNN* method [327] in Section 5.1.2 providing a disparity estimation from stereo images on the basis of CNNs. Furthermore, Section 5.2.1 of this thesis proposes novel error metrics to validate disparity estimation methods in a loosely coupled manner and with a special focus on their application for off-road vehicles in unstructured environments [327]. In addition, the Stereo Evaluation Toolbox (*SET*) approach [325] presented in Section 5.2.2 allows a loosely coupled validation of 3D reconstruction results from arbitrary stereo image disparity estimation methods.

Section 5.3 proposes multiple sensor data fusion approaches combining information from 2D image and 3D point cloud data. The achieved 3D–3D fusion results in the form of 3D point clouds can constitute a direct input to the subsequent mapping and planning steps or also be subject to high-level interpretation, depending on the design of the perception pipeline of the respective off-road vehicle.

5.1 Disparity Estimation from Stereo Images

Horizontal stereo camera setups were selected for 3D reconstruction in this thesis as they provide a wider overlap for the horizontal FoV. Kallwies et al. [147] confirm this choice demonstrating that horizontal outperforms vertical stereo image disparity estimation for autonomous off-road vehicles.

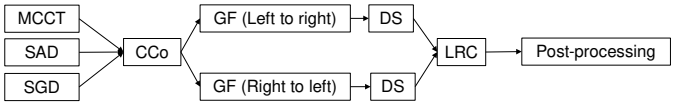


Figure 5.1 Processing pipeline of CCRADAR algorithm. MCCT: Modified Color Census Transform, *GF*: Guided Filter, *DS*: Disparity Selection.

5.1.1 Disparity Estimation on Hyperspectral Images

Hyperspectral image data provides more detailed spectral information than RGB images. As a results, it is expected that hyperspectral images achieve an enhanced depth reconstruction in unstructured environments. However, additional channels imply a higher computational effort, and local, correlation-based similarity measures are preferable as computations can be conducted in parallel and sped up notably.

5.1.1.1 CCRADAR for Hyperspectral Images

The local, correlation-based CCRADAR stereo method [139] was extended to hyperspectral image data in this thesis [265, 324]. To the best of the author's knowledge, a real-time capable approach for disparity estimation from stereo images on hyperspectral images to passively perceive unstructured environments was not known prior to the publication of [324]. Figure 5.1 provides an overview of the disparity estimation process in CCRADAR. The extended, hyperspectral CCRADAR method is demonstrated on Ximea xiSpec MQ022HG-IM-SM4X4-VIS cameras with 16 channels between 465 nm and 630 nm and a resolution of 2048×1088 px [324]. The Ximea stereo camera pair combines the intensity information from 16 channels into a 4×4 px mosaic representation that provides an image resolution of 512×272 mosaic pixels [324].

The local, hyperspectral CCRADAR algorithm operates on the mosaic pixels and performs numerous, simple similarity matching calculations in local window structures. This favors a parallel calculation on a General Purpose Computation on Graphics Processing Unit (GPGPU) on the basis of Compute Unified Device Architecture (CUDA) programming interface, as described in Section 5.1.1.2, to cope with the computationally expensive depth reconstruction due to many channels.

Different CCRADAR parameterizations and post-processing methods were evaluated to identify the most favorable CCRADAR configuration for unstructured environments. Table 5.1 provides an overview on the CCRADAR parameterizations for hyperspectral images from unstructured environments evaluated in this thesis and the configuration that achieved the most accurate 3D reconstruction results [324].

The 3D stereo clouds ($\mathcal{S}$) were generated from the estimated disparity images and compared against LiDAR clouds ($\mathcal{L}$) of the same, static scene for an accuracy evaluation: $\mathcal{S}$ was transformed in the coordinate system of the LiDAR sensor so that $\mathcal{S}$ and $\mathcal{L}$ were both present within the same coordinate system. Outliers were primarily introduced by depth estimation inaccuracies, and kNN outlier filtering explained in Section 4.2.1 allowed a more targeted accuracy assessment of $\mathcal{S}$.

The 3D reconstruction results $\mathcal{S}$ were assessed according to nine criteria. Here, ICP determines point-by-point correspondences between $\mathcal{S}$ and $\mathcal{L}$, and measures the 3D reconstruction accuracy in hyperspectral CCRADAR. ICP was chosen instead of GICP [323] as point-by-point distance measures from different sensor types were required here. The $\mathcal{S}$–$\mathcal{L}$ cloud pair was analyzed prior to ($\mathcal{S}_o$) and after ($\mathcal{S}_f$) outlier filtering on the basis of the following criteria:

- I ($\mathcal{S}_o$), V ($\mathcal{S}_f$): e_{fs} result of ICP registration with accurate $\mathcal{L}$ (target) and $\mathcal{S}$ (source) for accuracy assessment ($W_{I,V} = 0.25$).
- II ($\mathcal{S}_o$), VI ($\mathcal{S}_f$): correspondences in ICP ($W_{II,VI} = 0.075$).
- III ($\mathcal{S}_o$), VII ($\mathcal{S}_f$): L_2 deviation of $\mathbf{t}_{\text{ICP}}$ from $\mathbf{1}$ ($W_{III,VII} = 0.05$).
- IV ($\mathcal{S}_o$), VIII ($\mathcal{S}_f$): L_2 deviation of $\mathbf{r}_{\text{ICP}}$ from $\mathbf{1}$ ($W_{IV,VIII} = 0.05$).
- IX: percentage of kNN-filtered outliers ($W_{IX} = 0.15$).

$W_i, i \in [I; ...; IX]$ specifies the experimentally justified weighting factor for the respective criterion. The e_{fs} approximates the absolute depth estimation error for CCRADAR. A low absolute error (I, V) and a small number of outliers (IX) are the most important features of a point cloud in depth reconstruction for off-road vehicles, while a high number of correspondences between $\mathcal{S}$ and $\mathcal{L}$ (II, VI) indicates a reliable 3D reconstruction of the environment in $\mathcal{S}$. A high L_2 distance for $\mathbf{t}$ and $\mathbf{r}$ from the identity reference transformation (III, IV, VII, VIII) implies an inaccurate 3D reconstruction as the ICP registration optimizes the quadratic distances of corresponding point pairs. An accuracy-based, experimentally

justified ranking for the analyzed CCRADAR clouds was established for each criterion $i \in [I; \ldots; IX]$ on the basis of the measured values. The final scoring includes the achieved rank for each criterion i as a penalty score. The penalty points $P_{i,k}^{j}$ for each criterion i in CCRADAR configuration k on evaluation image j are combined with the weightings W_i:

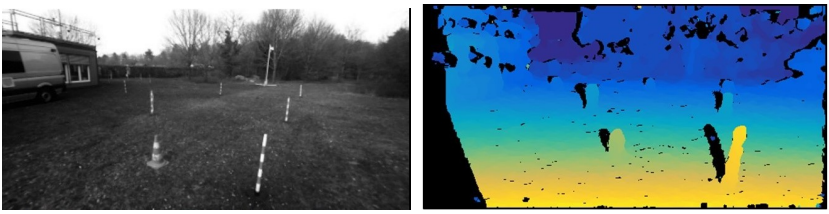
$$P_k^j = \sum_{i=I}^{IX} \left(P_{i,k}^j W_i \right). \tag{5.1}$$

The penalties P_k^j are summarized for all evaluation images to determine the CCRADAR configuration k with the lowest penalty P_k: $P_k = 2 \cdot P_k^1 + 2 \cdot P_k^2 + P_k^3$. Here, double importance was assigned to the images primarily representing unstructured environments ($j \in [1; 2]$). A low penalty P_k indicates a high 3D reconstruction accuracy of hyperspectral CCRADAR.

5.1.1.2 Proof of Concept: Real-Time Capable Stereo Image Disparity Estimation on Hyperspectral Images

All images are preprocessed and rectified prior to their input into hyperspectral CCRADAR. Detailed evaluation was conducted on three image pairs (see Figure B.2), and structured environments were included to take partially structured areas, such as walls or streets, into account. Furthermore, structured test objects in 5 – 10 m reference distance were utilized to assess the 3D reconstruction accuracy and a mean squared error of 0.0267 m^2 was achieved with the hyperspectral CCRADAR method proposed. Table 5.1 describes the CCRADAR configurations achieving the lowest penalties and, hence, the highest 3D reconstruction accuracy on hyperspectral images. Post-processing with median filtering did not prove useful as it smoothed transitions in disparity images and reduced the disparity estimation accuracy of hyperspectral CCRADAR. Figure 5.2 shows the evaluation image that was used to compare the results from the central processing unit (CPU) and the GPGPU processing with its corresponding disparity image. A quadratic L_2 error of 0.0267 m^2 was measured for the estimated distances to the test objects in 5 – 10 m. Furthermore, the 3D reconstruction from hyperspectral images was also evaluated with the *SET* approach discussed in Section 5.2.2.

Faster calculations with GPGPU parallelization require adaptations to the CCRADAR algorithm with memory access being the most critical



(a) Grayscale image. (b) CCRADAR disparity image.

Figure 5.2 Grayscale image of test scenery (a) and CCRADAR disparity image with final configuration (b). Invalid pixels are colored black.

issue. Therefore, access to the global memory was minimized by repetitively consolidating overlapping windows inside the local memory [265, 324]. Single instruction and multiple thread (SIMT) groups allowed the sharing of local memory allocations, and the provisioning of overlapping SIMT group elements further optimized memory access with tile structures including the edges of neighboring pixel structures of the SIMT window. The CPU implementation of CCRADAR used the OpenCV matrix structure row–column–channel, while the matrix structure was changed to channel–row–column on the GPGPU, and the channel information was imported block-wise to prevent stride inefficiency. Additional adaptions to reduce the calculation effort on the GPGPU included exchanging the L_2 norm with the L_1 norm, utilizing division instead of square root calculation, as well as a linear instead of an exponential weighting in the Census Transform.

An NVIDIA Quadro M6000 with 12 GB RAM, 317 GB/s throughput, and 3072 CUDA kernels was used for an exemplary parallelization. This can speed up the disparity estimation by more than 27× on the GPGPU compared with an Intel Xeon E5-2640 CPU that has eight cores and executes up to 16 threads in parallel [324]. Table 5.2 compares the calculation times on GPGPU and CPU with the acceleration factor n_G indicating the computation reduction on the GPGPU. Here, preprocessing required 52 ms, while rectification took 26 ms.

Parameter	Eval.	Top 4	Chosen
Weighting MCCT	0-1	0.70	0.70
Weighting SAD	0-1	0/0.15	0.15
Weighting SGD_x	0-1	0.10	0.10
Weighting SGD_y	0-1	0/0.05	0.05
Perform *GF*	false/true	true	true
GF regularization	0.001/0.004	0.004	0.004
GF window size	2-25	7/10	7
Perform LRC	false/true	true	true
Perform CBIV	false/true	false	false
Perform *MF*	false/true	false/true	false
MF window size	3-7	5	–

CBIV: Cross-based Iterative Voting according to [139].

Table 5.1 Evaluated hyperspectral CCRADAR parameterizations (Eval.), parameterizations of the four most accurate 3D reconstruction results (Top 4), and chosen, most accurate hyperspectral CCRADAR parameterization (Chosen).

5.1.2 *UEM-CNN*: Disparity Estimation with CNNs

The Unstructured Environment Matching-CNN [327] (*UEM-CNN*) approach proposes three CNN architectures for a local, correlation-based disparity estimation from grayscale stereo images in unstructured environments [327]. The fast MC-CNN architecture of Žbontar and LeCun [309, 310] was chosen as basic architecture due to its promising generalization performance in different domains. The considerably lower computational effort of fast MC-CNN outweighed the minor decrease in depth estimation accuracy relative to accurate MC-CNN. Moreover, Žbontar and LeCun [309, 310] state that fast MC-CNN is less sensitive to major differences between training and testing data. Both faster processing and reduced performance degradation for different domains are beneficial for unstructured environments as the availability of training, validation, and test data is limited here.

Luo et al. [177] propose an evolution of MC-CNN that interprets disparity estimation from stereo images as multi-class classification. The authors [177] evaluated different receptive field sizes from 9×9 to 37×37.

Step	$t_{C,80}$ in ms	$t_{C,174}$ in ms	$t_{G,80}$ in ms	$t_{G,174}$ in ms	$n_{G,80}$	$n_{G,174}$
Init	3.68	2.88	2.05	1.84	1.80	1.57
MCCT	502.79	486.29	4.76	5.05	105.56	96.22
SAD	228.34	404.16	1.04	1.34	218.76	301.95
SGD_x	155.60	272.86	2.80	3.31	55.62	82.55
SGD_y	144.50	272.88	3.39	3.74	42.64	72.85
CCo	43.66	94.65	4.66	8.06	9.36	11.74
GF (L → R)	434.32	913.87	25.56	47.80	16.99	19.12
DS (L → R)	25.89	49.00	1.37	1.39	20.28	35.15
GF (R → L)	432.46	902.03	25.47	47.73	16.98	18.90
DS (R → L)	26.55	48.46	1.37	1.43	19.33	33.95
LRC	0.74	0.67	0.85	0.74	–	–
Total	1998.53	3447.75	73.32	122.43	27.26	28.16

CCo: Cost Computation, *GF*: Guided Filter, *DS*: Disparity Selection.

Table 5.2 CPU (t_C) and GPGPU (t_G) calculation times, $\max(d) = \{80; 174\}$, $n_G = \frac{t_C}{t_G}$.

Luo et al. [177] do not estimate each disparity value independently in contrast to [309, 310] but combine disparity values for each pixel inside a probability distribution centered around the reference during training.

The three *UEM-CNN* architectures proposed in this thesis – *UEM-CNN*$_{base}$, *UEM-CNN*$_9$, and *UEM-CNN*$_{19}$ – estimate disparity on stereo camera images from unstructured environments. The *UEM-CNN* architectures proposed were trained on the KITTI 2012 training dataset [83], and five images from KITTI 2012 depicting unstructured environments were selected to assess the performance of *UEM-CNN* for unstructured environments. All *UEM-CNN* architectures require rectified, grayscale images. The images for training and testing were normalized to a mean intensity value of zero and a standard deviation of one. Furthermore, the images were subdivided into patches prior to the training process. Data filtering with an accuracy validation of the disparity information for extracted image patches was conducted prior to training, validation, and testing to exclude detrimental image patches, according to Section 6.2.1.

5.1.2.1 *UEM-CNN* Architecture

The proposed *UEM-CNN* networks mainly differ in the input patch size and in their training procedure. The two stereo image input patches are analyzed by Siamese layers with shared weights, and on the basis of a similarity metric learned in the training process. 3D tensors are generated after the fusion of the Siamese branches of the networks and contain the matching costs of image patches. Here, each element of the 3D tensor contains a similarity measure s_d for each possible disparity value d of the pixel p. This yields the matching cost

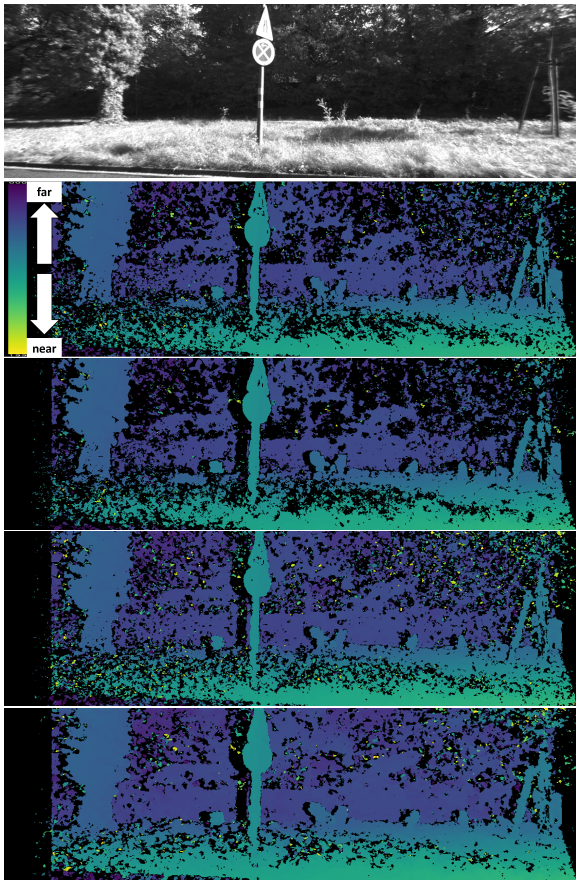
$$C(p,d) = -s_d\langle I^L(p), I^R(p-d)\rangle \tag{5.2}$$

$I^L(p)$ denotes the intensity of p from left input patch (L), while $I^R(p-d)$ denotes the respective intensity values of the input patch from the right image (R). The similarity measure is evaluated by the last network layer, and the disparity is chosen according to the winner-takes-it-all strategy. Hence, the disparity value with the lowest matching cost determines disparity of the respective pixel. Two feature vectors $\mathbf{u}_L$ and $\mathbf{v}_R$ constitute the output of the Siamese network layers from the two input patches. They are compared inside a cosine similarity metric for all *UEM-CNN* architectures:

$$\cos(\mathbf{u}_L, \mathbf{v}_R) = \frac{\mathbf{u}_L \cdot \mathbf{v}_R}{\|\mathbf{u}_L\| \cdot \|\mathbf{v}_R\|}. \tag{5.3}$$

The output of the last network layer during the training process is interpreted by a softmax that derives a probabilistic representation of all disparity values from the numeric output of the last layer. This facilitates the back-propagation during the training of the network. Validation and testing with argmax instead of softmax is possible, as validation and testing do not require differentiability, and proved useful to speed up the disparity estimation in all *UEM-CNN* architectures.

UEM-CNN$_{base}$ is inspired by the MC-CNN [309, 310] and works on 9×9 image patches. It is trained for the binary classification of image patches as matching or non-matching patches. *UEM-CNN*$_9$ and *UEM-CNN*$_{19}$ are trained for multi-class classification, as proposed in [177]. Here, the matching of pixel patches for stereo image disparity estimation is treated as a multi-class classification problem in contrast to the binary classification in *UEM-CNN*$_{base}$. Each possible disparity value represents one class



(a) KITTI image 88.

(b) *UEM-CNN*$_{base}$ 9×9.

(c) *UEM-CNN*$_{base}$ 19×19.

(d) *UEM-CNN*$_{9}$ 9×9.

(e) *UEM-CNN*$_{19}$ 19×19.

Figure 5.3 Disparity estimation on KITTI with $D_{1,\max} = 100$ [327]. *UEM-CNN*$_{base}$ with 19×19 (c) provided a slightly denser disparity estimation in comparison to *UEM-CNN*$_{base}$ 9×9 due to the increased receptive field. *UEM-CNN*$_{19}$ (e) yielded the most dense and accurate estimates, and particularly difficult, unstructured image parts, such as grass and bush structures, were well-reconstructed. Invalid disparities removed in post-processing are colored black.

and *UEM-CNN*$_9$ and *UEM-CNN*$_{19}$ analyze the local neighborhood of the image patches for a similarity assessment. Correlations with neighboring pixels are considered via the integration of all possible disparity values inside a probability distribution learned during training.

UEM-CNN$_9$ processes image patches of 9×9 px and allows a direct comparison to *UEM-CNN*$_{base}$. *UEM-CNN*$_{19}$ takes input patches of 19×19 px inspired by the evaluation results of [177] that achieved the most accurate results for a receptive field of 19×19. The comparison of *UEM-CNN*$_9$ and *UEM-CNN*$_{19}$ facilitates the evaluation of the most suitable receptive field size for the first layer of *UEM-CNN* for unstructured environments. The Siamese network branches with shared weights in *UEM-CNN*$_9$ and *UEM-CNN*$_{19}$ are fused using a dot product. This yields the cosine similarity measure specified in Equation 5.3. The maximum similarity indicates the disparity value with the highest probability for a correct matching of a pixel p with intensity $\boldsymbol{I}^L(p)$ and a pixel $p - d$ with $\boldsymbol{I}^R(p - d)$ equivalent to *UEM-CNN*$_{base}$.

UEM-CNN$_9$ and *UEM-CNN*$_{19}$ integrate all possible disparity values inside a probability distribution and achieved an implicit consideration of the correlation between different disparity values and the local pixel neighborhood. This probability distribution is learned during training: the left and reference image patches are sized according to the receptive field of the first layer, while the height of the right input patches corresponds to the height of the receptive field of the first layer, and the width of the right input patches is equal to the maximum disparity value.

5.1.2.2 Proof of Concept: *UEM-CNN*

UEM-CNN is demonstrated on KITTI2012 [83] and *IOSB-Reg* (see Section 7.4). The KITTI 2012 training set contains 194 grayscale images and corresponding reference data from a Velodyne HDL-64E [83] and was used for training and testing with a validation split of 0.2. The HDL-64E has a vertical FoV of 26.9° and reference disparities were only available for about 70 % of each image. Rectifying the images yielded a resolution of 1260×375 px. More than 600,000 image patches were extracted for a receptive field of 9×9 in *UEM-CNN*$_{base}$ and *UEM-CNN*$_9$ on KITTI, and a receptive field of 19×19 yielded more than 142,000 patches. The maximum disparity value $d(\max)$ was set to 100 for images of KITTI 2012,

according to [83], and to 255 for *IOSB-Reg* images to minimize disparity value quantization effects.

The five most challenging, unstructured images of the KITTI training dataset[1] were selected to validate the three presented *UEM-CNN* architectures for unstructured environments. Additional evaluation was conducted on the *IOSB-Reg* dataset. Here, challenging *IOSB-Reg* images were selected, such as image 03 with overexposed cobblestones, and image 13 dominated by green color and including far off, unstructured elements (see Figure D.3).

The *UEM-CNN* architectures presented were directly evaluated on the disparity maps to facilitate in-depth assessment of each architecture. Pixel-wise error metrics measure the number of pixels whose disparity values diverge from the ground truth with more than the error threshold. Well-known benchmarks for disparity estimation methods, such as the KITTI Vision Benchmark [82] and the Middlebury Stereo Evaluation [240], compare the relative amount of pixel errors within three pixel error (3PE), five pixel error (5PE), or eleven pixel error (11PE). These pixel-wise error metrics were also exploited to analyze and compare the proposed *UEM-CNN* architectures.

Post-processing was evaluated with different thresholds for LRC and different filter sizes for median filtering. Furthermore, different combinations of median filtering and LRC in relation to their order of application were tested and yielded the presented evaluation results.

Figure 5.3 and Figure 5.4 depict disparity estimation results on the KITTI validation data. Figure 5.5 shows the disparity estimation results of $\textit{UEM-CNN}_{19}$ on challenging *IOSB-Reg* dataset images. Table 5.3 shows a superior disparity estimation accuracy of all proposed *UEM-CNN* architectures in contrast to classic SGBM: the 3PE for unstructured KITTI validation images and the empirical mean $\overline{\text{3PE}}$ are lower for all *UEM-CNN* architectures, while the $\overline{\text{3PE}}$ in relation to the prediction density (PD) detailed in Section 5.2.1 ($\frac{\overline{\text{3PE}}}{PD}$) is higher for *UEM-CNN*. $\textit{UEM-CNN}_{base}$ achieved a slightly lower $\overline{\text{3PE}}$ than $\textit{UEM-CNN}_{9}$ and $\textit{UEM-CNN}_{19}$ on the five unstructured images. However, $\textit{UEM-CNN}_{base}$ presented a low *PD* after post-processing with median filtering and LRC and also yielded

[1] KITTI training images 09, 13, 30, 36, 45.

Metric, Image	SGBM	*UEM-CNN*$_{base}$	*UEM-CNN*$_{9}$	*UEM-CNN*$_{19}$
3PE, 09	22.03	11.89	10.94	**10.78**
3PE, 13	33.97	**16.88**	19.16	19.50
3PE, 30	60.48	19.80	21.48	**18.09**
3PE, 36	32.29	**18.70**	19.90	21.13
3PE, 45	31.12	17.80	**15.46**	16.76
$\overline{\text{3PE}}$, unstruct.	35.98	**17.01**	17.38	17.25
$\frac{\text{3PE}}{PD}$, unstruct.	39.87	34.71	27.59	**24.64**
$\overline{\text{3PE}}$, all img.	25.50	15.97	14.74	**14.26**
$\frac{\text{3PE}}{PD}$, all img.	26.84	32.59	23.40	**20.37**

Table 5.3 3PE of classic SGBM and *UEM-CNN* on the selected, unstructured images and on all KITTI 2012 images.

the lowest $\frac{\overline{\text{3PE}}}{PD}$ on the full KITTI validation dataset composed of random KITTI image patches. *UEM-CNN*$_{19}$ achieved the highest $\frac{\text{3PE}}{PD}$ results on the five images from unstructured environments and on the validation set in terms of $\overline{\text{3PE}}$. *UEM-CNN*$_{19}$ also showed a superior performance in terms of 3PE, 5PE, and 11PE: SGBM achieved 25.50 % $\overline{\text{3PE}}$, 17.21 % $\overline{\text{5PE}}$, and 9.84 % $\overline{\text{11PE}}$, while *UEM-CNN*$_{19}$ achieved 14.26 %, 6.71 %, and 3.50 %, respectively.

OpenCV SGBM [119][2] was selected to compare *UEM-CNN* to classic stereo image disparity estimation methods as the block matching approach of SGBM works similarly to the patch matching in *UEM-CNN*. OpenCV SGBM worked with three channel RGB images and the parameterization specified in Section B.1.1 that already proved useful for *IOSB-Reg* images in other applications within this thesis. Table 5.3 and Table 5.4 show that SGBM achieved a higher prediction density by an extensive interpolation of disparity estimates but the accompanying smoothness yielded a notably lower depth estimation accuracy than the proposed *UEM-CNN* networks. Disparity estimation errors became especially evident in unstructured environments with pasture or bushes, as

[2] cv::stereo::StereoBinarySGBM: https://docs.opencv.org/4.5.3/d1/d9f/classcv_1_1stereo_1_1StereoBinarySGBM.html, access on 04.11.2021.

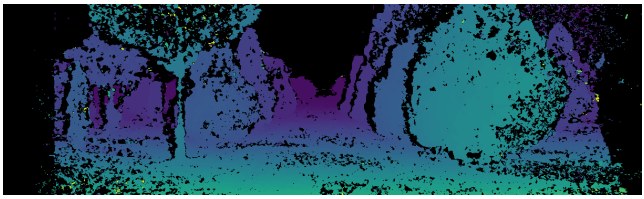
Disparity estimation method	$\overline{\text{3PE}}$ in %	$\overline{\text{R3PE}}$ in %	$\overline{PD}$ in %
SGBM LRC	25.50	17.14	95
UEM-CNN$_{base}$ no post-proc.	39.49	32.33	96
UEM-CNN$_{base}$ LRC	17.43	10.82	50
UEM-CNN$_{base}$ median	34.57	27.56	96
UEM-CNN$_{base}$ median, LRC	**15.97**	**9.59**	49
UEM-CNN$_{base}$ LRC, median	16.66	9.20	55
UEM-CNN$_{base}$ LRC, median, LRC	15.88	9.04	55
UEM-CNN$_{9}$ no post-proc.	31.37	24.7	96
UEM-CNN$_{9}$ median, LRC	**14.74**	**8.71**	63
UEM-CNN$_{19}$ no post-proc.	25.10	17.70	98
UEM-CNN$_{19}$ LRC	14.90	7.65	69
***UEM-CNN*$_{19}$ median, LRC**	**14.26**	**7.19**	70

Table 5.4 3PE, reference-weighted 3PE (R3PE), and *PD* according to Section 5.2.1 for SGBM and *UEM-CNN* on KITTI 2012. LRC and median filtering were conducted with 3 px.

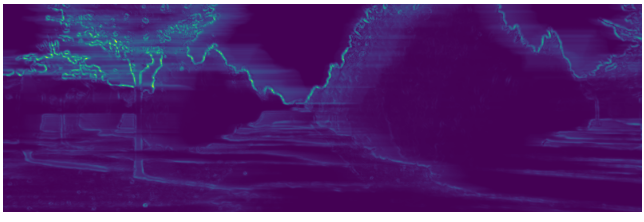
depicted in Figure 5.5 and Figure 5.10, which proved the limited suitability of SGBM for the 3D reconstruction from stereo images in unstructured environments.

Furthermore, an additional *UEM-CNN*$_{base}$ architecture with a receptive field of 19×19 instead of 9×9 was trained and evaluated to facilitate a comparison to *UEM-CNN*$_{19}$ with an equivalent receptive field size. As expected, the consideration of the local pixel neighborhood in *UEM-CNN*$_{9}$ and *UEM-CNN*$_{19}$ clearly outperformed *UEM-CNN*$_{base}$ with a 19×19 receptive field. Thus, further training and evaluation of *UEM-CNN*$_{base}$ with 19×19 patches was not conducted.

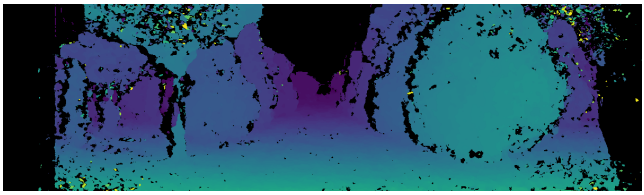
The consideration of the probability distribution of neighboring pixels in *UEM-CNN*$_{9}$ and *UEM-CNN*$_{19}$ generally provided a denser disparity estimation with higher accuracy. The inclusion of the local pixel neighborhood especially improved the disparity estimation in image parts with predominant coloring such as image 88 from KITTI 2012. This assumption was verified by comparing the special purpose *UEM-CNN*$_{base}$ with a 19×19 receptive field to *UEM-CNN*$_{19}$ illustrated in Figure 5.3.



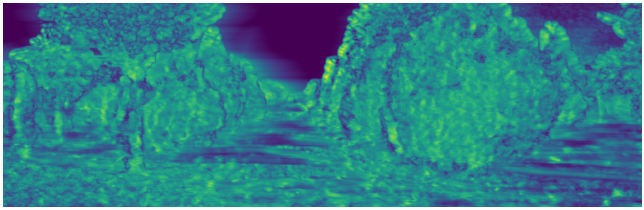
(a) Disparity map *UEM-CNN*$_{base}$.



(b) Confidence map *UEM-CNN*$_{base}$.



(c) Disparity map *UEM-CNN*$_{19}$.



(d) Confidence map *UEM-CNN*$_{19}$.

Figure 5.4 Disparity and left confidence maps for image 09 of KITTI 2012, $D_{1,\max} = 100$ [327]. Dark blue indicates low confidence, yellow symbolizes high confidence. *UEM-CNN*$_{19}$ achieved a higher confidence than *UEM-CNN*$_{base}$.

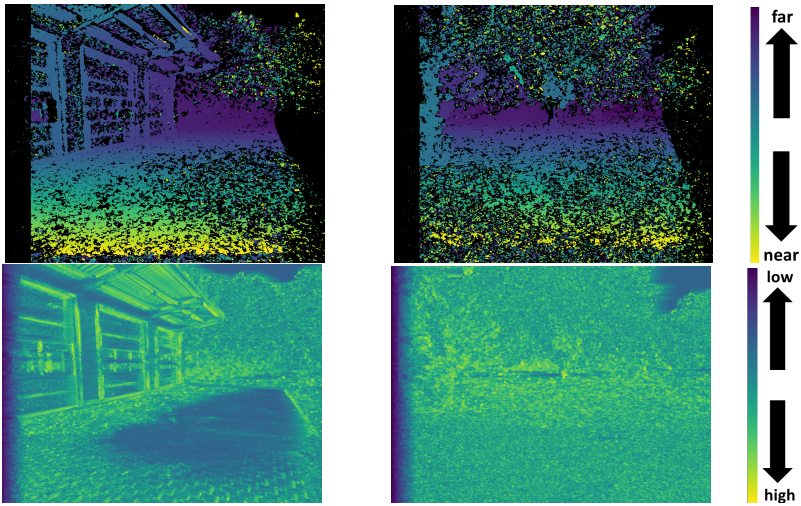


Figure 5.5 Disparity and left confidence maps for image pair 03 and 06 of *IOSB-Reg* [327], generated with *UEM-CNN*$_{19}$ using the left camera as reference, $D_{1,\max} = 255$. Invalid disparity values are removed during post-processing and the respective pixels are colored black. Yellow indicates high and blue low confidence.

Post-processing with LRC and median filtering are lightweight in terms of computational effort and presented very effective results in the identification of estimation errors and a suitable smoothing of the disparity estimation results. Table 5.4 shows the evaluation results of *UEM-CNN*$_{base}$, *UEM-CNN*$_9$, and *UEM-CNN*$_{19}$ for different post-processing configurations, while more extensive evaluation results are given in Table B.1. Experimental evaluation yielded the best combination of accuracy and prediction density for stereo image disparity estimation from unstructured environments with a median filter window of 3×3 px and an LRC threshold of 3 px. Consequently, all *UEM-CNN* architectures apply median filtering with a window of 3×3 px and LRC with a threshold of 3 px for testing and validation.

To conclude, the superior performances of *UEM-CNN*$_9$ and *UEM-CNN*$_{19}$ show that treating stereo image disparity estimation as a multi-class classification problem achieves more accurate results than the CNN

training for stereo matching with binary predictions. Furthermore, this thesis recommends the *UEM-CNN*$_{19}$ architecture for stereo image disparity estimation in unstructured environments as it yielded the most accurate estimation results.

5.2 Validating Stereo Image Disparity Estimation

Stereo image disparity estimation results can be assessed on both the disparity maps and the resulting 3D point clouds. The in-depth evaluation of disparity maps for off-road vehicles in unstructured environments is hereinafter, while 3D reconstruction assessment is described in Section 5.2.2.

5.2.1 Customized Error Metrics for Disparity Maps

Autonomous off-road vehicles must be capable of accurately detecting navigation obstacles and manipulation objects that can also be deduced from stereo camera 3D reconstruction. Here, incorrect detection of nearer objects constitutes a notably higher risk to the vehicle, and the detection of near objects is more important than the detection of objects farther away. Depending on the subsequent application of the estimated depth values, pixel error metrics such as 3PE do not contain sufficient information to assess the performance of stereo image disparity estimation methods in unstructured environments. For this reason, this thesis proposes eight additional error metrics to assess the disparity estimation results.

5.2.1.1 Novel Error Metrics for Unstructured Environments

The novel disparity error metrics presented focus on stereo image disparity estimation for off-road vehicles [327], and permit to determine potential optimizations for unstructured environments:

- Confidence maps,
- Tile error,
- Median-filtering the difference map,
- Doubling of negative errors/reference-weighted pixel error,
- Weighting related to the distance from the camera,

- Range-limit error weighting function,
- Prediction density (*PD*), and
- Pixel error in relation to prediction density ($\frac{\text{3PE}}{PD}$).

The difference map $\mathbf{D}_{\text{diff}}$ measures the deviation of the estimated disparity values ($\mathbf{D}_1$) from the reference disparity extracted from the LiDAR reference data ($\mathbf{D}_2$) for each pixel $[j,k]$ of an image pair with $M \times N$ pixels:

$$\mathbf{D}_{\text{diff}}[j,k] = \mathbf{D}_1[j,k] - \mathbf{D}_2[j,k] \tag{5.4}$$

The disparity is anti-proportional to the estimated depth and high disparity values indicate low depth values. Thus, if the estimated $\mathbf{D}_1[j,k]$ is lower than the reference $\mathbf{D}_2[j,k]$, the related pixel is estimated to be farther away than it is in reality, and negative values in $\mathbf{D}_{\text{diff}}$ can be dangerous in collision avoidance. Subsequently, the maximum disparity is abbreviated with $d_x(\max) = \max_{j=1,k=1}^{M,N}(\mathbf{D}_x[j,k])$, $x \in 1,2$ for clarity.

Confidence maps, as proposed in [281], facilitate a validation of the estimated disparities, error visualization, and the detection of problematic image characteristics for disparity maps. The proposed confidence maps are inspired by the peak ratio measure [218], where a high reliability of the disparity assignment is indicated by a high peak ratio. For both classic and CNN methods, the confidence can be obtained by determining the probability of a chosen disparity value in relation to other possible disparity values. For CNNs, the probability P of the disparity $\mathbf{D}_1[j,k]$ can be derived from the costs for $\mathbf{D}_1[j,k]$ in the activation function of the last CNN layer:

$$\text{softmax}(\mathbf{D}_1[j,k]) = \frac{\exp\left(\mathbf{D}_1[j,k]\right)}{\sum_{j,k}^{M,N} \exp\left(\mathbf{D}_1[j,k]\right)}. \tag{5.5}$$

Here, the confidence map visualizes the probability corresponding to the chosen disparity by applying an argmax function. It is illustrated together with the estimated disparity maps in Figure 5.4 and Figure 5.5.

The tile error metric rewards true estimations and punishes false estimations using a predefined window of 3×3 px. The minimum single pixel error value within this tile window determines the value of the respective pixels inside the tile, and correctly estimated disparities are rewarded higher than falsely estimated disparity values. This assesses

the disparity estimation of the evaluated method in smaller areas instead of single pixels and also allows a more global analysis if challenging images for disparity estimation are present.

The application of a median filter on $\mathbf{D}_{\text{diff}}$ smooths pixel-wise errors but preserves errors on the edges and in larger scale. The doubling of negative errors, also referred to as reference-weighting pixel error, penalizes depth estimations that assume pixels or areas to be farther away than they are in reality. The weighting of negative values in Equation 5.4 is doubled for this purpose. Table 5.4 demonstrates its application using a reference-weighted 3PE (R3PE). It highlights critical estimation errors and assesses disparity estimation methods with a special focus on collision avoidance.

The range-limit error weighting $\mathcal{W}$ allows an even more customized error weighting. Here, the elements of $\mathbf{D}_{\text{diff}}$ are multiplied with the corresponding elements of an $N \times M$ weight map $\mathcal{W}$:

$$\mathcal{W}[j,k] = \frac{1}{d_{2,\max}} \begin{cases} b, & \mathbf{D}_2[j,k] \leq T_L \\ b + O, & T_L < \mathbf{D}_2[j,k] < T_U \\ d_{2,\max}, & T_U \leq \mathbf{D}_2[j,k] \end{cases} \tag{5.6}$$

$$\mathbf{O}[j,k] = \quad (d_{2,\max} - b)\Big(\frac{\mathbf{D}_2[j,k] - T_L}{T_U - T_L}\Big)^{\chi}. \tag{5.7}$$

Offset b, upper threshold T_U, and lower threshold T_L, as well as power factor χ facilitate its customization. Here, b, T_U, and T_L limit the quadratic function to the relevant disparity range. The recommendations for b, T_U, and T_L depend on the sensor setup and the perceived environment, as the disparity values occur only in a limited range. A weighting related to the distance from the camera specifically takes the estimated distance from camera and robot into account and can be derived from Equation 5.6 with $\chi = 1$, $T_L = 0$, and $T_U = d_{2,\max}$. In this case, $\mathcal{W}$ contains high weighting values for high disparities close to the origin of the camera frame, and also rescales disparity values of the reference $\mathbf{D}_2$ using the offset b. In order to specially consider the theoretical depth estimation limitations in stereo camera setups according to Equation 3.17, the utilization of $\chi = 2$ proved useful (see Figure B.3).

Prediction density (*PD*) measures the ratio of pixels with a valid disparity estimate in a disparity map and quantifies the capability of different disparity estimation methods to provide a dense disparity map. Using a

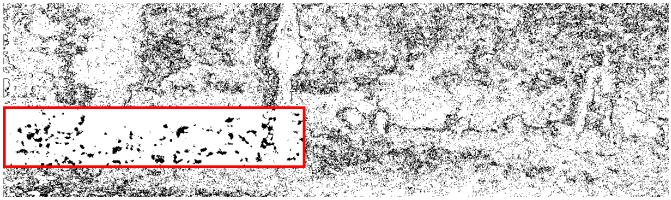
3PE for instance, a disparity estimate is valid if it deviates three pixels or less from the reference disparity. Here, *PD* only counts the number of pixels that lie inside the joint FoV of the stereo camera system, as only those can achieve a potentially correct disparity estimate with non-overlapping image areas being disregarded. Experimental evaluation showed that post-processing decreases the *PD* but generally increases the average estimation accuracy as expected.

The combination of *PD* and 3PE provides an additional error measure: the $\frac{\text{3PE}}{PD}$ measure relating the ability of a disparity estimation method to provide intrinsically valid estimates to the accuracy of its estimated disparity values. Thus, $\frac{\text{3PE}}{PD}$ combines the density and accuracy assessment of stereo image disparity estimation in one metric. Depending on the application scenario, an accurate disparity estimation is more important than a high *PD* with lower accuracy. For instance, *PD* is regarded as less important than accuracy for critical applications, such as the navigability analysis for autonomous off-road vehicles.

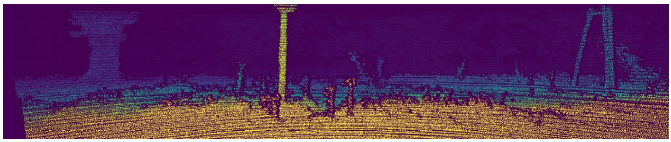
5.2.1.2 Proof of Concept: Novel Error Metrics

Figure 5.6 demonstrates a selection of the proposed disparity error metrics on image 88 of the KITTI 2012 dataset. Here, the tile error highlights image parts with a predominantly valid disparity estimation. Figure 5.5 shows the disparity and confidence maps of image 03 of the *IOSB-Reg* dataset. The yellow areas around edges and well-textured areas show a high confidence of the estimated disparities. Non-overlapping or partially overlapping image parts, such as the left margin of the left camera image, present the expected decreasing confidence values. The raw $\overline{\text{5PE}}$ on all images of the KITTI 2012 training set amounted to 26.80 % for *UEM-CNN*$_{19}$ without post-processing, while the tile $\overline{\text{5PE}}$ decreased to 21.70 % with a 3×3 pixel tile window with the seemingly correct reconstructions of the grass and bush areas in image 88 of KITTI being emphasized. Median filtering smoothed the difference map and single pixel estimation errors were discarded. This especially highlighted image parts with predominantly valid disparity estimation results. A 3×3 median filtering on all disparity estimation results of the KITTI 2012 training dataset yielded a $\overline{\text{5PE}}$ of 43.90 % for *UEM-CNN*$_{19}$ [327]. The effect of median filtering with a 3×3 filter mask is depicted in the highlighted part

(a) Raw 5PE: disparity errors with d >5PE are colored red.



(b) Difference map $\mathbf{D}_{\text{diff}}$ for *UEM-CNN*$_{19}$ with *MF*.



(c) Range-limit error function $\mathcal{W}$ mapped onto $\mathcal{L}$ (see Equation 5.6).

Figure 5.6 Evaluation of the proposed error metrics for *UEM-CNN* disparity estimation on Figure 5.3 [327]: (a) Green indicates estimated $d \leq$ 5PE. Especially areas with low and high exposure or low texture introduce high estimation errors. The red box in (b) highlights the effect of a 3×3 median filter. White pixels mark d errors $\leq$ 5PE. Dark coloring illustrates higher estimation errors. Yellow in (c) symbolizes close ranges with high weighting, dark blue indicates lower weights.

of Figure 5.6(b): it clearly emphasized that the problematic, overexposed grass parts in image 88 mainly produce high disparity estimation errors. If only non-occluded errors are considered, as proposed in [82], the $\overline{\text{5PE}}$ decreased to 6.70 %, compared to the raw $\overline{\text{5PE}}$ of 26.80 % which showed the dominant influence of occlusion problems in stereo image depth estimation.

The doubling of negative errors yielded a $\overline{\text{5PE}}$ of 29.00 % for *UEM-CNN*$_{19}$ on the KITTI 2012 training images. Disparity values estimated lower than the LiDAR reference are penalized quadratically. A too high depth was estimated for the overexposed grass parts as well as for low-textured parts of the bushes in image 88. The weighting related to their

distance from the sensor origin yielded a $\overline{5\text{PE}}$ of 24.30 %. The comparison of this weighted $\overline{5\text{PE}}$ for different stereo methods analyzes the depth estimation accuracy in relation to the distance to the camera. The range-limit error weighting provides a comparison method similar to $\overline{5\text{PE}}$, but with the possibility to customize the weighting parameters, as stated in Equation 5.6. Figure 5.6(c) shows an exemplary range-limit error weighting.

Furthermore, Table 5.4 compares the *PD* of SGBM and the proposed *UEM-CNN* architectures. Here, SGBM had a high $\overline{\text{PD}}$ of 95 % compared to *UEM-CNN*$_{base}$ with 49 %, while *UEM-CNN*$_{19}$ achieved a sufficiently high prediction density with a suitable depth estimation accuracy in terms of $\overline{3\text{PE}}$.

In conclusion, confidence maps presented a well-suited measure to indicate the reliability of the estimated disparity values and to identify difficult image areas for stereo image disparity estimation, such as overexposed cobblestones in Figure 5.5. Tile error and median filtering highlighted image areas that were very easy or very difficult to reconstruct, which facilitates a special focus on these areas in the analysis of the examined disparity estimation method.

5.2.2 *SET*: Stereo Evaluation Toolbox

The *SET* approach contributes to the interpretation and validation of mid-level perception results and answers questions such as: "Which camera system performs best with a defined reconstruction algorithm in a particular application environment?" and "How does a specific camera system influence reconstruction performance of an algorithm?" [325, p.1]. As the development of a stereo camera setup requires the consideration of many individual system characteristics, *SET* facilitates a well-founded selection of a suitable camera–algorithm combination for a specific application environment on the basis of comparable criteria. As a result, *SET* assesses the combination of all modular system components with their respective characteristics: camera specifications, image resolution, image noise, camera calibration, stereo image disparity estimation algorithm, and the specific application environment. Each of these individual characteristics has its measurements that can be utilized to optimize its individual performance. For a flexible integration in the perception–validation pipeline in this thesis, *SET* is designed for a fast integration of

arbitrary stereo camera systems and disparity estimation methods as it evaluates the generated stereo camera point clouds.

Several benchmarks for disparity estimation from stereo images exist, such as Middlebury Stereo Evaluation [240], KITTI Vision Benchmark [82], and others discussed in Section 2.6.4. They typically assess stereo image disparity estimation algorithms on provided artificial and partially stereo-beneficial images, and the benchmark evaluation does not consider the influences of different camera systems or application environments. As a result, most benchmark evaluations do not necessarily reflect the performance of stereo image disparity estimation algorithms in challenging application environments. The *SET* approach is proposed to overcome this and complements well-known stereo vision benchmarks with an analysis of the 3D reconstruction performance of stereo camera systems. Complementary to these benchmarks, the evaluation images for *SET* are captured in the targeted application scenario by a real stereo camera setup, and one generic overall score is provided for 3D stereo point clouds. This facilitates the comparison of different combinations of individual modules in the final application scenery.

SET is divided up into a static and a dynamic evaluation step. Static evaluation denotes the assessment of 3D stereo point clouds generated from a single stereo image pair in static scenes. In addition, the dynamic evaluation in *SET* [325] analyzes camera-based visual SLAM on its potential to provide accurate and suitable 3D reconstruction results for subsequent localization and mapping tasks (see Section B.2.1).

Summarizing, *SET* proposes a modular concept to assess camera systems in combination with their algorithms for 3D reconstruction and complements the existing, individual measures with one holistic 3D reconstruction score.

5.2.2.1 Static Evaluation: 3D Reconstruction Assessment

The static evaluation is separated in qualitative and quantitative criteria, as summarized in Table 5.5. Stereo image disparity estimation was evaluated in structured indoor, structured outdoor, and unstructured outdoor environments highlighting the generalization performance of the *SET* approach. The proposed qualitative criteria measure the suitability of the 3D reconstruction for human operators in use cases such as

Criterion	Identifier	w_i *SI*	w_i *UO*
Qualitative			
Cloud density	*CD*	0.20	0.50
Monochr. surfaces	*MoS*	0.30	0.15
Geometry	*Geo*	0.30	0.15
Consistency on edges	*CoE*	0.20	0.20
Quantitative			
Nearest neighbor	*NNS*	0.40	0.40
Mean dist. surfaces	*MDS*	0.30	0.40
Surface orientation	*SOe*	0.30	0.20

Table 5.5 Static, qualitative and quantitative evaluation criteria in *SET* with empirically justified, corresponding weightings for indoor (*SI*) and outdoor environments (*UO*) [325].

person indoor navigation, while the quantitative criteria provide objective evaluation results. As 3D LiDAR sensors generate highly accurate 3D information, 3D LiDAR data is utilized as ground truth reference data for the 3D reconstruction from stereo image disparity estimation. To this end, the LiDAR reference cloud ($\mathcal{L}$) and the 3D stereo camera cloud ($\mathcal{S}$) are registered within a common frame, as described in Chapter 4. The qualitative metrics are rated between 0 and 10. Here, a higher score indicates a better, more accurate and more useful, 3D reconstruction. A qualitative score close or equivalent to 10 indicates the best 3D reconstruction performance in comparison to the other evaluated stereo camera systems. Processing effort and runtime were measured for each camera–algorithm combination but not integrated as *SET* focuses on 3D reconstruction accuracy.

Cloud density (*CD*) plays an important role in the proper 3D reconstruction of the environment, especially in unstructured environments. Outdoors, *CD* notably influences the performance of stereo camera systems whereas the representation of monochromatic surfaces (*MoS*) and geometrical correctness (*Geo*) is of greater importance indoors. An up- or downsampling of the disparity map can lead to a higher or lower cloud density with an identical quality of $\mathcal{S}$. This entails a higher cloud

density score but also a lower qualitative user rating for the geometrical correctness. Hence, the overall qualitative score is not altered.

The consistency of disparity maps on the edges (*CoE*) is regarded as equally relevant in- and outdoors as the susceptibility to errors is similar. The weights w_i, according to Table 5.5, are multiplied by the obtained scores i to obtain the qualitative score

$$S_{\text{qual}} = \sum_i i \cdot w_i, i \in \{CD, MoS, Geo, CoE\}. \tag{5.8}$$

All quantitative criteria are related to $\mathcal{L}$. The outlier ratio $NNS = N_r/N_s$ determines the ratio between the number N_r of remaining points after outlier filtering and the number N_s of all points in $\mathcal{S}$ via a kNN search for all points in $\mathcal{S}$ within $\mathcal{L}$, as described in Section 4.2.1.

Figure 5.7 describes determining the mean distance of surfaces (*MDS*) measure, according to [325]. The RANSAC algorithm proved useful in estimating surfaces in $\mathcal{L}$ and $\mathcal{S}$ to determine the quantitative *MDS* and *SOe* measures. Surface orientations are estimated, as discussed in Section 3.6, with a radius of 0.40 m. *MDS* evaluates the L_1 distances $d_{i,j}$ between corresponding estimated surfaces k_j in $\mathcal{L}$ and the aligned reference points in $\mathcal{S}$ ($\mathbf{Q}_i$) with

$$\overline{d_{i,j}} = \frac{\sum_{i=0}^{m} |n_i^0|}{m}, \tag{5.9}$$

as illustrated in Figure 5.7. Here, m denotes the number of nearest neighbor (NN) points for a point $\mathbf{P}_i$ inside the corresponding surface k determined by the RANSAC algorithm. The experimentally justified perception range of stereo camera systems with sufficiently high accuracy for critical application scenarios is 10 m for current camera systems. Hence, the tolerable *MDS* error is set to 0.8 m in accordance with the error definition of [82], and the *MDS* measure is normalized with $MDS = \overline{d_{i,j}}/0.8\,\text{m}$.

The corresponding normal $\mathbf{n}_{i,\mathcal{S}}$ is compared to the surface normal $\mathbf{n}_{k_j}$ for each point in $\mathcal{S}$ inside the RANSAC plane model for a surface k_j in $\mathcal{L}$. The surface orientation measure (*SOe*) is normalized with 180° and summed up over all three axes with

$$SOe = \sum_{\mathbf{Q}_{i,j} \in k_j} \sum_{o=1}^{3} \frac{\left\| \mathbf{n}_{k_j}[o] - \mathbf{n}_{i,\mathcal{S}}[o] \right\|}{180^\circ} \tag{5.10}$$

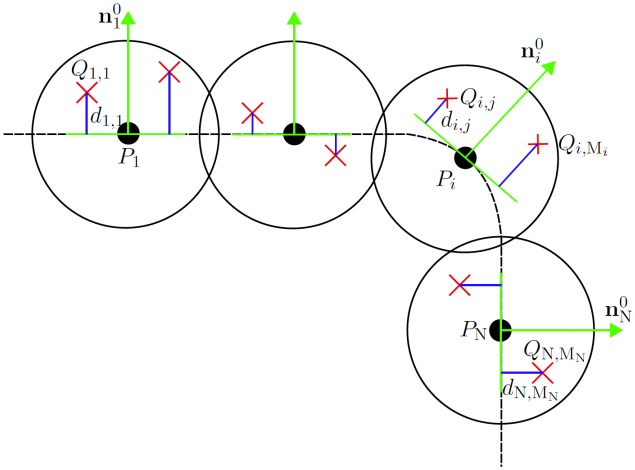


Figure 5.7 Mean distance of surfaces (*MDS*), according to [325]: $\mathbf{P}_i$ are LiDAR points, $\mathbf{Q}_{i,j}$ are NN stereo points of $\mathbf{P}_i$, $\mathbf{n}_i^0$ normal vectors, $d_{i,j}$ distance in direction of normals by projection on the normal vector $\mathbf{n}_i^0$.

to deduce an absolute measure for the orientation error of the stereo points $\mathbf{Q}_{i,j}$. An equivalent importance for the proximity of stereo points to the ground truth (*NNS*) for indoor and outdoor environments proved useful, as it directly describes the 3D reconstruction error of $\mathcal{S}$. Typically, a lower number of smooth surfaces are encountered outdoors than indoors making the comparison of their estimated orientation (*SOe*) less meaningful outdoors than the average distance between the approximated surface elements (*MDS*). In contrast, smooth surfaces such as walls are often present indoors, which justifies the same importance of the *MDS* and *SOe* criteria here.

Table 5.5 summarizes the experimentally justified weightings for each criterion. A low quantitative score highlights an accurate 3D reconstruction performance of a stereo camera system, and the quantitative score S_{quan} is calculated equivalent to S_{qual} with the weights w_i according to Table 5.5:

$$S_{\text{quan}} = \sum_i i \cdot w_i, i \in \{NNS, MDS, SOe\}. \tag{5.11}$$

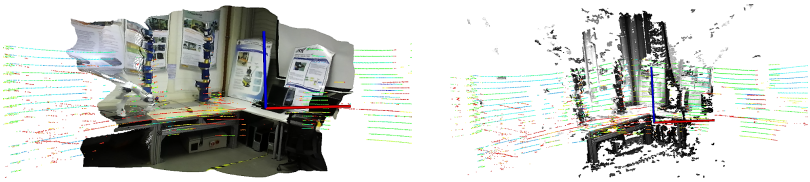
Criterion i	*SI*: ZED	*SI*: rc_v.	*SI*: MQ_{ORB}	*SI*: MQ_{SGBM}	*UO*: MQ022
CD (0–10)	10	8	6	9	9
MoS (0–10)	10	8	5	9	8
Geo (0–10)	2	9	10	9	10
CoE (0–10)	1	10	6	8	9
S_{qual}	5.8	8.7	6.9	**8.8**	9.0
NNS (0–1)	0.422	0.668	0.532	0.473	0.750
MDS (0–1)	0.128	0.105	0.101	0.094	0.204
SOe (0–1)	0.167	0.017	0.028	0.011	0.028
S_{quan}	0.257	0.304	0.246	**0.229**	0.560

Table 5.6 Static *SET* evaluation of the ZED, rc_visard (rc_v.), $MQ013_{ORB}$ (MQ_{ORB}), and $MQ013_{SGBM}$ (MQ_{SGBM}) stereo camera systems in structured indoor environments (*SI*) and the MQ022 stereo camera system in unstructured outdoor environments (*UO*) with a kNN filtering radius (*NNS*) of $\sqrt{0.05\,\text{m}}$. Figure 5.8 depicts the indoor scenes with $\mathcal{S}$ and $\mathcal{L}$.

Finally, the further processing of the stereo camera 3D reconstruction results determines the relative importance of the qualitative and quantitative scores. Subsequent utilization of the 3D reconstruction for a situation assessment for first responders might favor higher qualitative scores, while further processing of the 3D reconstruction for SLAM led to a higher importance of the quantitative scores.

5.2.2.2 Proof of Concept: *SET*

The *SET* approach was experimentally validated on multiple camera systems in structured indoor environments and unstructured outdoor environments: Stereolabs ZED, Roboception rc_visard 160, Ximea MQ013RG-E2, and Ximea MQ022HG-IM-SM4X4-VIS. ZED and rc_visard are off-the-shelf stereo camera systems, while the MQ013 and MQ022 are self-developed stereo camera systems with feature-based and correlation-based disparity estimation (see Table B.2 for technical details). ZED, rc_visard, and the MQ013 stereo camera system were evaluated on structured indoor environments, while the hyperspectral MQ022 stereo camera system was tested in unstructured outdoor environments, as de-



(a) ZED stereo. (b) MQ013 stereo with ORB-SLAM2.

Figure 5.8 Exemplary *SET* evaluation on structured indoor scene with Velodyne VLP-16 reference data [325]. *SET* scores can be found in Table 5.6.

scribed in Section 5.1.1. SGBM and feature-based stereo image disparity estimation with ORB features were evaluated for the MQ013 stereo camera system, whereas the MQ022 stereo camera systems rely on local, correlation-based CCRADAR (Section 5.1.1). Table 5.6 and Figure 5.8 show selected static evaluation results, while a more detailed experimental evaluation and implementation details for SGBM, ORB-SLAM2, and CCRADAR are provided in [318, 324, 325]. The MQ013 stereo system with SGBM disparity estimation yielded the most accurate and useful performance among the four stereo camera systems being compared in structured indoor environments. The static *SET* evaluation of the MQ022 stereo system in unstructured outdoor environments highlights the different character of both environments: only a low number of smooth surfaces exists in unstructured environments making surface estimations notably more difficult. The higher surface variations in unstructured environments were well-captured by single 3D LiDAR measurements but not by pixel-wise disparity estimates in larger distances from the origin of the stereo camera system.

5.3 Sensor Data Fusion

The perception of unstructured environments, especially for heavy construction machinery with manipulation capabilities, requires a 3D environmental perception. This facilitates accurate navigability analysis, object detection, and obstacle avoidance, as well as suitable workspace monitoring during manipulation tasks. Here, the fusion of sensor data from multiple sensors with complementary characteristics increases the

knowledge on the perceived environment and can also reduce sensor data uncertainty. While the fusion of similar-source LiDAR data is straightforward and integrated in the respective registration approach, as explained in Section 4.2, an appropriate and efficient fusion of cross-sensor sensor data is notably more complex, especially in unstructured environments.

Hence, this thesis proposes 3D–3D fusion methods for cross-source sensor data operating on the level of individual measurements in 3D space. These 3D measurements directly originate from active 3D sensors, from low-level perception results, or from mid-level perception results that also map color from 2D images into 3D space. Different fusion approaches finally providing one 3D point cloud from multiple sensor inputs are proposed with different levels of complexity, data validation, and computational effort for various sensor setups and off-road vehicles in unstructured environments. The proposed methods are demonstrated on the common perception sensor setup for off-road vehicles: RGB images from camera systems, dense 3D point clouds with limited geometric accuracy from stereo image disparity estimation, and sparse, highly accurate 3D LiDAR point clouds. Hence, the 3D–3D fusion methods proposed require the availability of a multi-sensor system with at least one calibrated stereo camera system or RGB-D camera and one 3D LiDAR sensor. The stereo camera or RGB-D camera point cloud ($\mathcal{S}$) contains geometric and color information, and $\mathcal{S}$ is fused with one or more 3D LiDAR point clouds ($\mathcal{L}$). Hence, $\mathcal{S}$ can complement $\mathcal{L}$ with dense depth estimates, and $\mathcal{S}$ and $\mathcal{L}$ are merged as 3D point clouds to alleviate information loss and reprojection errors. Here, the successful and accurate sensor calibration forms the basis for an accurate and fruitful fusion of sensor data (see Section 4.2 and 4.3). The subsequently analyzed fusion approaches rely on accurate sensor calibration with singularity-free transformations.

Object-oriented or iterative fusion approaches, as discussed in Section 2.4.3, are not suitable for unstructured environments mainly containing unknown topological structures and difficult-to-separate objects. The fusion of raw, unaltered 2D and 3D sensor data proposed in this thesis prevents information loss that can occur in object-oriented fusion approaches during the filtering or transformation for object extraction.

Wolf and Berns [296] assign a higher priority to LiDAR measurements within their voxel model of the environment so that 3D points from stereo

camera disparity estimation are eliminated if LiDAR measurements are available. The fusion methods $\mathcal{A}$ to $\mathcal{C}$ in this thesis pursue a similar strategy and include all LiDAR measurements in the fused cloud. However, LiDAR sensors can also provide erroneous measurements and are hence analyzed in the proposed fusion method $\mathcal{D}$. Naturally, confidence analysis, as discussed in Section 4.1, can contribute to a selection of valid and reliable measurements during the fusion process. In order to consider sensor confidence, this thesis also proposes a novel, confidence-based 3D–3D fusion approach considering the confidence of 2D images and 3D point clouds in a tightly coupled manner.

5.3.1 3D–3D Fusion of Cross-Source Sensor Data

This thesis proposes different 3D–3D fusion methods for 3D point clouds:

- $\mathcal{A}$: direct fusion of $\mathcal{L}$ and $\mathcal{S}$ colorless, with color from $\mathcal{S}$ for $\mathcal{L}$, or with color from the RGB image for $\mathcal{L}$,
- $\mathcal{B}$: kNN outlier filtering of inaccurate stereo or RGB-D depth estimates similar to Section 4.2,
- $\mathcal{C}$: threshold filtering of inaccurate stereo or RGB-D depth estimates with a minimum depth estimation accuracy,
- $\mathcal{D}$: confidence assessment for each 3D point in $\mathcal{L}$ and $\mathcal{S}$ (see Section 4.1).

$\mathcal{A}$ provides a basic and fast 3D–3D fusion similar to the merging process of similar-source LiDAR clouds in Section 4.2. Hence, $\mathcal{A}$ delivers large, dense 3D clouds without filtering possibly inaccurate and redundant 3D measurement points. Two options are possible for color assignment: the assignment of intensity/color information from the RGB image to $\mathcal{L}$ exploits the 2D–3D fusion detailed in Section 3.7, while intensity/color information from $\mathcal{S}$ assigns the color of the nearest neighbor of $\mathbf{p}_{s,\mathcal{L}} \in \mathcal{L}$ within the aligned $\mathcal{S}$ to $\mathbf{p}_{s,\mathcal{L}}$ if $||\mathbf{p}_{s,\mathcal{S}} - \mathbf{p}_{s,\mathcal{L}}||_1 \leq d_{\mathrm{NN}}$ is met. Here, color assignment from the RGB image showed more accurate results to color $\mathcal{L}$ as stereo camera 3D reconstruction accuracy decreases for larger distances.

The exclusion of too inaccurate depth estimates from $\mathcal{S}$ in the fusion methods $\mathcal{B}$ and $\mathcal{C}$ can increase the overall accuracy and also the reliability of the environment perception in 3D due to the accuracy limitations in $\mathcal{S}$ (see Section 3.8).

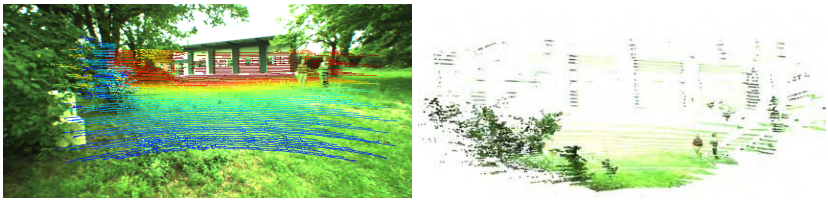
$\mathcal{B}$ requires a minimum number of nearest neighbors within a predefined maximum distance d_{NN} to verify the stereo depth estimates prior to their integration into the fused 3D cloud. Thus, method $\mathcal{B}$ only includes points $\mathbf{p}_{s,\mathcal{S}}$ fulfilling $||\mathbf{p}_{s,\mathcal{S}} - \mathbf{p}_{s,\mathcal{L}}||_1 \leq d_{\text{NN}}$ for $N_{\mathcal{L}} \geq N_{\text{min}}$ LiDAR points $\mathbf{p}_{s,\mathcal{L}}$ in the 3D–3D fusion cloud.

Workspace monitoring for autonomous vehicles with manipulation capabilities requires a dense 3D reconstruction in close range around the platform. Thus, $\mathcal{C}$ proposes the integration of close range measurements from $\mathcal{S}$ in $\mathcal{L}$. Depending on the selected stereo image disparity estimation method, the stereo camera setup, and the application environment, different values for the disparity estimation error ε_d, baseline $\boldsymbol{B}$, and focal length f determine the depth estimation accuracy ε_z. Method $\mathcal{C}$ keeps all 3D points of $\mathcal{L}$ and integrates 3D stereo measurements, if their ε_z is smaller than a predefined maximum depth estimation accuracy $\max(\varepsilon_z)$. Here, $\boldsymbol{B}$ and f are fixed for each stereo camera setup, and ε_d depending on the stereo algorithm is selected by the user in accordance with the required depth estimation accuracy. Naturally, the proposed methods $\mathcal{B}$ and $\mathcal{C}$ can also be combined to only include validated, close range measurement points from $\mathcal{S}$ in $\mathcal{L}$.

Finally, fusion method $\mathcal{D}$ integrates the concept of confidence measures for raw sensor data discussed in Section 4.1. Contrasting $\mathcal{A}$ to $\mathcal{C}$, the LiDAR cloud $\mathcal{L}$ is not assumed to be error-free, and $\mathcal{D}$ considers the confidence of both $\mathcal{L}$ and $\mathcal{S}$. A confidence threshold specifies the minimum confidence for each individual 3D measurement j of $\mathcal{L}$ ($c^{\text{3D}}_{\mathcal{L},j}$) and $\mathcal{S}$ ($c^{\text{3D}}_{\mathcal{S},j}$) to be met for its inclusion in the 3D–3D fusion cloud. Here, $c^{\text{3D}}_{\mathcal{L},j} \in [0,1]$ and $c^{\text{3D}}_{\mathcal{S},j} \in [0,1]$ are determined according to Section 4.1. The weak, medium, and strong filtering thresholds defined in Section 3.12 are applied: a weak threshold requires $c^{\text{3D}} \geq 0.6827$, a medium threshold $c^{\text{3D}} \geq 0.8664$, and a strong threshold eliminates all 3D measurements with $c^{\text{3D}} \geq 0.9545$.

5.3.2 Proof of Concept: 3D–3D Fusion

Figure 5.9 shows exemplary 2D–3D fusion results exploited to assign color information from the RGB image to $\mathcal{L}$. The proposed 3D–3D fusion methods are demonstrated on two selected, exemplary scenes of the



(a) (b)

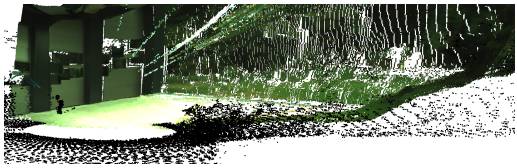
Figure 5.9 Fusion results: (a) Projection of the geometric 3D information from $\mathcal{L}$ onto exemplary *IOSB-Reg* image; (b) 2D–3D fusion result for (a) with image color (Velodyne HDL-64E cloud, JAI AD-130GE camera) with $\mathcal{L}$ a condensed point cloud from eleven Velodyne HDL-64E scans containing 3D LiDAR points inside the camera FoV only. Image (a) © Fraunhofer IOSB.

IOSB-Reg dataset: a primarily structured scene I (Figure 5.10(a)) and an unstructured scene II (Figure 5.10(d)). Figure 5.10 depicts the results of the basic fusion method $\mathcal{A}$ and demonstrates the 3D–3D fusion methods $\mathcal{B} - \mathcal{D}$ for II. Table 5.7 compares the quantitative results for the proposed fusion methods and indicates that a notably smaller number of points remain for the unstructured scenario (II) compared to a primarily structured scenario (I). The left camera on the IOSB.amp Q1 sensor setup was used as a reference camera with the camera matrix $\mathbf{K}$ and the registration result $\mathbf{T}_{s,\mathcal{L}}^{c}$. Selected examples for the proposed fusion methods are illustrated subsequently, and further examples are given in Section B.3.

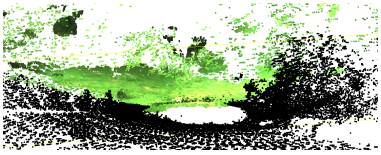
$\mathcal{A}$ with color from the RGB image facilitated an accurate assignment of color information in larger distances, as highlighted in Figure 5.9. The fusion of $\mathcal{L}$ and $\mathcal{S}$ without prior confidence assessment led to a high number of points in the point cloud. Here, most points were contributed from $\mathcal{S}$ and are thus subject to a limited depth resolution. Only 6.92 % of the 3D points within the partially structured scene I were LiDAR measurements. This highlights the need for a confidence-based fusion method to filter the integrated 3D points from $\mathcal{S}$ to keep the number of points sufficiently small for potential real-time processing in the subsequent mapping, planning, and control steps.

Method $\mathcal{C}$ only integrates points of the close range with a maximum depth inaccuracy ε_z, and $\max(\varepsilon_z) \in [0.04\,\text{m}, 0.10\,\text{m}, 0.25\,\text{m}, 0.50\,\text{m}]$ were evaluated to demonstrate the 3D–3D fusion with $\mathcal{C}$. Here, $\max(\varepsilon_z) =$

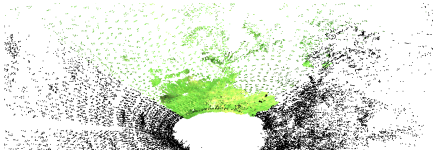
(a) Scene II.



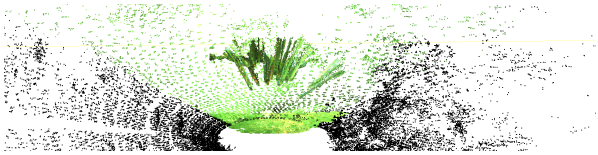
(b) $\mathcal{A}$ on I with $\mathcal{S}$ from SGBM with image color for $\mathcal{L}$.



(c) $\mathcal{B}^{(3)}$ for II.



(d) $\mathcal{B}^{(1)}$ and $\mathcal{C}^{(2)}$ for II.



(e) $\mathcal{D}$ with $\max(\varepsilon_z) = 0.25\,\text{m}$, scene II.

Figure 5.10 3D–3D fusion of $\mathcal{S}$ (SGBM) and $\mathcal{L}$ with $\mathcal{A} - \mathcal{D}$ for scene I (b) and II (a). Image (b) shows quantization and depth estimation inaccuracies for SGBM. 3D points without RGB color information are black. Images (c) and (d) demonstrate $\mathcal{B}$ where a higher N_{min} for $\mathcal{B}$ generally leaded to an exclusion of points from the far range of $\mathcal{S}$ and preserved more floor points than $\mathcal{C}$ and $\mathcal{D}$. kNN search for each point of $\mathcal{S}$ in $\mathcal{L}$ efficiently eliminated close-range disparity estimation errors for method $\mathcal{B}$.

0.04 m is similar to the measurement accuracy of the Velodyne HDL-64E with ±2 cm. However, for $\max(\varepsilon_z) = 0.04\,\text{m}$ ($z \leq 2.38\,\text{m}$) the minimum distance for an overlap for disparity estimation from stereo images hardly exceeded $z = 2.38\,\text{m}$. Realistic values for the close range in $\mathcal{C}$ result from $\max(\varepsilon_z) = 0.10\,\text{m}$ with $z \leq 3.76\,\text{m}$, $\max(\varepsilon_z) = 0.25\,\text{m}$ with $z \leq 5.96\,\text{m}$, and $\max(\varepsilon_z) = 0.50\,\text{m}$ with $z \leq 8.42\,\text{m}$.

The results for $\mathcal{B}$ in Table 5.7 show that kNN outlier filtering kept a higher ratio of points in the primarily structured scene I being more favorable for accurate stereo disparity estimation. For II, $\mathcal{B}$ mainly keeps ground floor points from $\mathcal{S}$, as depicted in Figure 5.10(b). Contrasting

Method	Parameterization	# in fusion cloud I	# in fusion cloud II
$\mathcal{B}^{(1)}$	$d_{\mathrm{NN}} = 0.10\,\mathrm{m}$, $N_{\mathrm{min}} = 10$	32.5 %	22.2 %
$\mathcal{B}^{(2)}$	$d_{\mathrm{NN}} = 0.50\,\mathrm{m}$, $N_{\mathrm{min}} = 5$	53.3 %	41.6 %
$\mathcal{B}^{(3)}$	$d_{\mathrm{NN}} = 0.50\,\mathrm{m}$, $N_{\mathrm{min}} = 10$	50.5 %	39.5 %
$\mathcal{C}^{(2)}$	$\max(\varepsilon_z) = 0.10\,\mathrm{m}$	24.5 %	43.5 %
$\mathcal{C}^{(3)}$	$\max(\varepsilon_z) = 0.25\,\mathrm{m}$	52.4 %	69.2 %
$\mathcal{C}^{(4)}$	$\max(\varepsilon_z) = 0.50\,\mathrm{m}$	72.5 %	77.7 %
$\mathcal{B}, \mathcal{C}$	$\mathcal{B}^{(1)}$, $\mathcal{C}^{(2)}$	**11.8 %**	**12.2 %**
$\mathcal{B}, \mathcal{C}$	$\mathcal{B}^{(1)}$, $\mathcal{C}^{(3)}$	25.0 %	22.0 %
$\mathcal{B}, \mathcal{C}$	$\mathcal{B}^{(1)}$, $\mathcal{C}^{(4)}$	28.4 %	24.2 %
$\mathcal{D}$	$\max(\varepsilon_z) = 0.10\,\mathrm{m}$	**10.0 %**	**31.7 %**
$\mathcal{D}$	$\max(\varepsilon_z) = 0.25\,\mathrm{m}$	34.5 %	54.4 %
$\mathcal{D}$	$\max(\varepsilon_z) = 0.50\,\mathrm{m}$	65.3 %	71.9 %
Primarily struct. scene I ($\overline{s}_I = 0.029$): 676,168 points in $\mathcal{L}$ and $\mathcal{S}$ with 6.9 % $\mathcal{L}$.			
Unstruct. scene II ($\overline{s}_{II} = 0.046$): 900,902 points in $\mathcal{L}$ and $\mathcal{S}$ with 5.6 % $\mathcal{L}$.			

Table 5.7 Comparison of the proposed 3D–3D fusion methods $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$. # indicates the ratio of $\mathcal{S}$ and $\mathcal{L}$ points that fulfill the accuracy requirements and were kept for the fusion cloud. Superscript indices denote particular parameterizations for $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$. $\mathcal{C}^{(1)}$ indicates $\max(\varepsilon_z) = 0.05\,\mathrm{m}$ and is neither recommended nor listed as a notable amount of 3D points still useful for dense workspace monitoring in combination with LiDAR measurements is eliminated.

$\mathcal{B}$, the results for $\mathcal{D}$ indicate that $\mathcal{D}$ naturally kept the close-range points from $\mathcal{S}$ and, hence, II contains a higher ratio of close-range points than I after filtering. The lowest ratio of $\mathcal{S}$ and $\mathcal{L}$ points for both scenes remained for $\mathcal{B}^{(1)}$ and $\mathcal{C}^{(2)}$.

Method $\mathcal{D}$ analyzes the confidence of each 3D point from $\mathcal{L}$ and $\mathcal{S}$. Here, the 2D confidence for scene I was determined to $c^{2\mathrm{D}}(\chi = 3.0) = 0.813$ and a disparity estimation from stereo images would only be conducted for a weak threshold. The 3D confidence of the stereo cloud consists of *PPC* and *PSC* elements, as explained in Section 4.1.4. The *PSC* elements $c^{3\mathrm{D}}_{\varepsilon_x}$ and $c^{3\mathrm{D}}_{\varepsilon_y}$ were determined as equal to 1.0 resulting in

$$c^{3\mathrm{D}}_{\mathrm{S},I} = 1.0 - 0.029 = 0.971, \tag{5.12}$$

$$c^{3\mathrm{D}}_{\mathrm{S},II} = 1.0 - 0.046 = 0.954. \tag{5.13}$$

Furthermore, $c_{\varepsilon_z}^{3D}$ with $\max(\varepsilon_z) = 0.04\,\text{m}$ according to Equation 4.6 yielded

$$c_{\varepsilon_z}^{3D} = 0.0, \qquad \text{for } z > 2.38\,\text{m}, \tag{5.14}$$

$$c_{\varepsilon_z}^{3D} = 1.0 - \frac{\varepsilon_z}{0.04\,\text{m}}, \qquad \text{for } z \leq 2.38\,\text{m}. \tag{5.15}$$

Hence, all 3D stereo points in I with z > 2.38 m had a *PPC* confidence of $c^{3D} = c^{2D} \cdot 0.743 = 0.604$ for $\max(\varepsilon_z) = 0.04\,\text{m}$ and are always excluded from the fused cloud as long as the LiDAR cloud is available. For a weak threshold, 3D stereo points require $c_{\varepsilon_z}^{3D} \geq 0.388$ to be included in the fusion cloud. This is equivalent to $\varepsilon \leq 0.612 \cdot \max(\varepsilon_z)$ implying

$$z \leq 1.86\,\text{m}, \quad \text{for } \max(\varepsilon_z) = 0.04\,\text{m}, \tag{5.16}$$

$$z \leq 2.95\,\text{m}, \quad \text{for } \max(\varepsilon_z) = 0.10\,\text{m}, \tag{5.17}$$

$$z \leq 4.66\,\text{m}, \quad \text{for } \max(\varepsilon_z) = 0.25\,\text{m}, \tag{5.18}$$

$$z \leq 6.59\,\text{m}, \quad \text{for } \max(\varepsilon_z) = 0.50\,\text{m}. \tag{5.19}$$

To conclude, 2D–3D fusion detailed in Section 3.7 generates 3D point clouds with camera-captured color information. Its advantage is that only one camera is required, and accurate 3D measurements of a LiDAR sensor can be combined with the color intensity values of a camera. Especially in real-time applications such as autonomous vehicles, requirements in terms of memory and processing time have to be met. Here, 2D–3D fusion provides a sparse, but colored 3D reconstruction of the environment with an approximately equivalent depth estimation accuracy for all 3D points.

3D–3D fusion requires a stereo or an RGB-D camera system within the multi-sensor system and can complement sparse 3D LiDAR measurements with dense but less accurate depth estimates. The colored 3D point cloud from 3D–3D fusion is notably larger and different measurement accuracies from stereo or RGB-D camera 3D reconstruction and LiDAR have to be considered additionally. The fusion methods $\mathcal{B}$ to $\mathcal{D}$ provide filtering options with different complexity levels if 3D measurements from LiDAR sensors and stereo or RGB-D camera 3D reconstruction are available. The suitability of a fusion method depends on its desired application with the trade-off between a higher density of the 3D reconstruction and the processing effort due to a higher number of points with different measurement accuracies requiring analysis in relation to the individual requirements and performance parameters of target application

and vehicle. For instance, 2D–3D fusion proved useful to integrated RGB color information in the 3D LiDAR cloud of the IOSB.amp Q1 platform that does not manipulate the environment and also has a smaller computational capacity as the IOSB.Alice platform. However, the environmental perception of IOSB.Alice can benefit from a confidence-based 3D–3D fusion of the sensor data from four LiDAR sensors and one stereo camera system mounted on the excavator's front: it can exploit notably greater computational power than most other off-road vehicles currently can, and the large amount of captured sensor data favors a filtering of both $\mathcal{L}$ and $\mathcal{S}$ 3D measurements, as proposed in method $\mathcal{D}$.

6 High-Level Perception

High-level perception interprets ordered and unordered 3D point clouds as "single-shot" 3D clouds and links perception and decision in the sensing–perception–decision pipeline for autonomous vehicles. Perception for autonomous off-road vehicles benefits from an accurate, semantic understanding of the scenery to enable autonomous systems to explore and especially to undertake manipulation tasks. Hence, determining one label for entire images and point clouds as well as object detection were not examined in this thesis.

This thesis focuses on the interpretation of geometric 3D point clouds without additional color information to provide generic high-level perception solutions. Here, semantic 3D segmentation requires a huge volume of training data with point-wise labeling information that comes with an immense effort if training data is not available and has to be generated first. The proposed semantic 3D segmentation interprets 3D point cloud data for an optimized navigability analysis, object detection, and obstacle avoidance in unstructured environments. Training and testing data from unstructured environments was not yet available in sufficient quantity at the time of writing this thesis and the semantic 3D segmentation methods were trained on data from mainly structured environments. In order to provide accurate and reliable semantic 3D segmentation for unstructured environments, Section 6.1 proposes a customized training approach for the semantic 3D segmentation with current state-of-the-art CNN architectures [326] as well as a domain transfer analysis that demonstrates how to optimize domain transfer with a special focus on unstructured environments.

The *IC-ACC* approach in Section 6.2.1 [326] constitutes a first step towards pre-modeling XAI with dataset assessment for 2D image and 3D point cloud data. It examines the data's information content (*IC*) and accuracy (*ACC*) as the input data exhibits the primary influence on the data-driven modeling of ANNs methods.

The X^{3}Seg approach in Section 6.2.2 examines semantic 3D segmentation results and facilitates a post-modeling explanation for semantically segmented 3D point clouds [330].

The presented semantic 3D segmentation methods and X^{3}Seg focus on 3D point clouds from rotating 3D LiDAR sensors. Nevertheless, the presented methods were designed for all types of 3D point clouds, e.g., from solid-state LiDAR sensors, 3D radar sensors, stereo camera systems, or ToF cameras, that provide a sufficient, measurement accuracy for segmentation.

6.1 Semantic Segmentation of 3D Point Clouds

Semantic 3D segmentation relies on CNNs as these mostly outperform classic methods (see Section 2.5.1). 3D point clouds are interpreted as "single-shot" scenes within this thesis, as moving objects are rarely encountered when decontaminating hazardous environments, while off-road transport for the material supply in defense applications does not require the capability of tracking dynamic objects for MULE[1] and Convoying[2].

Datasets available, such as SemanticKITTI [11], SemanticUSL [138], and Waymo [263], were captured using one sensor type. However, the domain transfer of CNNs trained on one source domain to other target domains, such as a sensor setup with multiple LiDAR sensors, is not trivial. Hence, this thesis investigates two strategies to cope with the limited data availability in unstructured environments: Section 6.1.2 proposes a customized training approach that can reduce the required amount of training data for a proper semantic segmentation performance, while Section 6.1.4 analyzes the domain transfer performance of semantic segmentation CNNs, whereby a promising domain transfer performance facilitates training in one domain, such as structured environments, and

[1] ELROB 2018: Transport–MULE: https://www.elrob.org/files/elrob2018/Transport_Mule_V3.pdf, access on 15.01.2022.

[2] ELROB 2018: Transport–Convoying: https://www.elrob.org/files/elrob2018/Transport_Convoy_V3.pdf, access on 15.01.2022.

the successful transfer to another domain, such as unstructured environments.

6.1.1 Semantic Segmentation Architectures

Generating training data in an unstructured target environments may not always be possible. For instance, training data for natural disasters, hazardous decontamination scenarios, or off-road transport in defense is scarcely available.

As a result, this thesis analyzes and optimizes state-of-the-art segmentation methods developed and trained for the semantic segmentation of 3D point clouds from structured environments for data from unstructured environments. It did not aim at the development of a new segmentation method from scratch, but at the deliberate evaluation of the segmentation performance for unstructured environments without any additional fine-tuning for the unstructured target domain to ensure the applicability of the analyzed methods in unknown scenarios to a certain extent.

A spherical projection of the sensor's 360° FoV onto a 2D range image achieved the fastest and most promising results in the study of [196]. In general, feature extraction on 2D range images facilitates the usage of fast 2D convolutions, while CNNs that perform the segmentation in 3D space still require a notably higher processing effort. Furthermore, Behley et al. [11] state that spherical projections, according to Equation 6.1, are beneficial for the segmentation of 3D LiDAR point clouds as the projection partially solves the sparsity problem of single scans. Thus, six CNN architectures performing the semantic segmentation on spherical 2D projections are analyzed hereinafter:

- SqueezeSeg (*Squ*) [298], SqueezeSeg-1024 (*Squ*-1024) [298],
- SqueezeSegV2 (*SquV2*) [299],
- DarkNet21Seg (*DN21*),
- DarkNet53Seg-512 (*DN53*-512), and DarkNet53Seg-2048 (*DN53*).

DN21, *DN53*-512, and *DN53* are RangeNet++ variants [65, 196, 225]. *DN53* is the most complex architecture with 53 layers, while *Squ* is the smallest and simplest architecture with 14 layers excluding CRF post-processing. Table 6.1 provides an overview of all analyzed segmentation architectures.

The analyzed architectures combine spherical projection and discretization into one step whereby the de-skewed 3D points clouds are directly converted into a range representation. Hereby, the range image representation implies discretization as the 3D data is mapped onto a grid structure similar to the pixel representation of 2D images.

LiDAR sensors transmit in the NIR spectral range and remission encodes the reflection characteristics of the objects or surfaces that trigger the measurement point $\mathbf{p}_i$. Hence, the 2D range image encodes 5D tensors with a receptive field size of $U \times V$ and $C = 5$ channels constituting the input into the segmentation architecture. Label predictions are conducted on the basis of a 5D tensor that encodes the geometric 3D information for each point in $\mathbf{p}_i = [x_i; y_i; z_i]^*$, the range $r_i = \left\|\mathbf{p}_i\right\|_2$, and the intensity I of the reflected laser beams in 2D range images. The spherical projection is calculated according to [196] and represented as a pixel grid with (u, v) tuples:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left(1 - \frac{\arctan(y,x)}{\pi}\right)U \\ \left(1 - \frac{\arcsin(\frac{z}{r}) + \tau_{\text{up}}}{\tau}\right)V \end{pmatrix}. \tag{6.1}$$

Here, U is the width and V the height of the desired range image in pixels, while u and v represent the image coordinates of the range image. The range image projection considers the LiDAR sensor's vertical FoV with $\tau = |\tau_{\text{up}}| + |\tau_{\text{down}}|$, e.g., with $\tau_{\text{up}} = 3°$ and $\tau_{\text{down}} = -25°$ for the Velodyne HDL-64E LiDAR sensor.

Contrasting [298] and [299], all CNN architectures interpret full 360° LiDAR point clouds from SemanticKITTI [196] captured with a Velodyne HDL-64E ($V = 64$). Different widths U were evaluated for the discretized grid representation, as summarized in Table 6.1.

The segmentation results are reprojected into 3D space after 2D segmentation on the range image. Here, the predicted label map output can be subject to the loss of low-level details due to the downsampling process in the first layers of the CNN. To cope with this, Wu et al. [298] apply a CRF on the label map output of the CNN, as proposed in [37], which is implemented as a recurrent neural network layer according to [313]. In contrast, Milioto et al. [196] propose kNN post-processing to overcome the problem of information loss as well as the reprojection

Architecture	# Param. in Mio.	V	U
SqueezeSeg-1024 (*Squ*-1024)	1	64	1024
SqueezeSeg (*Squ*)	1	64	2048
SqueezeSegV2 (*SquV2*)	1	64	2048
DarkNet21Seg (*DN21*)	25	64	2048
DarkNet53Seg-512 (*DN53*-512)	50	64	512
DarkNet53Seg (*DN53*)	50	64	2048

Table 6.1 Analyzed 3D segmentation methods [44, 196, 277, 298, 299].

uncertainty that occurs if two points lie on the same pixel grid point for spherical projection, but with different associated depths.

Semantic segmentation is the equivalent of multi-class classification and the cross-entropy loss is calculated from

$$L_{\text{CE}} = -\sum_{i=1}^{N} y_{O,i} \log P(O,i), \tag{6.2}$$

with N the total number of classes, $y_{O,i}$ the binary indicator for the correctness of a class label i for the respective observation O. $P(O,i)$ is the predicted probability that O is in class i. In order to evaluate the segmentation performance, intersection over union (IoU), also denoted as Jaccard Index, constitutes the state-of-the-art measure: $\overline{\text{IoU}}$ measures the IoU for all considered classes, while $\overline{\text{IoU}_i}$ is the per-class $\overline{\text{IoU}_i}$ for class i [196, 256]:

$$\overline{\text{IoU}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i}. \tag{6.3}$$

TP_i counts the number of true positives, FP_i the number of false positives, while FN_i denotes the number of false negatives for a class i and N is the total number of considered classes. As a second measure, the segmentation performance can be measured using the accuracy

$$\text{accuracy}_i = \frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i}. \tag{6.4}$$

The LiDAR-Bonnetal toolkit[3] was used for the training and testing of the semantic segmentation architectures within this thesis and all analyzed architectures were trained on data from the SemanticKITTI dataset [11]: 3D point clouds with point-wise labeling from one Velodyne HDL-64E LiDAR sensor.

6.1.2 Customized Training for 3D Segmentation CNNs

The presented customized training approach targets the increase in training performance to reduce the required volume of training data from unstructured environments. For this purpose, the *IC-ACC* method for pre-modeling explainability with dataset assessment discussed in Section 6.2.1 is applied to analyze the training data volume required from SemanticKITTI. A reduced dataset with seq. 02 to seq. 04 was utilized to evaluate the segmentation performance with a limited volume of training data and to yield a reference for the generation of future datasets for semantic 3D segmentation. As a next step, the proposed customized training approach can be utilized as a reference for a dedicated generation of training data from unstructured environments with reasonable effort, e.g., within the *GOOSE* dataset described in Section 7.5.

Here, the lower number of parameters in *Squ* compared to *DN53* facilitates a notably faster training. Thus, *Squ* with $U = 1024$ and $V = 64$ and a spherical projection for a horizontal FoV of 360° was selected to demonstrate the customized training methodology for a higher segmentation efficiency. Post-processing is not used to assess the segmentation performance independently and to inhibit a potential concealing of over-fitting. The SemanticKITTI point clouds are subdivided into structured (front, back) and unstructured (left, right) sectors to analyze the segmentation performance, as illustrated in Figure 6.2. The separation into four sectors yields $U = 256$ for each sector, and each sector contains a total number of $D_T = V \cdot U \cdot C = 256 \cdot 64 \cdot 5 = 81,920$ data points for $C = 5$ tensor elements.

Representative classes from the SemanticKITTI class structure were chosen and grouped into structured and unstructured classes: car, road,

[3] LiDAR-Bonnetal: https://github.com/PRBonn/lidar-bonnetal, access on 06.01.2022.

parking, pavement, building, fence, pole, and traffic sign belong to the group of structured classes, while the unstructured classes vegetation, terrain, and trunk belong to the nature category in SemanticKITTI. The training was split into different phases, and each phase was analyzed separately to consider its specific type and amount of input data. The first phase of the customized, iterative training process started with the data from one sector and trained for a predefined number of epochs. Subsequently, the segmentation architecture with pre-training on the first sector was trained with the data from the second 90° sector within a second training phase and so forth. Validation was conducted on a randomly reduced number of scenes from seq. 08 preserving the training to validation ration of 4:1. *IC-ACC* was applied to assess the information content (*IC*) and accuracy (*ACC*) of the point clouds, as described in Section 6.2.1.

A first study started with varying volumes of training data from the front (structured) and right (unstructured) sectors to analyze the development of training and validation performance for this domain change (see Table 6.2). An in-depth evaluation of over-fitting was conducted by gradually increasing the number of scenes (360° 3D point clouds) from $N_S = 350$ to $N_S = 2800$. This yielded the training and validation performances for $N_S \in \{350, 700, 1050, 1750, 2800\}$ presented in Table 6.2.

Generally, a high $\overline{\text{IoU}}$ in training and a low validation performance in terms of $\overline{\text{IoU}}$ indicate over-fitting. In consequence, a customized Δ_{TV} measure compares the training and validation performance in relation to the training data volume:

$$\Delta_{TV}(\overline{\text{IoU}}) = \frac{\overline{\text{IoU}}_V}{\overline{\text{IoU}}_T}. \tag{6.5}$$

Δ_{TV} for loss and accuracy is calculated accordingly. In order to evaluate the benefit of the proposed customized training approach, a customized metric η_{IoU} is proposed to measure training efficiency. It measures the $\Delta(\overline{\text{IoU}}_V)$ in relation to the amount of training data:

$$\eta_{\text{IoU}} = \frac{\Delta(\overline{\text{IoU}}_V)}{D_S \cdot N_S}. \tag{6.6}$$

The growth of the average validation IoU ($\overline{\text{IoU}}_V$) prior to and after the current iterative training step is hereby compared in $\Delta_{TV}(\overline{\text{IoU}})$. D_S calculates

the volume of data inside each scene using $D_S = U \cdot V \cdot C$ with $C = 5$ describing the number of features with $N = \{X, Y, Z, r, I\}$. The test of different values for N_S determined the minimum data volume required to prevent over-fitting. For this purpose, $N_S \in \{350, 700, 1050, 1750, 2800\}$ was evaluated and combined the front and right sectors into one common training dataset. The results shown in Table 6.2 indicate that the segmentation was not subject to over-fitting for training with $N_S = 2800$ scenes.

6.1.3 Proof of Concept: Customized Training for 3D Segmentation CNNs

The customized *IC-ACC* training approach is demonstrated on application examples for the perception of off-road vehicles from seq. 00 to seq. 10 of SemanticKITTI. Training with the full version of the SemanticKITTI dataset considers seq. 00 to 07, 09, and 10 and trains for 100 epochs, while the reduced version was trained on seq. 02 to 04 for 150 epochs and applied to evaluate the training efficiency using *IC-ACC*. The validation seq. 08 was excluded from training and utilized to measure the segmentation performance of *Squ* with the proposed customized training approach. The validation set for the reduced dataset was a randomly reduced number of scenes from seq. 08 preserving the training to validation ratio of 4:1. The mean surface variation $\bar{s}$ indicates the structured or unstructured character of the sectors and the *IC* of training data for unstructured environments. Combining all four sectors into one training dataset is identical to the training of *Squ* on the full dataset.

The following, experimentally justified training parameters were used to train *Squ* on SemanticKITTI: the learning rate was set to 0.001 with a decay of 0.995, the momentum of the stochastic gradient descent was set to 0.9, the weight decay to 0.0001, the batch size to 2, the class weighting was set to 0.001, and 12 kernel threads were utilized. Loss and validation were logged for each epoch to analyze the training process.

The nature classes vegetation, trunk, and terrain achieved $\overline{\text{IoU}} = 0.335$ on the reduced dataset, while the structured classes reached $\overline{\text{IoU}} = 0.266$ on the reduced dataset. This is remarkable as the nature classes are attributed to less than 30 % of the points present in the training data. It

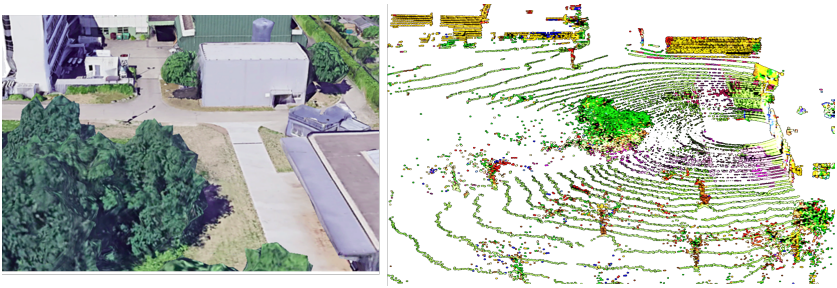


Figure 6.1 Semantic segmentation of Velodyne HDL-64E cloud from IOSB.amp Q1 with *Squ*; yellow: building, green: vegetation, brown: trunk, purple: road. Images © Fraunhofer IOSB.

	all	r	r, f	r, l	r, l, f	r, l, f, b
$\overline{\text{IoU}}$	0.284	0.135	0.230	0.180	0.250	0.281
$\bar{s}_T$	0.042	0.065	0.043	0.055	0.044	0.042
η_{IoU}	$3.1 \cdot 10^{-8}$	$6.0 \cdot 10^{-8}$	$5.0 \cdot 10^{-8}$	$3.0 \cdot 10^{-8}$	$3.7 \cdot 10^{-8}$	$3.0 \cdot 10^{-8}$

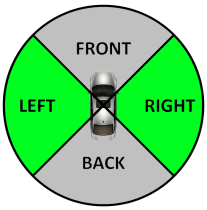
Table 6.2 Semantic segmentation performance of *Squ* on seq. 08 without post-processing, with $N_S = 2800$, and training for 150 epochs. The training efficiency η_{IoU} was measured for the training process proposed in sectors front (f), right (r), back (b), and left (l), and *IC* in terms of $\bar{s}_T$ was measured on the training data. The iterative training results on all four sectors highlight the impact of the separation into sectors.

can also indicate a higher IC making nature classes favorable in training and inference due to an increased unambiguousness. Figure 6.1 illustrates the segmentation results of *Squ* with the proposed customized training on SemanticKITTI for a 3D point cloud captured with the IOSB.amp Q1 platform.

Figure 6.2(a) shows that the over-fitting tendency decreased with a higher number of training samples as expected: Δ_{TV} increased for higher values of N_S which highlights the decreasing tendency for over-fitting. With $N_S \geq 1050$, the validation loss only doubled in comparison to the loss during training. For $N_S = 2800$, Δ_{TV} converged to one for $\overline{\text{IoU}}$

Data	Step	$\overline{\text{IoU}}$	Loss	Accuracy
$N_S = 350$	T	0.140	1.20	0.54
	V	0.119	2.79	0.45
	Δ_{TV}	0.85	0.43	0.83
$N_S = 700$	Δ_{TV}	0.880	0.40	0.91
$N_S = 1050$	Δ_{TV}	0.880	0.50	0.98
$N_S = 1750$	Δ_{TV}	0.880	0.51	0.82
$N_S = 2800$	T	0.172	1.22	0.62
	V	0.186	2.14	0.62
	Δ_{TV}	0.920	0.57	1.0

(a)



(b)

Figure 6.2 (a) Semantic segmentation performance (T: Training, V: validation) for a combined training on the front and right sectors of the reduced dataset (seq. 02–04) over 150 epochs; (b) Subdivision of 3D point clouds in SemanticKITTI: front and back are primarily structured, left and right are primarily unstructured with natural and grown structures.

and accuracy. This highlights that $\overline{\text{IoU}}$ and accuracy were approximately equal in training and validation. Hence, it can be assumed that the segmentation was not subject to over-fitting for training with $N_S = 2800$ scenes.

Table 6.2 illustrates the training efficiency η_{IoU} in the semantic segmentation for $N_S = 2800$. Combining all four – two primarily structured (f, b) and two primarily unstructured (l, r) – sectors provides a reference for the training efficiency with $N_S = 2800$. The η_{IoU} values in Table 6.2 were measured within the indicated sectors. The right sector achieved the highest training efficiency η_{IoU} and the *IC-ACC* analysis in Table 6.2 shows that the right sectors have the most unstructured character and hence the highest *IC*. This shows that a notably higher η_{IoU} can be achieved with a similar volume of training data if the training data has a different structure. For seq. 02–04 of SemanticKITTI, the $\overline{\text{IoU}}$ can be raised by more than 30 % by combining data with different surface variations compared to using the same volume of training data with a similar structure. Consequently, the composition of *IC*-efficient datasets can improve the performance of

ANN methods and reduce the volume of labeled training data required to achieve comparable results.

6.1.4 Domain Transfer

Research on CNNs for 3D semantic segmentation mostly focuses on structured environments due to the high research interest in autonomous driving on public roads. For instance, the analyzed CNNs were trained on 3D point clouds from the SemanticKITTI dataset [196] that includes data from unstructured environments as previously detailed but primarily contains data from structured environments. Thus, perception for unstructured environments can highly benefit from the transfer of CNNs that were trained on one specific source domain to other target domains. Typical domain changes in perception are the change of the application environment, the change of sensor mounting points or orientations in a current sensor setup, the mounting of additional LiDAR sensors, and the inclusion of new types of LiDAR sensors, as summarized in Table 6.3. Other domain changes include the transfer from simulation to the real-world or the transfer of perception solutions from a prototype or technology demonstrator into serial production [164, 294, 297].

Langer et al. [160] state that CNN models adapt to specific sensor parameters and characteristics of the environment, and that the transfer of trained models to another domain leads to a notable performance loss in semantic segmentation.

In order to cope with domain transfer loss, this thesis analyzes the measures for favorable domain transfers. Three options exist to achieve a satisfactory semantic segmentation performance in a new domain:

- Fine-tuning: record and label new training data, retraining,
- Domain adaption: synthetic adaption of the available training data to the new domain, retraining,
- Domain transfer: optimization of and preprocessing for pre-trained CNN architectures for a suitable generalization performance.

Domain changes may also occur unexpectedly for the critical applications discussed in this thesis. Consequently, this thesis focuses on domain transfer to avoid the generation of new training data and a synthetic adaption of existing training data both implying retraining the 3D segmentation CNNs in the case of domain changes. Here, the segmentation

performance of CNNs for different domain transfers is evaluated without any additional retraining in the new domain facilitating a well-founded analysis of their adaption to new, possibly unknown target domains.

A favorable domain transfer performance without retraining can be achieved by combining a CNN architecture with a proper generalization and a preferably high invariance between the input data of the source and target domain. This thesis proposes a novel combination of preprocessing techniques for 3D point clouds increasing both the domain invariance of the 3D point clouds themselves and the domain invariance of the spherical 2D projections subject to segmentation. To this end, the presented preprocessing methods aim at highly invariant spherical projections by synthetically approximating the source and target domain with preferably equivalent viewpoints, FoVs, sensor orientations, and sensor mounting positions.

The domain transfer performance of five CNN architectures is analyzed hereinafter: *Squ* [298], *SquV2* [299], *DN21*, *DN53*-512, and *DN53* [65, 196, 225] (see Table 6.1). The analyzed segmentation architectures were trained on sensor data from one specific source domain: one type of LiDAR (Velodyne HDL-64E) and one type of environment (SemanticKITTI) [11, 196]. The domain transfer performance was investigated for data from SemanticKITTI and SemanticUSL as well as for 3D point clouds from the IOSB.amp Q1 and IOSB.Alice platforms captured in mainly unstructured, off-road environments at the Fraunhofer IOSB in Karlsruhe. The domain transfer performance of the analyzed architectures was evaluated with the $\overline{\text{IoU}}$ and $\overline{\text{IoU}}_i$ and with and without post-processing, according to Equation 6.3.

Table 6.3 and Table 6.4 provide an overview of the domain transfers under analysis and discussion. The focus was placed on real-world data, and multiple domain specific variances were identified:

- Application environment: structured or unstructured character,
- FoV: individual sensors vs. sensor setup with multiple sensors,
- Viewpoint: different sensor poses or mounting points,
- Sensor orientation: different orientations in source and target domain,
- Sensor resolution, point density: different numbers of diodes,
- Noise characteristics: different sensor types.

Different experiments were conducted to analyze domain specific variances and their correlation to domain transfer performance. In order to achieve a preferably high domain invariance, different preprocessing steps are proposed in this thesis:

- Spherical projection (*SP*)
- Source alignment (*SA*),
- Shift to source (*StS*),
- and FoV adaption (*FoV*).

Preprocessing with *SP* is always conducted as it is inherent to all analyzed CNN architectures. *SP* reduces the domain gap for different types of individual sensors and sensor setups and is especially important if the target sensors are rotated relative to the source sensors as 3D ray paths are less dominant in the 2D grid representation of the spherical projection.

SA denotes the rotational alignment of the sensor origins for source and target domain. It virtually approximates their sensor orientation for both individual clouds and fused 3D point clouds from multiple LiDAR sensors. *SA* is mostly possible for autonomous off-road vehicles as they are equipped with highly accurate localization solutions, and two axes of the body frame are typically parallel to the ground plane. For instance, the body frame of the IOSB.Alice platform lies within the ground plane and in between the two chains. In the proof-of-concept discussed hereinafter, *SA* rotationally aligns the target clouds with the sensor orientation in the source domain, the ground plane in SemanticKITTI. Using the extrinsic calibration of the LiDAR sensors and their known orientation to the vehicle frame, *SA* can be conducted with minimal effort. Alternatively, the RANSAC algorithm can be applied to detect the ground plane, as explained in Section 4.2.3.

StS creates the translational alignment of the sensor origins for source and target domain. Similar to *SA*, it transforms the target sensor orientation to a virtual origin approximately equal to the source origin in terms of perspective and viewpoint for the subsequent spherical projection.

The grid size for the spherical projection is specified by the pre-trained network architecture and the projection of different sensor FoVs onto a grid of equivalent size leads to perspective distortions. To this end, the proposed *FoV* aims at the highest perspective similarity of the spherical projection despite different FoVs and requires an adaption of the vertical

ID	Domain transfer	Variance	Preproc.
I	Appl. environment: struct. to unstructured	$\bar{s}$, objects (types, separation), etc.	*SP*, class sel.
II	Sensor pose: different orientation, same LiDAR type	Viewpoint, ray paths in 3D	*SP*, *SA*, *StS*
III	Sensor type: type A to type B	FoV, noise, point density, remission	*SP*, *SA*, *StS*, *FoV*
IV	Sensor setup: single to multiple sensors	FoV, noise, point density, ray paths in 3D, viewpoint, refl. intensity	*SP*, *SA*, *StS*, *FoV*

Table 6.3 Overview of domain specific variances between source and target domain for the conducted domain transfer analysis in semantic 3D segmentation.

FoV with τ_{up} and τ_{down} according to Equation 6.1. Naturally, *FoV* is always required for domain transfers to other sensor types (III) and for sensor setups with multiple LiDAR sensors (IV).

Table 6.3 relates the proposed preprocessing methods to domain specific variances. Here, class selection describes the definition of a favorable class structure for unstructured environments. For instance, a class structure that only separates into drivable and non-drivable terrain can achieve a superior domain transfer performance for a navigability analysis as demonstrated below. However, exploration and manipulation of the environment require a more detailed class structure for a proper interpretation in high-level perception. The subsequent analysis demonstrates that the SemanticKITTI class structure or fine-grained class structures for unstructured environments, as proposed by Metzger et al. [193] and applied in Forkel et al. [71], provide notably more information with the downside of a less favorable domain transfer performance.

Unfortunately, suitable test data for a completely separate analysis of the domain transfers I–IV was not available. Hence, SemanticUSL, IOSB.amp Q1, the individual LiDAR sensors from IOSB.Alice, and the merged clouds from IOSB.Alice were selected, and different domain transfer scenarios were combined, as summarized in Table 6.4.

The transfer from SemanticKITTI to IOSB.Alice analyzes the domain transfer from a single LiDAR to a sensor setup with multiple LiDAR

ID	Data for Target Domain
I	SemanticUSL (Ouster OS1-64), IOSB.amp Q1 (Velodyne HDL-64E), individual IOSB.Alice clouds (OS0-64, OS0-128)
II	IOSB.amp Q1
III	SemanticUSL, individual IOSB.Alice clouds, incl. I and II
IV	Fused IOSB.Alice cloud (3× OS0-64, 1× OS0-128), incl. I–III

Table 6.4 Domain transfer analysis with source domain SemanticKITTI.

sensors of a different type. The sensor setup with multiple LiDARs on the IOSB.Alice platform provides one fused 3D point cloud from all four Ouster OS0 LiDAR sensors in the platform coordinate system. Thereby, the extrinsic sensor calibration as well as the calibration to the platform coordinate system allowed an accurate registration of the 3D point clouds from three OS0-64 and one OS0-128 LiDAR sensors with an FoV from −45° to 45°. SemanticUSL was captured with an Ouster OS1-64 LiDAR with an FoV of −22.5° to 22.5°. An outlier filtering for 3D points with potentially erroneous information was conducted for SemanticUSL, particularly in the origin of the coordinate system. The data for IOSB.amp Q1 and IOSB.Alice was labeled manually to evaluate the segmentation performance in the new target domains. The domain transfer was evaluated with special focus on the classes of the nature group mainly present in unstructured environments: vegetation, terrain, and trunk.

Remission information also contributes to the class label predictions, as detailed in Section 6.1.1. Consequently, the intensity values of the target domains have to match the range of the intensities in the source domain. SemanticKITTI contains normalized intensity values for all points but intensity normalization was not described in [11, 196]. Hence, two obvious methods for intensity normalization were also subject to evaluation on SemanticUSL: normalization can be conducted with a fixed maximum intensity value ($2^8 : I \in [0, \dots, 255]$, $2^9 : I \in [0, \dots, 511]$) for all clouds or with the maximum intensity of each individual cloud. Another alternative would be normalization with respect to the standard deviation but normalization for an arithmetic mean of zero is not feasible as I always needs to be positive.

Post-processing with CRF and kNN can increase the segmentation performance in some cases. CRF post-processing was only evaluated on *Squ* and *SquV2*, as proposed by Wu et al. [298, 299]. Milioto et al. [196] recommend kNN post-processing for *DN21* and *DN53* resulting in kNN post-processing being evaluated for *DN21*-512 to *DN53*.

Typically, the road is located in the middle of the spherical projection in the structured environments captured in SemanticKITTI, which suggests an additional over-fitting analysis for the examined, pre-trained network architectures: over-fitting on SemanticKITTI was assessed via a rotation of the clouds around the vertical axis of the coordinate system, which is equivalent to a shift in the direction of the width U of the spherical projection. Rotations of 90°and 180° were evaluated exemplarily to assess if the network architectures were influenced by the "typical" positions of the road and vegetation.

An alternative, contrasting approach to the domain transfer optimizations proposed is the identification of domain specific objects. However, this requires an in-depth analysis of each target domain and an a priori object detection. This may not always be feasible in unstructured application environments such as in the decontamination of landfill sites, and potentially encountered objects may not be clearly identifiable or separable. Furthermore, workspace monitoring in unstructured environments requires the point-wise understanding of the perceived 3D point clouds including the ground plane, which is why a high-level detection of individual objects is not sufficient. Instead of customized preprocessing for CNNs to generate preferably domain-invariant 2D range images for segmentation, Burkhardt et al. [28] and Schulz-Mirbach [248, 249] discuss 2D features invariant for certain transformations. However, the feature extractions in CNN-based semantic segmentation are modeled during the training step, and the requirements for invariant features according to [249] cannot be ensured. Another alternative to the proposed preprocessing would be the utilization of coordinate independent 3D CNNs for the semantic segmentation of 3D point clouds [42, 290]. However, similar to the point-wise segmentation in 3D space, the processing effort for coordinate independent CNNs is currently too high to achieve real-time capability in the environmental perception for off-road vehicles.

Nevertheless, it constitutes an interesting idea in combination with the increase in computation power within the next few years.

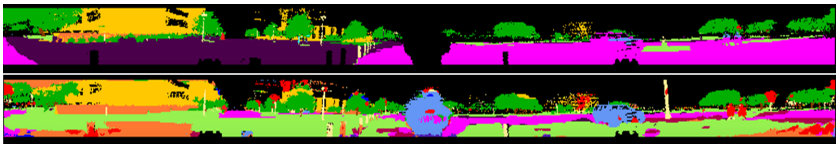
6.1.5 Proof of Concept: Domain Transfer

The experimental evaluation discussed hereinafter was mainly conducted within the master's thesis of Schmidl [339]. It has been published and presented[4] within the 1st Workshop on Scene Understanding in Unstructured Environments[5]. Table 6.3 and Table 6.4 summarize the analyzed domain transfer scenarios.

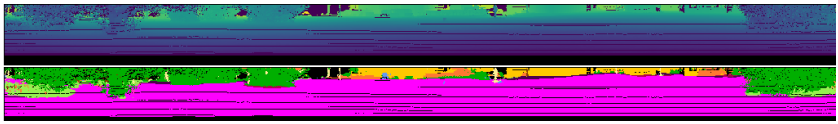
Intensity Normalization. The normalization with a fixed number of $2^9 = 512$ intensity values ($I \in [0, \dots, 511]$) for all clouds yielded $\overline{\text{IoU}} = 11.2\,\%$ for all classes with *DN53* on SemanticUSL (see Table 6.6). The normalization with the maximum intensity $\max(I)$ of each individual cloud performed better and yielded $\overline{\text{IoU}} = 12.0\,\%$. However, this requires the determination of $\max(I)$ for each cloud prior to segmentation, and intensity normalization for SemanticUSL was conducted with $\max(I) = 511$ for the presented domain transfer analysis to limit the required processing steps. Intensity normalization for the Ouster OS0 sensors on IOSB.Alice was conducted with $\max(I) = 255$ directly during data capture.

Complexity of CNN Architectures. Table 6.6 and Table 6.7 compare the domain transfer performance of the five discussed architectures from SemanticKITTI to SemanticUSL (I and III) and to IOSB.Alice (IV) on selected results that highlight the findings discussed hereinafter. The performance of *DN21*, *DN53*-512, and *DN53* decreased less than the performance of *Squ* and *SquV2*. As a result, the CNN architectures with the highest number of layers and parameters (*DN53*-512, *DN53*) showed the best domain transfer performance in I–IV. This leads to the assumption that CNN architectures with a higher number of parameters, such as *DN53*, yield better results in the analyzed domain transfers. This could be due to the fact that the higher number of parameters captures more correlations between points reducing the sensitivity to changes in the FoV and the perceived environment. In contrast to this, smaller, rather simple

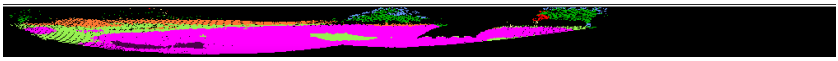
[4] https://www.youtube.com/watch?v=7aypd1QIqgw, access on 23.01.2022.
[5] https://unstructured-scene-understanding.com/program.html, access on 23.01.2022



(a) SemanticUSL: ground truth (above), predictions (below), FoV∈[−22.5°,22.5°].



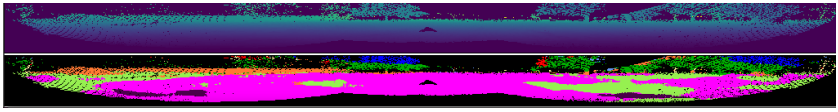
(b) IOSB.amp Q1: proj. measurements (above), predictions (below), FoV∈[−25°,3°].



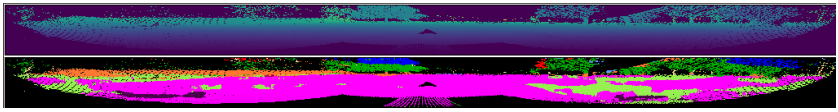
(c) IOSB.Alice: point-wise class predictions left LiDAR, FoV∈[−60°,25°].

(d) IOSB.Alice: class predictions rear LiDAR, FoV∈[−60°,25°].

(e) IOSB.Alice: class predictions boom LiDAR, FoV∈[−60°,25°].



(f) IOSB.Alice: left and right LiDAR sensors, FoV∈[−60°,25°].



(g) IOSB.Alice: left, right, and boom LiDAR sensors, FoV∈[−60°,25°].

Figure 6.3 Domain transfer results with *DN53* on 2D range images: *StS* was applied for IOSB.Alice ((c)–(g)) with -3.0 m. Class affiliation is indicated by coloring: pink: road, light green: terrain, dark green: vegetation, purple: pavement, brown: trunk, blue: car, orange: fence, yellow: building. Selected 3D point clouds are depicted in Figure C.1 for clarity. Images © Fraunhofer IOSB.

network architectures such as MC-CNN provide a favorable domain transfer performance from structured to unstructured environments, as discussed in Section 5.1.2. However, semantic 3D segmentation has a notably higher complexity compared to local disparity estimation from stereo camera images on small image patches. An evident assumption is that the apparently high number of layers and parameters in *DN53* is still low enough to impede the exact mapping of the characteristics of each domain. This can explain the favorable generalization and hence domain transfer performance of *DN53*. The future development of even deeper CNN architectures with notably more parameters will show if this assumption holds true.

FoV, StS. *FoV* and *StS* show a notable influence on the segmentation performance on SemanticUSL and the fused IOSB.Alice cloud from all sensors, as illustrated in Figure 6.4. *FoV* can achieve similar viewpoints for SemanticUSL, as depicted in Figure 6.4(b). Figure 6.3(a) depicts the segmentation results for Semantic USL (I, III) without *StS* and with an $\overline{\text{IoU}}$ of 11.2 %. The Clearpath Warthog robot used to record SemanticUSL is smaller than the VW Passat for SemanticKITTI. For SemanticUSL clouds, a proper segmentation could not be achieved without *FoV*, and an *StS* towards the source sensor origin (1.0 m higher above ground) notably increased the $\overline{\text{IoU}}$ of *DN53* to 14.6 %. Furthermore, the robot operator walked behind the Clearpath Warthog during the whole capture of SemanticUSL and was not eliminated from the data. Here, *StS* does not only generate a more similar viewpoint of the source and target domain, it also eliminated the operator not properly labeled within the ground truth. Hence, both changes caused the measured $\overline{\text{IoU}}$ increase.

The fused cloud from the four LiDAR sensors on IOSB.Alice (IV) presents the greatest FoV difference in the evaluated domain transfers. Consequently, the invariance increase that can be achieved with *FoV* preprocessing was evaluated on IV with two promising and experimentally justified FoV variants and with an identical *StS* of -3.0 m, as indicated in Table 6.5. The vertical FoV of the source domain (Velodyne HDL-64E) is –25° to 3° but experimental evaluation showed that a larger vertical FoV for IV achieved qualitatively and quantitatively more accurate segmentation results. Hence, an FoV from –50° to –5° was selected in accordance with the FoV of the Ouster OS1-64 in SemanticUSL and a second, notably

FOV_-	FOV_+	*StS*	$\overline{\text{IoU}}$ all	$\overline{\text{IoU}}$ veget.	$\overline{\text{IoU}}$ trunk	$\overline{\text{IoU}}$ terrain
–60°	25°	-3.0 m	3.5	43.6	0.1	22.0
–50°	–5°	-3.0 m	2.1	8.0	0.0	31.3

Table 6.5 Segmentation performance for domain transfer of *DN53* on fused 3D point clouds from IOSB.Alice (IV) with *SA* and *StS* (-3.0 m) preprocessing and without post-processing. The $\overline{\text{IoU}}$ measure for all classes (all) and selected classes is given in %.

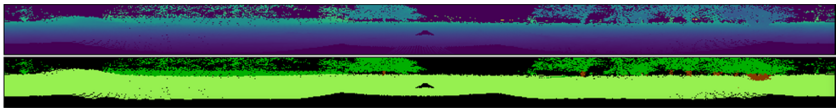
larger FoV from –60° to 25° was evaluated in comparison. Here, the larger vertical FoV∈[–60°,25°] (Figure 6.4(d)) increased the $\overline{\text{IoU}}$ by more than 70 % in contrast to an FoV∈[–50°,–5°] (Figure 6.4(c)). Furthermore, an FoV∈[–60°,25°] and *StS* with -3.0 m yielded better results for vegetation, as shown in Table 6.5.

Range Image Resolution. *DN53*-512 with a smaller range image resolution performed slightly better than *DN53* on SemanticUSL but *DN53* outperformed *DN53*-512 on IOSB.Alice data. To conclude, the domain transfer results in Table 6.6 and Table 6.7 indicate that a higher resolution of the spherical projection is beneficial for fused clouds from multiple LiDAR sensors.

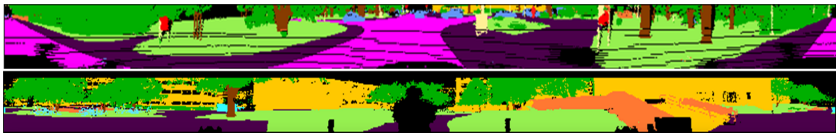
IOSB.Alice (IV). Table 6.7 shows selected $\overline{\text{IoU}}$ results for the conducted domain transfer experiments on the IOSB.Alice platform. It compares the individual segmentation of each LiDAR sensor in the sensor setup with the segmentation of multiple LiDAR clouds within one fused 3D cloud. Especially the segmentation performance of the boom LiDAR is notable as the difference between the $\overline{\text{IoU}}$ of *DN53* and *DN53*-512 becomes most evident. The rays of the boom LiDAR mounted below the excavator's boom have completely different paths in comparison to the source domain (see Figure 6.3(e)), and areas with high and with low point density especially exist in the 3D cloud of the boom LiDAR. As a result, it is assumed that the highly dense geometric information of the OS0-128 boom LiDAR is well-represented with a range image of $U = 2048$ and badly represented with $U = 512$, which leads to the assumption that a high range image resolution is especially beneficial if areas with different point density exist inside one cloud. *SP* was identified as the most beneficial preprocessing here as it eliminates the point accumulations

Architecture	$\overline{\text{IoU}}$	car	road	build.	veget.	trunk	terrain	pole
Segmentation performance in source domain K.								
Squ-kNN	31.9	76.1	85.9	68.8	71.6	26.8	66.0	28.2
SquV2-kNN	41.8	86.7	90.1	79.6	79.2	36.5	71.1	28.3
DN21	47.2	84.2	93.4	79.0	81.7	48.8	71.6	39.3
DN53-512-kNN	39.9	85.3	91.0	75.1	77.8	41.1	69.6	38.7
DN53-kNN	52.8	91.0	93.8	85.8	84.2	52.9	72.7	53.2
Relative performance loss in transfer from K to U (I–III).								
Squ	**86.9**	87.6	91.7	96.5	96.9	69.2	96.8	70.0
SquV2	85.4	73.7	87.3	94.6	95.1	75.4	90.5	80.2
DN21	81.4	61.8	79.1	97.6	59.4	68.9	89.8	76.6
DN53-512	68.0	45.8	67.4	91.6	42.1	41.9	77.8	63.9
DN53	**77.7**	46.6	83.8	94.7	**46.3**	**65.3**	**75.9**	73.2
Segmentation performance in target domain IOSB.amp Q1 (I, II).								
Squ-kNN	**2.2**	–	5.0	5.1	6.0	6.0	0.5	–
SquV2-kNN	6.3	–	5.2	58.8	41.6	12.7	1.3	–
DN21-kNN	8.5	–	5.2	74.9	76.1	2.0	2.8	–
DN53-512-kNN	6.9	–	5.4	52.3	72.2	0.0	0.6	–
DN53-kNN	**8.2**	–	**5.0**	**68.3**	**78.4**	0.6	**3.6**	–

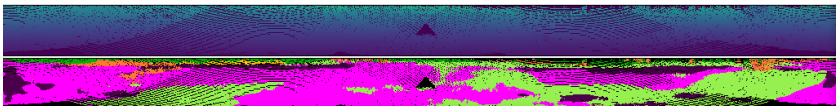
Table 6.6 Domain transfer evaluation for 3D segmentation CNNs with average and per class $\overline{\text{IoU}}$ in %: transfer from SemanticKITTI (K, seq. 8, source) to SemanticUSL (U, I–III), and IOSB.amp Q1 (I,II). *SP* and intensity normalization with $\max(I) = 511$ were applied, *StS* was not conducted.



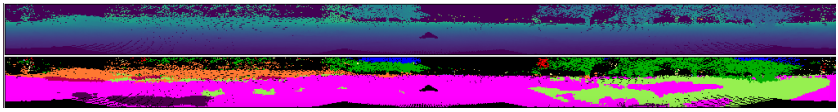
(a) IOSB.Alice: Range image (above), ground truth (below).



(b) Similar viewpoint with *FoV*: source **K** (above), target **U** (below).



(c) IOSB.Alice: FoV ∈ [−50°,−5°], $\overline{\text{IoU}}$ = 2.1 %.



(d) IOSB.Alice: FoV ∈ [−60°, 25°], $\overline{\text{IoU}}$ = 3.5 %.

Figure 6.4 2D range images for fused 3D cloud from IOSB.Alice with -3.0 m *StS*: image (a) shows the 2D range image and ground truth labeling with ∈ [−60°, 25°], while (b) shows exemplary range images of SemanticKITTI (**K**) and SemanticUSL (**U**) after *FoV* to [−22.5°, 22.5°]. The images (c) and (d) compare *FoV* and highlight typical misclassifications for navigable ground. Images © Fraunhofer IOSB.

on the ray paths, and particularly increased the domain invariance for rotated LiDAR sensors such as the examined boom sensor. Figure 6.4(a) shows range image and ground truth labeling of the IOSB.Alice cloud.

Naturally, the segmentation of the fused 3D cloud is faster than the segmentation of each individual LiDAR cloud. Most class predictions on the fused 3D cloud from the left and right LiDAR sensor on IOSB.Alice achieved a higher $\overline{\text{IoU}}$ for *DN53* than an individual segmentation of each cloud. However, the individual segmentation with *DN53*-512-kNN yielded higher $\overline{\text{IoU}}$ results than segmentation of the fused 3D clouds with kNN. It is hence assumed that the individual segmentation of the

LiDAR clouds yields more accurate segmentation results in the case of lower range image resolutions. Concluding, the segmentation of fused 3D clouds instead of the individual segmentation of each clouds is beneficial if the range image resolution is sufficiently high.

Class Occurrence, Misclassification, and Class Selection. The domain transfer loss was smaller for classes that were frequently present in the training data (see Table C.1 for ground truth label distributions in SemanticUSL). The domain transfer performance of *DN53*-kNN with $\overline{\text{IoU}} = 11.2\,\%$ on SemanticUSL and $\overline{\text{IoU}} = 3.6\,\%$ on the examined IOSB.Alice cloud is not yet sufficient for an accurate and reliable interpretation of the 3D measurement points. Table 6.6 further shows that for I and II low $\overline{\text{IoU}}$ values for terrain and road were achieved compared to high $\overline{\text{IoU}}$ achievements of building and vegetation. An in-depth analysis of the per class $\overline{\text{IoU}}$ showed that the analyzed architectures tend to cause misclassifications of navigable terrain (road, pavement, other-ground, terrain) in domain transfer, even if sensor type and sensor orientation in source and target domain are equivalent. The segmentation performance also decreased for other classes such as vegetation but notably less: $\overline{\text{IoU}} = 47.0\,\%$ were achieved for vegetation class predictions in domain transfer case IV on IOSB.Alice in comparison to $\overline{\text{IoU}} = 84.2\,\%$ on SemanticKITTI. However, this shows that a domain transfer without retraining is feasible with suitable preprocessing and low $\overline{\text{IoU}}$ for navigable ground: for instance, the low $\overline{\text{IoU}}$ for terrain in IV was mostly caused by misclassification as another class that also constitutes navigable ground for autonomous off-road vehicles, such as road and pavement.

Figure 6.3 highlights these class interchanges on data from IOSB.Alice, IOSB.amp Q1, and SemanticUSL: terrain and road were confused in all segmentation results depicted in Figure 6.3, while pavement and road were erroneously classified as terrain in Figure 6.3(a) and Figure 6.3(b), and fence and vegetation were confused in Figure 6.3(c)–(g). Furthermore, in the case of the unstructured off-road environment around the Fraunhofer IOSB shown in Figure 6.3(b), the entire ground consisting of terrain and road was predicted as road, while class estimates for vegetation on the left side and for building were correct. It becomes clear that a suitable class selection and the suitable combination of all navigable ground classes into one class can notably optimize the quantitative semantic

segmentation performance in unstructured environments, particularly for domain transfers from structured environments.

To conclude, classes whose geometry is clearly distinguishable, such as building or vegetation, show a target domain $\overline{\text{IoU}}$ close to their source domain $\overline{\text{IoU}}$. However, geometrically similar classes, such as terrain, road, pavement, and other-ground, were often interchanged but they all present navigable ground structures for off-road vehicles. Here, the X^3Seg approach presented in Section 6.2.2 detects notable similarities between fence and vegetation in SemanticKITTI and also finds that the fence class often contains vegetation elements in the SemanticKITTI source domain, which naturally favors the probability of their interchangeability in semantic segmentation. This shows that the integration of post-modeling XAI methods in the domain transfer can further help to understand the segmentation results on a target domain in the future.

Post-Processing. Table 6.6 compares CRF and kNN post-processing. Both CRF and kNN post-processing increased the $\overline{\text{IoU}}$ on the source domain SemanticKITTI for all analyzed architectures. However, CRF post-processing for *Squ* and *SquV2* impaired the segmentation of SemanticUSL data and benefited the segmentation of IOSB.Alice data. Post-processing with kNN slightly increased the $\overline{\text{IoU}}$ of *SquV2*, *DN21*, and *DN53* in most classes of SemanticUSL (I,III), in the IOSB.Alice clouds from individual OS0 LiDAR sensors, and also in the fused 3D cloud from IOSB.Alice. Summarizing, kNN post-processing slightly increased the $\overline{\text{IoU}}$ in the target domain for the analyzed architectures and is consequently recommended for the analyzed domain transfers.

Over-Fitting. The $\overline{\text{IoU}}$ of *DN53*-kNN decreased from 11.2 % to 10.9 % on SemanticUSL for a rotation of 90°and 180°, which can indicate a slight over-fitting on SemanticKITTI. *Squ* and *SquV2* also showed this tendency for over-fitting but *DN21* was only slightly influenced by the rotation of the point clouds. *DN53*-512 did not show any impaired performance for a rotation of the point clouds. In addition, the fence class is rather present on the side of the range images in SemanticKITTI, as discussed above, and it is assumed that this contributed to the confusion between fence and pavement in SemanticUSL, as shown on the left side of Figure 6.3(a).

Favorable Domain Transfer with Preprocessing. To conclude, favorable domain transfers to different sensor types, to sensor setups with

Architecture	LiDAR sensors	$\overline{\text{IoU}}$	vegetation	terrain
SquV2-kNN[1]	all	1.1	7.4	7.1
DN21	all	0.5	2.5	7.4
DN53-512-kNN	all	1.9	27.2	7.8
DN53	all	3.5	43.6	22.0
DN53-kNN	all	3.6	**47.0**	22.0
DN21	boom	0.6	5.3	5.4
DN21	rear	1.9	11.3	24.4
DN53-512-kNN	left	1.9	28.9	6.6
DN53-512-kNN	boom	**0.1**	2.2	0.1
DN53-512-kNN	rear	4.1	51.4	23.8
DN53-kNN	left	2.8	27.2	25.1
DN53-kNN	right	3.0	39.8	17.1
DN53-kNN	boom	**4.7**	30.3	59.8
DN53-kNN	rear	4.7	56.5	32.3
DN53-512-kNN	left, right	1.4	22.6	4.9
DN53-kNN	left, right	**3.3**	38.7	23.5
DN53-512-kNN	left, right, boom	1.1	17.7	2.7
DN53-kNN	left, right, boom	**3.3**	39.1	22.8

[1] *Squ* and *Squ*-kNN achieved $\overline{\text{IoU}}_{\text{all}} = 0.0$, *Squ*-CRF achieved $\overline{\text{IoU}}_{\text{all}} = 0.2$

Table 6.7 Selected segmentation performance results in terms of $\overline{\text{IoU}}$ in % on IOSB.Alice LiDAR clouds (IV) with *SA* and *StS* (-3.0 m) preprocessing and $\max(I) = 2^8 = 255$, FoV $\in[-60°,25°]$.

multiple sensors, and to other application environments can be achieved by a high domain invariance. At first, this requires a similar viewpoint (*FoV*) and perspective (*SP*) for the spherical projections subject to segmentation in state-of-the-art semantic segmentation CNNs for 3D point clouds. It was demonstrated that domain transfer from an individual LiDAR sensor to fused 3D clouds from multiple LiDAR sensors can benefit from a high resolution of the range image, such as in *DN53*. The experimental evaluation conducted also showed that a favorable domain transfer is more probable for classes frequently present in the training dataset if they have sufficiently different geometric characteristics. To the best of the author's knowledge, *SA*, *StS*, *FoV*, and *SP* constitute a novel combination of preprocessing methods optimizing the domain transfer

for semantic 3D segmentation CNNs. Their proposed combination increased the domain invariance, and also showed that the utilization of CNN architectures with a higher number of parameters, such as *DN53*, contributes to a better segmentation performance in the target domain.

6.2 Explainable Artificial Intelligence

The explainability of the behavior and decisions of ML systems in XAI targets their transparency by examining the what and why of the effectiveness and success of ML systems for a given task [161]. Analogies to psychology are obvious due to the similarity of ANNs to natural neural networks, discussed in Section 2.1, and two main tendencies can be identified: the desired tendency is that the examined ML system has learned a valid strategy that generalizes well; the second tendency is that the ML system bases its decisions "on a spurious correlation in the training data" [161, p.2] which is designated as a Clever Hans behavior in psychology [217]. Concluding, the XAI approaches proposed hereinafter verify that ML methods have learned a valid strategy and help to understand and explain the performance of ML methods, which ensures their capability to deliver accurate and stable predictions.

In 2019, the AI HLEG determined ethics guidelines for trustworthy AI systems including three fundamental characteristics: AI systems have to be lawful, technically and socially robust, and comply with ethical principles and values[6]. These characteristics lead to the European Commission's key requirements for the development of trustworthy AI systems: "Human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity, non-discrimination and fairness, environmental and societal well-being, and accountability"[7]. Here, transparency also includes the capability to explain and retrace the decisions made by an ML system. Due to the black box nature of ML methods, a deterministic behavior inherent to classic, non-ML methods cannot be guaranteed intrinsically. A definite solution for the problem of

[6] Ethics Guidelines for Trustworthy AI: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, pp. 4, access on 24.04.2022.

[7] Ethics Guidelines for Trustworthy AI, pp.4.

transparency in autonomous systems, and in particular in autonomous ML systems, is neither functional, nor properly established, incidents such as the accident with the Tesla autopilot prove this[8]. As a result, the proposed XAI approaches in this thesis also facilitate a first step towards transparent ML methods in the perception for autonomous off-road vehicles.

Perception for autonomous off-road vehicles, deals with a large volume of information inside each data sample. Both 2D images and 3D point clouds require large receptive fields for neural network processing, and the number of single data elements can easily exceed 2,000,000 for 2D images (1080×1920 px) and 50,000 for 3D point clouds (3D points in Velodyne HDL-64E cloud). ANNs interpreting the perceived data such as in semantic segmentation also require an encoder-decoder structure to consider local and global characteristics. Hence, a development of inherently explainable models is rather impossible here due to the large number of network layers and parameters. Consequently, this thesis focuses on pre-modeling and post-modeling XAI approaches.

6.2.1 *IC-ACC*: Pre-Modeling XAI with Dataset Assessment

The input data exhibits the primary influence on the data-driven modeling of ANN methods. Hence, proper ANN training requires a sufficient volume and diversity of the information inside the data as well as high accuracy of the reference data for the supervised training of ML methods.

The research on data analysis for ANN methods is greatly underrepresented in relation to the extensive research on ANN methods themselves, and the in-depth examination of training, validation, and test data conducted hereinafter showed that state-of-the-art datasets do not always provide an error-free ground truth.

In order to contribute to the closure of this gap in perception, *IC-ACC* proposes a generalized, step-by-step exploratory data analysis facilitating a better insight into the dataset in the pre-modeling stage. This benefits the development of powerful and trustworthy ML and hence AI systems as detrimental data can be eliminated before the ANN performance

[8] Tesla Germany GmbH: An Update on Last Week's Accident, https://www.tesla.com/de_DE/blog/update-last-week%E2%80%99s-accident, access on 06.11.2021.

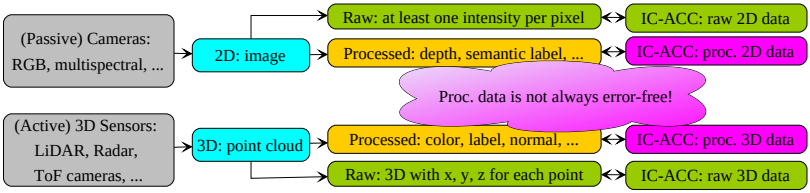


Figure 6.5 Data classification in image processing: raw data is assumed error-free and only *IC* is measured, for processed data *IC* and *ACC* are assessed.

is negatively influenced. Furthermore, Section 6.1.2 demonstrates that *IC-ACC*-customized data to train, validate, and test ML methods can contribute to a more efficient training of ANNs.

IC-ACC aims at answering a frequently posed question in ANN research on the definition of good data and validates training, validation, and test data in a loosely coupled manner as detrimental data can be eliminated before a negative influence on the performance of classic or ML methods occurs. *IC-ACC* further allows the deduction of hypotheses on the cause and reason of the observed *IC* and *ACC* characteristics and proposes a guideline to generate efficient and accurate data for specific, data-driven ANN methods as the availability of training, validation, and test data is often limited in unstructured environments.

Contrasting most works on exploratory data analysis detailed in Section 2.5.3, *IC-ACC* focuses on a generalized assessment of training data for image processing ANN methods that permits an in-depth assessment of data characteristics highly relevant for a strong and reliable performance of the developed ANN. Only classic methods are utilized to assess the data quality in *IC-ACC* as an analysis with ML methods would require an additional assessment due to their inherent black box nature. Naturally, *IC-ACC* is also applicable for the training and evaluation data of classic methods.

The assessed data in *IC-ACC* is classified depending on the target application of the ANN. The 2D imaging domain can be subdivided into segmentation, depth estimation, object detection and tracking, and classification, while 3D imaging can be separated into segmentation, object detection and tracking, shape classification, and registration ap-

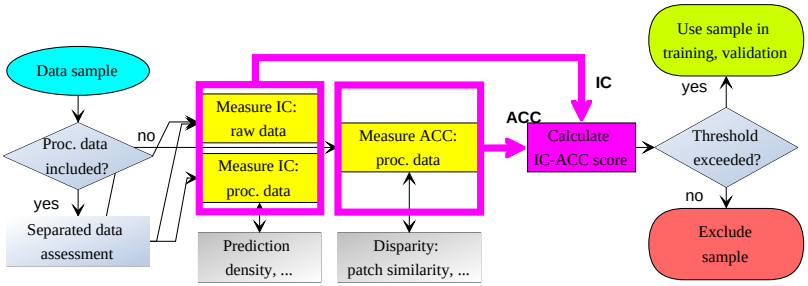


Figure 6.6 *IC-ACC*: separated assessment of raw and processed data, the *IC-ACC* score decides whether to include the sample in the final dataset.

proaches [11, 99]. The workflow of the proposed *IC-ACC* method is illustrated in Figure 6.6, and Table 6.8 summarizes the proposed *IC* and *ACC* measures. 2D and 3D data divides into raw and processed data, as illustrated in Figure 6.5: raw data is the output of a sensor system after the application of the intrinsic calibration corrections and is assumed to be error-free in *IC-ACC*, while processed data designates the reference data that is obtained in processing the raw data. Raw 2D data consists of single images or image patches, while 3D point clouds constitute raw 3D data. Supervised training requires processed data as a reference to define, calculate, and optimize the loss during training, while unsupervised training only needs raw data. Thus, *IC-ACC* analyzes the processed data *ACC* as data processing can be subject to errors.

6.2.1.1 Information Content (*IC*) of Raw and Processed Data

Quality Measures for Raw 2D Data. Some images or image patches are detrimental in the training process of ANNs instead of helpful, as described in Section 5.1.2 and [327]. Disparity estimation from stereo images with CNNs for instance benefits from training image patches with high, non-repetitive texture information. Here, *IC* quality measures can assess the potential of a single image patch and help to improve the matching performance as detrimental patches can be identified and excluded prior to training. The *IC* of raw 2D image data can be measured with different quality measures (see also Section 3.5): Shannon entropy,

DoG, HOG, as well as SIFT, SURF, and FAST descriptors. However, a high *IC* is not necessarily related to a high amount of detectable edges which is why DOG and HoG are less suitable to assess the quality of images. Furthermore, features and descriptors do not provide a global description of the image or image patch as they only describe the rather local image area included in the respective descriptor.

Shannon entropy as the most basic measure provides a pure indication of the texture of an image patch and proved to be a very strong measure for the *IC* of 2D images [326]. Consequently, *IC-ACC* measures the *IC* of raw 2D image data with the Shannon entropy $\boldsymbol{H}$ according to the Equation 3.10. Here, a high $\boldsymbol{H}$ indicates a high number of different intensity values and thus a high *IC*, while a low $\boldsymbol{H}$ indicates a high similarity of the pixel intensities. Naturally, a high similarity of the pixel intensities indicates a low probability that the inclusion of this patch improves the performance of the stereo disparity estimation network. Subsequently, the quality measures for single image patches can also be utilized to additionally compare the similarity of two image patches similar to the X^3Seg approach discussed in Section 6.2.2.

Quality Measures for Raw 3D Data. The point density and geometric structure measures were identified as the most conclusive criteria for the *IC* of raw 3D data (IC_{r3D}) in [327]. The point density in homogenized coordinates (ξ) presents the most promising measure for the density of active 3D measurements. A uniform point distribution ξ in cylindrical coordinates according to Equation 3.15 illustrates a proper representation of all cloud sectors inside a point cloud [329]. As a result, ξ, $\overline{\xi}$, and $\sigma^2(\xi)$ can be applied to compare different samples, as described in Section 3.6 and according to Equation 3.12. The geometric structure of a point cloud can be described with its surface variation s, as detailed in Section 3.6, and the future application environment defines if a high or a low *IC* measure in terms of surface variation was achieved. A high $\overline{s}$ in more complex, unstructured environments indicates a high *IC*, while a high *IC* in structured environments is synonymous to a clear structure and thus a low $\overline{s}$ (see Section 6.2.1.5).

Information Content *(IC)* of Processed Data. The *IC* of processed data depends on the prediction density and diversity of the information supplemented to the raw data during the processing step. Prediction

density for 2D data (IC_{p2D}) is related to the number of pixels, and an example for 2D prediction density is provided in stereo image disparity estimation: a high prediction density indicates a high percentage of valid depth estimates and thus a decent quality of the reference data [327]. For 3D data, the number of points inside a point cloud (IC_{p3D}) is used accordingly. The diversity of the information is measured with the Shannon entropy according to Equation 3.10.

6.2.1.2 Accuracy (*ACC*) of Processed Data

ACC estimates the confidence and error characteristics for available, processed reference data with indirect measures. As a consequence, the *ACC* estimate in *IC-ACC* can determine the suitability of the reference data as ground truth for the supervised training and validation of ANN methods. This can overcome the common lack of a verified, error-free ground truth to compare against. The confidence assessment detailed in Section 4.1 analyzes the accuracy of raw sensor data. Consequently, an *ACC* analysis of raw sensor data is not included in *IC-ACC*.

Contrasting the presented *IC* measures, the evaluation of *ACC* has to be adapted to the type of the processed information to some extent, and two groups can be distinguished: data used to train an ANN for similarity matching (ACC_{2Ds}, ACC_{3Ds}) and data for interpretation (ACC_{2Di}, ACC_{3Di}). Here, similarity matching includes stereo image disparity estimation that registers 2D image patches to derive disparity values in 2D (ACC_{2Ds}) as well as the registration of 3D point clouds (ACC_{3Ds}). Segmentation, object detection and tracking, and classification aim at the interpretation of imaging data and thus belong to ACC_{2Di} and ACC_{3Di}.

ACC_{2Ds} and ACC_{3Ds} examine the similarity of source and target to be matched. For instance, the similarity of 2D samples for stereo camera image disparity estimation is measured with SSIM and the NRMSE, as proposed in [326]. In 3D–3D registration, the correct transformations that perfectly align each source–target pair constitute the reference data subject to *ACC* analysis: the target, for instance the 3D point cloud of one LiDAR sensor, remains in its original representation and thus also in its sensor coordinate system, while the source, a 3D point cloud of a second LiDAR sensor, is transformed by applying the reference data. A high

IC-ACC element	Measure
IC_{r2D}: raw 2D	Shannon entropy H
IC_{r3D}: raw 3D	Surface variation $\bar{s}$, relative density μ
IC_{p2D}: processed 2D	Prediction density (pixels), diversity via H
IC_{p3D}: processed 3D	Prediction density (points), diversity via H
ACC_{2Ds}: similarity	NRMSE, SSIM
ACC_{2Di}: interpretation	Qualitative visual assessment, label smoothness
ACC_{3Ds}: similarity	L_2 norm
ACC_{3Di}: interpretation	Qualitative visual assessment, label smoothness

Table 6.8 Exploratory data analysis in *IC-ACC* with proposed measures.

similarity of both aligned clouds indicates a high *ACC* and difference measures such as L_1 norm, L_2 norm, or NRMSE are applicable.

Similarity Assessment for Stereo Image Disparity Estimation. The training of an ANN for stereo image disparity estimation in supervised manner requires preferably equivalent image patches from corresponding left–right image pairs as raw data as well as processed reference data in the form of accurate reference disparities. The disparity information is encoded in the grayscale intensity values and defines the horizontal shift inside the same pixel row of the respective, rectified images. Erroneous information can be contained inside the reference data even if the disparity value was extracted from very accurate LiDAR measurements, as demonstrated hereinafter for KITTI 2012 [83]. Different similarity measures are proposed in this thesis to detect the erroneous assignment of image patches prior to their utilization in CNN training for stereo image disparity estimation:

- Manhattan metric (L_1 norm) according to Equation 6.7,
- L_2 norm according to Equation 6.7,
- NRMSE,
- SSIM,
- Cross-correlation.

The proposed measures identify non-similar image patches by applying the respective similarity measures at each valid disparity location. The

L_p norm can be applied to measure the distance of two intensity vectors $\mathbf{i}_n$ and $\mathbf{i}_m$ of two $N \times N$ patches m and n:

$$L_p(m,n) = \|(\mathbf{i}_m[j,k] - \mathbf{i}_n[j,k])\|_p = \sqrt[p]{\sum_{j=1}^{N}\sum_{k=1}^{N} |\mathbf{i}_m[j,k] - \mathbf{i}_n[j,k]|^p}, \quad (6.7)$$

with $p \in [1,2]$ for L_1 and L_2 norm. L_1 evaluates the differences in the pixel intensity values linearly, while the L_2 norm evaluates the pixel-wise difference of the intensity values inside the patches quadratically. NRMSE measures intensity differences with

$$\text{NRMSE}(m,n) = \frac{L_2(m,n)}{\sqrt{\sum_{j=1}^{N}\sum_{k=1}^{N} \mathbf{i}_m[j,k]}}. \quad (6.8)$$

The normalization of the RMSE in NRMSE yields an exposure-invariant assessment, and NRMSE proved useful if the focus is laid on relative instead of absolute differences.

The calculation of the cross-correlation metric requires the representation of the intensities in m and n as $N \times N$ matrices. Shifting is not required, and the cross-correlation $R_{m,n}(0)$ is calculated for the overlay in contrast to 2D image registration techniques:

$$R_{m,n}(0) = \sum_{j=1}^{N}\sum_{k=1}^{N} (\mathbf{i}_m[j,k] \cdot \mathbf{i}_n[j,k]). \quad (6.9)$$

The mean SSIM can be utilized to compare the similarity of the patches m and n, as described in [288]. SSIM assesses the structural details inside the image patches, and an implementation is provided in the Scikit-Image library[9].

Experimental evaluation showed that the cross-correlation metric is sensitive to noise and different exposures which is often detrimental to assess the similarity of two image patches. As the resilience to noise and different exposure characteristics is crucial in stereo matching, cross-correlation is not applicable here. An alternative to the cross-correlation

[9] SciKit-Image library: https://scikit-image.org/docs/dev/api/skimage.measure.html, access on 04.11.2021.

metric would be the usage of the cross-correlation coefficient which is invariant to noise and exposure. However, its evaluation was not conducted within this thesis as NRMSE yielded satisfactory results for similarity measurement, as discussed subsequently.

NRMSE contains information similar to the L_1 and L_2 norms but in a normalized and thus exposure-invariant manner. It also considers the translational errors induced by erroneous reference rotations in relation to the scale, and small translational errors close to the origin of the point clouds are considered with the same magnitude as a large translational error in large distance. Therefore, NRMSE is selected as the first similarity measure. Structural details form the image texture, which is the most important matching criterion for image patches in disparity estimation, and SSIM is selected as a second similarity measure.

In order to measure the *ACC* for processed, reference data, the disparity values are applied onto the patch pairs. To conclude, the similarity of the patch is compared using a combination of SSIM and NRMSE of the pixel intensity values.

In addition, it is possible to assess the *IC* for the difference measure between the 2D or 3D input data samples after applying the processed reference data. For instance, if the intensity values of two identical image patches are pixel-wise subtracted, this yields a white or black patch with $H = 0.0$.

Similar to the *IC* measure of raw 2D data, strong, medium, and weak thresholds, as described in Table 6.9, can also be established in the similarity assessment for disparity estimation. The weak, medium, and strong filtering criteria are chosen to approximately preserve 95 %, 90 %, or 75 % of the grayscale image patches from the KITTI 2012 training set.

2D Reference Data Assessment: Interpretation. The processed data for ANN methods interpreting 2D or 3D data (ACC_{p2Di}, ACC_{p3Di}) consists of labeling information. Labels can be present in different levels: image- or cloud-wise labels in classification, labels for groups of pixels or points in 2D or 3D bounding boxes for object detection and tracking, and pixel- or point-wise labels in semantic segmentation. One obvious strategy of determining ACC_{2Di} and ACC_{3Di} is to check a small number of random samples manually and deduce a qualitative statement. This is a time-consuming, but often a straightforward strategy for experts, and provides

Metric	Strong Filter	Medium Filter	Weak Filter
NRMSE	$< 0.75 \implies 90.3\,\%$	$< 0.9 \implies 94.5\,\%$	$< 1.1 \implies 97.7\,\%$
SSIM	$> 0.3 \implies 89.2\,\%$	$> 0.2 \implies 96.0\,\%$	$> 0.1 \implies 98.6\,\%$
Entropy H	$> 3.8 \implies 90.2\,\%$	$> 3.0 \implies 96.7\,\%$	$> 2.5 \implies 98.5\,\%$
Combined	75.4 %	89.5 %	95.5 %

Table 6.9 Application of strong, medium, and weak dataset filtering thresholds on grayscale image patches from the KITTI 2012 training set. Resulting percentages of valid patch pairs after the application of the respective weak, medium, and strong filters demonstrate their filtering performance.

a suitable and appropriate option to derive *ACC* measures, especially for classification with one label per 2D image or 3D cloud. Typically, human annotators assign labels with the assistance of labeling tools, and errors tend to occur particularly in border regions or transitions between objects. Furthermore, objects are rarely represented by a small number of points or pixels, and a high number of different labels in a small area or space can indicate noisy and inaccurate labeling data. As a first step towards a verifiable and quantitative *ACC* measure, pixel- and point-wise labels can be examined for smoothness, and thus for the existence of outliers with a kNN search, as explained in Section 5.1.1. A qualitative visual assessment of the label smoothness is also possible with a scoring from 0 to 10 where 10 indicates the highest *ACC*.

6.2.1.3 Deriving the *IC-ACC* Score

The *IC-ACC* score facilitates the choice whether to include or not to include a data sample into the dataset for training, validation, and testing. Naturally, an *IC-ACC* analysis of unsupervised learning approaches only performs an *IC* analysis on the raw data. Table 6.8 provides an overview of all *IC-ACC* elements. Each *IC* and *ACC* measure is normalized to $[0, 1]$ individually with the maximum value of the respective measure, as demonstrated in Section 6.2.1.5, and the *IC-ACC* score is calculated with

$$IC\text{-}ACC = 1/3 \cdot (IC_{\mathrm{rtD}} + IC_{\mathrm{ptD}} + ACC_{\mathrm{tD}}), \quad t \in \{2, 3\}. \tag{6.10}$$

If more than one measure is included in an *IC-ACC* element, the empirical mean of both measures is considered.

This thesis proposes the utilization of weak, medium, and strong threshold levels according to Section 3.12: if a data sample achieves more than 68.27 % of the possible maximum *IC-ACC* score of 1.0, it is included in the dataset for training, validation, and testing if a weak threshold is applied. The medium threshold is set to 0.8664 ($\mu \pm 1.5\sigma$), while the strong threshold is equivalent to 0.9545 ($\mu \pm 2\sigma$). These threshold levels categorize the analyzed samples on the basis of their *IC-ACC* scores and facilitate the adjustment of the required *IC* and *ACC* characteristics of data samples included in the dataset. Detrimental samples within one dataset can be detected and eliminated via the *IC-ACC* score for each data sample, while a comparison of different datasets is possible via the calculation of the *IC-ACC* score for all elements of each respective dataset. Another possibility is to calculate the *IC-ACC* score for all elements of all available datasets and to determine the normalization parameters with regard to elements of each respective dataset. Subsequently, the same threshold level (weak, medium, or strong) is applied to each dataset subject to comparison. Finally, the remaining good data samples of all datasets can be compared with each other.

6.2.1.4 Proof of Concept: *IC-ACC* for 2D Data in Disparity Estimation

A high Shannon entropy for 2D images indicates a high *IC* for 2D raw data samples, while a low NRMSE and a high SSIM demonstrate a high similarity of two image patches and thus a high *ACC* for processed 2D data. *IC-ACC* is demonstrated on training data for the *UEM-CNN* architecture discussed in Section 5.1.2. The data samples for disparity estimation consist of patch pairs formed by an evaluated patch and its associated patch captured by the other camera. The Shannon entropy H is calculated for each image patch individually, and the NRMSE and SSIM measures are calculated for pairs of image patches with the respective reference disparity values. NRMSE values close to zero indicate a good match, a high SSIM (SSIM $\in [-1, 1]$) indicates a high structural similarity and, consequently, well-matching patches. The patch pairs for training, validation, and testing were filtered with a combination of the selected NRMSE, SSIM, and Shannon entropy, as specified in Table 6.9, and each

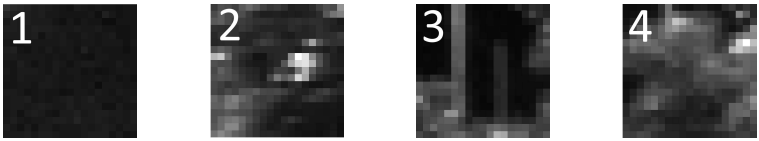


Figure 6.7 IC_{r2D} and ACC_{2Ds} for 19×19 pixel patches to train a CNN for disparity estimation from stereo images: 1 has a low *IC* with $H_1 = 2.5$, whereas $H_2 = 6.24$ and $H_3 = 5.75$ indicate a high *IC*; for pair 3-4 SSIM = -0.04 is sufficient, but the NRMSE = 1.15 is too high and does not meet the similarity requirements wherefore the reference disparity is rated as inaccurate. Images © Fraunhofer IOSB.

image patch had to exceed all three required thresholds to be included in the dataset for *UEM-CNN*.

Table 6.9 demonstrates the filtering results for detrimental patches with strong, medium, and weak thresholds on the KITTI 2012 grayscale training images. The proposed threshold levels facilitate the specification of a certain quality and accuracy requirements for the analyzed dataset with $\boldsymbol{IC}_{\mathrm{r2D}}$ and $\boldsymbol{ACC}_{\mathrm{2Ds}}$. Detrimental patches and poorly matching patch pairs can be eliminated, as illustrated in Figure 6.7 for the KITTI 2012 training dataset: one patch with a low *IC* due to a low Shannon entropy and two patches with a high *IC*.

The prediction density measure for processed 2D data (IC_{p2D}) can be illustrated on the comparison of disparity maps from SGBM, as described in Section 3.3, and *UEM-CNN*. SGBM achieved a prediction density of around 97 % but with a high percentage of erroneously predicted values. In contrast to this, *UEM-CNN* achieved a notably higher accuracy in disparity estimation with a lower prediction density of 63 %. To conclude, the proposed *IC-ACC* analysis proved useful for filtering detrimental image patches and improved the training, validation, and testing accuracy of *UEM-CNN* with the KITTI 2012 training dataset.

6.2.1.5 Proof of Concept: *IC-ACC* for 3D Point Cloud Data

The *IC* analysis for raw 3D data is demonstrated on seq. 00–10 of the SemanticKITTI dataset [11]. A high *IC* for 3D data is indicated by a high density in combination with a high surface variation $\overline{s}$ for unstructured

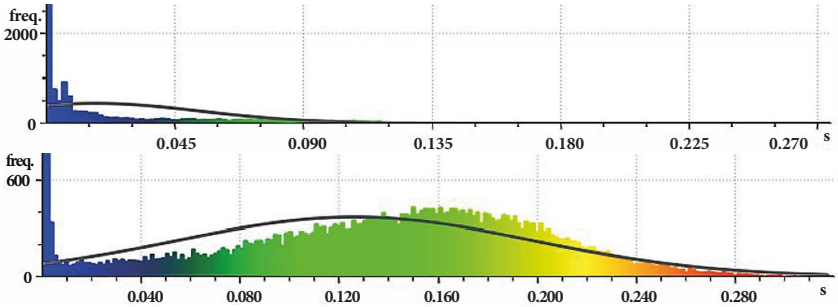


Figure 6.8 IC_{r3D}: histogram of surface variation s in the front (above) and right sector (below) of scene 245, seq. 04 [326]. A radius of 0.40 m proved useful for normal estimation [323]. The frequency of bin 0 was clipped for a clearer visualization (13,515 (front), 3783 (right)). The distribution of s was approximated with a single Gaussian density function as the focus is on a holistic distribution of s. The low $\overline{s} = 0.018$ of the front sector highlights its structured character, while the high $\overline{s} = 0.128$ of the right sector shows its unstructured character. The accumulation of low s measures, especially in structured environments, clearly diminishes $\overline{s}$.

environments, while a high density in combination with low $\overline{s}$ indicates a high *IC* for structured environments.

A subdivision into sectors benefits the *IC* analysis if clearly separable sectors can be identified inside the point clouds. As SemanticKITTI was captured with a vehicle platform, four sectors of 90° can be identified according to the sensor position on the vehicles, as depicted in Figure 6.2. The sectors are axisymmetric to the axes of the LiDAR coordinate system: front (f), right (r), back (b), and left (l). The surface variation $\overline{s}$ of the left and right sectors is notably higher than $\overline{s}$ of front and back for seq. 02–04: $\overline{s}_{l,02-04} = 0.046$, $\overline{s}_{r,02-04} = 0.065$, $\overline{s}_{f,02-04} = 0.021$, and $\overline{s}_{b,02-04} = 0.023$, while $\overline{s}_{all,02-04} = 0.040$ was measured for reference.

This justifies the separation into structured and unstructured sectors on the example of SemanticKITTI clouds. The $\overline{s}$ measures of seq. 03 exemplary highlight the predominating structured and unstructured characters of the subdivided sectors: $\overline{s}_{all,03} = 0.041$, $\overline{s}_{f,03} = 0.031$, $\overline{s}_{b,03} = 0.023$, $\overline{s}_{l,03} = 0.051$, and $\overline{s}_{r,03} = 0.051$. Only 9.77 % of the labels in seq. 06 represent classes from unstructured environments compared to 23.91 %

$\overline{s}$	Vegetation (V)	Trunk (T)	Unstruct. (V,T)	Terrain
$\overline{s}_{01} = 0.051$	23.87	0.04	23.91	13.83
$\overline{s}_{06} = 0.027$	9.31	0.46	9.77	26.10
$\overline{s}_{09} = 0.051$	29.29	0.67	29.96	8.88

Table 6.10 Relative pointwise class distributions for the lowest (seq. 06) and highest (seq. 01, 09) $\overline{s}$ values measured in seq. 0-10 of SemanticKITTI (in %). Only two of the 28 classes in SemanticKITTI predominantly represent unstructured elements: vegetation and trunk. Terrain in urban and suburban areas mostly includes cultivated and rather structured terrain.

and 29.96 % in seq. 01 and 09. The higher $\overline{s}$ measurements for seq. 01 and 09 given in Table 6.10 additionally justify the selection of $\overline{s}$ to indicate the structured or unstructured character of a point cloud. In addition, a higher $\overline{s}$ proved to be a suitable measure for a higher *IC* of training and testing data for ML methods in unstructured environments. The left and right sectors of SemanticKITTI show a higher *IC* for unstructured environments, as depicted in Figure 6.8, which shows the estimates of s for all individual 3D points in scene 245 from seq. 04. Table 6.10 shows the lowest and highest measured $\overline{s}$ with the respective class distributions for the nature group in SemanticKITTI.

An exemplary *IC-ACC* assessment is demonstrated for the comparison of two 3D clouds and summarized in Table 6.11: scene 245 of seq. 04 (245,04) with a medium $\overline{s}$ and scene 778 of seq. 09 (779,09) with a high $\overline{s}$. The point density for IC_{r3D} was calculated with $N = 12$ bins which maps 30° in one bin and yielded $\mu_{245,04} = 0.083$ and $\mu_{778,09} = 0.083$. ACC_{3Di} is demonstrated in qualitative manner as the renowned SemanticKITTI dataset comes with a high labeling accuracy and label smoothness that could both be verified manually. $N_I = 28$ was set for H with 28 classes in SemanticKITTI to measure IC_{p3D}. A label is provided for each point which yielded a prediction density of 100 %.

Classes that are not present have a likelihood of occurrence of $p(i) = 0$ for IC_{p3D} and the IC_{p3D} for all scenes in seq. 04 with 34,059,667 points was determined as $H_{04} = -\left(\sum_{28} p(i) \cdot \log_2(p(i))\right) = 2.406$. The IC_{p3D} for 245, 04 was approximately equal to H_{04} as it contains fewer classes but the labels are more uniformly distributed. Seq. 01 was recorded on a

	Raw measures		Normalization	
Measure	245, 04	778, 09	245, 04	778, 09
IC_{r3D}: $\overline{s}$	0.027	0.047	$\frac{0.027}{0.047} = 0.574$	$\frac{0.047}{0.047} = 1.0$
IC_{r3D}: μ	0.083	0.083	$\frac{0.083}{0.083} = 1.0$	$\frac{0.083}{0.083} = 1.0$
IC_{p3D}: H	2.405[*1]	2.903[*2]	$\frac{2.405}{2.903} = 0.828$	$\frac{2.903}{2.903} = 1.0$
IC_{p3D}: pred. density	100 %	100 %	$\frac{100\,\%}{100\,\%} = 1.0$	$\frac{100\,\%}{100\,\%} = 1.0$
ACC_{3Di}: qual.	10, 10	10, 10	$\frac{10}{10} = 1.0$	$\frac{10}{10} = 1.0$
IC-ACC score	–	–	0.90	1.0

[*1] Seq. 04: 19 classes; most frequent: road (33.79 %), veget. (32.78 %).

[*1] Sc. 245, seq. 04: 14 classes; most frequent: road (35.46 %), veget. (20.36 %).

[*2] Seq. 09: most see Table 6.10. Sc. 778, seq. 09: most freq.: veget. (26.89 %),

[*2] building (17.87 %), road (17.61 %).

Table 6.11 *IC-ACC* assessment of scene 245, seq. 04 and scene 778, seq. 09.

motorway where motorway borders as well as the central strip mainly consist of vegetation. Here, $H_{01} = 2.267$ was measured for seq. 01 with the most frequent classes road (40.51 %), vegetation (23.87 %), and terrain (13.83 %). The H of seq. 01 is smaller but the high $\overline{s}$ of this collection of point clouds proved useful if a CNN shall be trained for the semantic segmentation in unstructured environments.

Deriving the *IC-ACC* score. All 3D measures given in Table 6.8 were summarized to generate the 3D *IC-ACC* score according to Equation 6.10. The normalization references were derived from the maximum of the compared sequences, such as for the surface variation s with $\max(\overline{s}) = \overline{s}_{778,09} = 0.047$. Prediction density and ACC_{p3D} are identical for both scenes as both belong to the same dataset. This yielded 0.0266/0.0465 = 0.572 for $\overline{s}_{245,04}$ and 0.083/0.083 = 1 for the density in IC_{r3D} and a diversity measure of 2.4048/2.9031 = 0.828 was derived for IC_{p3D}. Identical comparisons were performed for scene 778 of seq. 09 as demonstrated in Table 6.11. Finally, the *IC-ACC* score for (245,04) is determined to

$$\textit{IC-ACC}_{245,04} = 1/3 \cdot \big((0.572 + 1.0)/2 + (0.828 + 1)/2 + 1.0\big) = 0.90, \quad (6.11)$$

while *IC-ACC* score for (779,09) equals to $IC\text{-}ACC_{778,09} = 1.0$. Both samples exceed the requirement with 86.64 % < 90.0 % if a weak or medium threshold is applied. However, only (778,09) would be included in the final dataset for a strong threshold.

6.2.1.6 Guidelines for Data Generation

Naturally, guidelines for future data generation can be derived from the proposed *IC-ACC* method. It is recommended to ensure that the captured data achieves a high *IC* and a high *ACC* as this fulfills the central requirement to generate good training, validation, and testing data: it does contain neither too similar, nor too little, nor erroneous information. Here, the targeted application environment of the ANN method defines the desired surface variation for 3D data as previously stated. If 3D data for applications in unstructured environments, such as from off-road vehicles, shall be captured, a high $\overline{s}$ is recommended, while indoor scenes can benefit from a low $\overline{s}$ measure. Furthermore, *ACC* measures on test samples can verify a high *ACC* for the full dataset.

6.2.2 X^{3}Seg: Post-Modeling, Model-Agnostic XAI for 3D Semantic Segmentation

The explanation of class predictions for 2D pixels and 3D points is subject to research in post-modeling XAI methods with heat- or class-activation maps [236, 252, 282]. X^{3}Seg contributes to a straightforward and model-agnostic explanation (X) of point-wise class predictions in 3D (3) semantic segmentation (Seg) [330]. It complements model-specific methods with model-agnostic explanations to understand class predictions and contributes to more trustworthy AI systems.

X^{3}Seg comprises three different methods: encompassing X^{3}Seg, selective X^{3}Seg, and predictive X^{3}Seg. Each of these methods focuses on a holistic explanation of class predictions and also regards topology and spatial arrangement of coherent 3D point sets: spatial relations with neighboring points are identified and related to the ground truth data from the training dataset (encompassing X^{3}Seg, selective X^{3}Seg) and to other predictions (predictive X^{3}Seg). Inspection of the most similar (prototypes) and least similar (criticism) coherent point sets leads to an

understanding of the segmentation results as it highlights the most relevant features of 3D point sets. Thereby, coherent point sets are spatially connected sets of the same class that form a coherent 3D structure and are thus very likely to belong to the same object or area. The interpretation of 3D point clouds is examined without color information in a first step to facilitate the interpretation of raw sensor data and with this the parallel processing of low-, mid-, and high-level perception methods without mutual dependencies. Each of the three methods in X^{3}Seg consists of two major steps:

- Generate database for prototypes and criticism (sample database),
- Similarity measurement of the explanation target to the respective sample database elements.

Furthermore, the in-depth assessment of this generated sample database examines the class distributions in the training and testing data, the suitability of the class definitions, and also the quality of the labeling process itself. In addition, the identification of prototypes and criticism from the database elements of the predicted class also validates class predictions: the probability for a correct class estimation is high if well-matching prototypes are present in the sample database, it is low if only ill-fitting prototypes are determined or if prototypes and criticism are too similar for a reliable distinction.

In summary, X^{3}Seg identifies prototypes and criticism for the 3D point set whose segmentation result is selected for explanation by a human operator (explanation target) and provides an understanding of the analyzed class predictions via their similarity to the explanation target. Consequently, the affiliation of (sample) database elements to prototypes or criticisms depends on their similarity to the 3D structure subject to explanation. The central question of similarity is addressed in-depth hereinafter with different similarity measures for selective X^{3}Seg in 2D and encompassing and predictive X^{3}Seg in 3D.

The X^{3}Seg approach is demonstrated on class predictions for the SemanticKITTI dataset [83] with a state-of-the-art semantic 3D segmentation method. However, X^{3}Seg is nevertheless applicable for arbitrary data-driven [94, 221, 267] and traditional approaches [3]. X^{3}Seg particularly focuses on off-road scenarios such as decontamination in hazardous environments to address the challenge of a reliable environment percep-

tion for autonomous heavy construction machinery [216]. As a result, the SemanticKITTI sequences with the highest amount of unstructured elements according to Section 6.2.1 are selected to demonstrate X^{3}Seg on class predictions of DarkNet53Seg-2048-kNN (*DN53*-kNN, see Table 6.13). Structured object classes, such as car and road, achieve a higher $\overline{\text{IoU}}$ than unstructured classes, such as trunk or vegetation, and smaller objects, such as pole or trunk. In order to determine the reasons for these differing performances, the focus of X^{3}Seg was primarily placed on these naturally grown objects that dominate unstructured environments.

What is a good explanation? A good, model-agnostic explanation highlights descriptive correlations between in- and output data to explain AI predictions. Consequently, a good explanation allows a human to comprehend predictions of an AI system on a high level. Qualitative user studies for predefined evaluation scenarios and with determined boundary conditions present a suitable assessment of explanation quality as the expressiveness of explanations is hard to describe using individual measures and key figures. The subsequent demonstration of X^{3}Seg shows that the explanation of clearly separable, coherent 3D point sets with a distinct geometry is more intuitive for a human operator than the explanation of grown, merged structures, such as vegetation. Furthermore, the distinctive symmetry of 3D objects represented in coherent 3D point sets benefits an intuitive explanation for humans.

6.2.2.1 Encompassing, Selective, and Predictive X^{3}Seg

Encompassing X^{3}Seg selects example-based explanations from the entire training dataset of the examined segmentation method. This process identifies prototypes and criticism by evaluating the similarity of the explanation target to each coherent 3D point set extracted from the training dataset. Naturally, this requires processing a high volume of data but ensures a high explanatory power and also allows an in-depth assessment of similarity metrics to identify suitable example-based explanations.

Selective X^{3}Seg provides a basic and fast explanation of class predictions with a small number of representative prototypes and criticism samples. The ProtoDash method [100] is applied to identify representative samples and thus limits the search space for prototypes and criticism to a small, previously selected number of representative 3D point sets

from the encompassing dataset. The representative, coherent 3D point sets are transformed into a suitable 1D representation, which implies a generalization of the 3D prototypes to some extent. This is desired in selective X^{3}Seg to compare the potential of a limited amount of prototypes to the holistic assessment in encompassing X^{3}Seg.

Contrasting encompassing and selective X^{3}Seg, predictive X^{3}Seg does not work with ground truth data: it identifies prototypes and criticism among coherent point sets with point-wise class predictions – the inference results of the examined segmentation method. Thus, 3D point sets with potentially wrong class predictions are also examined which is especially beneficial when analyzing prediction failures.

6.2.2.2 Generating the Sample Database

X^{3}Seg requires the input of point clouds with 3D points $\mathbf{p} = [x; y; z]^*$ and a label for each $\mathbf{p}$. The sample databases are composed as follows: encompassing X^{3}Seg includes all coherent 3D point sets extracted from the training dataset, selective X^{3}Seg reduces the encompassing sample database to a small number of representative 3D structures with ProtoDash, and predictive X^{3}Seg extracts coherent 3D point sets from the class predictions of the analyzed segmentation method.

Extraction of Coherent 3D Point Sets. Figure 6.9 illustrates the extraction of the coherent 3D point sets that form the respective sample databases. The data is subdivided according to class labels and explanation targets, and elements of the sample databases are extracted with an identical workflow and identical parameters for consistency. Outlier filtering using a kNN search [323] or voxelization of the subdivided data [329] is optional and achieved a higher similarity of cross-source point clouds as well as a higher insensitivity to noise with the downside of information loss.

Different clustering methods with different parameterizations were evaluated in X^{3}Seg: Euclidean clustering and clustering with VCCS supervoxels [209] were evaluated to extract coherent 3D point sets using the PCL implementations[10]. The assignment of a unique file ID with

[10] Euclidean Cluster Extraction: https://pointclouds.org/documentation/tutorials/cluster_extraction.html, access on 29.12.2021.

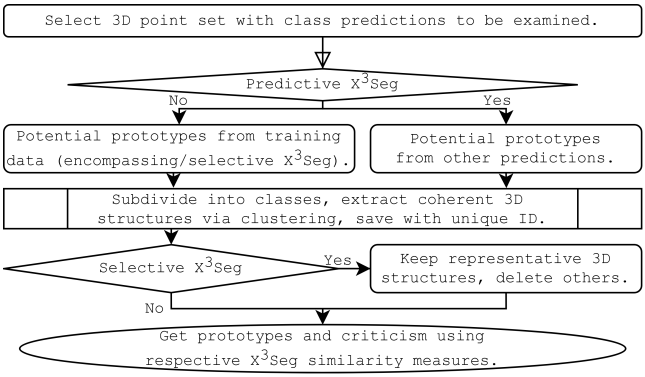


Figure 6.9 Encompassing, selective, and predictive X^{3}Seg at a glance [330].

clustering method as well as date and time of extraction allowed the allocation of each point set to its scene of origin as well as the traceability of the evaluated parameter settings during evaluation and tuning of X^{3}Seg.

Experimental evaluation justified that the estimated 3D shape of point sets plays a central role in cluster formation with clustering methods of higher complexity, such as [209] and [283]. However, the supvervoxel clustering of point sets from unstructured environment classes by exploiting the estimated shape did not prove useful, while Euclidean clustering with experimentally justified parameters yielded satisfying results and was chosen to determine coherent 3D point sets. Here, the cluster tolerance parameter as well as the minimum number of points required to form a cluster showed great impact on the Euclidean clustering result, while a high maximum number of points per cluster was of negligible importance. Generally, the minimum cluster size had to be adapted to the approximate size of the point set to be extracted, and it was thus determined in adaptive manner depending on the respective class and the cluster size in a maximum distance that still allows an expressive explanation. For point sets of limited extent such as traffic signs, trunks, or cars, the minimum size has to be smaller compared to wide spread areas, such as roads, buildings, or vegetation, and subcategories for clustering were formed: one for small, bounded 3D point sets, and one for rather extensive 3D point sets.

ProtoDash for Selective X^{3}Seg. The fast ProtoDash algorithm [100, 101] proved useful for selective X^{3}Seg in determining a small set of samples – the sample database of selective X^{3}Seg($\mathcal{X}^{\in}$) – that optimally represent another, notably larger, dataset, which is the sample database of encompassing X^{3}Seg($\mathcal{X}^{\infty}$). Contrasting the MMD-critic approach of [149], ProtoDash is faster, and non-negative weights are calculated for each sample in $\mathcal{X}^{\infty}$. Furthermore, criticism samples can be determined for the selected subset in $\mathcal{X}^{\in}$.

The selected ProtoDash implementation requires the input data to be represented as 1D arrays. Hence, all 3D point sets from the encompassing database were transformed into a 1D representation and saved separately according to their classes with their corresponding unique file ID. PCA proved useful to center the analyzed 3D point set on its two primary components and was hence selected instead of other possible projections for the projection of explanation targets and sample database elements in 2D space. Here, other projection methods did not provide a well-centered projection, such as a spherical projection according to Equation 6.1, as it is related to the sensor origin and would require an additional preprocessing step. Thus, PCA is applied on each coherent point set from the encompassing database and identifies the two axes to project the respective 3D point set in 2D, as described in Section 3.6. A discrete, quadratic grid of a previously defined resolution is created in 2D by binning the points in two dimensions. Here, minimum and maximum along each axis yield the ranges for binning and the resolution defines the number of bins for each axis. Each cell counts its occupancy by the number of inlying points, and the cell values are inserted into a 1D array line-by-line to map the 2D grid into 1D [336]. This takes spatial correlations into account as demonstrated on the MNIST dataset in [100]. Different grid sizes from 25×25 to 200×200 were evaluated on 3D point sets belonging to the trunk class. Here, the grid size presents a trade-off between accuracy and over-determinacy as well as in terms of calculation time. A grid was chosen instead of an octree structure and other options to preserve geometrical correlation between individual points from the user perspective. A discretization in 3D using a voxelization is also possible in an identical manner with binning in three axes. However, this was not utilized as it results in a high number of empty voxels with zeros in the

final 1D array and showed a detrimental influence on the determination of relevant 3D samples for prototypes and criticism with ProtoDash.

$\mathcal{X}^{\infty}$ is equal to the set of all 3D clusters belonging to one class in the database of encompassing X^{3}Seg. Consequently, ProtoDash was executed for each of the 19 well-defined classes to generate the selective X^{3}Seg database from SemanticKITTI. The target dataset $\mathcal{X}^{\in}$ is a subset of the source dataset $\mathcal{X}^{\infty}$. Hence, $\mathcal{X}^{\in} \in \mathcal{X}^{\infty}$ is the subset containing the most representative 3D samples determined out of $\mathcal{X}^{\infty}$ by ProtoDash. As discussed in Section 2.5.3, the MMD metric measures the difference between two data distributions $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$ [97, 98]. A subset $\mathcal{X}^{\in} \in \mathcal{X}^{\infty}$ optimally represents the dataset $\mathcal{X}^{\infty}$ if its MMD converges towards zero. Here, $\mathcal{X}$ is the function space that contains $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$, while $\mathcal{k} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the kernel function and $\mathcal{K}$ the reproducing kernel Hilbert space belonging to $\mathcal{k}$, as further detailed in [97, 98].

Finally, the discretized 1D arrays represent the two compared data distributions in selective X^{3}Seg and the MMD metric is given by

$$MMD(\mathcal{K}, P, Q) = \sup_{h \in \mathcal{K}} \left(E_{\mathbf{x} \sim p}[h(\mathbf{x})] - E_{\mathbf{y} \sim q}[h(\mathbf{y})] \right) = \sup_{h \in \mathcal{K}} \langle h, \mu_p - \mu_q \rangle \quad (6.12)$$

with $\mu_p = E_{\mathbf{x} \sim p}[\phi_{\mathbf{x}}]$, and p and q Borel probability measures defined on $\mathcal{X}$, as further detailed in Gretton et al. [97]. Further adaption and approximation of the MMD metric discussed in [101] converts the minimization problem of [97, 98] into the maximization of a goal function converging towards its maximum for an optimal sample selection in $\mathcal{X}^{\in}$ from $\mathcal{X}^{\infty}$. ProtoDash determines n representative samples from $\mathcal{X}^{\infty}$ to form $\mathcal{X}^{\in}$ in n iterations with each iteration starting with the calculation of the goal function that uses the prototypes from $\mathcal{X}^{\infty}$ currently included in $\mathcal{X}^{\in}$. The sample with the highest gradient in relation to the current goal function is the next potentially relevant point cloud sample from $\mathcal{X}^{\infty}$, and each iteration is concluded with the calculation of the prototype weights. As the 3D cloud samples are mapped onto a 2D grid by PCA and subsequently arranged into a 1D line vector, the kernel width σ_G determines the 1D Gaussian kernel

$$G(\mathbf{x}, \sigma_G) = \frac{1}{\sqrt{2\pi}\sigma_G} \exp - \frac{\mathbf{x}^2}{2{\sigma_G}^2}. \quad (6.13)$$

Here, the desired number of prototypes for $\mathcal{X}^{\in}$ and the width σ_G of the Gaussian kernel mainly influenced the prototype selection for $\mathcal{X}^{\in}$.

6.2.2.3 Similarity Measures for Coherent 3D Point Sets

Transformative similarity measures assess each 3D point set separately and compare these measurements for two point sets. They comprise s_X, c_X, v_X in encompassing and predictive X^{3}Seg and all metrics in selective X^{3}Seg. Direct similarity measures perform a registration of two point sets and interpret the registration result for similarity, such as e_X in encompassing and predictive X^{3}Seg. The number of points in coherent point sets as well as their absolute spatial extent was not considered for similarity as it showed a high dependence on the distance to objects and the resolution of the 3D sensor which limited the generalization of X^{3}Seg for multiple sensors and diverse environments. Scale proved to be a relevant, geometric feature of 3D point sets as it also correlates to the resolution, especially in sparse point clouds. As a result, an invariance to scale is not desirable for the proposed similarity measures as both close-range and far-off prototypes have to be present in the database to facilitate a proper explanation.

Encompassing and Predictive X^{3}Seg. Surface variation s_X, covariance c_X, and singular values v_X of the explanation target are compared to each 3D point set in the sample database. A high surface variation indicates a high curvature and sharp feature regions [135] and it can also determine the structured or unstructured character of a 3D point set and its information content [326]. The covariance measure c_X compares the normalized 3×3 covariance matrices, while v_X compares the singular values extracted from its singular value decomposition. The weighting of each measure $i \in \{s_X, v_X, c_X, e_X\}$ with the empirical mean $\mu(i)$ of all evaluated, static prototype classes ($\sum$ st) facilitated a straightforward analysis of quantitative results ($i/\mu(i)_{\sum \text{st}}$, see Table 6.12). The s_X measure evaluates the L_1 distance between the surface variation of the explanation target s_X^e and each database element s_X^d calculated according to Equation 3.11:

$$s_X = L_1(s_X^e - s_X^d), \quad \text{and} \quad S_s = \frac{s}{\mu(s)_{\sum \text{st}}}. \tag{6.14}$$

Here, $\mu(s)_{\sum \text{st}}$ is the average of the s_X measures for all evaluated database elements from all present classes.

Fehr et al. [66] build a covariance descriptor for feature extraction and subsequent object detection and recognition in point clouds, and c_X evaluates the normalized covariance matrices globally, similar to [66], with one covariance similarity measure c_X per analyzed 3D point set:

$$c_X = L_1(c_X^{e_{(j,l)}} - c_X^{d_{(j,l)}}), \quad \text{and} \quad S_c = \frac{c_X}{\mu(c_X)_{\sum \text{st}}},\ j,l \in \{1,2,3\}. \tag{6.15}$$

Here, $j,l \in \{1,2,3\}$ indicate the respective matrix elements of the covariance matrix. The object orientation provides important geometric information for the subsequent interpretation of the 3D data in robotic perception and the utilization of rotation-sensitive similarity metrics, such as covariance matrices [104], is beneficial. For instance, objects of the trunk class typically have their most prominent extension in height contrasting drivable areas, such as road or terrain, and invariance to rotations is not desired for all similarity metrics in X^{3}Seg. Fehr et al. [66] state that covariance matrices do not comply with Euclidean geometry due to their positive definite character, and that this can be met with the usage of special distance metrics, such as geodesic distance [72] or log Riemannian metric [4]. However, X^{3}Seg focuses on the intuitive and holistic analysis of 3D point sets, and covariance proved useful as one out of four similarity metrics in encompassing and predictive X^{3}Seg.

The singular values measure v_X and the score S_v are calculated with

$$v_X = L_1(v_X^{e_j} - v_j^d), \quad \text{and} \quad S_v = \frac{v_X}{\mu(v_X)_{\sum \text{st}}},\ j \in \{1,2,3\}. \tag{6.16}$$

To summarize, a low L_1 distance with a low S_s, S_c, and S_v indicates a high similarity. The fourth, direct measure is the Euclidean fitness score e_X after GICP registration [250], and S_e is calculated equivalently to S_s, S_c, and S_v. The regular ICP algorithm was not used as the ring structure from rotating 3D LiDAR sensors impeded proper point-wise registration [323]. The transformative similarity metrics s_X, c_X, and v_X evaluate the relative spatial point distribution and are scale-invariant, while e_X is sensitive to scale. Furthermore, s_X is rotation-invariant, while c_X and v_X are sensitive to different orientations.

An overall similarity score S_{ID} combines all measures to determine the similarity of each database element to the explanation target (ID) with normalized weights:

$$S_{\text{ID}} = \sum w_i \cdot S_i, i \in \{s_X, v_X, c_X, e_X\}, \quad \text{with} \quad 1/\sum_i w_i = 1.0. \tag{6.17}$$

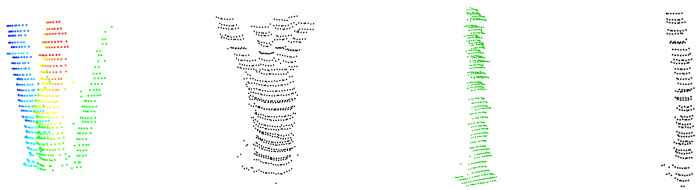
Here, the X subscript emphasizes that s_X, v_X, c_X, e_X are similarity measures in X^{3}Seg, e.g., in distinction from the surface variation s. Good matches have a low S_{ID}, while criticism exhibits a high S_{ID}. A high w_i shall indicate a high relevance of the respective metric i with scoring S_i to distinguish the regarded prototypes, and two possibilities were analyzed to derive the weighting w_i: w_i can be derived from a combination of the $\mu(S_i), i \in \{s_X, c_X, v_X, e_X\}$ results and from the occurrences of same-class prototypes in the 100 most similar prototypes of all classes for each i, or the statistical relevance of each measure can be determined by the empirical mean μ and variance σ^2 of all prototype measures inside the database with Equation 4.20 as a second option. The determination of w_i for the similarity measures with $\mu(S_i)$ and the occurrences of same-class prototypes in the 100 most similar prototypes yielded more comprehensible explanation results for encompassing and selective X^{3}Seg. In addition, the determination of w_i for the similarity measures according to a combination of the $\mu(S_i)$ results and with an in-depth analysis of the 100 most similar prototypes partially considered the statistical relevance within $\mu(S_i)$.

Selective X^{3}Seg. The similarity is measured for each point set independently in the 2D domain and four transformative similarity measures are applied in selective X^{3}Seg. Contrasting encompassing and predictive X^{3}Seg, PCA prior to similarity evaluation ensures the rotation-invariance of the presented similarity measures to a great extent. This was beneficial in selective X^{3}Seg, as it works with a notably lower number of prototypes compared to encompassing and predictive X^{3}Seg. Selective X^{3}Seg applies four similarity measures: the normalized 2×2 covariance matrices ($c_{X,2}$), the first and second principal components ($p_{X,1}$, $p_{X,2}$), and the ratio between the first and second principal component (r_X) indicating the relative spatial extent. In order to assess the similarity of explanation target and each database element, the L_2 norms of each similarity measure are evaluated. Here, $c_{X,2}$ is defined as the Frobenius

	$S_i = i/\mu(i)_{\sum \text{st}}$ in % for selected classes								
	best	40	50	51	70	71	72	80	$\mu(i)_{\sum \text{st}}$
	EX trunk (0, sc. 87, seq. 08)[*1*2], database seq. 09, $r_n = 0.10$ m.								
S_s	0.01	169.7	81.5	83.6	98.1	44.9	127.8	88.7	0.037
S_c	3.76	89.5	129.0	71.8	125.4	54.1	79.2	46.7	78.3
S_v	0.15	76.4	140.3	80.4	131.8	53.8	79.0	73.8	93.2
S_e	0.01	57.9	135.4	94.1	129.5	73.2	67.5	54.9	485.5
	EX trunk (0, sc. 87, seq. 08)[*1], database seq. 04, $r_n = 0.40$ m								
S_s	0.07	170.5	77.5	97.4	77.1	43.5	121.8	54.1	0.037
S_c	7.70	109.2	199.8	105.3	128.7	39.1	59.8	39.9	102.1
S_v	0.02	105.0	214.6	134.5	179.6	31.5	47.9	51.9	132.7
S_e	0.01	99.87	163.9	137.1	117.6	105.6	58.1	64.0	557.3
	EX car[*3] (0, sc. 26, seq. 08), database seq. 09, $r_n = 0.10$ m								
S_s	10^{-3}	118.5	37.5	44.5	168.1	96.5	84.8	60.2	0.032
S_c	2.07	86.3	136.0	90.7	127.7	50.2	76.8	44.0	72.32
S_v	0.23	68.8	154.7	81.9	138.9	36.8	69.3	46.8	93.39
S_e	10^{-3}	69.6	143.0	95.7	121.9	64.7	74.6	43.3	444.4
	PX trunk (0, sc. 965, seq. 08)[*4], database 08_P, $r_n = 0.40$ m								
S_s	0.03	180.9	98.8	103.2	74.5	39.2	146.7	103.9	0.042
S_c	1.59	43.8	161.4	76.7	143.6	48.3	58.0	13.2	73.13
S_v	0.04	29.4	165.4	67.2	152.7	49.4	49.6	6.6	118.7
S_e	10^{-4}	27.8	134.8	68.5	157.0	67.5	48.1	14.2	368.4

[*1] Figure 6.11(a).
[*2] $r_n = 0.40$ m: identical results except $S_s = 96.4$ for 70.
[*3] Car (10): $S_s = 0.51$, $S_c = 19.8$, $S_v = 1.39$, $S_e = 1.73$. [*4]Figure 6.11(e).

Table 6.12 Quantitative explanations with encompassing (EX) and predictive X^{3}Seg (PX): similarity measures for best-matching prototype (same class) and prototypes of other classes in % of $\sum$ st indicating the μ of all 19 static classes. The lowest overall similarity score for each explanation target indicates the highest similarity and is given in the column labeled with best. The columns titled with class numbers, as detailed in Table 6.13, show the average similarity scores for each class.



(a) (b) (c) (d)

Figure 6.10 Selective X^{3}Seg (SX) explanations for the trunk class: the best prototypes have the lowest S_{ID}: (a) upper trunk explanation target, (b) best upper trunk prototype for (a), (c) lower trunk explanation target, (d) best lower trunk prototype for (c). Upper trunk prototype with ramifications: sc. 116, seq. 08; lower trunk prototype: sc. 142, seq. 08. Images ©Fraunhofer IOSB.

norm F of the normalized 2×2 covariance matrices, as described in Section 3.11 and Equation 3.29. The first principal component $p_{X,1}$ describes the variance along the primary axis of each 3D point set, and the second principal component $p_{X,2}$ represents the width of the 3D point set. For the trunk class, $p_{X,2}$ indicates the diameter of the trunk as well as potential ramification, while experimental evaluation showed that ramifications have no influence on $p_{X,1}$. The third principal component typically points in the direction of the sensor origin for rotating 3D LiDAR sensors and correlates with the smallest eigenvalue and the estimated normal in surface estimation as discussed in Section 3.6. Finally, each similarity metric is calculated independently for each combination of potential prototype and explanation target, and the potential prototypes are sorted according to their similarity results for each of the four similarity measures $i \in \{c_{X,2}, p_{X,1}, p_{X,2}, r_X\}$. Equivalent to encompassing and selective X^{3}Seg, w_i facilitates different weighting of the proposed similarity measures, and S_{ID} yields the overall similarity with

$$S_{\mathrm{ID}} = \sum w_i \cdot S_i, i \in \{c_{X,2}, p_{X,1}, p_{X,2}, r_X\}, \tag{6.18}$$

whereby the determination of w_i according to the ranking position among all sample database elements in terms of i proved useful.

6.2.2.4 Proof of Concept: X^{3}Seg

X^{3}Seg is demonstrated on the semantic segmentation of 3D point clouds from SemanticKITTI [11, 82], and class predictions were generated with *DN53*-kNN, as detailed in Section 6.1.1. The training dataset consists

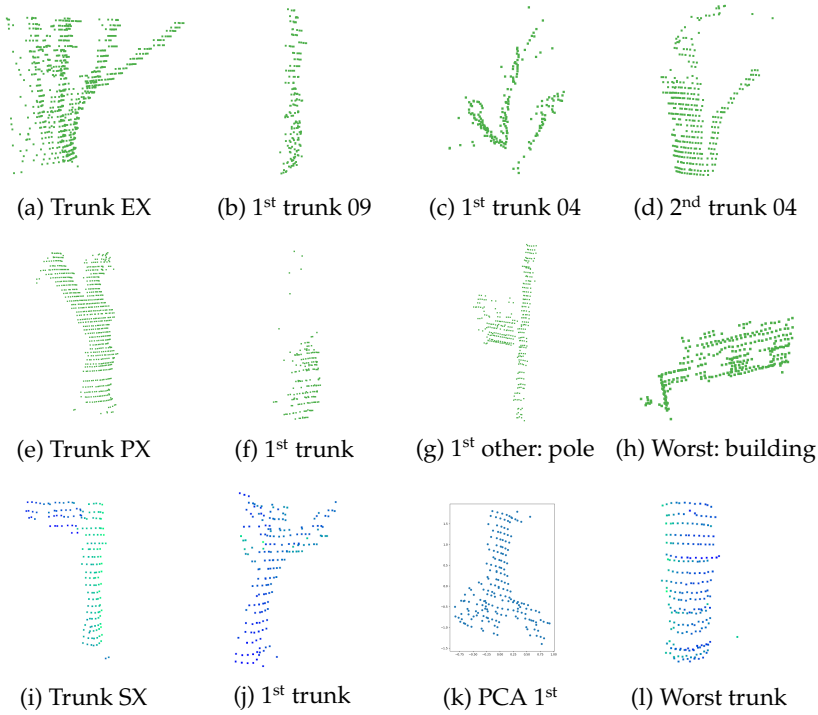


Figure 6.11 Model-agnostic explanations with encompassing (a–d, EX), predictive (e–h, PX), and selective X^{3}Seg (i–l, SX) [330]: (a), (e), (i) explanation targets; (b)–(d), (f), (j) best same-class prototypes ($S_b = 0.113$, $S_c = 0.178$, $S_d = 0.192$, $S_f = 0.016$); (g) best different-class prototypes ($S_g = 0.014$); (h), (l) criticism ($S_h = 4.154$).

of seq. 00 to 07, 09, and 10, and led to the inference results with *DN53*-kNN on seq. 08 being subject to explanation with X^{3}Seg. Table 6.12 presents quantitative, Figure 6.11 qualitative results for the trunk and road classes, and Figure 6.12 depicts a qualitative explanation result for

Figure 6.12 EX for car class predictions: explanation target (red, sc. 40, seq. 08) and best car prototype (blue, sc. 0, seq. 04) among the top 2 for e_X, v_X, c_X.

an exemplary explanation target of the car class[11] with the best-matching same-class prototype[12]. The best-matching prototype for all explanation targets given in Table 6.12 belongs to the class of the explanation target validating the corresponding class predictions. Furthermore, the given similarity scores show that the similarity of cloud samples from the trunk and road classes is very low (high similarity score S_i), while the similarity of trunk and vegetation is notably higher (low S_i). These similarities and dissimilarities allow a probability estimation for erroneous class predictions due to too high geometric similarities between different classes. To conclude, a high S_i indicates a high probability for class confusions.

Sample Database and Similarity Measures. The complete training set consists of 22,184 scenes [11]. The encompassing database contains prototypes for the 19 static classes from seq. 01, 04, and 09 of the training dataset for a clear and thorough evaluation, as illustrated in Table 6.13. Seq. 01 and 09 were chosen due to their unstructured character [326], while seq. 04 was captured in a mixed street scene. Predictive X^{3}Seg analyzes *DN53*-kNN predictions on seq. 08.

Preprocessing. SemanticKITTI consists of 3D point clouds from KITTI 2012 [82] with 2D label maps to train segmentation architectures. Consequently, class predictions in the inference step yielded 2D label maps for each scene, and a label was assigned to each 3D point **p**. For the evaluation of X^{3}Seg, all input points were converted to a customized point type[13] to allow a unique identification of the origin with XYZ encoding the geometric information of each **p** and L defining the label.

Extraction of Coherent 3D Point Sets. Different values for the cluster tolerance and the minimum number of points were evaluated. The

[11] sc. 40, seq. 08, unique ID: 080040100000202012121801
[12] sc. 0, seq. 04, unique ID: 040000100000202012111139
[13] `iosb::PointXYZLSeqScene`

Seq.[*1*2]	10	40	50	51	70	71	72	80	$\sum$ st[*3]
# 01	0	3151	154	3716	5230	23	2216	159	13,554
# 04	135	216	204	348	1540	118	550	149	4098
# 08_P	3389	1468	3876	661	10,859	1972	3357	464	28,796
# 09	4021	1124	5954	3528	12,908	1865	4477	533	39,644
$\mu(\text{IoU})_{08,P}$	0.91	0.94	0.86	0.54	0.84	0.53	0.73	0.53	0.53

[*1]Car (10), road (40), building (50), fence (51), veget. (70), trunk (71), terrain (72), pole (80). [*2]Scenes in seq.: 01: 1100; 04: 271; 08: 0–1000; 09: 1591. [*3] $\sum$ st: bi-/motorcycle, bus, person, truck, bicyclist, parking, pavement, lane-marking, traffic-sign.

Table 6.13 Composition of the sample databases for encompassing and predictive X^{3}Seg from 19 static classes: car, road, building, fence, vegetation, trunk, terrain, and pole (see also [*1]). Here, # designates the number of database elements. The predictions $\mu(\text{IoU})_{08,P}$, indicated by subscript *P* were obtained on seq. 08 with *DN53*-kNN [196].

cluster minimum is a trade-off between expressiveness and explanatory power. A high number of points extracted highly detailed 3D point sets as prototypes but limited the spectrum of possible explanations and generalization. A too low minimum reduced the expressiveness of explanations due to inconclusive representations of 3D point sets, which led to a limited recognition capability for commonly learned features. Prototype extraction for trunk on seq. 04 yielded 141 prototypes for at least 150 points and 118 prototypes with 180 points. However, identical car prototypes were extracted for 150 and 180 points in seq. 04. The extraction of explanation targets with class predictions from *DN53*-kNN from scene 0 to 124 in seq. 08 yielded 514 trunk clusters if a minimum of 150 points was required and 426 clusters for a minimum of 180 points. Euclidean clustering extracted 13,184 trunk clusters from the full training dataset (seq. 00 to 07, 09, and 10)[14]. Furthermore, different cluster tolerances were evaluated from 0.5 m to 5.0 m, and a cluster tolerance of 1.0 m with two subgroups for an adaptive selection of the cluster minimum yielded

[14] PCL, Euclidean Clustering: https://github.com/PointCloudLibrary/pcl/blob/master/apps/cloud_composer/tools/euclidean_clustering.cpp, access on 07.11.2021.

consistent and satisfactory results on SemanticKITTI. A minimum of 180 points for small point sets, such as trunk or car, proved useful, while prototypes and criticism for large sets, such as buildings, needed a minimum of 300 points per cluster to provide satisfactory explanation targets and sample database elements.

Validation of Similarity Measures. Coherent 3D point sets within the SemanticKITTI dataset appear in multiple, consecutive scenes due to the 10 Hz capture frequency of the Velodyne LiDAR [82]. Here, extracting a 3D point set from one scene and searching in consecutive scenes determined different representations of the same objects thus validating the proposed similarity measures in X^{3}Seg.

Similarity in Encompassing and Predictive X^{3}Seg. The similarity of the explanation target was compared to each database element using the determined results for s_X, v_X, c_X, and e_X. The normal estimation for s_X on the basis of all points inside a radius of $r_n = 0.40$ m proved useful [326], and Table 6.12 shows the comparison to $r_n = 0.10$ m. The exclusive consideration of s_X, v_X, c_X, and e_X identified different prototypes and criticism, while the weighted combination in S_{ID} allowed a stable and holistic similarity assessment that was also independent of the sample resolution in the 3D cloud. The empirical mean results $\mu(i)$, $i \in \{s_X, c_X, v_X, e_X\}$, were analyzed to determine w_i in encompassing and predictive X^{3}Seg. For the same class, $\mu(i)$, $i \in \{s_X, c_X, v_X\}$ yielded an average of approximately 50 % of $S_i = i/\mu(i)_{\sum \text{st}}$ (Table 6.12). As a result, they provided a satisfactory description of the characteristics of coherent 3D point sets to measure similarity, and s_X proved to be the strongest similarity metric. The occurrences of prototypes of the same class in the top 100 most similar prototypes of all classes in terms of s_X, v_X, c_X, and e_X yielded the S_{ID} weights w_i: $w_{s_X} = 0.45$, $w_{c_X} = 0.20$, $w_{v_X} = 0.30$, $w_{e_X} = 0.05$. Extensive evaluation of GICP for e_X in X^{3}Seg showed that it is unlikely to get stuck in local minima even in the case of objects of different orientations. It is assumed that this was caused by the registration of two bounded and coherent 3D point sets being favorable for GICP [250].

Selective X^{3}Seg Parameterization. Selective X^{3}Seg proved to be particularly applicable in explaining predictions with high $\overline{\text{IoU}}$). An imple-

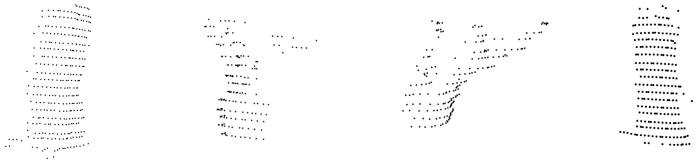
mentation of ProtoDash was provided within the IBM AIX360 toolkit[15] and required the input of the two databases $\mathcal{X}^{\infty}$ and $\mathcal{X}^{\in}$ as 1D arrays. Multiple parameterizations for the selection of representative database elements with ProtoDash were evaluated:

- Grid resolution for 2D discretization: 25×25 to 200×200,
- Width of 1D Gaussian kernel σ_G in ProtoDash: $\sigma_G = 1$ to $\sigma_G = 50$,
- Representative 3D samples in $\mathcal{X}^{\in}$ (ProtoDash): $\#\mathrm{P} \in \{25, \dots, 500\}$.

Experimental evaluation justified that the required number of prototypes for a beneficial explanation in selective X^{3}Seg depends on the complexity of the class of the explanation target. Selecting 100 prototypes for the trunk class described the differences in trunk clusters properly, and more than 100 prototypes did not further aid the expressiveness in the explanation of trunk class predictions. An expressive explanation of more simple geometric structure classes, such as building or road, was achieved with 25 prototypes in $\mathcal{X}^{\in}$. Figure 6.10 depicts two exemplary concise trunk predictions that were used to evaluate different ProtoDash parameterizations for selective X^{3}Seg. A grid resolution of 100×100 determined satisfactory, highly diverse prototypes for small objects, such as trunk. A performance degradation of ProtoDash due to an excessive number of empty grid cells resulting in many zeros and little variation of the 1D arrays was not observed. Larger objects and area structures, such as buildings or roads, may require a higher resolution of the grid for a sufficiently accurate discrete representation. Figure 6.13 shows the 2D representation and compares the grid resolutions 25×25 to 100×100 for the two trunk explanation targets depicted in Figure 6.10. It becomes clear that a higher complexity of the explanation target as shown by the trunk structure with ramifications in Figure 6.13 requires a sufficiently high resolution of the grid to identify well-matching prototypes from $\mathcal{X}^{\in}$. For structures that are less complex than the lower trunk structure analyzed in Figure 6.10, the smaller grid size did not impair the proper selection of the most similar prototype.

Different values for σ_G were evaluated on explanation targets of different complexity, and the best-matching prototypes were analyzed for

[15] IBM Research Trusted AI: AI Explainability 360, http://aix360.mybluemix.net/, access on 06.12.2021.



(a) 25×25. (b) 100×100. (c) $\sigma_G = 2$. (d) $\sigma_G = 10$.

Figure 6.13 Selective X^{3}Seg prototypes: the images (a) and (b) contrast a low (25×25) and a high (100×100) resolution grid for the lower trunk explanation target (see Figure 6.10(a)). Different resolutions have a notable influence on the best-matching prototypes determined by $\mathcal{X}^{\in}$ with #=100 representative samples identified by ProtoDash. Image (c) and (d) compare the best-matching prototypes for $\sigma_G \in \{2, 10\}$. Images ©Fraunhofer IOSB.

different trunk explanation targets, among them the two exemplary trunks depicted in Figure 6.10. The influence of the kernel size σ_G on the prototypes determined by ProtoDash increased with a higher complexity of the explanation target, such as ramifications for the trunks in Figure 6.10. A high σ_G introduced an averaging character into the prototype selection process, especially for explanation targets of higher complexity. Similar to the grid size, a higher complexity of the explanation target required less generalization for a high explanation quality. For the more complex, upper trunk structure in Figure 6.10, $\sigma_G = 10$ identified a best prototype qualitatively worse than $\sigma_G = 2$ due to the higher generalization character. Hence, a kernel size of $\sigma_G = 2$ was selected in ProtoDash to determine the sample database $\mathcal{X}^{\in}$ in selective X^{3}Seg. This allowed the identification of well-matching prototypes for explanation targets of higher complexity. A low $c_{X,2}$ highlights a high similarity between the explanation target and the best prototype from $\mathcal{X}^{\in}$, and the similarity of the best prototypes in terms of $c_{X,2}$ decreased with increasing σ_G, as illustrated in the selection of the best-matching prototypes in Figure 6.13. Furthermore, Figure 6.13 describes the influence of σ_G and different grid resolutions on the identified best-matching prototypes.

Intuitively, a higher number of prototypes #P led to a better understanding of the class predictions in X^{3}Seg due to a higher variety of prototypes, which enabled the identification of a very similar prototype in the training data. Experimental evaluation justified this intuitive assumption: all

similarity measures indicated an identical or higher similarity of the explanation target and the best-matching prototype identified. Figure 6.10 depicts the best-matching prototypes for the selected lower and upper trunk structures. However, the number of prototypes in $\mathcal{X}^{\in}$ essentially influenced the calculation effort to identify prototypes and criticism for the current explanation target. Furthermore, the most diverse and thus also most relevant prototypes to represent the huge amount of available prototypes in the database of encompassing X^{3}Seg with the small subset in $\mathcal{X}^{\in}$ were always identified among the first 100 prototypes. Hence, a higher number of prototypes only led to the selection of more prototypes in addition to the extremes initially chosen from the dataset $\mathcal{X}^{\infty}$.

To conclude, a guideline for the number of prototypes from $\mathcal{X}^{\in}$ to represent $\mathcal{X}^{\infty}$ properly can be derived as follows: a high number of prototypes in $\mathcal{X}^{\in}$ is required to properly represent $\mathcal{X}^{\infty}$ for a high complexity, diversity, and size of $\mathcal{X}^{\infty}$. A selection of #P = 100 trunk prototypes for $\mathcal{X}^{\in}$ yielded holistic, well-funded, and satisfactory explanation results for understanding trunk class predictions with selective X^{3}Seg.

Similarity in Selective X^{3}Seg. The similarity metrics $c_{X,2}$, $p_{X,1}$, $p_{X,2}$, and r_X were evaluated with a 100×100 2D grid, PCA without standardization, $\sigma_G = 2$, and #P = 100. All similarity metrics indicate a high similarity with low values, while criticism has high values for $c_{X,2}$, $p_{X,1}$, $p_{X,2}$, and r_X. Each proposed similarity metric evaluates a different geometrical aspect of the 3D point sets: the best-matching prototype for the upper trunk explanation target in terms of $p_{X,1}$ may not have had ramifications but did have a similar orientation of the main component of the trunk structure, while the best-matching prototype in terms of $p_{X,2}$ had ramifications similar to the explanation target (see also Figure C.7). To conclude, all four similarity measures proved useful in identifying 3D point sets as prototypes and criticism. Furthermore, the combination of their complementary analysis of the geometrical characteristics of each prototype provided a valuable similarity measure for the analyzed 3D explanation targets. A selection of three trunk prototypes, including the two explanation targets depicted in Figure 6.10, was analyzed in-depth to validate the selected metrics. Similar to encompassing and predictive X^{3}Seg, the weights w_i, $i \in \{c_{X,2}, p_{X,1}, p_{X,2}, r_X\}$ were derived from the occurrences in the best-matching prototypes in terms of $c_{X,2}$,

$p_{X,1}$, $p_{X,2}$, and r_X. In contrast to encompassing and predictive X^{3}Seg, only the ten most similar (top 10) prototypes for each individual similarity metric were analyzed as the top 100 prototypes would include all prototypes in $\mathcal{X}^{\in}$. Here, $p_{X,2}$ achieved the highest top 10 accordance with 44.5 %, $c_{X,2}$ achieved 33.3 %, $p_{X,1}$ 30.0 %, and r_X 36.7 %. Both $p_{X,1}$ and $p_{X,2}$ evaluate the primary components and constitute a connected, correlating similarity metric. To conclude, each similarity metric in selective X^{3}Seg presented an approximately similar importance and suitability. Consequentially, the weights w_i for S_{ID} were determined as follows in selective X^{3}Seg: $w_{c_{X,2}} = 1/3$, $w_{p_{X,1}} = 1/6$, $w_{p_{X,2}} = 1/6$, $w_{r_X} = 1/3$. S_{ID} provided stable explanation results, also for different scale, rotation, and noise characteristics of the prototypes in selective X^{3}Seg. Contrasting this, encompassing and predictive X^{3}Seg allowed more differentiated explanations due to more extensive databases, especially for the understanding of inaccurate predictions and the identification of similarities.

6.2.2.5 Understanding Inaccurate Class Predictions

Trunk class predictions exhibit a low $\overline{\mathrm{IoU}}$ in SemanticKITTI (Table 6.13). Here, correlations to the best prototypes of other classes as well as the similarity to other classes, as described in Table 6.12, provide helpful insights, especially in how to analyze a low $\overline{\mathrm{IoU}}$ for single classes. In order to examine erroneous trunk predictions with encompassing X^{3}Seg, the sample database was composed of all prototypes from seq. 01, excluding the 23 trunks. X^{3}Seg highlighted notable correlations of trunk to fence (51), vegetation (70), and pole (80), and the nine best prototypes in terms of s_X, two of the ten best for c_X, and the best-matching prototype with v_X were from the fence class. Furthermore, all ten best-matching prototypes in terms of e_X came from the vegetation class. This can be explained by the fact that seq. 01 was captured on a motorway that had crash barriers on either sides labeled as fence and containing vegetation elements. Furthermore, accurate class predictions for point sets representing natural, grown elements, such as trunk, are challenging due to an often mixed occurrence with other classes and a low dissimilarity to other natural elements. Here, class definitions that notably distinguish or unite coherent point sets often occurring together can provide more

accurate predictions, especially for segmentation methods without color information, such as *DN53*.

The similarity between classes described by $\mu(i)$ in Table 6.12 also provides a likelihood estimate for classification errors. For instance, classifying a coherent 3D point set of a trunk as road or building proves to be very unlikely, whereas the classification as pole is shown to be notably more probable. Figure 6.11(g), the best-matching prototype from other classes, was of the pole class and $\overline{\text{IoU}}$ for pole is low. Furthermore, the qualitative similarity between Figure 6.11(e), (f), and (g) highlights the difficulty for accurate class predictions. In critical applications, X^3Seg facilitates a first risk assessment: the determination of classes with the highest similarity can provide an indication on the danger and impairment of false predictions. This also allows a likelihood estimate for misclassifications that are highly relevant when distinguishing navigable ground and obstacles for the navigation of autonomous off-road vehicles.

7 Application Scenarios

The decontamination of hazardous environments and off-road transport in defense constitute the two main application scenarios for autonomous off-road vehicles in unstructured environments in this thesis. The thoughts, methods, and approaches within this chapter focus on the practical application of the previously explained scientific contributions to low-level, mid-level, and high-level perception, and naturally influenced their development and testing.

Section 7.1 reviews the integration of the proposed methods in the processing chain for autonomous off-road vehicles in decontamination and defense. Section 7.2 details the perception–validation coupling proposed and describes the technology demonstrators utilized for the proof-of-concept demonstrations within this thesis with exemplary, modular perception pipelines. Furthermore, Section 7.3 indicates generalization possibilities for the proposed methods.

The highly limited availability of data for test and verification purposes has severely impacted the research on unstructured environment perception. The *IOSB-Reg* dataset proposed in Section 7.4 and the **G**erman **O**utdoor **O**ff-road data**se**t (*GOOSE*) discussed in Section 7.5 aim to close this gap.

Trustworthy and reliable decisions in the sensing–perception–decision system architecture require accurate and suitable environmental perception. Hereby, planning for autonomous off-road vehicles mostly relies on "single-shot" 3D perception, as it is provided by the perception and validation methods. Hence, the planning constraint [328] discussed in Section 7.6 contributes to the decision step and constitutes an important connecting point to the perception methods proposed within the sensing–perception–decision chain.

7.1 Decontamination and Defense

The combination of a very challenging environment with a very limited availability of data for training, testing, and verification makes the introduction of ML into decontamination and defense very difficult. Walther et al. [331] and Woock et al. [332] discuss the decontamination of hazardous environments with autonomous off-road vehicles. Woock et al. [332] provide an overview of robotic technologies for the landfill industry with a special focus on the processing chain of autonomous heavy construction machines. The authors [332] state that the first step towards an autonomy of autonomous off-road vehicles is the sensor equipment for environment perception. Sensor data must be referenced extrinsically and to the coordinate system of the construction machine to fuse the sensor measurements for joint and complementary environment perception. Furthermore, the autonomous system requires knowledge about itself such as the tool position that can be reconstructed from the joint angles and the kinematic model of chassis, cabin, boom, and dipper stick. As a result, the successful perception of the environment facilitates the planning of meaningful and targeted movements for the tool and the complete machine itself, while a motion control system jointly monitors the position of all machine components and the status of the environment during task execution. An intervention can be carried out where necessary to react to a situation that has changed in the meantime. Last but not least, a sequence control system ensures that individual subtasks are processed sensibly one after another and that the construction machine does not endanger itself or proceed inefficiently [332].

In defense logistics, MULE and Convoying require fewer humans in danger zones and can also overcome the lack of manpower in off-road transport. Recent advances in ML technology extended the capabilities of autonomous systems in off-road scenarios, especially by an optimized perception performance. Two key challenges can be identified for AI and ML especially in defense[1]. The first challenge is the development of trusted AI systems. Trusted AI systems have to fulfill the three funda-

[1] "AI for Defense" workshop (24.09.2020), https://eda.europa.eu/news-and-events/news/2020/09/28/eda-workshop-with-industry-on-artificial-intelligence, access on 23.01.2022.

mental requirements identified by the AI HLEG[2]: lawfulness, compliance with ethics, and social and technical robustness. The second challenge lies in the limited availability of data, particularly in defense. Here, the development of common standards to generate, process, and save data, as discussed in Section 7.5, facilitates the collaboration in developing and benchmarking software and algorithms. Benchmarks for classic and ML methods promote advances in image processing, such as the well-known KITTI Vision Benchmark [82] or the ImageNet competition [233].

7.2 Perception–Validation Coupling, Technology Demonstrators, and Perception Pipeline

This thesis recommends a tight coupling of perception and validation for confidence and data assessment as well as for registration. This allows immediate actions in case of inaccurate or erroneous, detrimental data, as discussed in Section 4.1, 4.3, 5.3, and 6.2.1. The validation of the proposed perception methods is loosely coupled with the analyzed perception method. A tight coupling of perception and validation for all proposed perception methods would impair the flexibility of the pipeline design and notably increase the processing effort for the environment perception.

Loosely coupled validation is conducted in a post-modeling manner – during its evaluation and fine-tuning process or after the completed development of the method. This facilitates a consistent, and mainly model-agnostic assessment of the proposed perception methods. Furthermore, a tight perception–validation coupling for autonomous off-road vehicles requiring real-time capability is only possible with GPGPU parallelization currently, as exemplarily demonstrated in Section 5.1.1.2.

Potential consequences for the subsequent planning and control steps in case of unsatisfactory results from the tightly coupled validation methods are the slower execution of autonomous navigation and manipulation tasks or the requiring for teleoperation by a human operator.

[2] Ethics Guidelines for Trustworthy AI: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, access on 24.10.2021.

Here, the integration of loosely and tightly coupled validation facilitates confidence-aware, adaptive perception for an enhanced planning and control behavior of autonomous off-road vehicles.

A modular system architecture for autonomous off-road vehicles ensures a fast and generic deployment of the developed modules for many autonomous platforms ranging from small indoor platforms to heavy construction machinery. Furthermore, the algorithm toolbox (ATB) of the Fraunhofer IOSB [60] provides a modular concept to equip mobile robots with autonomy capabilities based on ROS. ROS provides a broad and modular basis of software libraries and tools to build robot applications with a common standard and framework for software exchange in research and industry all over the world. The ATB combines all modules required for the autonomy of mobile robots such as excavators, off-road trucks, and small electrically powered platforms. Sensor drivers, as well as algorithms for localization, mapping, planning, and performing manipulation tasks can be combined in a flexible way, depending on the requirements of the platform and the desired functionalities and similar to the flexible and modular structure of the perception–validation pipeline proposed in this thesis.

The ATB and customized hardware and software equipment enable the autonomous operation of IOSB.BoB, IOSB.amp Q1, IOSB.amp Q2, IOSB.Alice, and TULF depicted in Figure 1.3 for decontamination [216] and defense applications. A 1.8 t Wacker Neuson crawler excavator is used for the IOSB.BoB platform, while a 24 t Liebherr R924 crawler excavator is utilized for IOSB.Alice. Both excavators are equipped for the autonomous remediation of landfill sites and capable of autonomous navigation and manipulation. The optical multi-sensor system of IOSB.BoB consists of three Velodyne VLP-16 3D LiDAR sensors, while IOSB.Alice is equipped with four LiDAR sensors: three Ouster OS0-64 and one Ouster OS0-128, as well as with two multispectral JAI FS3200-10GE cameras in stereo-setup. The IOSB.amp Q1 platform, also referred to as Mustang, was used to capture the *IOSB-Reg* dataset discussed hereinafter. It is equipped with a Velodyne HDL-64E 3D LiDAR and two JAI AD-130GE

cameras[3]. The AD-130GE stereo camera setup captures 1296 × 966 px RGB and NIR images simultaneously with identical exposure times. The perception setup for the Bundeswehr's TULF consists of a hyperspectral stereo-setup with Ximea MQ022HG-IM-SM4X4-VIS[4] cameras and two Velodyne HDL-32E LiDAR sensors for perception purposes. In addition to the optical multi-sensor system, all platforms are at least equipped with an inertial measurement unit and a satellite positioning system for localization. An accurate localization for all platforms was provided by the ATB localization and mapping module [60].

Two exemplary perception–validation pipelines are described to illustrate the combination of individual methods for the perception of unstructured environments: the basic perception setup for IOSB.BoB and the more complex perception setups for IOSB.Alice and IOSB.amp Q1. Loosely and tightly coupled validation methods are optional and can be integrated according to demand. Tightly coupled confidence assessment is recommended if a stereo camera setup is integrated in addition to the LiDAR sensor for 3D–3D fusion, as proposed in Section 5.3.

The IOSB.BoB platform demonstrates a basic perception solution with workspace monitoring by multiple rotating 3D LiDAR sensors. A stereo camera setup is not recommended as a substitute for the 3D LiDAR sensors due to the limited FoV and the lower depth estimation accuracy not being able to guarantee highly accurate workspace monitoring at the moment. Registration exploiting the extrinsic calibration of the LiDAR sensors and their registration to the robotic platform were performed with the similar-source registration method detailed in Section 4.2. The interpretation of the perceived data with high-level perception and validation methods is not necessarily required as navigability analysis and obstacle avoidance are also possible without a high-level interpretation of the 3D data.

The autonomous IOSB.Alice and IOSB.amp Q1 off-road vehicles exhibit more complex, holistic perception pipelines with perception and validation methods from each level. The registration of the multi-sensor

[3] JAI: Datasheet AD-130-GE, https://www.jai.com/downloads/datasheet-ad-130ge, access on 07.11.2021.

[4] Hyperspectral Snapshot USB3 camera 16 bands, https://www.ximea.com/files/brochures/xiSpec-Hyperspectral-cameras-2015-brochure.pdf, access on 25.11.2021.

systems relies on the *UCSR* approach discussed Section 4.3 with a tightly coupled validation of the registration results. Grayscale, RGB, or hyperspectral stereo camera systems with disparity estimation from stereo images and loosely coupled validation can be integrated optionally. Semantic 3D segmentation provides a point-by-point interpretation of 3D point clouds, as detailed in Section 6.1. Dataset assessment with *IC-ACC* provides a tightly coupled, pre-modeling validation for the ML methods in low-level registration, mid-level stereo processing, and high-level segmentation. The post-modeling analysis of the segmentation results with X^3Seg allows an in-depth understanding and a loosely coupled validation of the segmentation results, as described in Section 6.2.2.

7.3 Generalization of the Proposed Methods

Perception and validation methods for 2D images from different camera systems were already explained in this thesis. The proposed perception and validation methods for 3D data can also be used for other sensors providing depth or 3D measurements, such as ToF cameras, light field cameras, RGB-D cameras, and high resolution 3D radar sensors [227]. The registration methods listed for similar-source and cross-source data in Section 4.2 and 4.3 are also applicable for similar-source or cross-source 3D data from active cameras, light field cameras, and radar with minor adjustments. Validation methods, such as the disparity error metrics (see Sec. 5.2.1) or SET (see Sec. 5.2.2), also apply to disparity maps from light field cameras and for 3D clouds from ToF and light field cameras. Furthermore, the application of selected perception and validation methods on data from structured environments demonstrates the generalization of the proposed methods for other domains. Potential subsequent processing steps, such as mapping and planning, are also discussed to address the application of the proposed methods in other use cases exceeding the proof-of-concept demonstrations in this thesis.

7.4 *IOSB-Reg* Dataset

The *IOSB-Reg* dataset provides data from primarily unstructured environments to train, validate, and test the classic and ML registration

methods discussed in Section 4.3. It was captured with the IOSB.amp Q1 platform on the test site for autonomous platforms at the IOSB headquarters in Karlsruhe between August 2019 and March 2020 and includes different seasons. 176 3D LiDAR point clouds and 2D image pairs from a stereo camera setup with corresponding point clouds and intrinsic and extrinsic calibration information were captured in ROS bagfiles. Four exemplary RGB images are depicted in Figure D.3. Selected images include structured test elements to investigate the correlation of registration and the structured or unstructured character of a scene.

Image–LiDAR synchronization for the KITTI dataset [80] relies on a reed contact to provide hardware-triggering for the cameras. This is only possible for a limited number of LiDAR sensors with rotating parts such as Velodyne HDL-32E and HDL-64E. Contrasting this, the IOSB.amp Q1 platform for the capture of *IOSB-Reg* extracted the current rotation angles of Velodyne and Ouster LiDAR sensors from their data packages, and a micro controller generated an analog hardware-trigger signal. The 2D RGB images were thus captured while the Velodyne recorded the camera FoVs. A fixed exposure can lead to overexposure or underexposure in highly variable light conditions, and auto-exposure commonly yielded better results in outdoor environments.

The JAI AD-131GE cameras comply with the EMVA GenICam standard[5]. As a result, Aravis, a generic driver for GenICam compliant cameras, and camera_aravis[6] were used for image capture in ROS. The camera images were rectified using OpenCV *stereoRectify*[7], and the image size was preserved in rectification. Images and point clouds of static scenes were captured in the coordinate system of the sensor itself, and the ATB [60] provided localization information with 100 Hz.

Augmentation techniques can multiply the size of the dataset, and rotation, flipping, and shifting proved useful to augment registration data from unstructured environments. Other techniques such as horizontal

[5] European Machine Vision Association, Generic Interface for Cameras - GenICam, https://www.emva.org/standards-technology/genicam/, access on 30.10.2021.

[6] Camera_aravis: An Ethernet camera driver for ROS, http://wiki.ros.org/camera_aravis, access on 30.10.2021.

[7] Camera Calibration and 3D Reconstruction: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, access on 17.01.2022.

translation, scaling, and cropping were not applied as they alter the input format of the data or require pixel interpolation. Dataset augmentation was introduced at runtime during training as the augmentations are computationally lightweight in contrast to their memory size, e.g., with the Keras Image Data Generator[8] for image augmentation.

7.5 *GOOSE*: German Outdoor Off-Road Dataset

Since 2021, TAS, the Fraunhofer IOSB Karlsruhe, and the WTD41 have been working together on *GOOSE*, a dataset for unmanned ground systems with financial support of the BAAINBw U6.2[9]. Limits, difficulties, and optimization potential of state-of-the-art datasets and of the *IOSB-Reg* dataset were analyzed in the conceptual development phase in 2021. Here, the KITTI dataset [83] was selected as a first reference point for data capture and processing.

GOOSE addresses the limited data availability from unstructured environments and consequently supports the development of robust and reliable AI systems from the outset. It provides data for the development of classic and ML methods for autonomous systems in the ROS standard with a special focus on high quality as well as validation of the data during dataset generation. *GOOSE* is one of the first datasets with data from different institutions, sensor setups, and platforms. This sets the course for trustworthy and reliable AI systems right from the start in terms of pre-modeling XAI. Contrasting other datasets, *GOOSE* includes data from different off-road vehicles: MuCAR-3 of TAS[10], IOSB.Alice, and Mustang. Uniform standards for recording, storing, and processing sensor data have been introduced. Further validation and quality-checking methods for the recorded data are also being developed for *GOOSE* and extend the dataset assessment detailed in Section 6.2.1. *GOOSE* is

[8] Keras API reference: Image data preprocessing, https://keras.io/api/preprocessing/image/, access on 07.11.2021.

[9] "VIII. SGW-Forum Unbemannte Systeme", 26.10.2021: "Das GOOSE-Dataset: Ein gemeinsamer Datensatz für KI-Anwendungen", https://veranstaltungen.dwt-sgw.de/anlage?i=2132&c=zTeBRWkSXkRXkSY&t=954853&n=000021_agenda.pdf, access on 17.01.2021.

[10] MuCAR-3, https://www.unibw.de/tas/ausstattung/mucar-3, access on 07.11.2021.

designed to contain labeled images and point clouds from unstructured environments as well as all relevant information for the development of AI procedures for autonomous systems, such as localization and synchronized time stamps. In addition, a customized *GOOSE* labeling policy was specified by TAS and IOSB for all data that shall be included in *GOOSE*. For this, the minimum sensor requirements to capture *GOOSE* data are one 2D camera, one 3D LiDAR sensor, and one localization providing a NavSatFix message.

GOOSE is designed as training, validation, and test dataset for ML approaches and also as a reference dataset for existing, non-ML approaches. The highly detailed specification for the captured data and meta data facilitate an equivalent data capture, storing, and processing for different contributors with different off-road vehicles and sensor setups. *GOOSE* is structured in three levels for a clear hierarchy that provides a fast and easy overview for users in the robotic community. "Setup" constitutes the highest level and summarizes data for identical hardware and sensor setups of a platform. The compatibility for training, validation, and testing is ensured for data within the same setup and without the need for code adaptions. "Scenario" defines the intermediate level of the *GOOSE* hierarchy and all data samples from the same scenario with a predefined task and identical environmental conditions but potentially different weather conditions. "Sequence" constitutes the lowest hierarchy level, and one sequence is represented by a single ROS bagfile. For instance, an autonomous earth excavation process lasting for several days comprises multiple sequences that belong to the same scenario. Meta data is automatically generated for each sequence and contains all relevant information for using the dataset, such as sensor types and manufacturers, intrinsic and extrinsic sensor calibrations, descriptions of the sensor coordinate systems, and the estimated accuracy of the sensor data timestamps inside a `yaml` file. An automatic offline validation of the generated `yaml` file against the specification ensures the compatibility of the captured data with the *GOOSE* specifications. Predefined tags for weather, environment, and platform and sensor setup enable fast identification of useful scenarios and sequences for users.

However, it becomes difficult with a large number of topics to check whether every necessary topic is present and to control the individual

data parameters. A customized validation tool was developed to analyze the sensor data for completeness and fulfillment of the predefined minimum requirements during the capturing process (online) and afterwards (offline) for this purpose, as illustrated in Figure D.4. Color codes and supporting text facilitate the detection of problematic topics that could negatively impact the recorded data. Minimum requirements like publishing frequency or criteria for data quality can be defined for each desired topic within a customized configuration file.

Many publicly available tools for the pixel-wise annotation of 2D images and the point-wise annotation of 3D point clouds are available. An experimental evaluation of selected tools was conducted to analyze available labeling tools for *GOOSE*, as detailed in Section D.1. As a result, a labeling tool for *GOOSE* was not developed as the state-of-the-art labeling tools available already provided satisfactory results.

7.6 Cost Valley for Constrained Planning

Planning extends the proposed perception pipeline towards the mapping and planning for autonomous off-road platforms. Here, the work of Forkel et al. [71] on probabilistic terrain estimation highlights the close connection of perception and planning for the identification of drivable areas. To the best of the author's knowledge, the proposed cost valley approach [328] constitutes a novel planning constraint optimizing the autonomous driving behavior of off-road vehicles in transport scenarios encountered in military logistics.

The track in off-road scenarios is not defined by street borders and considerate behavior of other vehicles cannot be assumed in contrast to autonomous driving in structured, urban environments. The main challenges for an optimal driving behavior in unstructured environments are the navigability of the path and obstacle avoidance. The cost valley allows the avoidance of static and dynamic obstacles while simultaneously keeping a predetermined track in an accurate manner. It additionally facilitates two evolved transport behaviors for autonomous vehicles of

different kinematics, weight, and size in military logistics: MULE[11] and Convoying[12]. The experimental evaluation of the presented constrained planning approach [328] was carried out on IOSB.amp Q1 and TULF, as depicted in Figure 1.3. In addition to MULE and Convoying, Following scenarios present the third, highly relevant transport behavior for autonomous vehicles in off-road environments. To address this challenge, Albrecht et al. [321] present an enhanced convoying functionality with two operations modes: one for exact, and one for flexible Following, and the interested reader is referred to [321] for further details about Following scenarios.

7.6.1 Line Simplification, Grid Structure, and Cost Valley

A grid map [64] with obstacle and navigability information is required as input for the cost valley. The grid map enables discretized mapping to prepare the input data for search-based planning approaches (see Section 2.6). The cost valley can flexibly be integrated in existing processing pipelines for autonomous driving, as presented in [40, 60]. The valley is integrated as an additional layer of the grid map and the final grid structure contains all information needed for planning. The generation of the cost valley is divided up into three steps, as illustrated in Figure 7.1: track specification, waypoint optimization, and navigation.

Track specification denotes the recording of globally referenced waypoints. A global reference is required to reuse all processing results. The recorded track for MULE can be generated with a teach-in procedure for the teleoperation of the vehicle. Waypoints for Convoying are directly transmitted from the leading vehicle during operation. Like this, only the distance between leader and follower is optimized, and the waypoint list is much shorter. Finally, a highly dense track is generated without the need for manual specification of the waypoints for both MULE and Convoying.

[11] ELROB 2018: Transport–MULE: https://www.elrob.org/files/elrob2018/Transport_Mule_V3.pdf, access on 17.01.2022.

[12] ELROB 2018: Transport–Convoying: https://www.elrob.org/files/elrob2018/Transport_Convoy_V3.pdf, access on 17.01.2022.

Mule	Convoying
1. Track Specification	
Asynchronous Mode (Post-Processing):	**Synchronous Mode (Live-Processing):**
Record waypoints and save to file.	Leader sends waypoints to autonomous follower.
2. Waypoint Optimization	
Online or offline:	**Online:**
Start DP.	Start DP if $D_{lwp} >$ defined minimum.
3. Navigation	
Find grid cells in optimized track with Breshenham's, calculate EDT, apply w_f and w_m for cost valley.	

Figure 7.1 Cost valley processing pipeline for MULE and Convoying [328]. DP designates the Ramer-Douglas-Peucker approach.

Waypoint optimization processes the densely recorded track to facilitate a smooth navigation. The removal of too much waypoints from the recorded track can result in a decelerated path calculation, especially in difficult passages, and the driving behavior of the vehicle can be impaired. Waypoint lists that are too dense can lead to planning on shorter distances and thus to unsteady driving. Line simplification in MULE is directly performed on the densely recorded track. Convoying requires a live-processing of the waypoints. Thus, line simplification is only performed if a previously specified distance is exceeded to limit calculation operations. Two line simplification algorithms were evaluated for the cost valley approach: the well-known, global Ramer-Douglas-Peucker (RDP) approach [51, 222] operating on point-to-edge distance tolerances and a customized, local approach described in Section D.2. Other types of line simplification methods such as Nth Point, Opheim or perpendicular distance provide a more independent simplification approach. However, Shi and Cheung [255] state that the limitation of search areas results in worse mean and maximum distances. RDP line simplification clearly outperformed the customized, local approach in the conducted evaluation and was chosen for the proposed cost valley approach. RDP iteratively generates polygons to simplify the given curve. The polygon consists of a small number of vertices lying on the analyzed curve. The path hull is the maximum distance of the curve from the approximated polygon. It is specified as a fitting criterion and threshold in the approximation of the two-dimensional curve. The curve is represented by an ordered set C_{RDP} of $N + 1$ consecutive points. In the case of an open polygon P_{RDP},

the first and the last point are not equal. This is mostly true for waypoint optimization. P_{RDP} has N edges, and the corresponding points are interpreted as vertices $\mathbf{p}_i$. The aim is to find a simplified representation P'_{RDP} equal to a set of vertices C'_{RDP} with $N' < N$ edges and vertices $\mathbf{p}'_i$ inside ordered subsets $C_{k,\mathrm{RDP}}$ of C_{RDP}:

$$C_{k,\mathrm{RDP}} = \{\mathbf{p}_i, \mathbf{p}_{i+1}, ..., \mathbf{p}_j\}; \quad \mathbf{p}_i = \mathbf{p}'_{k-1}, \mathbf{p}_j = \mathbf{p}'_k. \tag{7.1}$$

The points $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ represent consecutive vertices in P_{RDP} with $i \in [0, N]$, and $\mathbf{p}'_{k-1}$ and $\mathbf{p}'_k$ consecutive vertices in P'_{RDP} with $k \in [1, N']$. C'_{RDP} divides the curve to be approximated into consecutive segments with the subsets $C_{k,\mathrm{RDP}}$ containing the vertices of k-th curve element [222]. This yields $\bigcup_{k\in[1,N']} C_{k,\mathrm{RDP}} = C_{\mathrm{RDP}}$ and

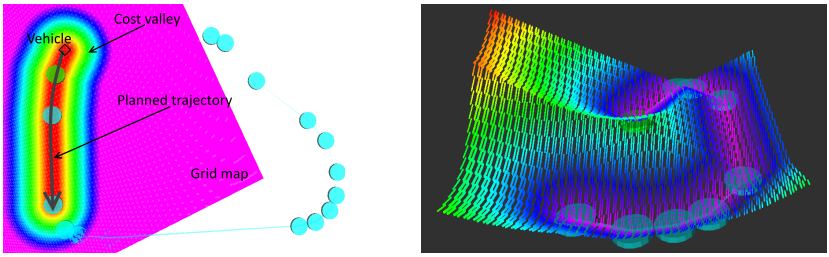
$$\left(\bigcup_{k\in[1,N'-1]} (C_{k,\mathrm{RDP}} \cap C_{k+1,\mathrm{RDP}}) \right) \cup \{\mathbf{p}_0\} \cup \{\mathbf{p}_N\} = C'_{\mathrm{RDP}} \tag{7.2}$$

for open polygons. If P_{RDP} with vertices $\mathbf{p}_i$, $i \in [0, N]$, is given in a set C_{RDP}, the simplified P'_{RDP} satisfies the criterion function

$$f(C_{k,\mathrm{RDP}}) = \max(L_1(\mathbf{p}_i \langle \overline{\mathbf{p}'_{k-1}\mathbf{p}'_k} \rangle)) \leq \varepsilon. \tag{7.3}$$

The constant threshold parameter for the size of the path hull is defined by $\mathbf{p}_i \in C_{k,\mathrm{RDP}}$, and $\overline{\mathbf{p}'_{k-1}\mathbf{p}'_k}$ denotes the line segment from $\mathbf{p}'_{k-1}$ to $\mathbf{p}_k$.

Navigation uses the optimized waypoint list to generate the cost valley – the core of the method presented. The accuracy required while keeping the predefined track can be specified with two parameters: the free width w_f and the maximum width w_m. Here, w_f defines the valley width around the optimized track navigable without additional costs, determines the valley bottom and ensures track-keeping as tightly as required. The maximum width w_m sets the permitted area for evasive maneuvers and thus the total width of the valley. Finally, the cost valley is calculated as a grid map layer of the input map with obstacle and navigability information, as defined by w_f and w_m. The cost valley is completely cost-based and an absolute potential is not calculated in contrast to potential field approaches, as described in [306]. The direct path between the waypoints constitutes the valley center and consequently lowest costs. Problems with local minima that may occur in potential field approaches were not encountered.



(a) Cost valley on TULF. (b) EDT for cost valley.

Figure 7.2 Cost valley evaluation on TULF: Figure (a) depicts the real-time experimental validation on the TULF platform: the track is optimized at starting time, $w_f = 3$ m (edge of red area), $w_m = 30$ m (edge of dark blue area). The planned trajectory is emphasized in dark gray; (b) shows the EDT with optimized waypoints (blue) and the direct path between waypoints (pink) [328].

Joy [142] states that the Breshenham's Algorithm provides a fast and accurate determination of the cells a path passes inside a grid map. Euclidean Distance Transform (EDT) is a fast, popular, and well-suited method for distance calculation to the nearest obstacle in a grid structure commonly used for obstacle avoidance in planning, as described in [185, 215]. EDT is used within the cost valley approach to explore the distances to the closest grid cells containing a higher value than the defined minimum [185]. The first step in the calculation of the cost valley is to ensure that all waypoints lie inside the input grid. If this is not the case, the input grid is enlarged to contain all required waypoints. Next, the cells passed by the optimized track are identified using Breshenham's algorithm in a fast implementation, as proposed in [142]. All cells that contain track elements are defined as obstacles from the view of the EDT [185, 215]. Here, EDT yields the distances of the track elements from other cells and the EDT layer shown in Figure 7.2 is temporarily integrated inside the grid structure. The utilization of the fast and well-proven EDT implementation of [185] ensured real-time capability. The EDT result is normalized to [0, 1] to correspond to obstacle costs of the input grid structure. Finally, the cost valley is created by applying w_f and w_m, as illustrated in Figure 7.2 and Figure 7.3.

The proposed valley approach also facilitates autonomous turning when the last waypoint is reached. This is made possible by a higher w_f

around the last waypoint defined according to the vehicle geometry. Furthermore, the valley approach also allows an autonomous return to the valley after an intervention by the human safety driver (see Figure D.6).

The cost valley planning constraint is evaluated according to the planning benchmarking of Cohen et al. [41]: computation times, path length, smoothness of plans, clearance, and success rate. The computation time requirements are met if real-time capability is achieved. Contrasting planning in structured environments, the path length is less important than the navigability of the unstructured, rough terrain. Sufficient clearance is achieved if all obstacles can be avoided. The success rate and smoothness are measured qualitatively by a successful completion of the test scenarios. Special focus in off-road driving is placed on critical passages and curves that could not be passed successfully with classic waypoint navigation.

7.6.2 Proof of Concept: Cost Valley

The cost valley approach was experimentally validated on IOSB.amp Q1 and TULF – two off-road vehicles with different kinematics, weight, and size for defense applications. For the evaluation on the TULF platform, the speed was limited to 25 $\frac{\text{km}}{\text{h}}$ to allow potential safety interventions. The experiments were carried out in mainly unstructured environment containing trees, bushes, metal structures, and dirt roads. A small part of the test track crossed between buildings on a paved road to examine the performance in semi-urban environment. The experimental results provided in this thesis focus on the MULE behavior due to its higher complexity with longer predetermined tracks in waypoint optimization and navigation.

Track specification was executed at 10 Hz in the Universal Transverse Mercator coordinate system for global referencing. The MULE teach-in acquired more than 10,000 waypoints over 1 km.

Waypoint optimization with RDP achieved a better performance than the analyzed local approach due to its global character. Radius selection for the path hull on the basis of the vehicle geometry and terrain to be navigated proved useful: the radius has to be set large enough for smooth planning and small enough to reconstruct narrow passages and curves sufficiently accurate. The test scenarios covered up to 2.5 km in MULE

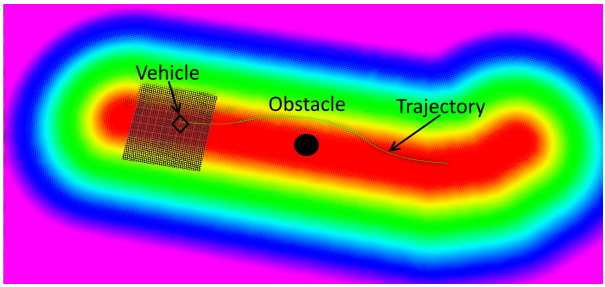


Figure 7.3 TULF ($w_f = 3\,\text{m}$, $w_m = 30\,\text{m}$): Obstacle avoidance. The vehicle exits $w_f = 3\,\text{m}$ for an avoidance maneuver around a barrel blocking the way (black circle) and returns to the cost valley center [328].

and 1 km in Convoying. Empirical studies showed that a radius of 0.8 m for the TULF platform and 0.5 m for IOSB.amp Q1 achieved the most accurate and successful driving performance results.

Experimental evaluation further demonstrated that the MULE and Convoying behaviors achieved a superior performance with the integration of the cost valley constraint. Figure 7.2 shows all waypoints of a chosen experimental MULE driving scenario on a track approximately 1 km in length. The valley was calculated over the next waypoints and not over all waypoints to maintain real-time capability. Figure 7.3 illustrates the obstacle avoidance around a barrel and the track was followed as accurately as possible in MULE. Some narrow passages in the testing environment were not passed with the TULF platform in autonomous operation with classic waypoint navigation. These narrow passages could be passed in autonomous operation with an integration of the proposed cost valley. The predefined track was followed accurately, and obstacles were avoided. The valley limited the space of possible paths, and less calculation time was needed in the planning step. With a parallel processing on multiple CPU cores, the additional generation of the cost valley increased the total runtime only slightly. This maintained the real-time capability of the testing platforms that rely on the ATB [60] for their autonomous operation.

8 Conclusion

8.1 Summary

Autonomous vehicles need to perceive and "understand" their environment to interact with it in a controlled and safe way. In contrast to structured environments such as production buildings or urban surroundings, perception of unstructured environments constituting the typical operation environment for autonomous off-road vehicles is greatly underrepresented in research. Unstructured environments are challenging due to difficult-to-separate objects and an often inhomogeneous structure of their natural and grown geometries. To this end, this doctoral thesis presents novel and customized classic and machine learning perception methods for unstructured environments and combines them within a holistic, three-level pipeline for autonomous off-road vehicles: low-level, mid-level, and high-level perception. The classic and ML perception methods proposed in this work complement each other. The accompanying validation methods proposed for each level facilitate environment perception for off-road vehicles with a better understanding of their unstructured operation environments – especially for heavy construction machinery in the remediation of landfill sites and unmanned land systems for off-road transport in defense.

All proposed perception and validation methods were designed as individual modules within the proposed three-level pipeline. Hence, their flexible combination allows different pipeline designs for a variety of off-road vehicles and use cases depending on respective requirements and constraints, such as a sufficient volume of training data. Here, the combination of classic and ML methods with accompanying, classic validation methods paves the way for a comprehensible and trustworthy perception of unstructured environments in these critical application scenarios.

The proposed low-level perception methods comprise a novel confidence analysis process for raw sensor data and registration methods for visual multi-sensor systems. The proposed confidence estimation for 2D images and 3D point clouds permits loosely coupled validation or provides input to a tightly coupled validation within a confidence-based data fusion in mid-level perception. The registration approaches presented do not rely on calibration targets but only on the structure of the surroundings. A semi-automatic registration approach facilitates the registration of multiple, similar-source LiDAR sensors, while the *UCSR* registration framework for cross-source sensor data combines the customized *cc23*, *cnn23*, and *graph33* registration methods for sensor data from unstructured environments and provides confidence-based registration results for sensors with differing measurement principles.

For mid-level perception, this thesis presents two novel stereo image disparity estimation methods specially customized for unstructured environments: the classic CCRADAR method extended for hyperspectral images that only requires a minimal volume of testing data and the *UEM-CNN* method that relies on convolutional neural networks (CNNs) for disparity estimation. Novel disparity estimation error measures for unstructured environments and the *SET* evaluation toolbox for 3D reconstruction results from stereo image disparity estimation provide a loosely coupled validation for the disparity estimation methods presented as well as for other stereo image disparity estimation methods.

Depending on the capabilities and the deployment scenario of autonomous off-road vehicles, a highly detailed navigability analysis, object detection, and obstacle avoidance are required, which implies a semantic 3D segmentation within the high-level perception of unstructured environments. Here, the limited data availability becomes especially evident as evaluation and test data cannot always be provided for critical applications, such as the remediation of landfill sites. Hence, this thesis has analyzed the domain transfer of state-of-the-art semantic 3D segmentation methods and presents recommendations for an enhanced domain transfer performance as well as the customized *IC-ACC* training approach to reduce the required amount of training data. In addition, high-level perception proposed in this thesis discusses the explanation of predictions from ML methods so that human operators are able to

understand and judge their performance. This is especially important for data-driven ML methods as they can learn erroneous behavior from erroneous training data. The understanding and comprehensibility of their predictions constitutes a crucial step towards trustworthy and reliable AI methods. The pre-modeling *IC-ACC* method presented provides a generalized, exploratory data analysis for ANN methods in image processing. Here, information content (*IC*) and accuracy (*ACC*) are examined to filter detrimental data and to compose efficient datasets that contribute to the reduction of the data amount required to train neural networks. The X^3Seg approach facilitates a post-modeling, model-agnostic explanation of semantic 3D segmentation results in unstructured environments. It contributes to the understanding of class predictions in the semantic 3D segmentation by highlighting descriptive, model-agnostic correlations between in- and output data.

The presented proofs-of-concept with data from unstructured environments demonstrate the applicability of all proposed perception and validation methods and show the suitability of the scientific contributions explained previously in the two main application scenarios in this thesis – decontamination of hazardous environments and off-road transport in defense. To summarize, the perception–validation pipeline proposed within this thesis facilitates a flexible combination of perception solutions for autonomous off-road vehicles and has been successfully implemented on technology demonstrators such as the two IOSB.Alice and IOSB.BoB excavators and the TULF off-road truck. The combination of the complementary classic and ML perception methods with the presented validation methods ensures an accurate and reliable perception of unstructured environments for autonomous off-road vehicles.

8.2 Outlook

This doctoral thesis presents perception solutions for unstructured environments. Nevertheless, this thesis cannot address all facets of perception required in any single case.

The accompanying classic perception and validation methods already validate the proposed ML perception methods. However, future work for all ML methods should tackle the issue of testing against adversarial

attacks, as discussed for optical flow estimation by Ranjan et al. [224]. In addition, synthetic training could facilitate the evaluation of safety-critical corner cases where real-world data is hardly available, such as from natural disasters.

An extension of the proposed low-level confidence analysis towards merging multiple 3D measurements points into a single 3D point with a high confidence or also to increase the confidence of each respective 3D measurement would contribute an even tighter coupling of cross-source sensor measurements. This could be achieved by examining if 2D image pixels and 3D points within the common FoV of two or more sensors are observed by all or only by some of the applicable sensors. NIR cameras and LiDAR sensors often operate within the same spectral range and the fusion of their perception results provides another option for 2D–3D registration, in addition to *cc23* and *cnn23*. Hence, extending the *UCSR* registration framework with an NIR-based registration method would benefit overall registration accuracy.

For high-level perception, the extension of the proposed semantic segmentation from "single-shot" 3D point clouds to instance segmentation, the filtering of temporally inconsistent semantic class predictions, and the tracking of identified, countable objects for knowledge on dynamic objects with motion profiles would increase the robustness of the interpretation results. Especially in defense, this would extend the application scenarios of off-road vehicles for tasks exceeding off-road transport.

Future work in domain transfer should include further analysis of the model performance in data collected with more sensor configurations and LiDAR sensors with a higher or lower number of diodes as well as the integration of post-modeling XAI methods such as X^3Seg. The additional development of a coarse class structure for navigability analysis and a fine-grained class structure for exploration and manipulation could further benefit the domain transfer performance in CNN-based semantic 3D segmentation.

Appendix

A Low-Level Perception

A.1 Sensor Data Confidence

The confidence assessment detailed in Section 4.1 analyzes the accuracy (*ACC*) of raw sensor data and complements the *IC-ACC* approach presented in Section 6.2.1, as illustrated in Figure A.1. In the context of confidence assessment and reliability analysis, Hughes [128] discusses four key questions that are addressed by the proposed confidence estimation approach as follows:

1. "Where in the system is the sensor performance considered?"[128, p. 2]: the sensor performance is directly estimated on raw sensor data and consequently considered in a tightly-coupled manner to exclude detrimental sensor data prior to its input to perception methods.
2. "Upon what is the uncertainty/reliability judged?"[128, p. 2]: different confidence/accuracy measures are proposed to estimate the reliability of the captured raw sensor data, and high confidence indicates high reliability and low uncertainty.
3. "How is the uncertainty/reliability measured?"[128, p. 2]: the confidence/reliability is measured on the basis of local and global, per pixel/point (*PPC*) and per sensor/scene (*PSC*) criteria.
4. "And what is the effect of its judgment?"[128, p. 2]: sensor data with a low confidence is expected to be detrimental and is consequently not considered in the subsequent processing steps.

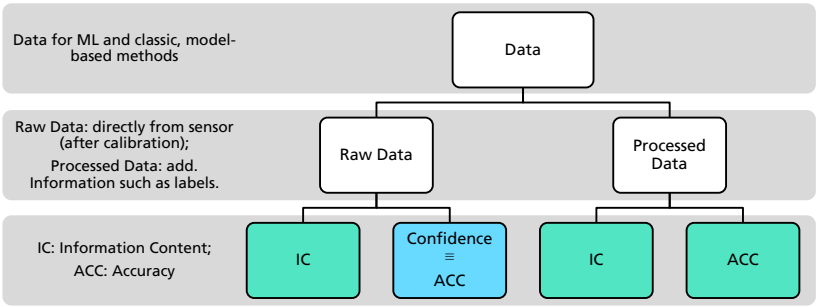


Figure A.1 Confidence (blue), accuracy, and information content assessment proposed within this thesis. The *IC* of raw data and the *IC* and *ACC* of processed sensor data is colored green and analyzed with the *IC-ACC* approach in Section 6.2.1.

A.2 3D–3D Similar-Source Registration

The discussed analysis of one-to-many correspondences in the similar-source 3D–3D registration with GICP shows that the elimination of the target points with more than 50 one-to-many reduces the e_{fs} from $2.167\,\mathrm{m}^2$ to $0.060\,\mathrm{m}^2$, while the allowance of up to 100 one-to-many per target point did not further decrease the e_{fs}. Here, Table A.1 quantitatively illustrates the influence of the analysis of one-to-many correspondences in the similar-source 3D–3D registration with GICP.

Number of one-to-many correspondences	Ratio
0 – 20	0.72
21 – 100	0.12
101 – 200	0.09
201 – 300	0.00
301 – 400	0.03
401 – 600	0.04

Table A.1 Distribution of the number of one-to-many correspondences in similar-source registration of the left (source) $\mathcal{L}$ to the right (target) $\mathcal{L}$ from Velodyne VLP-16 sensors mounted to IOSB.BoB in partially unstructured outdoor environments.

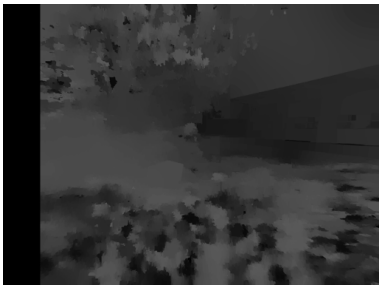
A.3 *UCSR*: Confidence-Based Registration Framework for Cross-Source Sensor Data

A.3.1 *cc23*: Classic 2D–3D Cross-Source Registration

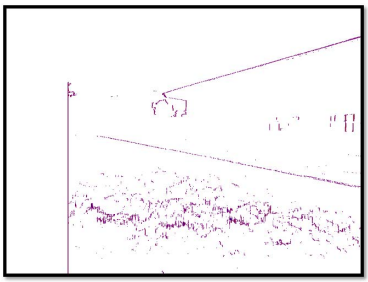
This appendix contains additional material to Section 4.3.2 and demonstrates the classic 2.5D–3D cross-source registration with *cc23* exploiting 2.5D disparity maps instead of 2D RGB images. Disparity images were generated using SGBM, as described in Section B.1.1. The expected advantage of using disparity maps in intensity feature extraction is their invariance to image contrast, while their disadvantage are the rather smoother transitions between the intensity values representing estimated disparities in comparison to RGB images. Experimental evaluation showed that the extraction of intensity features is less fruitful, and this approach did not yield sufficiently accurate and unsatisfactory registration results in unstructured environments, where smooth depth transitions and difficult-to-separate objects dominate. Two exemplary input disparity maps, one for structured and one for unstructured environment, illustrate the problem of insufficient offset in depth transitions in unstructured environments. Furthermore, errors in disparity estimation add up to potential inaccuracies in registration, which can further decrease the achievable registration accuracy. The differences between data from structured and unstructured environments as well as errors occurring in disparity estimation are emphasized in Figure A.2, which compares

(a) Original 2D RGB image (structured).

(b) Original 2D RGB image.

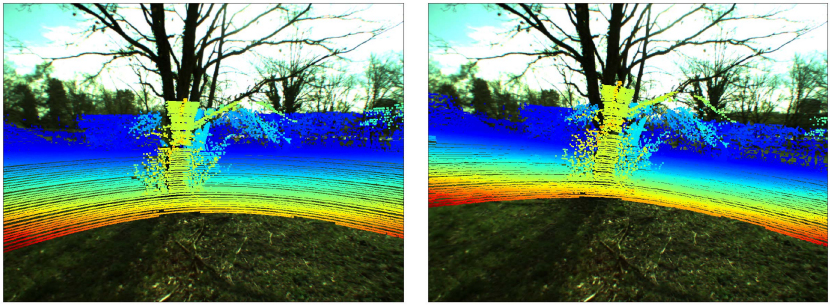


(c) SGBM disparity image of (c).



(d) 2D int. features of disparity image (d).

Figure A.2 Left reference images of the *IOSB-Reg* dataset ((a), (b)) with exemplary stereo image disparity estimation result with SGBM ($D_{1,\max} = 255$, image (c)). Clear depth offsets are visible for structured elements, whereas unstructured image areas generally present rather smooth disparity and hence depth transitions. Images ©Fraunhofer IOSB.



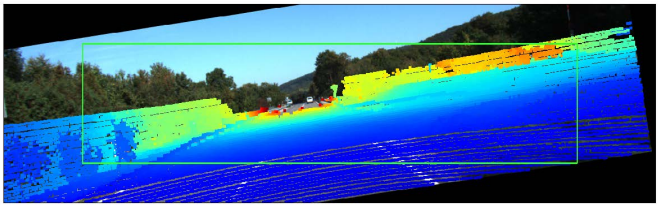
(a) Visual overlay validates ground truth. (b) Decalibrated input.

Figure A.3 Visual overlay for ground truth validation of *IOSB-Reg* and decalibrated input to *cnn23*. Images ©Fraunhofer IOSB.

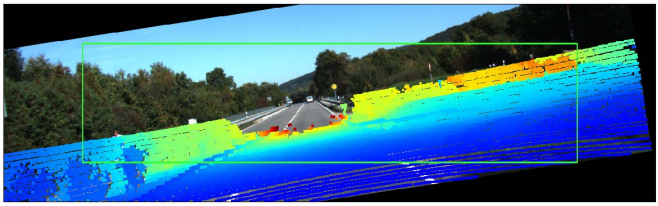
the disparity estimation in structured and unstructured environments used as an input to the *cc23* variant extracting 2D intensity features from disparity maps. For this purpose, Figure A.2 shows an exemplary SGBM disparity image generated from the *IOSB-Reg* dataset with structured as well as unstructured image areas. Figure A.2(d) exemplarily illustrates partially erroneous stereo image disparity estimation results of SGBM introducing additional difficulties in 3D–3D registration and also in 2.5D cross-source registration with the proposed *cc23* variant in unstructured environments like Figure A.2(b).

A.3.2 *cnn23*: 2D–3D Cross-Source Registration with Neural Networks

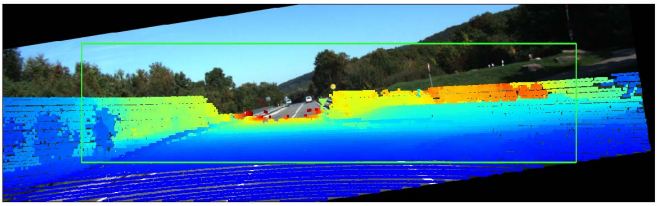
This appendix contains additional material to Section 4.3.3. Figure A.3(a) validates the ground truth for *IOSB-Reg* capture by visual overlay of the 2D range image projection of a 3D LiDAR point cloud onto a 2D RGB image in unstructured environment prior to data augmentation for *cnn23*, and Figure A.3(b) shows an exemplary, small decalibration of the RGB image and the 2D projected range image. Figure A.4 and Figure 4.11 compare the *cnn23* registration results with data augmentation according to Schneider et al. [246] to an RGB image to range image registration with *cnn23* trained with the enhanced augmentation (*cnn23-U*) proposed in



(a) Ground truth.



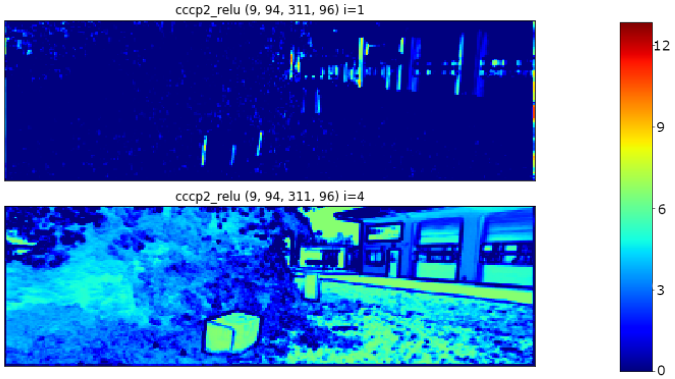
(b) Decalibration.



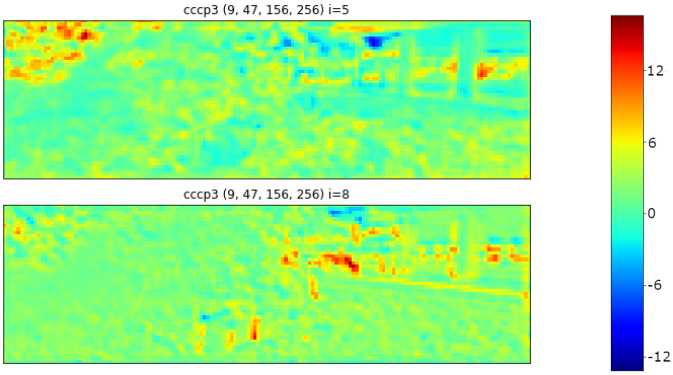
(c) Erroneous registration due to over-fitting.

Figure A.4 Over-fitting on KITTI data for 2D–3D registration with *cnn23* trained as proposed in [246]. Without an enhanced augmentation (*cnn23-U*), *cnn23-N* tended to adjust the LiDAR depth image to the RGB image by a horizontal orientation of the LiDAR depth image with the highest depth values in the approximate image center as shown in image (c). Hence, neither rotation nor translation was determined correctly without the proposed uniform data augmentation. Images © Fraunhofer IOSB.

(a) 2D RGB input to *cnn23*.

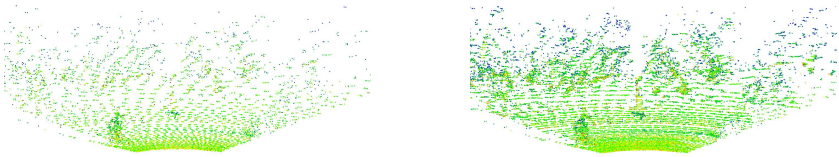


(b) 1st layer of *cnn23* after ReLU activation and prior to max. pooling.



(c) ReLU activations on layer 10 prior to ReLU activation and max. pooling.

Figure A.5 Image (b) and (c) show selected filter activations for *cnn23* feature extraction on RGB image (a) to register image and point cloud for an exemplary *IOSB-Reg* scene. Preference for structured elements is indicated by higher weightings for the pixels/activations representing trunk or ground floor.



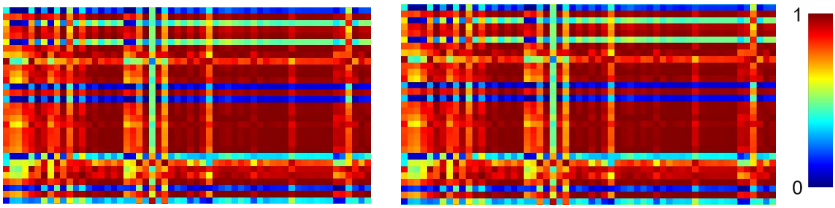
(a) Original LiDAR cloud. (b) Dense LiDAR cloud.

Figure A.6 Static condensation of a 3D LiDAR cloud from a Velodyne HDL-64E sensor: the original LiDAR cloud (a) from one sensor rotation with 10 Hz is condensated by overlaying and aggregating the five previous and the five subsequently captured LiDAR clouds (b) within a time interval of 1.1 s.

this thesis. Figure A.5(b) shows filter activation weights prior to ReLU activation and maximum pooling, while Figure A.5(c) depicts the filter activations after ReLU activation but prior to maximum pooling for the RGB feature extraction on Figure A.5(a). Both filter activations highlight the preference for structured elements in the *cnn23* registration method that only relies on the structure of the surroundings. Figure A.5 illustrates filter activations prior to and after ReLU activation and maximum pooling from different feature extraction layers inside the *cnn23* architecture. Natural and grown structures, such as grass or mounds in unstructured environments, can contribute to the feature extraction process from the 2D image but most *cnn23* filter activations for feature extraction show a preference for structured elements compared to unstructured elements with inhomogeneous structure and primarily dominated by similar textures and difficult-to-separate objects.

A.3.3 *graph33*: Classic 3D–3D Cross-Source Registration

This appendix contains additional material to Section 4.3.4. Figure A.6 visualizes the densification of LiDAR clouds in a static scene increasing the number of 3D points for correspondence matching with the stereo camera point cloud in *graph33* registration. Figure A.7 shows the affinity matrices of two additional descriptors added to *graph33*: distance to origin ($\mathbf{A}_R$) and normal orientations to the voxel centroid ($\mathbf{A}_N$).



(b) Angle to z ($\mathbf{A}_Z$). (b) Normal of centroid ($\mathbf{A}_N$).

Figure A.7 Affinity matrices of additional $\mathbf{A}_Z$ and $\mathbf{A}_N$ descriptors in *graph33* (supplementary to the descriptors depicted in Figure 4.17). The vertical axis represents the source nodes, the horizontal axis the target nodes. The color scaling highlights the affinity values between source and target nodes where zero indicates no affinity between source and target nodes and a value of one – colored in red – describes complete affinity between two nodes.

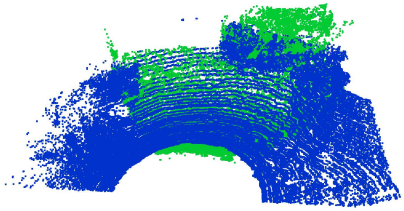
A.3.4 *dsm33*: 3D–3D Cross-Source Registration with Neural Networks

The *dsm33* method facilitates the 3D–3D registration of cross-source data with CNNs, as described in Section 4.3.5. The proof of concept described hereinafter was primarily conducted by Leitritz [335] under supervision of this thesis' author.

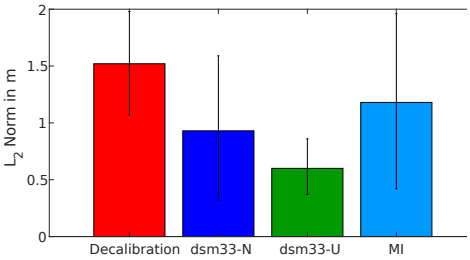
A.3.4.1 Proof of Concept: *dsm33*

Different hyper-parameter settings for network training were evaluated for *dsm33* in [335] as the perception data from unstructured environments notably differs from the medical data targeted in [109]: input size of the 3D images, number of filters per layer, dropout, and maximum decalibration, and only one hyper-parameter setting was changed in each training process. The learned similarity metric's performance and the efficacy of the differential optimization method for *dsm33* were evaluated. Furthermore, the capture range of the registration method – the maximally tolerated decalibration to achieve a valid registration result – was analyzed, and all evaluation steps were conducted with a special focus on data from unstructured environments. 146 3D point clouds of the *IOSB-Reg* dataset were used to train *dsm33*, and 15 randomly selected

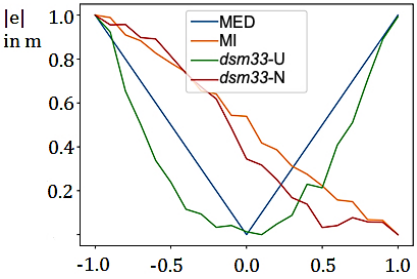
(a) 2D RGB reference image.



(b) *dsm33-U* registration result.



(c) *dsm33-U* registration accuracy.



(d) Translational decalibration in *y* axis.

Figure A.8 Registration results with *dsm33-U* and differential evolution for a level M decalibration with a **t** offset of [-0.49 m; 0.18 m; 0.089 m] and an **r** offset of [7.76°; 5.54°; –3.46°]. (d) compares *dsm33-U*, *dsm33-N*, and mutual information (MI) as similarity metrics to estimate the translational decalibration in direction of the *y* axis on *IOSB-Reg* with the associated normalized mean Euclidean error ($|e|$ showing the ground decalibration (MED) that shall be estimated by *dsm33*. Here, only *dsm33-U* provided a correct estimation of the translational decalibration which highlights the importance to augment the training data with uniformly distributed decalibrations using the presented enhanced augmentation (*dsm33-U*). Images (a) and (b) ©Fraunhofer IOSB.

3D cloud pairs were reserved for validation and 15 for testing. As the model size of *dsm33* is smaller than the model of [109], the *IOSB-Reg* dataset with 146 data points provided a sufficient amount of training data compared to 539 data points that achieved representative results in [109].

The dense stereo camera clouds $\mathcal{S}$ were generated with the classic SGBM approach [118] according to B.1.1 and selected as the source, while $\mathcal{L}$ was chosen as the registration target. SGBM was applied instead of the *UEM-CNN* architectures as SGBM provided a higher prediction density. The lower disparity estimation accuracy and higher prediction density of SGBM were favorable to evaluate *dsm33* as it aims at a stable registration even with higher depth estimation errors. ML methods trained for less accurate input data are naturally capable of working with more accurate data, but the reverse is seldom possible. In contrast to [109], *dsm33* works with equal dimensions of the input data in 3D space. This facilitated an increased augmentation dimension of *dsm33*, and it was sufficient to use augmented data from unstructured environments solely.

Haskins et al. [109] parameterized DINO with a population size of 5 and terminated it after 13 generations, and the differential evolution results were subject to subsequent BFGS optimization. The *dsm33* method utilizes a truncated differential evolution with a population size of 15 agents and termination after 100 generations with a coarse initial guess in the range of the decalibration levels S to L. All other parameter selections followed the suitable and well-tested recommendations in SciPy[1] and Open3D [315] for ICP, voxelization of the point clouds, and truncated differential evolution.

The training data for the final configuration of *dsm33* was augmented for a uniform distribution of level M decalibrations (see Section 3.11). The memory requirements of the model were adjusted for training on smaller computer architectures by smaller voxelized 3D images with $d_{vx} \times d_{vy} \times d_{vz} = 256 \times 256 \times 32$ to achieve reasonable training times on the available hardware. The number of 3D convolutional layers and the number of neurons in the fully connected layer were reduced by

[1] SciPy: Open source scientific tools for Python, http://www.scipy.org/, access on 03.11.2021.

50 % compared to the $512 \times 512 \times 32$ voxelization proposed in [109]. The batch size was set to 1 to prevent batch normalization, which reduced the number of network parameters by 92 % from 285,193,745 to 22,564,873, and *dsm33* trained for 438,000 iterations (79 hours on RTX 2080S).

Figure A.8 compares the registration accuracy of *dsm33* with differential evolution optimization to the use of MI with DINO and classic ICP. The resulting L_{dsm33} measures the achieved registration accuracy of source and target cloud similar to the GICP e_{fs} score and amounted to $L_{\text{dsm33}} = 0.356\,\text{m}$ on the validation set of *dsm33*. The training error was approximately ten times lower with $L_{\text{dsm33}} = 0.039\,\text{m}$ which can indicate over-fitting due to a limited amount of training data, and the extension of the training dataset with additionally captured $\mathcal{S}$ and $\mathcal{L}$ data can help to further analyze and also prevent over-fitting.

The maximum number of differential evolution iterations was set to 100, and convergence was mostly reached at around 60 iterations. The inference in *dsm33* was conducted about 10,000 times for each complete *dsm33* registration. Initially, the L_2 values in differential evolution have a high variation as random parameter populations are generated. Each subsequent generation utilizes the best solution of the previous generation as a starting point which yielded a lower variation of the Euclidean distance during the optimization process. In contrast to other CNN-based registration methods, an accurate registration result in *dsm33* is not necessarily indicated by a low validation error as long as a high decalibration results in a high decalibration estimate and vice versa. The L_2 norm and the F norm are applied to measure the registration performance of *dsm33*, and Figure A.8 shows the registration accuracies in terms of L_2 for the 3D–3D registration of the selected 3D validation dataset for *dsm33-U*, *dsm33-N*, and MI with differential evolution optimization. Here, *dsm33-U* performed best with the lowest L_2 norm and the lowest $\sigma(\sqrt{L_2})$. Figure A.8 shows an exemplary registration result on the *IOSB-Reg* dataset with *dsm33-U* and the subsequent differential evolution.

A.3.5 Comparison of Individual Cross-Source Registration Methods

Table A.2 illustrates the registration performance of *dsm33* variant that achieved the most accurate registration results (*dsm33-U*) for a level L decalibration in comparison with *cnn23* and the classic ICP method.

Measure	*cnn23*	*dsm33-U*	ICP	Decal.
Registration accuracy in terms of L_2.				
$\mu(L_2)$	1.895	0.916	2.281	3.267
$\sigma(L_2)$	1.115	0.376	0.990	0.876
$\min(L_2)$	0.291	0.313	0.748	1.345
$\max(L_2)$	4.227	1.693	4.345	4.983
Registration accuracy in terms of F.				
$\mu(F)$	0.496	0.664	1.478	0.284
$\sigma(F)$	0.262	0.287	0.820	0.070
$\min(F)$	0.119	0.261	0.531	0.155
$\max(F)$	0.991	1.453	3.233	0.402
Registration accuracy for selected DoFs.				
$\overline{\Delta \mathbf{t}}$ along z axis	0.146	0.097	0.807	0.427
$\overline{\Delta \mathbf{r}}$ around x axis	3.107	0.905	3.770	10.379
$\overline{\Delta \mathbf{r}}$ around z axis	6.647	2.558	6.712	8.765
$\overline{\Delta \mathbf{t}}$	0.201	0.310	0.739	0.437
$\overline{\Delta \mathbf{r}}$	5.166	1.338	4.167	10.067

L_2 is given in m. Rotations are given in degrees.

Table A.2 Registration results of *cnn23*, *dsm33-U*, and ICP for level L decalibrations on the *IOSB-Reg* dataset.

A.4 2D Image Fusion and Visual SLAM with RGB–NIR and HDR Images

This appendix contains additional material to Section 4.4 and demonstrates the application of the achieved 2D image fusion results in visual SLAM.

Visual SLAM is discussed here, as mapping and localization build on the environment perception results, generate a map, and determine a

position inside a predefined coordinate system. If this coordinate system is related to a previously captured map, localization can be interpreted a registration problem in a wider sense. SLAM approaches align the acquired sensor data to the map and hereby determine the position in relation to the present map such as in SLAM-driven point cloud registration [150]. For autonomous vehicles, SLAM and the applied registration methods need to work in real-time. Consequently, visual SLAM approaches such as scan matching in 3D on the basis of 3D LiDAR data [59] rely on direct similar-source registration methods. Thereby, "single-shot" LiDAR clouds [59] are registered to the present map that is typically saved as a 3D point cloud to prevent information loss. In 3D scan matching, the relative motion between two consecutive scans is estimated and contributes to a multi-sensor SLAM solution similar to the measurements of odometry sensors. Emter and Petereit [59] utilize the GICP method to register consecutive LiDAR scans and to determine the LiDAR odometry measurement as visual 3D SLAM result. As an extension to visual SLAM, feature-based stereo image disparity estimation can be combined with LiDAR measurements as proposed by Gräter et al. [96]. The authors target one of the main drawbacks of stereo image disparity estimation, its dependency on accurate extrinsic camera calibration, by extracting the depth information from LiDAR as well as motion estimation from keyframe-based bundle adjustment. Landmarks are weighted according to their semantic classification and low weightings for unstructured vegetation landmarks in contrast to high weightings for structured landmarks highlight the difficulties encountered in the perception of unstructured environments.

In general, visual odometry estimates the relative motion between the capture of two consecutive images or point clouds on the basis of the feature extraction and matching. On the basis of these visual odometry results, visual SLAM tracks the feature points through consecutive key frames to generate a map and the determined, relative odometry measurements provide the localization for the integration of the captured images in the generated map. Contrasting 3D SLAM, visual 2D SLAM mainly relies on 2D camera images and visual 2D SLAM methods such as ORB-SLAM [200] and ORB-SLAM2 [201] constitute registration scenarios with visual input information. In general, feature-based visual SLAM relies

on the extraction of 2D features, such as SIFT, SURF, or ORB described in Section 2.3.2, and conducts a 2D registration process via a sparse feature representation of the original 2D images. While ORB-SLAM provides a visual SLAM solution for monocular cameras and derives the relative motion from two consecutive monocular images, ORB-SLAM2 works with monocular, stereo, and RGB-D data. In the Visual Odometry/SLAM Evaluation 2012[2], ORB-SLAM2 ranged 62 of 141 with less than 0.1 s of computation [82]. ORB-SLAM3 by Campos et al. [31] furthermore provides visual, visual-inertial, and multi-map SLAM for monocular, stereo and RGB-D cameras with pinhole or fish-eye lens models. Campos et al. [31] state that in contrast to previous feature-based approaches, the Maximum-a-Posterior estimation in ORB-SLAM3 yields robust operation in indoor and outdoor environments with a notably higher accuracy as the ability to process multiple maps facilitates the generation of a new map when the current localization is lost. As an alternative approach, Albrecht and Heide [318, 320] present the integration of inertial measurements and the direct usage of HDR images generated by MEF with an evaluations in person indoor navigation that increases robustness of ORB-SLAM2 as ORB-SLAM3 [31] was not yet published at this point in time.

To conclude, the utilization of multi-spectral and HDR cameras can help to improve the reliability, accuracy, and success rate for visual 2D SLAM, especially in the transition between indoor and outdoor environments and naturally also in unstructured outdoor environments. HDR cameras can also circumvent the challenge of illumination changes in the transition from indoor to potentially brighter outdoor environments or when facing windows with a higher brightness by capturing two images per activation and per camera: one image with a low, and one image with a high exposure time. Prism-mounted HDR cameras yield those two images from the same viewpoint and with identical camera intrinsics.

Multi-spectral prism cameras provide simultaneously captured RGB and NIR images with exactly the same FoV, which allows the direct com-

[2] Visual Odometry / SLAM Evaluation 2012: http://www.cvlibs.net/datasets/kitti/eval_odometry.php, access on 06.12.2021.

parison of visual SLAM results on RGB, NIR, and fused RGB–NIR images. Especially in difficult lighting conditions, multi-layered image representation in visual SLAM can improve the performance of ORB-SLAM2 as demonstrated by Wang et al. [287] and also as discussed in Section 2.3.2. The 2D image approach proposed in this thesis pursues a similar strategy and combines image information from different spectral channels into one image with MEF. MEF as described in Section 4.4 was utilized to fuse RGB and NIR images and the RGB, NIR, and fused RGB–NIR images constituted the input for the visual SLAM with the ORB-SLAM2 algorithm [201] to examine a potential increase in reliability, accuracy, and success rate. ORB-SLAM2 was selected to demonstrate the benefit of the proposed 2D image fusion approach in this thesis as it was open-source available and experimentally well-validated. The target environment to examine and compare the performance of visual 2D SLAM with RGB, NIR, and fused RGB–NIR images was chosen as a combination of structured and semi-structured environments. This analysis of visual SLAM methods in indoor and rather structured outdoor environments highlights the generalization possibilities of the proposed perception methods for other types of environments and the visual SLAM results on the basis of ORB-SLAM2 constitute a proof of concept for visual SLAM with fused RGB–NIR images in primarily structured environments.

The RGB and NIR images for the image fusion in Section 4.4.2 and for the subsequent input into the visual SLAM approach were also captured with the sensor setup of the *IOSB-Reg* dataset (see Section 7.4) and three SLAM scenarios are evaluated: the first indoor scenario covered a short, straight path in direction of the window shown in Figure 4.21, a second loop-closure scenario analyzed transitions from outdoor to indoor and back again, while the third scenario was the longest with turns around several corners and included the same transition from outdoor to indoor as the loop-closure scenario.

The conducted visual SLAM analysis showed that NIR images alone did not provide sufficient information for feature-based visual SLAM, neither in indoor nor in outdoor environments.

The 2D path for the second scenario with RGB and RGB–NIR stereo images includes loop-closure. However, loop-closure was not detected for both RGB and RGB–NIR, but the starting point was reached without loss

of localization in contrast to the utilization of HDR images [320]. Thus, the transition from outdoor to indoor as well as from indoor to outdoor was achieved without losing the visual SLAM localization presented. Concluding, fused RGB–NIR images proved useful for visual SLAM in unstructured outdoor environments due to the imaging characteristics of the NIR spectrum and the achieved results. The successful application of fused RGB–NIR images in feature-based, visual SLAM also validated the utilization of the MEF method to generate RGB–NIR images proposed in Section 4.4.

Hence, the fusion of the two synchronized image streams constitute a suitable input for feature-based visual 2D SLAM ORB-SLAM2 [201, 320]. The utilization of several HDR fusion methods for grayscale HDR cameras as an input to visual 2D SLAM is examined by Albrecht and Heide [320]. Evaluation was conducted in person indoor navigation using a person carried HDR stereo camera system in realistic indoor application scenarios. The authors [320] show that the approach is capable to handle changing illumination conditions and can increase the reliability of localization and mapping in visual 2D SLAM.

B Mid-Level Perception

B.1 Stereo Image Disparity Estimation

B.1.1 Disparity Estimation from Stereo Images

Ray-tracing or the perspective transformation matrix $\mathbf{Q}_p$ can be applied to identify the corresponding $x([i,j])$ and $y([i,j])$ coordinates for the calculated disparity $d([i,j])$ of pixel $[i,j]$ in the reference camera frame. A rectified 2D image of a horizontal stereo camera setup with the disparities $d([i,j])$ is projected into 3D space with

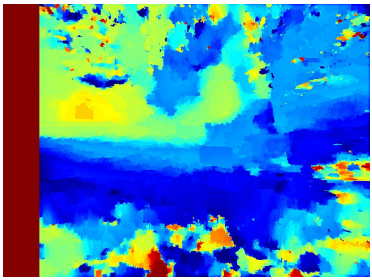
$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = \mathbf{Q}_p \cdot \begin{pmatrix} i \\ j \\ d([i,j]) \\ 1 \end{pmatrix}. \tag{B.1}$$

The optical centers o_{x_1}, o_{x_2}, o_y are given in the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ of the rectified camera coordinate systems, and the calibration of a the stereo camera system for zero disparity yields an equivalent o_x for both optical centers inside the projection matrices.

For stereo image disparity estimation, the following parameterization of SGBM achieved a convenient performance on unstructured images of the JAI AD-130GE cameras: a minimum disparity of 0, a maximum disparity of 160, a block size of 3, a disparity smoothness $P1 = 216$ and $P2 = 864$, and a uniqueness ratio of 5 %. Pre-filtering was used with a *preFilterCap* of 63 and LRC was conducted with a one pixel threshold. A weighted least squares filter with a $\lambda_{\mathrm{LS}} = 8000$ and $\sigma_{\mathrm{LS}} = 1.5$ was applied for additional refinement. Figure B.1 shows an exemplary left and right image of the IOSB.amp Q1 stereo-setup as well as an exemplary SGBM disparity estimation result and the corresponding stereo camera point cloud.

(a) Left (reference) RGB image.

(b) Right RGB image.



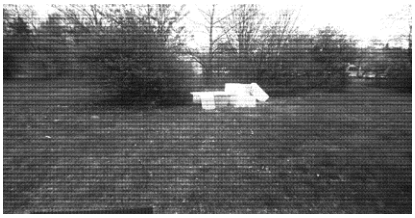
(c) SGBM disparity image.

(d) SGBM stereo camera point cloud.

Figure B.1 Left (a) and right (b) stereo image with disparity map (c) and corresponding 3D point cloud (d) from SGBM stereo image disparity estimation. Disparity estimates that were identified as inaccurate during post-processing with LRC are colored in red. Images ©Fraunhofer IOSB.

B.1.2 Disparity Estimation on Hyperspectral Images

Figure B.2(a)-(c) depict the evaluation images for the hyperspectral stereo image disparity estimation with CCRADAR, as detailed in Section 5.1.1. Table B.2(d) shows the group dimensions used on the M6000. The evaluation of the CUDA implementation was carried out with the NVIDIA Visual profiler allowing the analysis of registry allocation and memory requirements. As native CUDA context on the M6000 contains 32 elements, the integral multiples of 32 and of the image resolution with mosaic pixels (512×272) were evaluated. For hyperspectral images with 16 channels, one mosaic pixel needs 16 bytes as a basis for memory allocation and the similarity measures for stereo image disparity estimation



(a) Vegetation with test objects ($j = 1$).

(b) Mixed urban zone ($j = 2$).

(c) Pure urban zone ($j = 3$).

Processing step	Group dimension
Census Transform	16×8
SAD	128×1
SGD_x	16×16
SGD_y	16×16
Cost Comb.	8×64 - 512×1
Disp. Selection	16×16
Guided Filter	256×1

(d) Group dimensions for M6000.

Figure B.2 Images (a)-(c) depict the evaluation images for hyperspectral stereo image disparity estimation [324]: image (a) and (b) represent mainly unstructured environments, while image (c) represents structured outdoor environments. Table (d) describes the chosen dimensions of the SIMT group for hyperspectral CCRADAR on the NVIDIA M6000 GPGPU.

are conducted on 16 channels. Thereby, a partitioning with eight contexts allowed efficient latency hiding, as detailed in [285].

B.1.3 *UEM-CNN*: Stereo Disparity Estimation with CNNs

Figure B.3 shows an exemplary range-limit error weighting $\mathcal{W}$ scaled for the reference disparity values $\mathbf{D}_2[j,k] \in [0,255]$. Table B.1 extends the accuracy analysis results for stereo image disparity estimation with *UEM-CNN* described in Table 5.4. Figure B.4 depicts the non-occluded 5PE error on image 88 of the KITTI 2012 training dataset depicted in Figure 5.6.

Method	3PE	R3PE	PD
SGBM, LRC, weighted LS	25.50	17.14	95
UEM-CNN$_{base}$ raw	39.49	32.33	96
UEM-CNN$_{base}$ LRC	17.43	10.82	50
UEM-CNN$_{base}$ median	34.57	27.56	96
***UEM-CNN*$_{base}$ median, LRC**	15.97	9.59	49
UEM-CNN$_{base}$ LRC, median	16.66	9.20	55
UEM-CNN$_{base}$ LRC, median, LRC	15.88	9.04	55
UEM-CNN$_9$ raw	31.37	24.7	96
UEM-CNN$_9$ LRC	16.63	10.32	63
UEM-CNN$_9$ median	28.00	21.50	96
***UEM-CNN*$_9$ median, LRC**	14.74	8.71	63
UEM-CNN$_9$ LRC, median	16.03	9.34	61
UEM-CNN$_9$ LRC, median, LRC	14.64	8.36	63
UEM-CNN$_{19}$ raw	25.10	17.70	98
UEM-CNN$_{19}$ LRC	14.90	7.65	69
UEM-CNN$_{19}$ median	23.92	16.69	98
***UEM-CNN*$_{19}$ median, LRC**	14.26	7.19	70
UEM-CNN$_{19}$ LRC, median	14.81	7.48	70
UEM-CNN$_{19}$ LRC, median, LRC	14.35	7.16	70

Table B.1 3PE, reference weighted 3PE (R3PE), and PD for classic SGBM and *UEM-CNN* on KITTI 2012. Raw indicates no post-processing, LRC and median filtering were conducted with 3 px.

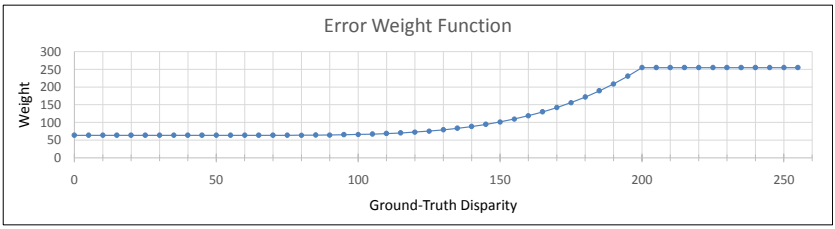


Figure B.3 Exemplary range-limit error weighting function W: $D_2 \in [0, 255]$, $b = 63.75$, which is 25 % of 255, $L = 50$, and $U = 200$. This parameterization achieved a convenient error assessment for stereo image disparity estimation on greyscale images from unstructured environments.

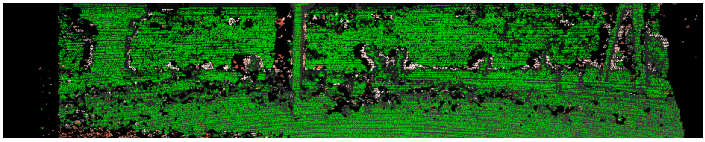


Figure B.4 Non-occluded 5PE error in addition to the disparity error metrics demonstrated on image 88 of the KITTI 2012 training dataset in Figure 5.6. Green coloring shows estimated disparities with less than 5PE and errors exceeding 5PE are colored red with higher intensity indicating higher error values. Especially low exposed and low textured areas introduced high disparity estimation errors in the analyzed unstructured environments.

B.2 *SET*: Stereo Evaluation Toolbox

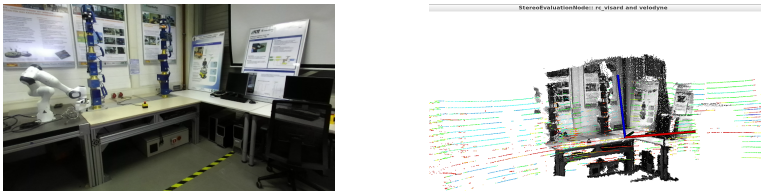
Table B.2 gives the technical details of all camera systems that were evaluated with *SET*. Further details to *SET* can be found in [325].

B.2.1 Dynamic Evaluation: Visual SLAM Assessment

SET can also be applied to compare the results of off-the-shelf stereo camera systems with integrated SLAM solutions to customized SLAM solutions with feature- or correlation-based algorithms on different camera systems. ORB-SLAM2 was chosen to demonstrate the comparison of

Type (# Channels)	Image Res.	Depth Res.	B	Stereo Alg.	SLAM Alg.
ZED, filling (3)	3840×1080	1280×720	0.120 m	custom	custom
rc_visard 160 (1)	1280×960	640×480	0.160 m	custom	custom
MQ013RG (1)	1280×1024	1280×1024	0.385 m	ORB, SGM	ORB-SLAM2
MQ022HG (16)	2048×1088	512×272	0.740 m	CCRADAR	–

Table B.2 Camera systems included in experimental evaluation with *SET*.



(a) (b)

Figure B.5 Image a) depicts an exemplary structured indoor scene for static SET evaluation, and image (b) shows stereo disparity estimation results of the rc_visard camera system after registration to the Velodyne VLP-16 reference data.

off-the-shelf visual SLAM solutions to customized SLAM solutions with different camera systems as elaborated in [325].

The dynamic evaluation was demonstrated for indoor environments as a typical application environment for visual SLAM in person indoor navigation [318, 320]. However, the proposed approach is also applicable for visual SLAM in outdoor environments. The proposed evaluation

Criterion	Identifier	Calculation	Weight
Absolute path length	PL	$\max(\lvert PL_{\text{SLAM}}/PL_{\text{GT}} - 1\rvert, 1)$	0.40
Max. distance from path	$D_{\max}$	$\max_{j=1}^{N}(d_j)/0.5\,\text{m}$	0.15
Min. distance from path	$D_{\min}$	$\min_{j=1}^{N}(d_j)/0.5\,\text{m}$	0.15
Average distance from path	$\overline{D}$	$\sum_{j=1}^{N} d_j/(N \cdot 0.5\,\text{m})$	0.30

Table B.3 Criteria and corresponding weightings for the dynamic evaluation of visual SLAM solutions in indoor environments.

criteria are purely quantitative and Table B.3 provides an overview with identifiers, calculation methods, and weights. The ground truth PL_{GT} can be obtained from building plans, if available, or from manual measurements. Path length compares the total path length in visual SLAM to the ground truth path length. Here, the SLAM path length (PL) is the sum of the path elements d_j between the estimated camera positions $\mathbf{Q}_j$:

$$PL_{\text{SLAM}} = \sum_{j=1}^{N-1} |\mathbf{Q}_{j+1} - \mathbf{Q}_j|, \tag{B.2}$$

as illustrated in Figure B.6. $D_{\max}$, $D_{\min}$, and $\overline{D}$ were normalized with 0.5 m, which proved useful as a tolerable maximum in experimental validation [318, 325]. PL measures scaling accuracy and noise of the overall SLAM solution, while $D_{\max}$, $D_{\min}$, and $\overline{D}$ assess the number of outliers as well as the accuracy of the individual estimated camera positions in comparison to the ground truth path. The combination of the four proposed criteria allowed a separate assessment of inaccuracies due to outliers and noise and the overall performance rating of the SLAM solution. Finally, the dynamic *SET* score is calculated with

$$S_{\text{dyn}} = \sum_{i} i \cdot w_i, i \in \{PL, D_{\max}, D_{\min}, \overline{D}\}. \tag{B.3}$$

B.2.2 Proof of Concept: Visual SLAM Assessment with *SET*

The dynamic *SET* evaluation was demonstrated on three typical indoor scenarios as described in [325]. The results for visual SLAM performance given in Table B.4 originate from the second scenario that passed a straight corridor with monochromatic walls and also included a turn around one corner. Figure B.5 shows exemplary results of the dynamic evaluation. In the dynamic *SET* evaluation, the rc_visard camera yielded the most accurate visual SLAM performance among the three evaluated systems for visual SLAM in indoor environments.

Criterion	ZED	rc_visard	$\text{MQ013}_{\text{ORB}}$
PL (0–1)	0.095	0.602	0.059
$D_{\max}$ (0–1)	0.312	0.497	0.529
$D_{\min}$ (0–1)	0.003	0.012	0.0
$\bar{D}$ (0–1)	0.155	0.149	0.265
Score	**0.132**	**0.362**	**0.182**

Table B.4 Application of *SET* in performance assessment of the stereo systems described in Table B.2 in visual SLAM according to [325].

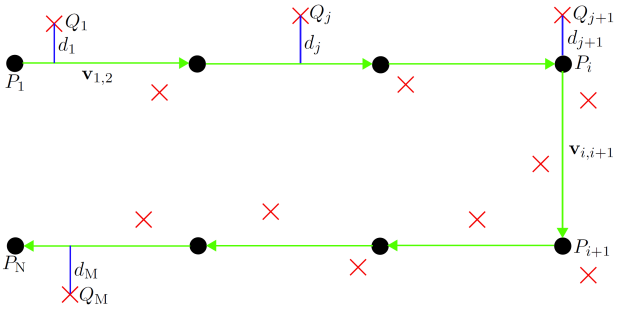


Figure B.6 Evaluation of visual SLAM: Q_j camera position from visual SLAM, P_i GT path, d_j denotes the perpendicular distance to the ground truth according to [325].

B.3 Sensor Data Fusion

Table B.5 contrasts the complementary characteristics of the analyzed 2D and 3D cross-source sensor data. The given, complementary characteristics can be combined depending on the available sensors within the visual multi-sensor system with 2D–3D or 3D–3D fusion detailed in Section 5.3.

Characteristic	RGB	3D Stereo	3D LiDAR	2D–3D Fusion	3D–3D Fusion
Intensity	RGB	RGB	Refl.	RGB, refl.	RGB, refl.
Color	Dense	Dense	–	Dense	Dense
Geometry	–	Dense	Sparse	Sparse	Dense
Depth acc.	–	$\sim 1/z^2$	High	High	Mixed

Table B.5 Complementary characteristics in 2D and 3D cross-source data: intensity, passive measurements of color intensity from passive cameras (color), geometric measurements for 3D reconstruction (geometry), and depth estimation accuracy (depth acc.); LiDAR sensors measure the intensity of the reflections in the NIR spectral range (refl.).

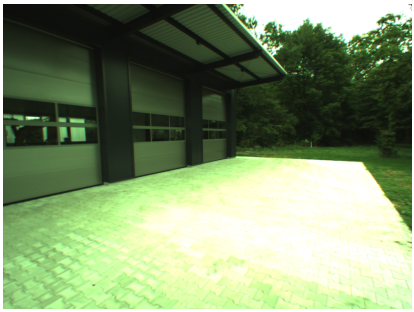
Figure B.7 shows selected 2D–3D fusion results on exemplary cross-source sensor data of the *IOSB-Reg* dataset generated on the basis of the given intrinsic and extrinsic calibration. Figure B.8 depicts 3D–3D fusion results for method $\mathcal{C}$ and a combination of method $\mathcal{B}$ and $\mathcal{C}$. The combination of $\mathcal{B}$ and $\mathcal{C}$ filters a notably higher amount of points from the stereo point cloud than the utilization of method $\mathcal{C}$.

Furthermore, the integration of the implicit 2.5 D surface fusion approach of Dutschk et al. [55] in the confidence-based 3D–3D fusion can help to increase the surface reconstruction accuracy of the fused 3D cloud in future works. The implicit surface approximation estimates and fuses surfaces depending on the uncertainty σ^2 of the model:

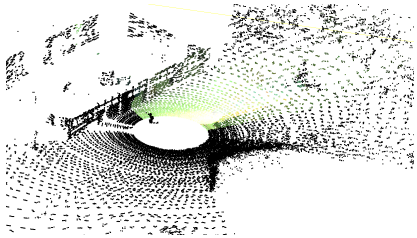
$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{N} (f_k(\mathbf{x}) - \mathbf{n}_j^T(\mathbf{x} - \mathbf{x}_j))^2 \tag{B.4}$$

with N implicit surfaces and $\mathbf{n}_j^T$ the transposed normal vector at the test point $\mathbf{x}_j$. Finally, the fusion of the surfaces is performed using the weighting factor $w_i = \frac{1}{\sigma^2}$ as

$$f_k(\mathbf{x}) = \frac{\sum_i w_i f_{i,k}(\mathbf{x})}{\sum_i w_i}. \tag{B.5}$$

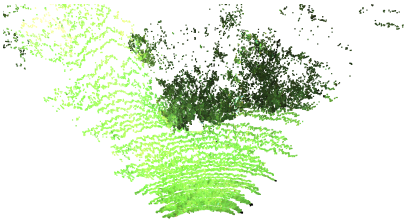


(a) Original 2D RGB image.



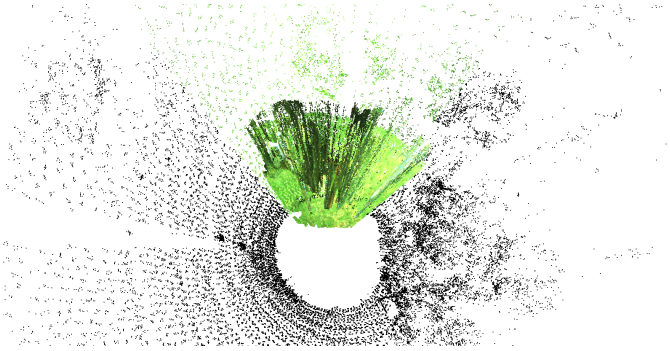
(b) 2D–3D fusion.

(c) 2D–3D fusion: front view of condensed, FoV-filtered cloud.

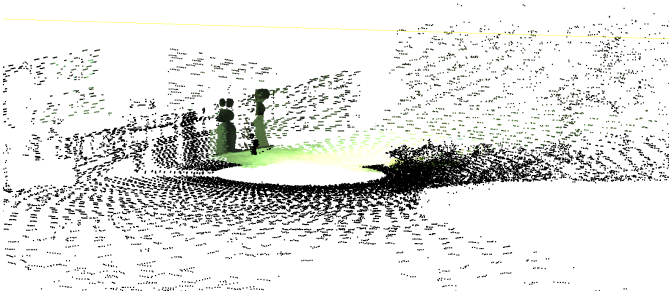


(d) 2D–3D fusion: bird's eye view of (b).

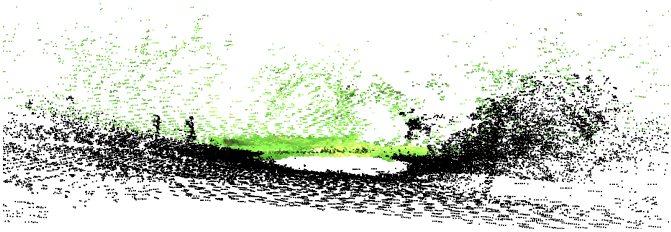
Figure B.7 Projection of the geometric 3D information from the LiDAR point cloud onto the 2D image for an exemplary image from the *IOSB-Reg* dataset: original RGB image from JAI AD-130GE camera (a) and corresponding 2D–3D fusion result with a single Velodyne HDL-64E point cloud (b). (c) shows the 2D–3D fusion result of a condensed Velodyne HDL-64E point cloud from eleven single scans in static configuration after filtering the FoV with available color information, while (d) depicts a bird's eye view on 2D–3D fusion result with condensed and FoV-filtered cloud. 3D points without color information from the image are colored in black.



(a) 3D–3D fusion $\mathcal{C}$ for unstructured scenery (II).



(b) 3D–3D fusion with the combination of $\mathcal{B}$ and $\mathcal{C}$ for scene I.



(c) 3D–3D fusion with method $\mathcal{B}$ and $\mathcal{C}$ for scene II.

Figure B.8 3D–3D fusion results of stereo camera point clouds and Velodyne LiDAR point clouds for scenes I and II: (a) uses the range-based method $\mathcal{C}$ with $r \leq 10\,\text{m}$ for scene II (694,452 points), (b) and (c) combine method $\mathcal{B}$ ($d_{\text{NN}} = 0.1\,\text{m}$, $N_{\text{min}} = 10$) and $\mathcal{C}$ with $r \leq 10\,\text{m}$ for the partially structured scene I (b) and for the unstructured scene II (c).

C High-Level Perception

C.1 Semantic Segmentation of 3D Point Clouds

The proposed high-level perception methods interpret ordered and unordered 3D point clouds as "single-shot" 3D clouds without additional color information.

C.1.1 Classic Segmentation of 3D Data

Classic 2D and 3D segmentation techniques include spatial clustering and region growing schemes [6, 108]. Classic segmentation methods are well-established tools and seldomly subject to research nowadays [318, 323]. Classic segmentation techniques are mainly utilized as pre- and post-processing tools within other methods. Consequently, classic segmentation techniques are only discussed in the context of low-level perception in this thesis. For instance, the Random Sample Consensus (RANSAC) algorithm [39] can be applied to estimate point sets with simple geometric characteristics such as in plane segmentation. RANSAC can also be applied to detect the approximately flat ground plane in outdoor environments, as discussed in Section 4.2.3. Albrecht and Heide [318] apply RANSAC to identify and remove floor and ceiling in structured indoor environments.

C.1.2 Computational Effort for Semantic 3D Segmentation

One Velodyne HDL-64E frame typically includes around 100,000 points and frames are captured with 10 Hz on the technology demonstrators in this thesis. The subsequent analysis compares the computational effort of an exemplary classification method, which can be extended towards semantic segmentation and works in 3D space (PointNet), to the computation effort of a semantic segmentation methods that works on the basis of

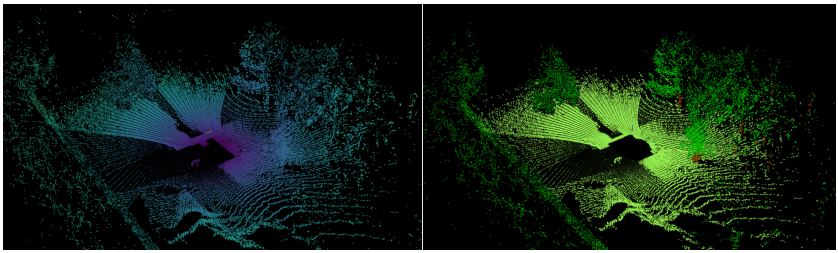
2D range images (SqueezeSeg). PointNet requires $440 \cdot 10^6$ floating point operations for the classification of one sample in average [221] and empirical tests of Qi et al. [221] show the potential to process 10^6 points per second with an NVIDIA 1080Ti GPU. For reference, an NVIDIA GeForce GTX TITAN X[1] has a theoretical performance of $6691 \cdot 10^{12}$ floating point operations per second (FLOPS), while an NVIDIA GeForce GTX 1080 has a maximum theoretical processing power of $8873 \cdot 10^{12}$ single precision FLOPS – even more than the reference GPU that was used to evaluate the feature extraction in 2D space for SqueezeSeg [298]. Contrasting the computational effort of PointNet for classification only, SqueezeSeg required 13.5 ms in average on a TITAN X GPU in the experiments of [298] for the semantic segmentation of 360° LiDAR point clouds from Velodyne HDL-64E LiDAR sensors including post-processing. Concluding, PointNet required approximately one second to process one million points for classification on a faster GPU and without any post-processing. This clearly shows that the application of 2D convolutions that implies the semantic segmentation on the basis of 2D range images is much more favorable in robotic perception by now.

C.1.3 Domain Transfer

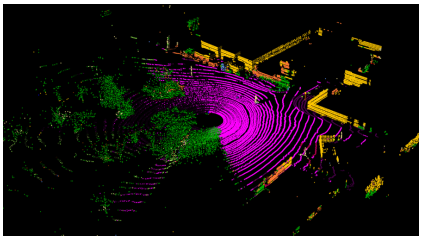
Figure C.1 depicts the ground truth labeling of the analyzed IOSB.Alice cloud and illustrates selected 3D point clouds for the 2D range images depicted in Figure 6.3.

Table C.1 compares the relative label distribution of SemanticKITTI, the full SemanticUSL dataset, as well as of selected scenes of SemanticUSL. Figure C.1 furthermore depicts selected 3D point clouds that correspond to the semantic segmentation results given in Figure 6.3 that were achieved by *DN53* after domain transfer to IOSB.Alice.

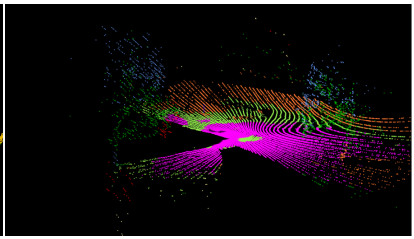
[1] https://www.techpowerup.com/gpu-specs/geforce-gtx-titan-x.c2632, access on 06.12.2021.



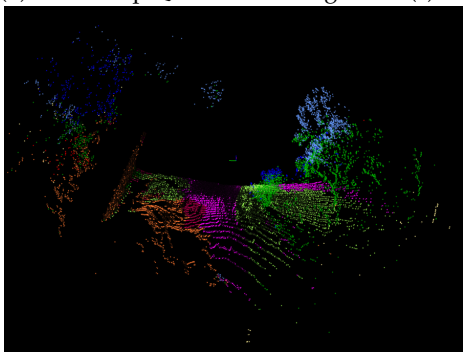
Left: Raw 3D point cloud IOSB.Alice, right: ground truth labeling [339]



To Figure 6.3(b): IOSB.amp Q1.



To Figure 6.3(c): IOSB.Alice left.



To Figure 6.3(d): IOSB.Alice rear.

Figure C.1 Selected, respective 3D point clouds for IOSB.Alice domain transfer analysis: raw, fused 3D point cloud and ground truth labeling, and domain transfer results on spherical projections (range images) with *DN53* in Figure 6.3.

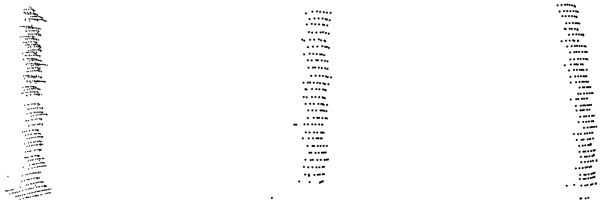
Dataset	car	truck	other-vehicle	person	road	parking	sidewalk	other-ground	building	**fence**	**vegetation**	**trunk**	**terrain**	pole	traffic-sign
K	6.2	0.1	0.4	0.1	17.6	1.2	12.1	0.1	11.4	**2.5**	**29.2**	**1.1**	**12.8**	0.3	0.1
U 3,12,21,32	1.3	<0.1	0.0	0.3	17.3	2.1	13.5	<0.1	18.4	**6.7**	**21.0**	**0.8**	**9.7**	0.4	0.1
U 3	0.3	0.0	0.0	0.4	23.7	1.7	8.6	<0.1	15.5	17.3	18.4	0.7	5.3	0.5	0.1
U 12	0.0	0.0	0.0	<0.1	0.0	0.0	17.0	0.0	25.3	5.9	21.6	1.3	20.7	0.1	0.0
U 21	3.6	0.1	0.0	0.5	25.4	7.5	10.8	0.0	20.4	0.2	20.0	0.3	1.7	0.5	0.4
U 32	2.1	0.0	0.0	0.2	23.1	0.0	17.7	0.0	11.4	1.4	24.3	0.9	9.4	0.4	0.1

Table C.1 Relative label distribution of SemanticKITTI (K) (seq. 8), the full SemanticUSL (U) dataset, and of individual SemanticUSL scenes (3, 12, 21, 32). The relative label distributions for the following classes are not displayed as they are seldom encountered in the analyzed unstructured environments or have a negligible share: bicycle, bicyclist, motorcycle, motorcyclist, other-vehicle, unlabeled.

C.2 Explainable Artificial Intelligence

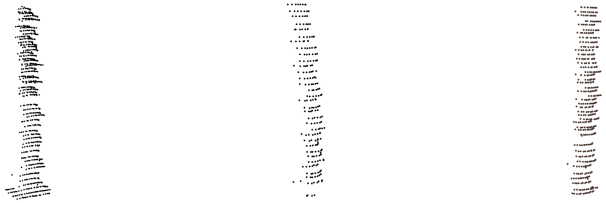
C.2.1 X^{3}Seg: Post-Modeling, Model-Agnostic XAI for 3D Semantic Segmentation

Figure C.2 depicts the lower trunk explanation target and the identified best-matching prototypes for two different grid sizes, and Figure C.3 shows the best-matching selective X^{3}Seg prototypes for two different values of σ_G. Figure C.7 provides an overview of the best-matching prototypes of the analyzed lower and upper trunk structures according to each individual metric. As already discussed, experimental evaluation justified the assumption that a higher number of prototypes #P lead to a better understanding of the class predictions in X^{3}Seg due to a higher variety of prototypes which allowed the identification of a very similar prototype from the training data. Figure C.5 and Figure C.4 show the best-matching prototypes with #P $\in \{25, 100, 500\}$.



(a) Explanation target. (b) 25×25 grid. (c) 100×100 grid.

Figure C.2 Comparison of a low and a high resolution grid for the lower trunk explanation target. Different resolution did not show a notable influence on the best-matching prototypes determined out of 100 prototypes in $\mathcal{X}^{\in}$. Images ©Fraunhofer IOSB.

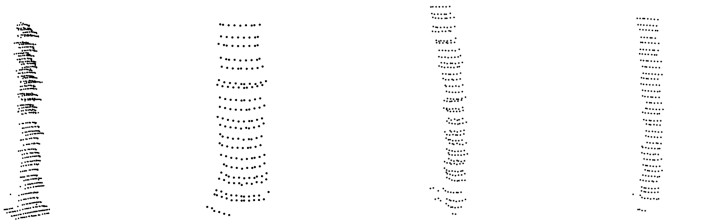


(a) Explanation target. (b) σ_G=2. (c) σ_G=10.

Figure C.3 Comparison of an exemplary explanation target of a lower trunk structure with the best matching prototypes from $\mathcal{X}^{\in}$ with (b) σ_G=2 and (c) σ_G=10. Images ©Fraunhofer IOSB.

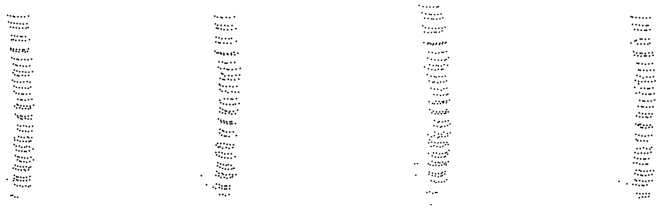
(a) Explanation target (b) #P = 25 (c) #P = 100 (d) #P = 500

Figure C.4 Evaluation results for different number of prototypes #P in $\mathcal{X}^{\in}$: (a) explanation target upper trunk, (b) best prototype for #P = 25, (c) best prototype for #P = 100, (d) best prototype for #P = 500 [336]. Images ©Fraunhofer IOSB.



(a) Explanation target (b) #P = 25 (c) #P = 100 (d) #P = 500

Figure C.5 Evaluation results for different number of prototypes #P in $\mathcal{X}^{\in}$: (a) explanation target lower trunk, (b) best prototype for #P = 25, (c) best prototype for #P = 100, (d) best prototype for #P = 500 [336]. Images ©Fraunhofer IOSB.



(a) (b) (c) (d)

Figure C.6 Best matching prototypes according to individual metrics for exemplary lower trunk explanation target depicted in Figure 6.10 with selective X^{3}Seg: (a) best SX $c_{X,2}$, (b) best SX $p_{X,1}$, (c) best SX $p_{X,2}$, and (d) best SX r_X [336]. Images ©Fraunhofer IOSB.

(a) (b) (c) (d)

Figure C.7 Best matching prototypes according to individual metrics for exemplary upper trunk explanation target depicted in Figure 6.10 with selective X^{3}Seg: (a) best SX $c_{X,2}$, (b) best SX $p_{X,1}$, (c) best SX $p_{X,2}$, and (d) best SX r_X [336]. Images ©Fraunhofer IOSB.

D Application Scenarios

Figure D.1 illustrates the three cornerstones of autonomous systems according to [173] that were adapted for autonomous off-road vehicles in this thesis. Figure D.2 depicts the 3D model of the IOSB.Alice platform

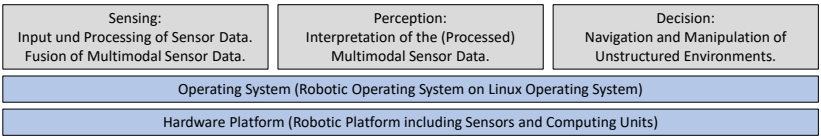


Figure D.1 System architecture of autonomous vehicle systems derived from [173].

in the ROS visualization tool Rviz with the calibrated and fused LiDAR sensor data for accurate workspace monitoring.

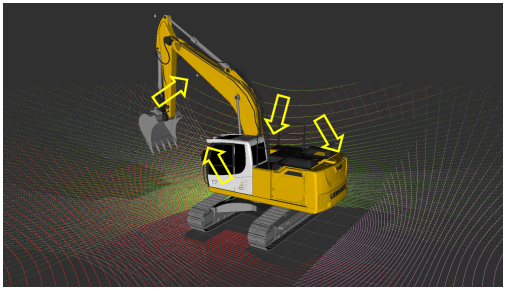


Figure D.2 Commercially available excavator platform IOSB.Alice. The four Ouster OS-0 LiDAR sensors with a FoV from −45° to 45° providing 3D perception of the environment are highlighted. © Fraunhofer IOSB

D.1 Data Generation for Unstructured Environments

Figure D.3 shows some additional *IOSB-Reg* images in addition to the images utilized for proof-of-concept demonstration within this thesis.

Image 04.

Image 06.

Image 13.

Image 16.

Figure D.3 Selected images of the *IOSB-Reg* dataset. © Fraunhofer IOSB

The 3D LiDAR measurements have to be referenced to the coordinate system of the reference camera to generate disparity images from the depth information of the sensor. Then, the 2D depth map and its associated disparity image can be used as an accurate ground truth for stereo image disparity estimation. Horizontal and vertical FoV filtering for the LiDAR clouds is required to provide depth images that contain exactly the same area, and the horizontal FOV is already filtered during the preprocessing. The overlap of the vertical FoVs of camera and LiDAR can

be determined using the camera intrinsics, as discussed in the 2D–3D fusion of RGB images and LiDAR data in Section 3.7. The accumulation step was performed as proposed in the KITTI dataset [80]: Geiger et al. [83] merged five LiDAR point clouds before and after the respective frame after an ICP registration and achieved an average density of approximately 50 % for a ground truth that can be compared against stereo depth estimation results. Hence, five clouds prior and five clouds after the current capture were fused over a time interval of 1.1 s for dense point clouds in *IOSB-Reg*. The positions of the vehicle and the sensors itself in the local reference frame were exactly known due to an accurate localization provided by the ATB [60]. Furthermore, vehicle movement during accumulation did not pose a problem as inertially corrected point clouds were used in contrast to the accumulation proposed in [80] relying on ICP registration. Labelbox[1], PixelAnnotation Tool[2], LabelImg[3], BMW-

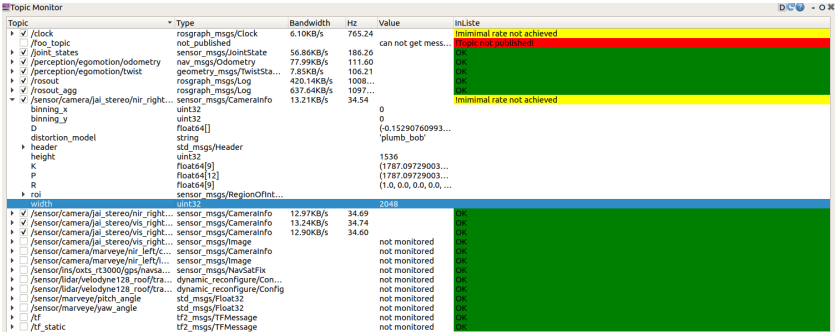


Figure D.4 Customized data validation tool for *GOOSE* on the basis of the rqt topic_monitor tool. Green coloring indicates that the topic in question is published correctly and all the specified minimum requirements are met, while a yellow coloring of the respective field highlights that the minimum publication rate or another quality criterion is not met. If the field is highlighted in red, the topic is not published correctly.

Labeltool-Lite[4], and CVAT Computer Vision Annotation Tool[5] allow the labeling of 2D images. Labelbox provides model-assisted labeling with bounding boxes, polygons, polylines, and points, and also allows the import of computer generated labels. The PixelAnnotation Tool has an integrated automatic labeling procedure which can notably speed up the labeling process. LabelImg and BMW-Labeltool-Lite exclusively generate 2D bounding boxes, the Cvat tool can add bounding boxes, polygons, and polylines with corresponding labels. The CVAT tool allows and interactive annotation of image and video data with bounding boxes, polygons, and polylines. Labelstudio[6] enables the labeling of audio, text, images, videos, and time series. 2D images can be labeled using bounding boxes, polygons, and polylines. Monica et al. [198] and Behley et al. [11] propose annotation tools for 3D point clouds. The RViz Cloud Annotation Tool [198] was developed on the basis of the Rviz visualization tool of the ROS environment. Labeling is conducted by manually selecting sparse control points belonging to objects respectively previously defined labels in unorganized point clouds. The sparse control points are extended to a full semantic labeling by a classic segmentation using a shortest path tree search on the neighborhood graph. The Rviz-based tool of [198] developed for close-range, structured indoor environment allows the specification of own class labels, and the labeled point clouds can be saved as *pcl::PointXYZRGBALabel* clouds with a numerical representation for each label. Point_labeler [11] was provided together with the SemanticKITTI dataset [11] as an open source tool for offline use. It facilitates the point-by-point labeling of a single 3D point cloud or a stream of

[1] D. Rasmuson et al., Labelbox: Apache-2.0 License, www.labelbox.com, https://github.com/Labelbox/labelbox, access on 14.04.2022.

[2] A. Bréhéret: PixelAnnotationTool, https://github.com/abreheret/PixelAnnotationTool, access on 14.04.2022.

[3] D. Tzutalin: LabelImg, MIT license, https://github.com/tzutalin/labelImg, access on 14.04.2022.

[4] R. Anwar and E. Saller, BMW-InnovationLab: BMW-Labeltool-Lite, Apache-2.0 License, https://github.com/BMW-InnovationLab/BMW-Labeltool-Lite, access on 14.04.2022.

[5] B. Sekachev; A. Zhavoronkov; M. Zhiltsov; and D. Kalinin: Computer Vision Annotation Tool (CVAT), https://github.com/openvinotoolkit/cvat, access on 14.04.2022.

[6] M. Tkachenko; M. Malyuk; N. Shevchenko; A. Holmanyuk; and N. Liubimov: Label Studio, Apache-2.0 License, https://github.com/heartexlabs/label-studio, access on 14.04.2022.

point clouds with customized labels and customized labels can be defined within a *json* file. Experimental evaluation showed that point_labeler is the most suitable tool for the labeling of cloud sequences. 3D-Bat [316] facilitates the labeling of 2D images and 3D point clouds with bounding boxes in semi-automatic manner. 3D-Bat is web-based and also allows the addition of instance IDs for objects on roads as it targets autonomous driving applications in structured environments. Additional, predefined bird's-eye, side, and front views help to visualize objects from different perspectives. 2D images and 3D point clouds can also be labeled using the semantic segmentation editor[7]. The integrated Cityscapes class definition can be exchanged with arbitrary, customized class definitions within the supplied yaml file. The editor is browser-based and can be used online and offline. 2D images are labeled pixel-by-pixel in a bitmap image editor with the assistance of polygons to label several points at once. 3D point clouds are required in PCD format with *pcd::PointXYZ* or *pcd::PointXYZRGB* point types, and the tool outputs PCD files with point order and format equivalent to the input files and corresponding labels and object IDs are added in integer format.

D.2 Cost Valley for Constrained Planning

This appendix contains additional material to Section 7.6. A customized, local approach for waypoint optimization was evaluated together with RDP comparing the first order derivative (d'_a) of the currently processed, 2D path element d_a to the first order derivatives (d'_b) of consecutive path elements d_b. Their difference was calculated with $\Delta(d') = |d'_a - d'_b|$ and path elements were kept if $\Delta(d')$ was larger than the predefined max $\Delta(d')$, as further elaborated in [328]. The local and the RDP line simplification method were compared on several recorded tracks with different lengths and RDP clearly outperformed the local approach.

Figure D.5 and Figure D.6 depict additional results of the real-world navigation evaluation on the TULF platform: Figure D.5 depicts an in-

[7] D. Mandrioli et al., Hitachi Automotive And Industry Lab: Semantic segmentation editor, https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor, access on 07.11.2021.

termediate result of the real-world navigation evaluation on the TULF platform, and Figure D.6 illustrates the return of the TULF platform to the cost valley after an intervention of the human safety driver.

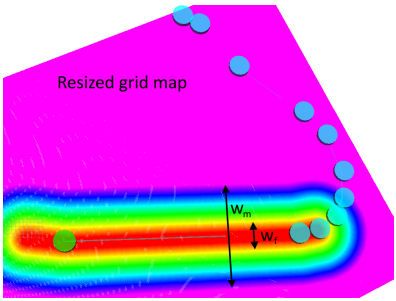


Figure D.5 TULF: First waypoints completed, planning along approx. 300 m to next waypoints, ϵ is 0.8 m, w_f is 3.0 m, w_m is 30 m.

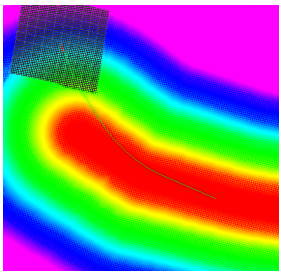


Figure D.6 TULF: Returning to the valley after intervention of safety driver, w_f is 3 m, w_m is 30 m.

Bibliography

[1] **A. Agrawal, A. Nakazawa, and H. Takemura**. *MMM-classification of 3D Range Data*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2009), pp. 2003–2008.

[2] **P. F. Alcantarilla, A. Bartoli, and A. J. Davison**. *KAZE features*. In: *European Conference on Computer Vision (ECCV)* (2012), pp. 214–227.

[3] **M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl**. *Nearest neighbor classification in 3D protein databases*. In: *International Conference on Intelligent Systems for Molecular Biology* (1999).

[4] **V. Arsigny, P. Fillard, X. Pennec, and N. Ayache**. *Log-Euclidean metrics for fast and simple calculus on diffusion tensors*. In: *Magnetic resonance in medicine* 56.2 (2006), pp. 411–421.

[5] **V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, and A. Mojsilović**. *One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques*. 2019. arXiv preprint: 1909.03012.

[6] **T. Asano and N. Yokoya**. *Image segmentation schema for low-level computer vision*. In: *Pattern Recognition* 14.1-6 (1981), pp. 267–273.

[7] **E. P. Baltsavias**. *Airborne laser scanning: basic relations and formulas*. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 54.2-3 (1999), pp. 199–214.

[8] **A. Barla, F. Odone, and A. Verri**. *Histogram intersection kernel for image classification*. In: *International Conference on Image Processing (ICIP)* (2003).

[9] **A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera**. *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI*. In: *Information Fusion* 58 (2020), pp. 82–115.

[10] **H. Bay, A. Ess, T. Tuytelaars, and L. van Gool**. *Speeded-up robust features (SURF)*. In: *European Conference on Computer Vision (ECCV)* 110.3 (2006), pp. 346–359.

[11] **J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall**. *SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences*. In: *IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).

[12] **J. T. Behrens**. *Principles and procedures of exploratory data analysis*. In: *Psychological Methods* 2.2 (1997), pp. 131–160.

[13] **M. Bergerman, S. Singh, and B. Hamner**. *Results with autonomous vehicles operating in specialty crops*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2012).

[14] **J. Berkmann and T. Caelli**. *Computation of surface geometry and segmentation using covariance techniques*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.11 (1994), pp. 1114–1116.

[15] **P. J. Besl and N. D. McKay**. *A method for registration of 3-D shapes*. In: *Sensor fusion IV* 1611 (1992), S. 586–606.

[16] **J. Beyerer, F. Puente León, and C. Frese**. *Machine vision*. Berlin Heidelberg: Springer, 2016.

[17] **J. Beyerer, M. Heizmann, J. Sander, and I. Gheta**. *Bayesian methods for image fusion*. In: *Image fusion: Algorithms and Applications* (2008).

[18] **D. Bhargava, S. Vyas, and A. Bansal**. *7 - Comparative analysis of classification techniques for brain magnetic resonance imaging images*. In: *Advances in Computational Techniques for Biomedical Image Analysis*. Ed. by **D. Koundal and S. Gupta**. Academic Press, 2020, pp. 133–144.

[19] **M. Bleyer, C. Rhemann, and C. Rother**. *Extracting 3D scene-consistent object proposals and depth from stereo images*. In: *European Conference on Computer Vision (ECCV)* (2012).

[20] **M. Bleyer, C. Rother, and P. Kohli**. *Surface stereo with soft segmentation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[21] **M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha**. *Object stereo–Joint stereo matching and object segmentation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).

[22] **A. Boulch, J. Guerry, B. Le Saux, and N. Audebert**. *SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks*. In: *Computers & Graphics* 71 (2018), pp. 189–198.

[23] **Y. Boykov, O. Veksler, and R. Zabih**. *Fast approximate energy minimization via graph cuts*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11 (2001), pp. 1222–1239.

[24] **W. S. Boyle and G. E. Smith**. *Charge coupled semiconductor devices*. In: *Bell System Technical Journal* 49.4 (1970), pp. 587–593.

[25] **G. S. Broten and H. C. Wood**. *Determining the confidence levels of sensor outputs using neural networks*. In: *CNS Proceedings of the 16th Annual Conference, Volume I and II* (1995).

[26] **D. C. Brown**. *Close-range camera calibration*. In: *Photogrammetric Engineering* 37.8 (1971), pp. 855–866.

[27] **M. Buehler, K. Iagnemma, and S. Singh**. *Special issues on the 2007 darpa urban challenge*. In: *Journal of Field Robotics* 25.8 (2008).

[28] **H. Burkhardt, A. Fenske, and H. Schulz-Mirbach**. *Invariants for the recognition of planar contour and gray-scale images / Invarianzen zur Erkennung ebener Kontur- und Graubilder*. In: *tm - Technisches Messen* 59.10 (1992), pp. 398–407.

[29] **H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom**. *nuscenes: A multimodal dataset for autonomous driving*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 11621–11631.

[30] **M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua**. *BRIEF: Computing a local binary descriptor very fast*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7 (2011), pp. 1281–1298.

[31] **C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós**. *ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM*. 2020. arXiv preprint: 2007.11898v1.

[32] **J. Castorena, U. S. Kamilov, and P. T. Boufounos**. *Autocalibration of LiDAR and optical cameras via edge alignment*. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), pp. 2862–2866.

[33] **D. Cattaneo, M. Vaghi, A. L. Ballardini, S. Fontana, D. G. Sorrenti, and W. Burgard**. *CMRNet: Camera to LiDAR-map registration*. In: *IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), pp. 1283–1289.

[34] **J. Čech, J. Sanchez-Riera, and R. Horaud**. *Scene flow estimation by growing correspondence seeds*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 3129–3136.

[35] **J. Čech and R. Sara**. *Efficient sampling of disparity space for fast and accurate matching*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2007), pp. 1–8.

[36] **N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss**. *Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields*. In: *The International Journal of Robotics Research* 36.10 (2017), pp. 1045–1052.

[37] **L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille**. *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[38] **Y. Chen and G. Medoni**. *Object modelling by registration of multiple range images*. In: *IEEE International Conference on Intelligent Robots and Automation* (1991).

[39] **S. Choi, T. Kim, and W. Yu**. *Performance evaluation of RANSAC family*. In: *Journal of Computer Vision* 24.3 (1997), pp. 271–300.

[40] **J. Clemens, T. Reineking, and T. Kluth**. *An evidential approach to SLAM, path planning, and active exploration*. In: *International Journal of Approximate Reasoning* 73 (2016).

[41] **B. Cohen, I. A. Şucan, and S. Chitta**. *A generic infrastructure for benchmarking motion planners*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2012).

[42] **T. S. Cohen, M. Geiger, J. Koehler, and M. Welling**. *Spherical CNNs*. 2018. arXiv preprint: 1801.10130v3.

[43] **M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele**. *The Cityscapes dataset for semantic urban scene understanding*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 3213–3223.

[44] **T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy**. *SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds*. In: *Advances in Visual Computing, Lecture Notes in Computer Science* 12510 (2020), pp. 207–222.

[45] **N. Dalal and B. Triggs**. *Histograms of oriented gradients for human detection*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 886–893.

[46] **A. Das and D. Kempe**. *Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection*. 2011. arXiv preprint: 1102.3975v2.

[47] **J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei**. *ImageNet: A large-scale hierarchical image database*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 248–255.

[48] **Y. Deng, A. Rangarajan, S. Eisenschenk, and B. C. Vemuri**. *A Riemannian framework for matching point clouds represented by the Schrodinger distance transform*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[49] **A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna**. *LiDAR-camera calibration using 3D–3D point correspondences*. 2017. arXiv preprint: 1705.09785v1.

[50] **D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel**. *Path planning for autonomous vehicles in unknown semi-structured environments*. In: *The International Journal of Robotics Research* 29.5 (2010), pp. 485–501.

[51] **D. H. Douglas and T. K. Peucker**. *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10.2 (1973).

[52] **S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier**. *Sparse stereo disparity map densification using hierarchical image segmentation*. In: *International Symposium on Mathematical Morphology and its Applications to Signal and Image Processing* (2017), pp. 172–184.

[53] **F. Dürr, H. Weigel, M. Mählisch, and J. Beyerer**. *Iterative deep fusion for 3D semantic segmentation*. In: *Fourth IEEE International Conference on Robotic Computing (IRC)* (2020), pp. 391–397.

[54] **B. Dutschk, P. Ernst, and M. Heizmann**. *Registrierung von multimodalen Sensordaten für die Oberflächeninspektion / Registration of multimodal sensor data for surface inspection*. In: *tm-Technisches Messen* 86.s1 (2019), pp. 72–76.

[55] **B. Dutschk, M. Pordzik, and M. Heizmann**. *Einsatz von Gaußprozessen und Weighted Least-Squares-Verfahren für die Fusion von konfokaler Mikroskopie und Weißlichtinterferometrie / Gaussian processes and weighted least squares methods for fusion of confocal microscopy and white light interferometry*. In: *tm-Technisches Messen* 85.s1 (2018), s7–s13.

[56] **G. El Masry, N. Wang, A. ElSayed, and M. Ngadi**. *Hyperspectral imaging for nondestructive determination of some quality attributes for strawberry*. In: *Journal of Food Engineering* 81.1 (2007), pp. 98–107.

[57] **G. Elbaz, T. Avraham, and A. Fischer**. *3D point cloud registration for localization using a deep neural network auto-encoder*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4631–4640.

[58] **W. Elmenreich**. *An introduction to sensor fusion. Research Report 47/2001*. https://www.researchgate.net/profile/Wilfried_Elmenreich/publication/267771481_An_Introduction_to_Sensor_Fusion/links/55d2e45908ae0a3417222dd9.pdf, access on 05.01.2022. Institut für Technische Informatik, Vienna University of Technology, 2002.

[59] **T. Emter and J. Petereit**. *3D SLAM with scan matching and factor graph optimization*. In: *50th International Symposium on Robotics (ISR)* (2018), pp. 1–8.

[60] **T. Emter, C. Frese, A. Zube, and J. Petereit**. *Algorithm toolbox for autonomous mobile robotic systems*. In: *ATZoffhighway Worldwide* 10.3 (2017), pp. 48–53.

[61] **European Defence Agency**. *Artificial Intelligence: Joint quest for future defence applications*. In: *European Defence Matters' magazine* 19 (2020). https://eda.europa.eu/docs/default-source/eda-magazine/edm19_web.pdf, access on 05.01.2022, p. 34ff.

[62] **M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman**. *The Pascal Visual Object Classes (VOC) challenge*. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338.

[63] **F. Falchi**. *About deep learning, intuition and thinking*. In: *ERCIM News* 116 (2019), p. 14.

[64] **P. Fankhauser and M. Hutter**. *A universal grid map library: Implementation and use case for rough terrain navigation*. In: *Robot Operating System (ROS) – The Complete Reference* 1 (2016).

[65] **A. Farhadi and J. Redmon**. *Yolov3: An incremental improvement*. 2018. arXiv preprint: 1804.02767.

[66] **D. Fehr, W. J. Beksi, D. Zermas, and N. Papanikolopoulos**. *Covariance based point cloud descriptors for object detection and recognition*. In: *Computer Vision and Image Understanding* 142 (2016), pp. 80–93.

[67] **B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars**. *Unsupervised visual domain adaptation using subspace alignment*. In: *IEEE international conference on computer vision (ICCV)* (2013), pp. 2960–2967.

[68] **M. A. Fischler and R. C. Bolles**. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[69] **A. W. Fitzgibbon**. *Robust registration of 2D and 3D Point Sets*. In: *Image and Vision Computing* 21.13-14 (2003), pp. 1145–1153.

[70] **J. D. Foley, F. D. Van, A. van Dam, S. K. Feiner, J. F. Hughes, J. Hughes, and E. Angel**. *Computer graphics: Principles and practice*. Vol. 12110. Addison-Wesley Professional, 1996.

[71] **B. Forkel, J. Kallwies, and H.-J. Wünsche**. *Probabilistic terrain estimation for autonomous off-road driving*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2021).

[72] **W. Förstner and B. Moonen**. *A metric for covariance matrices*. In: *Geodesy - The Challenge of the 3rd Millennium* (2003), pp. 299–309.

[73] **E. R. Fossum**. *CMOS image sensors: Electronic camera on a chip*. In: *Proceedings of International Electron Devices Meeting* (1995).

[74] **V. Fremont and P. Bonnifait**. *Extrinsic calibration between a multi-layer lidar and a camera*. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2008), pp. 214–219.

[75] **C. Frese, A. Fetzner, and C. Frey**. *Multi-sensor obstacle tracking for safe human-robot interaction*. In: *41st International Symposium on Robotics (ISR)* (2014).

[76] **J. Frolik, M. Abdelrahman, and P. Kandasamy**. *A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data*. In: *IEEE Transactions on Instrumentation and Measurement* 50.6 (2001), pp. 1761–1769.

[77] **K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf**. *Kernel measures of conditional dependence*. In: *Advances in Neural Information Processing Systems* 20 (2008), pp. 179–186.

[78] **C. Gao and J. R. Spletzer**. *On-line calibration of multiple LIDARs on a mobile vehicle platform*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2010), pp. 279–284.

[79] **N. Gat and C. A. Torrance**. *Real-time multi-and hyper-spectral imaging for remote sensing and machine vision: an overview*. In: *ASAE Annual International Meeting* (1998).

[80] **A. Geiger, P. Lenz, and R. Urtasun**. *Are we ready for autonomous driving? The KITTI Vision Benchmark Suite*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).

[81] **A. Geiger, M. Roser, and R. Urtasun**. *Efficient large-scale stereo matching*. In: *Asian Conference on Computer Vision* (2010).

[82] **A. Geiger, F. Moosmann, O. Car, and B. Schuster**. *A toolbox for automatic calibration of range and camera sensors using a single shot*. In: *International Conference on Robotics and Automation (ICRA)* (2012).

[83] **A. Geiger, P. Lenz, C. Stiller, and R. Urtasun**. *Vision meets robotics: The KITTI dataset*. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[84] **A. Gelman**. *A Bayesian formulation of exploratory data analysis and goodness–of–fit testing*. In: *International Statistical Review* 71.2 (2003), pp. 369–382.

[85] **M. Gey**. *Instrumentelle Analytik und Bioanalytik Biosubstanzen, Trennmethoden, Strukturanalytik, Applikationen*. 3. Auflage. Berlin and Heidelberg: Springer Spektrum, 2015.

[86] **J. Geyer *et al.*** *A2D2: Audi Autonomous Driving Dataset*. https://www.a2d2.audi/, access on 06.01.2022. 2020. arXiv preprint: 2004.06320v1.

[87] **I. Gheta, M. Heizmann, and J. Beyerer**. *Object oriented environment model for autonomous systems*. In: *Proceedings of the Second Skövde Workshop on Information Fusion Topics, Skövde Studies in Informatics* (2008), pp. 9–12.

[88] **I. Gheta, M. Heizmann, A. Belkin, and J. Beyerer**. *World modeling for autonomous systems. Lecture Notes in Computer Science (LNCS)*. In: *KI 2010: Advances in Artificial Intelligence* 6359 (2010), pp. 176–183.

[89] **R. Girshick**. *Fast R-CNN*. In: *IEEE International Conference on Computer Vision (ICCV)* (2015).

[90] **X. Glorot and Y. Bengio**. *Understanding the difficulty of training deep feedforward neural networks*. In: *Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256.

[91] **G. H. Golub and C. Reinsch**. *Singular value decomposition and least squares solutions*. Springer, 1971, pp. 134–151.

[92] **I. Goodfellow, Y. Bengio, and A. Courville**. *Deep learning*. Cambridge, Massachusetts and London, England: MIT press, 2016.

[93] **G. Görz, U. Schmid, and T. Braun**. *Handbuch der Künstlichen Intelligenz*. 6th ed. De Gruyter, 2020.

[94] **B. Graham, M. Engelcke, and L. van der Maaten**. *3D semantic segmentation with submanifold sparse convolutional networks*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 9224–9232.

[95] **J. Gräter, T. Strauss, and M. Lauer**. *Photometric laser scanner to camera calibration for low resolution sensors*. In: *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (2016), pp. 1552–1557.

[96] **J. Gräter, A. Wilczynski, and M. Lauer**. *LIMO: Lidar-Monocular Visual Odometry*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 7872–7879.

[97] **A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola**. *A kernel method for the two-sample-problem*. In: *Advances in Neural Information Processing Systems* 19 (2006), pp. 513–520.

[98] **A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola**. *A kernel two-sample test*. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[99] **Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun**. *Deep learning for 3D point clouds: A survey*. 2019. arXiv preprint: 1912.12033.

[100] **K. S. Gurumoorthy, A. Dhurandhar, and G. Cecchi**. *Protodash: Fast interpretable prototype selection*. 2017. arXiv preprint: 1707.01212.

[101] **K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi, and C. Aggarwal**. *Efficient data representation by selecting prototypes with importance weights*. In: *IEEE International Conference on Data Mining (ICDM)* (2019), pp. 260–269.

[102] **B. Hall**. *Lie Groups, Lie Algebras, and Representations*. XiV. New York: Springer, 2003.

[103] **M. Hall-Beyer**. *Practical guidelines for choosing GLCM textures to use in landscape classification tasks over a range of moderate spatial scales*. In: *International Journal of Remote Sensing* 38.5 (2017), pp. 1312–1338.

[104] **O. C. Hamsici and A. M. Martinez**. *Rotation invariant kernels and their application to shape analysis*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.11 (2009), pp. 1985–1999.

[105] **A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison**. *gvnn: Neural network library for geometric computer vision*. In: *European Conference on Computer Vision – ECCV 2016 Workshops* (2016).

[106] **C. Hane, L. Ladicky, and M. Pollefeys**. *Direction matters: Depth estimation with a surface normal classifier*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[107] **M. Hansard, S. Lee, O. Choi, and R. P. Horaud**. *Time-of-flight cameras: Principles, methods and applications*. Springer Science & Business Media, 2012.

[108] **R. M. Haralick and L. G. Shapiro**. *Image segmentation techniques*. In: *Computer Vision, Graphics, and Image Processing* 29.1 (1985), pp. 100–132.

[109] **G. Haskins, J. Kruecker, U. Kruger, S. Xu, P. A. Pinto, B. J. Wood, and P. Yan**. *Learning deep similarity metric for 3D MR-TRUS image registration*. In: *International journal of computer assisted radiology and surgery* 14.3 (2019), pp. 417–425.

[110] **R. Häusler and R. Klette**. *Disparity confidence measures on engineered and outdoor data*. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (2012), pp. 624–631.

[111] **R. Häusler and R. Klette**. *Evaluation of stereo confidence measures on synthetic and recorded image data*. In: *International Conference on Informatics, Electronics and Vision (ICIEV)* (2012), pp. 963–968.

[112] **K. He, J. Sun, and X. Tang**. *Guided image filtering*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (2012), pp. 1397–1409.

[113] **K. He, X. Zhang, S. Ren, and J. Sun**. *Deep residual learning for image recognition*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[114] **S. Heiler and P. Michels**. *Deskriptive und explorative Datenanalyse*. 1. Auflage. München: GRIN Verlag, 2020.

[115] **M. Heizmann, I. Gheta, F. Puente León, and J. Beyerer**. *Informationsfusion zur Umgebungsexploration*. In: *Verteilte Messsysteme*. KIT Scientific Publishing, 2010, pp. 133–152.

[116] **M. Heizmann, I. Gheta, F. Puente León, and J. Beyerer**. *Sensoreinsatzplanung und Informationsfusion zur Umgebungsexploration*. In: *tm - Technisches Messen* 77.10 (2010).

[117] **C. Hertzberg, R. Wagner, U. Frese, and L. Schröder**. *Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds*. In: *Information Fusion* 14.1 (2013), pp. 57–77.

[118] **H. Hirschmüller**. *Accurate and efficient stereo processing by semi-global matching and mutual information*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[119] **H. Hirschmüller**. *Stereo processing by semiglobal matching and mutual information*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2007), pp. 328–341.

[120] **H. Hirschmüller**. *Semi-global matching - motivation, developments and application*. In: *Photogrammetric Week* (2011), pp. 173–184.

[121] **A. E. Hodler, M. Needham, and J. Graham**. *Artificial intelligence & graph technology: enhancing AI with context & connections*. access on 26.01.2022. 2020. White Paper: \url{https://neo4j.com/whitepapers/artificial-intelligence-graph-technology/}.

[122] **J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell**. *Cycada: Cycle-consistent adversarial domain adaptation*. In: *International Conference on Machine Learning (ICML)* (2018), pp. 1989–1998.

[123] **D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke**. *Registration with the point cloud library: A modular framework for aligning in 3-D*. In: *IEEE Robotics and Automation Magazine* 22.4 (2015), pp. 110–124.

[124] **X. Hu and P. Mordohai**. *A quantitative evaluation of confidence measures for stereo vision*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2121–2133.

[125] **X. Hu and A. Ensor**. *Fourier spectrum image texture analysis*. In: *International Conference on Image and Vision Computing New Zealand (IVCNZ)* (2018), pp. 1–6.

[126] **X. Huang, J. Zhang, L. Fan, Q. Wu, and C. Yuan**. *A systematic approach for cross-source point cloud registration by preserving macro and micro structures*. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3261–3276.

[127] **K. Hughes and N. Ranganathan**. *A model for determining sensor confidence*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (1993), pp. 136–141.

[128] **K. F. Hughes**. *Sensor confidence in sensor integration tasks: a model for sensor performance measurement*. In: *Applications of Artificial Intelligence 1993: Machine Vision and Robotics* (1993), pp. 277–286.

[129] **F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer**. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size*. 2016. arXiv preprint: 1602.07360.

[130] **S. Ioffe and C. Szegedy**. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In: *32nd International Conference on Machine Learning* (2015), pp. 448–456.

[131] **G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna**. *CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks*. In: *International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 1110–1117.

[132] **A. K. Jain, J. Mao, and K. M. Mohiuddin**. *Artificial neural networks: A tutorial*. In: *Computer* 29.3 (1996), pp. 31–44.

[133] **M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez**. *xmuda: Cross-modal unsupervised domain adaptation for 3D semantic segmentation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 12605–12614.

[134] **S. Jayasuriya, A. Pediredla, S. Sivaramakrishnan, A. Molnar, and A. Veeraraghavan**. *Depth fields: Extending light field techniques to time-of-flight imaging*. In: *International Conference on 3D Vision (3DV)* (2015), pp. 1–9.

[135] **C. C. Jia, C. J. Wang, T. Yang, B. H. Fan, and F. G. He**. *A 3D point cloud filtering algorithm based on surface variation factor classification*. In: *Procedia Computer Science* 154 (2019), pp. 54–61.

[136] **B. Jian and B. C. Vemuri**. *Robust point set registration using Gaussian mixture models*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pp. 1633–1645.

[137] **P. Jiang and S. Saripalli**. *LiDARNet: A boundary-aware domain ddaptation model for point cloud semantic segmentation*. 2020. arXiv preprint: 2003.01174v3.

[138] **P. Jiang, P. Osteen, M. Wigness, and S. Saripalli**. *RELLIS-3D dataset: Data, benchmarks and analysis*. 2020. arXiv preprint: 2011.12954v3.

[139] **J. Jiao, R. Wang, W. Wang, S. Dong, Z. Wang, and W. Gao**. *Local stereo matching with improved matching cost and disparity refinement*. In: *IEEE MultiMedia* 4 (2014), pp. 16–27.

[140] **A. E. Johnson and M. Hebert**. *Using spin images for efficient object recognition in cluttered 3D scenes*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 433–449.

[141] **I. T. Jolliffe and J. Cadima**. *Principal component analysis: a review and recent developments*. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016).

[142] **K. I. Joy**. *Breshenham's algorithm*. In: *Visualization and Graphics Research Group, Department of Computer Science, University of California, Davis* (1999).

[143] **B. Julesz and J. R. Bergen**. *Human factors and behavioral science: Textons, the fundamental elements in preattentive vision and perception of textures*. In: *Bell System Technical Journal* 62.6 (1983), pp. 1619–1645.

[144] **D. Kahneman**. *Thinking, fast and slow*. PENGUIN UK, 2012.

[145] **D. Kahneman and G. Klein**. *Conditions for intuitive expertise: a failure to disagree*. In: *American Psychologist* 64.6 (2009), pp. 515–526.

[146] **J. Kallwies and H.-J. Wuensche**. *Effective combination of vertical and horizontal stereo vision*. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), pp. 1992–2000.

[147] **J. Kallwies, T. Engler, B. Forkel, and H.-J. Wuensche**. *Triple-SGM: Stereo processing using semi-global matching with cost fusion*. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2020), pp. 192–200.

[148] **S. Khan, H. Rahmani, and S. A. A. Shah**. *A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision*. Morgan & Claypool Publishers, 2018.

[149] **B. Kim, R. Khanna, and O. O. Koyejo**. *Examples are not enough, learn to criticize! criticism for interpretability*. In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 2280–2288.

[150] **P. Kim, J. Chen, and Y. K. Cho**. *SLAM-driven robotic mapping and registration of 3D point clouds*. In: *09265805* 89 (2018), pp. 38–48.

[151] **D. P. Kingma and J. Ba**. *Adam: A method for stochastic optimization*. In: *3rd International Conference for Learning Representations* (2015).

[152] **G. J. Klir and B. Yuan**. *Fuzzy sets and fuzzy logic: theory and applications*. In: *Possibility Theory versus Probability Theory* 32.2 (1996).

[153] **C. Klüver and J. Klüver**. *IT-Management durch KI-Methoden und andere naturanaloge Verfahren*. Wiesbaden: Vieweg+Teubner, 2011.

[154] **M. Kölle, D. Laupheimer, S. Schmohl, N. Haala, F. Rottensteiner, J. D. Wegner, and H. Ledoux**. *The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo*. In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* 1 (2021).

[155] **V. Kolmogorov and R. Zabih**. *Computing visual correspondence with occlusions using graph cuts*. In: *IEEE International Conference on Computer Vision (ICCV)* 2 (2001), pp. 508–515.

[156] **S. Kolski, D. Ferguson, M. Bellino, and R. Y. Siegwart**. *Autonomous driving in structured and unstructured environments*. In: *Intelligent Vehicles Symposium (IV)* (2006), pp. 558–563.

[157] **M. F. Kragh, P. Christiansen, M. S. Laursen, M. Larsen, K. A. Steen, O. Green, H. Karstoft, and R. N. Jørgensen**. *FieldSAFE: Dataset for obstacle detection in agriculture*. In: *Sensors* 17.11 (2017).

[158] **J. Kümmerle, T. Kühner, and M. Lauer**. *Automatic calibration of multiple cameras and depth sensors with a spherical target*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 1–8.

[159] **L. Landrieu and M. Simonovsky**. *Large-scale point cloud semantic segmentation with superpoint graphs*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 4558–4567.

[160] **F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss**. *Domain transfer for semantic segmentation of LiDAR data using deep neural networks*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), pp. 8263–8270.

[161] **S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller**. *Unmasking clever hans predictors and assessing what machines really learn*. In: *Nature communications* 10.1 (2019), pp. 1–8.

[162] **Y. LeCun, Y. Bengio, and G. Hinton**. *Deep learning*. In: *nature* 521.7553 (2015), p. 436.

[163] **K.-H. Lee, G. Ros, J. Li, and A. Gaidon**. *SPIGAN: Privileged adversarial learning from simulation*. In: *International Conference on Learning Representations (ICLR)* (2019).

[164] **M. Lehmann, C. Wittpahl, H. B. Zakour, and A. Braun**. *Modeling realistic optical aberrations to reuse existing drive scene recordings for autonomous driving validation*. In: *Journal of Electronic Imaging* 28.01 (2019).

[165] **J. Levinson and S. Thrun**. *Automatic online calibration of cameras and lasers*. In: *Robotics: Science and Systems IX* (2013), p. 7.

[166] **H. Li and L. Zhang**. *Multi-exposure fusion with CNN features*. In: *25th IEEE International Conference on Image Processing (ICIP)* (2018), pp. 1723–1727.

[167] **M. Li and P. Vitányi**. *An introduction to Kolmogorov complexity and its applications*. 2. ed. Graduate Texts in Computer Science. New York: Springer, 1997.

[168] **Y. Li, Y. Ruichek, and C. Cappelle**. *3D triangulation based extrinsic calibration between a stereo vision system and a LiDAR*. In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (2011).

[169] **T. Lillesand, R. W. Kiefer, and J. Chipman**. *Remote sensing and image interpretation*. 5th ed. John Wiley & Sons, 2004.

[170] **M. Lin, Q. Chen, and S. Yan**. *Network in network*. 2013. arXiv preprint: 1312.4400.

[171] **M. Lindauer and F. Hutter**. *Best practices for scientific research on neural architecture search*. In: *Journal of Machine Learning Research* 21.243 (2020), pp. 1–18.

[172] **H. Liu, Y. Liu, X. Gu, Y. Wu, F. Qu, and L. Huang**. *A deep-learning based multi-modality sensor calibration method for USV*. In: *IEEE Fourth International Conference on Multimedia Big Data (BigMM)* (2018), pp. 1–5.

[173] **S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot**. *Creating autonomous vehicle systems. Synthesis Lectures on Computer Science*. Vol. 6. Morgan & Claypool Publishers, 2017.

[174] **J. Long, E. Shelhamer, and T. Darrell**. *Fully convolutional networks for semantic segmentation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 3431–3440.

[175] **D. G. Lowe**. *Object recognition from local scale-invariant features*. In: *IEEE International Conference on Computer Vision (ICCV)* (1999), pp. 1150–1157.

[176] **D. G. Lowe**. *Distinctive image features from scale-invariant keypoints*. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[177] **W. Luo, A. G. Schwing, and R. Urtasun**. *Efficient deep learning for stereo matching*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5695–5703.

[178] **D. Lyons, A. Hekler, B. Noack, and U. D. Hanebeck**. *Maße für Wahrscheinlichkeitsdichten in der informationstheoretischen Sensoreinsatzplanung*. In: *Verteilte Messsysteme*. KIT Scientific Publishing, 2010.

[179] **J. Ma and G. Plonka**. *A review of curvelets and recent applications*. In: *IEEE Signal Processing Magazine* 27.2 (2010), pp. 118–133.

[180] **P. C. Mahalanobis**. *On the generalised distance in statistics*. In: *Proceedings of the National Institute of Science of India* 1 (1936), pp. 49–55.

[181] **S. Mani, A. Sankaran, S. Tamilselvam, and A. Sethi**. *Coverage testing of deep Learning models using dataset characterization*. 2019. arXiv preprint: 1911.07309.

[182] **F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman**. *Averaging quaternions*. In: *Journal of Guidance, Control, and Dynamics* 30.4 (2007), pp. 1193–1197.

[183] **L. E. Marks**. *The unity of the senses: Interrelations among the modalities*. Academic Press, 2014.

[184] **D. Marr and E. Hildreth**. *Theory of edge detection*. eng. In: *0950-1193* 207.1167 (1980), pp. 187–217.

[185] **C. R. Maurer, R. Qi, and V. Raghavan**. *A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.2 (2003).

[186] **J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale**. *Online self-calibration for robotic systems*. In: *The International Journal of Robotics Research* (2016).

[187] **N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox**. *A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4040–4048.

[188] **N. Mellado, D. Aiger, and N. J. Mitra**. *Super 4pcs fast global pointcloud registration via smart indexing*. In: *Computer Graphics Forum* 33.5 (2014).

[189] **N. Mellado, M. Dellepiane, and R. Scopigno**. *Relative scale estimation and 3D registration of multi-modal geometry using Growing Least Squares*. In: *1077-2626* 22.9 (2015), pp. 2160–2173.

[190] **H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, M. Urban, M. Burkart, M. Dippel, M. Lindauer, and F. Hutter**. *Towards automatically-tuned deep neural networks*. In: *Automated Machine Learning* (2019), pp. 135–149.

[191] **M. Menze and A. Geiger**. *Object scene flow for autonomous vehicles*. In: *IEEE International Conference on Computer Vision (ICCV)* (2015).

[192] **T. Mertens, J. Kautz, and F. van Reeth**. *Exposure fusion*. In: *15th Pacific Conference on Computer Graphics and Applications (PG'07)* (2007), pp. 382–390.

[193] **K. A. Metzger, P. Mortimer, and H.-J. Wuensche**. *A fine-grained dataset and its efficient semantic segmentation for unstructured driving scenarios*. In: *25th International Conference on Pattern Recognition (ICPR)* (2020).

[194] **J.-A. Meyer and D. Filliat**. *Map-based navigation in mobile robots: a review of map-learning and path-planning strategies*. In: *Cognitive Systems Research* 4 (2003).

[195] **S. Miao, Z. J. Wang, and R. Liao**. *A CNN regression approach for real-time 2D/3D registration*. In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1352–1363.

[196] **A. Milioto, I. Vizzo, J. Behley, and C. Stachniss**. *RangeNet++: Fast and accurate LiDAR semantic segmentation*. In: *IEEE/RSJ Internal Conference on Intelligent Robots and Systems (IROS)* (2019).

[197] **M. L. Minsky**. *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. PhD thesis. Princeton University, 1954.

[198] **R. Monica, J. Aleotti, M. Zillich, and M. Vincze**. *Multi-label point cloud annotation by selection of sparse control points*. In: *IEEE International Conference on 3D Vision (3DV)* (2017), pp. 301–308.

[199] **A. Motten and L. Claesen**. *Low-cost real-time stereo vision hardware with binary confidence metric and disparity refinement*. In: *IEEE International Conference on Multimedia Technology (ICMT)* (2011), pp. 3559–3562.

[200] **R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos**. *ORB-SLAM: a versatile and accurate monocular SLAM system*. In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163.

[201] **R. Mur-Artal and J. D. Tardós**. *ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras*. In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262.

[202] **A. Myronenko and X. Song**. *Point set registration: Coherent point drift*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), pp. 2262–2275.

[203] **G. Nam and M. H. Kim**. *Multispectral photometric stereo for acquiring high-fidelity surface normals*. In: *IEEE Computer Graphics and Applications* 34.6 (2014), pp. 57–68.

[204] **B. Naujoks, P. Burger, and H.-J. Wuensche**. *Combining deep learning and model-based methods for robust real-time semantic landmark detection*. In: *22nd International Conference on Information Fusion (FUSION)* (2019).

[205] **K. Ozawa, I. Sato, and M. Yamaguchi**. *Hyperspectral photometric stereo for a single capture*. In: *Journal of the Optical Society of America, Optics, Image Science, and Vision* 34.3 (2017), pp. 384–394.

[206] **G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice**. *Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information*. In: *Twenty-Sixth AAAI Conference on Artificial Intelligence* 1 (2012).

[207] **G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice**. *Automatic extrinsic calibration of vision and lidar by maximizing mutual information*. In: *Journal of Field Robotics* 32.5 (2015), pp. 696–722.

[208] **J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan**. *Cascade residual learning: A two-stage convolutional neural network for stereo matching*. In: *IEEE/CVF International Conference on Computer Vision Workshops (ICCV Workshops)* 7 (2017).

[209] **J. Papon, A. Abramov, M. Schoeler, and F. Worgotter**. *Voxel cloud connectivity segmentation-supervoxels for point clouds*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 2027–2034.

[210] **K. Park, S. Kim, and K. Sohn**. *High-precision depth estimation using uncalibrated LiDAR and stereo fusion*. In: *IEEE Transactions on Intelligent Transportation Systems* (2019).

[211] **W. J. Park and J.-B. Park**. *History and application of artificial neural networks in dentistry*. In: *European Journal of Dentistry* 12.4 (2018), pp. 594–601.

[212] **Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim**. *Calibration between color camera and 3D LiDAR instruments with a polygonal planar board*. In: *Sensors* 14.3 (2014), pp. 5333–5353.

[213] **M. Pauly, R. Keiser, and M. Gross**. *Multi–scale feature extraction on point–sampled surfaces*. In: *Computer Graphics Forum* 22.3 (2003), pp. 281–289.

[214] **N. Pears, Y. Liu, and P. Bunting**. *3D imaging, analysis and applications*. Vol. 3. Springer, 2012.

[215] **J. Petereit**. *Adaptive state × time lattices: A contribution to mobile robot motion planning in unstructured dynamic environments*. PhD thesis. 2017.

[216] **J. Petereit, J. Beyerer, T. Asfour, S. Gentes, B. Hein, U. D. Hanebeck, F. Kirchner, R. Dillmann, H. H. Götting, and M. Weiser**. *ROBDEKON: Robotic systems for decontamination in hazardous environments*. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (2019).

[217] **O. Pfungst**. *Clever Hans: (the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart and Winston, 1911.

[218] **M. Poggi, G. Agresti, F. Tosi, P. Zanuttigh, and S. Mattoccia**. *Confidence estimation for ToF and stereo sensors and its application to depth data fusion*. In: *IEEE Sensors Journal* 20.3 (2019), pp. 1411–1421.

[219] **A. Pujol-Miro, J. Ruiz-Hidalgo, and J. R. Casas**. *Registration of images to unorganized 3D point clouds using contour cues*. In: *25th European Signal Processing Conference (EUSIPCO)* (2017), pp. 81–85.

[220] **C. R. Qi, H. Su, K. Mo, and L. J. Guibas**. *Pointnet: Deep learning on point sets for 3D classification and segmentation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 652–660.

[221] **C. R. Qi, L. Yi, H. Su, and L. J. Guibas**. *Pointnet++: Deep hierarchical feature learning on point sets in a metric space*. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (2017), pp. 5099–5108.

[222] **U. Ramer**. *An iterative procedure for the polygonal approximation of plane curves*. In: *0146-664X* 1.3 (1972). http://www.sciencedirect.com/science/article/pii/S0146664X72800170, access on 05.01.2022, pp. 244–256.

[223] **R. Ranftl, S. Gehrig, T. Pock, and H. Bischof**. *Pushing the limits of stereo using variational stereo estimation*. In: *IEEE Intelligent Vehicles Symposium (IV)* (2012).

[224] **A. Ranjan, J. Janai, A. Geiger, and M. J. Black**. *Attacking optical flow*. In: *IEEE International Conference on Computer Vision (ICCV)* (2019), pp. 2404–2413.

[225] **J. Redmon and A. Farhadi**. *YOLO9000: better, faster, stronger*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[226] **O. Reinoso and L. Payá**. *Special issue on mobile robots navigation*. In: *Applied Sciences* 10.4 (2020), p. 1317.

[227] **K. Retan, F. Loshaj, and M. Heizmann**. *Radar odometry on SE(3) with constant velocity motion prior*. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6386–6393.

[228] **G. Riegler, A. Osman Ulusoy, and A. Geiger**. *Octnet: Learning deep 3D representations at high resolutions*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 3577–3586.

[229] **M. A. Robertson, S. Borman, and R. L. Stevenson**. *Dynamic range improvement through multiple exposures*. In: *IEEE International Conference on Image Processing (ICIP)* 3 (1999), pp. 159–163.

[230] **E. Rosten and T. Drummond**. *Machine learning for high-speed corner detection*. In: *European Conference on Computer Vision (ECCV)* 3951 (2006), pp. 430–443.

[231] **E. Rosten, R. Porter, and T. Drummond**. *Faster and better: A machine learning approach to corner detection*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.1 (2010), pp. 105–119.

[232] **E. Rublee, V. Rabaud, K. Konolige, and G. Bradski**. *ORB: An efficient alternative to SIFT or SURF*. In: *IEEE international conference on computer vision (ICCV)* (2011), pp. 2564–2571.

[233] **O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei**. *ImageNet large scale visual recognition challenge*. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[234] **R. B. Rusu, N. Blodow, and M. Beetz**. *Fast Point Feature Histograms (FPFH) for 3D registration*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2009), pp. 3212–3217.

[235] **R. B. Rusu, N. Blodow, and M. Beetz**. *Persistent point feature histograms for 3D point clouds*. In: *10th International Conference on Intelligent Autonomous Systems (IAS-10)* (2008).

[236] **W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller**. *Evaluating the visualization of what a deep neural network has learned*. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11 (2016), pp. 2660–2673.

[237] **A. D. Sappa, J. A. Carvajal, C. A. Aguilera, M. Oliveira, D. Romero, and B. X. Vintimilla**. *Wavelet-based visible and infrared image fusion: A comparative study*. In: *Sensors* 16.6 (2016).

[238] **D. Schacter**. *Psychology*. New York: Worth Publishers, 2011.

[239] **D. Scharstein and C. Pal**. *Learning conditional random fields for stereo*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2007), pp. 1–8.

[240] **D. Scharstein and R. Szeliski**. *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*. In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 7–42.

[241] **D. Scharstein and R. Szeliski**. *High-accuracy stereo depth maps using structured light*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 1 (2003).

[242] **D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling**. *High-resolution stereo datasets with subpixel-accurate ground truth*. In: *German Conference on Pattern Recognition (GCPR)* (2014), pp. 31–42.

[243] **D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling**. *High-resolution stereo datasets with subpixel-accurate ground truth*. In: *Lecture Notes in Computer Science book series (LNCS)* 8753 (2014).

[244] **R. M. Scheffel and A. A. Fröhlich**. *Increasing sensor reliability through confidence attribution*. In: *Journal of the Brazilian Computer Society* 25.1 (2019), pp. 1–20.

[245] **F. Schimpf**. *Fusion of LiDAR and stereo point clouds using Bayesian networks*. Master's thesis. Georg-August-Universität Goettingen, 2018.

[246] **N. Schneider, F. Piewak, C. Stiller, and U. Franke**. *Regnet: Multimodal sensor registration using deep neural networks*. In: *IEEE Intelligent Vehicles Symposium (IV)* (2017), pp. 1803–1810.

[247] **S. Schneider, T. Luettel, and H.-J. Wuensche**. *Odometry-based online extrinsic sensor calibration*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2013), pp. 1287–1292.

[248] **H. Schulz-Mirbach**. *On the existence of complete invariant feature spaces in pattern recognition*. In: *11th IAPR International Conference on Pattern Recognition* (1992), pp. 178–182.

[249] **H. Schulz-Mirbach**. *Constructing invariant features by averaging techniques*. In: *12th IAPR International Conference on Pattern Recognition* (1994), pp. 387–390.

[250] **A. V. Segal, D. Haehnel, and S. Thrun**. *Generalized-ICP*. In: *Robotics: Science and Systems* 2.4 (2009).

[251] **A. Seki and M. Pollefeys**. *Patch based confidence prediction for dense disparity map*. In: *British Machine Vision Conference* (2016).

[252] **R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra**. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. In: *IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 618–626.

[253] **C. M. Shakarji**. *Least-squares fitting algorithms of the NIST algorithm testing system*. In: *Journal of Research of the National Institute of Standards and Technology* 103.6 (1998), pp. 633–641.

[254] **C. E. Shannon**. *A mathematical theory of communication*. In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[255] **W. Shi and C. Cheung**. *Performance evaluation of line simplification algorithms for vector generalization*. In: *The Cartographic Journal* 43.1 (2006).

[256] **M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand**. *RTSeg: Real-time semantic segmentation comparative study*. In: *IEEE International Conference on Image Processing (ICIP)* (2018).

[257] **N. Silberman, D. Hoiem, P. Kohli, and R. Fergus**. *Indoor segmentation and support inference from RGBD images*. In: *European Conference on Computer Vision (ECCV)* (2012).

[258] **H. A. Simon**. *What is an explanation of behavior?* In: *Psychological Science* 3.3 (1992), pp. 150–161.

[259] **T. Stathaki**. *Image fusion: Algorithms and applications*. Elsevier Academic Press, 2008.

[260] **R. Storn and K. Price**. *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*. In: *Journal of Global Optimization* 11.4 (1997), pp. 341–359.

[261] **H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz**. *Splatnet: Sparse lattice networks for point cloud processing*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 2530–2539.

[262] **D. Sun, X. Yang, M.-Y. Liu, and J. Kautz**. *Pwc-net: CNNs for optical flow using pyramid, warping, and cost volume*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 8934–8943.

[263] **P. Sun *et al.*** *Scalability in perception for autonomous driving: Waymo Open Dataset*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[264] **D.-W. Sun**. *Hyperspectral imaging for food quality analysis and control*. Elsevier, 2010.

[265] **C. Sünkenberg**. *Echtzeitfähige Parallelisierung, GPU-Implementierung und Evaluation eines Stereokorrespondenzalgorithmus auf multispektralen Bilddaten*. Student Research Project. Karlsruhe Institute of Technology, 2017.

[266] **T. Tao, J. C. Koo, and H. R. Choi**. *A fast block matching algorthim for stereo correspondence*. In: *IEEE Conference on Cybernetics and Intelligent Systems* (2008), pp. 38–41.

[267] **L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese**. *Segcloud: Semantic segmentation of 3D point clouds*. In: *IEEE International conference on 3D vision (3DV)* (2017), pp. 537–547.

[268] **S. Thrun, W. Burgard, and D. Fox**. *Probalistic robotics*. MIT press, 2006.

[269] **A. Torsello, E. Rodola, and A. Albarelli**. *Multiview registration via graph diffusion of dual quaternions*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 2441–2448.

[270] **Y. Touati, A. Ali-Chérif, and B. Daachi**. *Energy management in wireless sensor networks*. 2017.

[271] **M. Trierscheid, J. Pellenz, D. Paulus, and D. Balthasar**. *Hyperspectral imaging or victim detection with rescue robots*. In: *IEEE International Workshop on Safety, Security and Rescue Robotics* (2008), pp. 7–12.

[272] **J. W. Tukey**. *Exploratory data analysis*. New York: Addison-Wesley, 1977.

[273] **A. M. Turing**. *On computable numbers, with an application to the Entscheidungsproblem*. In: *Proceedings of the London Mathematical Society* 42 (1936), pp. 230–265.

[274] **A. M. Turing**. *Mind*. In: *Mind* 59.236 (1950), pp. 433–460.

[275] **D. Uhlig and M. Heizmann**. *Multi-Stereo-Deflektometrie mit einer Lichtfeldkamera / Multi-stereo deflectometry with a light-field camera*. In: *tm - Technisches Messen* 85.s1 (2018).

[276] **D. Uhlig and M. Heizmann**. *Model-independent light field reconstruction using a generic camera calibration*. In: *tm - Technisches Messen* 88.6 (2021), pp. 361–373.

[277] **O. Unal, L. van Gool, and D. Dai**. *Improving point cloud semantic segmentation by learning 3D object detection*. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2021).

[278] **A. Valada, G. Oliveira, T. Brox, and W. Burgard**. *Deep multispectral semantic scene understanding of forested environments using multimodal fusion*. In: *International Symposium on Experimental Robotics (ISER)* (2016).

[279] **J. Veitch-Michaelis**. *Fusion of LIDAR with stereo camera data - an assessment*. https://discovery.ucl.ac.uk/1536083/1/thesis.pdf, access on 03.01.2022. PhD thesis. University College London, 2016.

[280] **M. Velas, M. Španěl, Z. Materna, and A. Herout**. *Calibration of RGB camera with Velodyne LiDAR*. In: *International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)* (2014), pp. 135–144.

[281] **R. O. H. Veld, T. Jaschke, M. Bätz, L. Palmieri, and J. Keinert**. *A novel confidence measure for disparity maps by pixel-wise cost function analysis*. In: *25th IEEE International Conference on Image Processing (ICIP)* (2018), pp. 644–648.

[282] **K. Vinogradova, A. Dibrov, and G. Myers**. *Towards interpretable semantic segmentation via gradient-weighted class activation mapping*. 2020. arXiv preprint: 2002.11434.

[283] **G. Vinué, A. Simó, and S. Alemany**. *The k-means algorithm for 3D shapes with an application to apparel design*. In: *Advances in Data Analysis and Classification* 10.1 (2016), pp. 103–132.

[284] **D. G. Viswanathan**. *Features from accelerated segment test (FAST)*. In: *10th workshop on Image Analysis for Multimedia Interactive Services* (2009).

[285] **V. Volkov**. *Understanding Latency Hiding on GPUs*. Technical Report No. UCB/EECS-2016-143. http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-143.html, access on 20.04.2022.

[286] **J.-D. Wang and H.-C. Liu**. *An approach to evaluate the fitness of one class structure via dynamic centroids*. In: *Expert Systems with Applications* 38.11 (2011), pp. 13764–13772.

[287] **X. Wang, M. Christie, and E. Marchand**. *Multiple layers of contrasted images for robust feature-based visual tracking*. In: *25th IEEE International Conference on Image Processing (ICIP)* (2018), pp. 241–245.

[288] **Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli**. *Image quality assessment: From error visibility to structural similarity*. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[289] **Z. Wang and F. Lu**. *VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes*. In: *IEEE Transactions on Visualization and Computer Graphics* 26.9 (2019).

[290] **M. Weiler, P. Forré, E. Verlinde, and M. Welling**. *Coordinate independent convolutional networks – isometry and gauge equivariant convolutions on Riemannian manifolds*. 2021. arXiv preprint: 2106.06020v1.

[291] **W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis**. *Multi-modal volume registration by maximization of mutual information*. In: *Medical image analysis* 1.1 (1996), pp. 35–51.

[292] **J. Wendel**. *Integrierte Navigationssysteme*. Oldenbourg Wissenschaftsverlag, 2011.

[293] **C. Wheatstone**. *XVIII. Contributions to the physiology of vision. Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision*. In: *Philosophical Transactions of the Royal Society of London* 128 (1838), pp. 371–394.

[294] **C. Wittpahl, H. B. Zakour, M. Lehmann, and A. Braun**. *Realistic image degradation with measured PSF*. In: *Electronic Imaging* 2018.17 (2018).

[295] **W. Wohlkinger and M. Vincze**. *Ensemble of shape functions for 3d object classification*. In: *IEEE International Conference on Robotics and Biomimetics* (2011), pp. 2987–2992.

[296] **P. Wolf and K. Berns**. *Data-fusion for robust off-road perception considering data quality of uncertain sensors*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021).

[297] **P. Wolf, T. Groll, S. Hemer, and K. Berns**. *Evolution of robotic simulators: Using UE4 to enable real-world quality testing of complex autonomous robots in unstructured environments*. In: *International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)* (2020), pp. 271–278.

[298] **B. Wu, A. Wan, X. Yue, and K. Keutzer**. *SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud*. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2018), pp. 1887–1893.

[299] **B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer**. *Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud*. In: *International Conference on Robotics and Automation (ICRA)* (2019), pp. 4376–4382.

[300] **C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka**. *SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation*. In: *European Conference on Computer Vision (ECCV)* 12373 (2020), pp. 1–19.

[301] **K. Yamaguchi, D. McAllester, and R. Urtasun**. *Efficient joint segmentation, occlusion labeling, stereo and flow estimation*. In: *European Conference on Computer Vision (ECCV)* (2014), pp. 756–771.

[302] **Y. Yang, A. Yuille, and J. Lu**. *Local, global, and multilevel stereo matching*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (1993).

[303] **C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang**. *BiSeNet: Bilateral segmentation network for real-time semantic segmentation*. In: *European Conference on Computer Vision (ECCV)* (2018), pp. 325–341.

[304] **F. Yu, D. Wang, E. Shelhamer, and T. Darrell**. *Deep layer aggregation*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[305] **F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell**. *BDD100K: A diverse driving dataset for heterogeneous multitask learning*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 2636–2645.

[306] **M. N. Zafar and J. C. Mohanta**. *Methodology for path planning and optimization of mobile robots: A review*. In: *Procedia Computer Science* 133 (2018), pp. 141–152.

[307] **S. Zagoruyko and N. Komodakis**. *Learning to compare image patches via convolutional neural networks*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[308] **P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo**. *Time-of-flight and structured light depth cameras*. In: *Technology and Applications* (2016).

[309] **J. Zbontar and Y. LeCun**. *Computing the stereo matching cost with a convolutional neural network*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[310] **J. Zbontar and Y. LeCun**. *Stereo matching by training a convolutional neural network to compare image patches*. In: *Journal of Machine Learning Research* 17.1 (2016), pp. 2287–2318.

[311] **Z. Zhang**. *Iterative point matching for registration of free-form curves and surfaces*. In: *International Journal of Computer Vision* (1994).

[312] **B. Zhao**. *Understanding sources of variation to improve the reproducibility of radiomics*. In: *Frontiers in Oncology* 11 (2021), p. 826.

[313] **S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr**. *Conditional random fields as recurrent neural networks*. In: *IEEE/CVF International Conference on Computer Vision (ICCV)* (2015), pp. 1529–1537.

[314] **F. Zhou and F. de La Torre**. *Factorized graph matching*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 127–134.

[315] **Q.-Y. Zhou, J. Park, and V. Koltun**. *Open3D: A modern library for 3D data processing*. http://www.open3d.org, access on 05.01.2022. 2018. arXiv preprint: 1801.09847v1.

[316] **W. Zimmer, A. Rangesh, and M. Trivedi**. *3D BAT: A semi-automatic, web-based 3D annotation toolbox for full-surround, multi-modal data streams*. 2019. arXiv preprint: 1905.00525v1.

[317] **A. R. Zubair and O. A. Alo**. *Grey level co-occurrence matrix (GLCM) cased second order statistics for image texture analysis*. In: *International Journal of Science and Engineering Investigations* 8.93 (2019), pp. 64–73.

List of Publications

[318] **A. Albrecht and N. Heide**. *Improving stereo vision based SLAM by integrating inertial measurements for person indoor navigation*. In: *4th International Conference on Control, Automation and Robotics (ICCAR)* (2018).

[319] **A. Albrecht and N. Heide**. *Mapping and automatic post-processing of indoor environments by extending visual SLAM*. In: *International Conference on Audio, Language and Image Processing (ICALIP)* (2018).

[320] **A. Albrecht and N. F. Heide**. *Improving feature-based visual SLAM in Person Indoor Navigation with HDR Imaging*. In: *International Conference on Information Communication and Signal Processing (ICICSP)* (2019).

[321] **A. Albrecht, N. F. Heide, C. Frese, and A. Zube**. *Generic convoying functionality for autonomous vehicles in unstructured outdoor environments*. In: *IEEE Intelligent Vehicles Symposium (IV)* (2020).

[322] **N. Heide**. *Calibration of multiple 3D laser scanners to a common vehicle coordinate system*. Master's thesis. Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 2017.

[323] **N. Heide, T. Emter, and J. Petereit**. *Calibration of multiple 3D LiDAR sensors to a common vehicle frame*. In: *50th International Symposium on Robotics (ISR)* (2018).

[324] **N. Heide, C. Frese, T. Emter, and J. Petereit**. *Real-time hyperspectral stereo processing for the generation of 3D depth information*. In: *IEEE International Conference on Image Processing (ICIP)* (2018).

[325] **N. F. Heide, A. Albrecht, and M. Heizmann**. *SET: Stereo evaluation toolbox for combined performance assessment of camera systems, 3D reconstruction and visual SLAM*. In: *International Conference on Information Communication and Signal Processing (ICICSP)* (2019).

[326] **N. F. Heide, A. Albrecht, and M. Heizmann**. *A step towards explainable artificial neural networks in image processing by dataset assessment*. In: *Forum Bildverarbeitung* (2020).

[327] **N. F. Heide, S. Gamer, and M. Heizmann**. *UEM-CNN: Enhanced stereo matching for unstructured environments with dataset filtering and novel error metrics*. In: *52nd International Symposium on Robotics (ISR)* (2020).

[328] **N. F. Heide, A. Albrecht, T. Emter, and J. Petereit**. *Performance optimization of autonomous platforms in unstructured outdoor environments using a novel constrained planning approach*. In: *IEEE Intelligent Vehicles Symposium (IV)* (2019).

[329] **N. F. Heide, P. Woock, M. Sauer, T. Leitritz, and M. Heizmann**. *UCSR: Registration and fusion of cross-source 2D and 3D sensor data in unstructured environments*. In: *23rd International Conference on Information Fusion (FUSION)* (2020).

[330] **N. F. Heide, E. Müller, J. Petereit, and M. Heizmann**. *X^3Seg: Model-agnostic explanations for the semantic segmentation of 3D point clouds with prototypes and criticism*. In: *IEEE International Conference on Image Processing (ICIP)* (2021).

[331] **C. Walther, C. Eisenhut, M. Etzhold, J. Petereit, C. Frese, K. Pfeiffer, N. F. Heide, M. Aboubakr, T. Emter, A. Albrecht, A. Zube, and A. Wenzel**. *AKIT - Teilprojekt: Intelligente Unterstützungsfunktionen für Navigation und Manipulation für mobile Arbeitsmaschinen: Verbundprojekt: Autonomie-KIT für seriennahe Arbeitsfahrzeuge zur vernetzten und assistierten Bergung von Gefahrenquellen*. In: *Zivile Sicherheit - Innovative Rettungs- und Sicherheitssysteme: Schlussbericht* (2020). https://www.tib.eu/de/suchen/id/TIBKAT:1734046082/, access on 25.01.2022.

[332] **P. Woock, N. F. Heide, and D. Kühn**. *Robotersysteme für die Dekontamination in menschenfeindlichen Umgebungen*. In: *Leipziger Deponiefachtagung (LDFT-2020)* (2020).

List of Supervised Theses

[333] **S. Gamer**. *3D reconstruction of unstructured environments using convolutional neural networks for stereo matching*. Master's thesis. Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 2019.

[334] **C. Hanisch**. *Sensor data confidence assessment*. Master's thesis. Pforzheim University, 2022.

[335] **T. Leitritz**. *Registration of camera images and 3D LiDAR data using neural networks, Registrierung von Kamerabildern und LiDAR-Punktwolken unstrukturierter Umgebungen mithilfe von neuronalen Netzen*. Master's thesis. Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 2020.

[336] **E. Müller**. *A contribution to the explainability of an AI-based semantic segmentation for 3D point clouds*. Bachelor's thesis. Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 2021.

[337] **H. Padusinski**. *Optimization of point cloud based acquisition of unstructured environments for mobile robots using AI-based semantic segmentation*. Master's thesis. Technische Hochschule Nürnberg Georg Simon Ohm, 2020.

[338] **M. Sauer**. *Registration and fusion of camera images and depth information for 3D reconstruction of unstructured environments*. Master's thesis. Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 2020.

[339] **J. Schmidl**. *A contribution to the domain transfer of artificial neural networks for the semantic segmentation of 3D point clouds*. Master's thesis. Pforzheim University, 2021.