Karlsruher Schriften zur Anthropomatik

Band 44

Sebastian Bullinger

Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion

KIT Scientific Publishing

Sebastian Bullinger

Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion

Karlsruher Schriften zur Anthropomatik
Band 44
Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion

by
Sebastian Bullinger

KIT Scientific Publishing

Karlsruher Institut für Technologie
Institut für Anthropomatik und Robotik

Image-Based 3D Reconstruction of Dynamic Objects
Using Instance-Aware Multibody Structure from Motion

Zur Erlangung des akademischen Grades eines Doktor-Ingenieurs
von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Sebastian Bullinger

Tag der mündlichen Prüfung: 25. November 2019
Hauptreferent: Prof. Dr.-Ing. habil. Rainer Stiefelhagen
Korreferent: Prof. Dr.-Ing. habil. Jürgen Beyerer

Impressum

KIT Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2020 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489
ISBN 978-3-7315-1012-3
DOI 10.5445/KSP/1000105589

Abstract

Computing three-dimensional reconstructions of *dynamic* scenes is one of the fundamental problems in computer vision. For many applications this task can be reduced to the determination of three-dimensional object motion trajectories w.r.t. mainly static environment structures. This approach simplifies the reconstruction problem by constraining projective ambiguities of different scene components.

Image-based reconstruction approaches such as *Multibody Structure from Motion (MSfM)* represent an appealing choice to reconstruct dynamic scenes given suitable conditions like sufficiently textured surfaces and non-degenerated camera trajectories. The underlying assumption of MSfM is that the scene may be represented by a *multibody system*, *i.e.*, that the scene consists of multiple non-deformable components, which may undergo independent translational and rotational displacements.

Existing MSfM approaches use epipolar constraints or motion segmentation to determine component specific feature correspondences to reconstruct independently moving components. Such methods are agnostic to semantics and fail in certain scenarios like stationary or parallel moving objects. It is difficult to identify capabilities and limitations of existing approaches, because of the lack of image-based dynamic object reconstruction baseline algorithms and benchmark datasets.

We propose a novel MSfM algorithm for moving object reconstruction that incorporates *(instance-aware) semantic segmentation* and *multiple view geometry* methods. The proposed MSfM pipeline includes a *Multiple Object Tracking (MOT)* algorithm that tracks two-dimensional object shapes on pixel level to determine object specific feature correspondences. We consider non-object structures for the environment reconstruction.

The proposed MSfM method allows the reconstruction of three-dimensional object shapes and object motion trajectories. We leverage camera poses w.r.t. object reconstructions and corresponding instance-aware semantic segmentations to determine object points consistent with image observations. The generated point clouds are suitable for object mesh computations. In order to compute a three-dimensional object trajectory we combine corresponding camera poses in the object and in the background reconstruction. We present different algorithms to reconstruct object motion trajectories in monocular and stereo image sequences. In the monocular case, three-dimensional object trajectories are defined up to scale. In order to resolve this ambiguity, we propose two different constraints to estimate the scale ratio between object and environment reconstructions.

To facilitate the benchmarking of new and existing approaches, we additionally created two publicly available datasets for moving object reconstruction. The first dataset comprises real-world image sequences of a moving vehicle and a corresponding vehicle laser scan suitable for evaluation of object shape reconstructions. The second dataset contains synthetic sequences of different vehicles in an urban environment. The ground truth includes vehicle shapes as well as vehicle and camera poses per frame. This dataset allows to quantitatively evaluate shape and trajectory reconstructions of moving objects.

Using the created datasets, we evaluate our algorithms on outdoor scenarios of driving vehicles with challenging properties such as small object sizes, reflecting surfaces as well as illumination and view dependent appearance changes. We show that the proposed semantic constraint for object shape reconstruction produces meshes that are robust w.r.t. reflections and appearance changes. The quantitative evaluation of the trajectory reconstruction algorithms shows that the scale ambiguity of (monocular) image-based reconstructions poses a challenging problem. The usage of stereo image sequences resolves this ambiguity and results in more accurate and robust reconstructions. By quantitatively evaluating the proposed algorithms on our datasets we provide a reference for future research in the area of moving object reconstruction.

Zusammenfassung

Das Berechnen von dreidimensionalen Rekonstruktionen *dynamischer* Szenen ist eines der grundlegenden Probleme im Bereich des Maschinellen Sehens. Für viele Anwendungen kann diese Aufgabe auf eine Bestimmung der dreidimensionalen Bewegungsbahnen von Objekten bzgl. einer primär statischen Umgebung reduziert werden. Dieser Ansatz vereinfacht das Rekonstruktionsproblem durch eine Einschränkung der projektiven Mehrdeutigkeiten der verschiedenen Szenenkomponenten.

Multibody Structure from Motion (MSfM) Ansätze erlauben, unter geeigneten Bedingungen wie beispielsweise nicht-degenerierten Kamerabewegungen und ausreichend strukturierten Oberflächen, dynamische Szenen zu rekonstruieren. Die zugrundeliegende Annahme von MSfM ist, dass eine Szene durch ein *Mehrkörpersystem* dargestellt werden kann, d.h. die Szene besteht aus mehreren nicht-deformierbaren Komponenten, welche unabhängige Translationen bzw. Rotationen aufweisen können.

Existierende MSfM Ansätze nutzen beispielsweise Bewegungssegmentierungen oder Zwangsbedingungen der Epipolargeometrie, um komponentenspezifische Merkmalskorrespondenzen zu bestimmen und damit unabhängig bewegende Komponenten zu rekonstruieren. Diese Methoden erfassen keine Semantik und scheitern in bestimmten Szenarien wie beispielsweise stationäre oder parallel bewegende Objekte. Aufgrund von fehlenden bildbasierten Referenzalgorithmen für die Rekonstruktion dynamischer Objekte und entsprechenden Datensätzen ist es schwierig Fähigkeiten und Einschränkungen existierender Verfahren zu identifizieren.

Wir präsentieren einen neuartigen MSfM Algorithmus zur Rekonstruktion dynamischer Objekte, welcher *(instanzbewusste) semantische Segmentierungen* and *Multiple View Geometry* Methoden einbindet. Die vorgeschlagene MSfM Pipeline schließt eine Komponente zur Verfolgung von mehreren Objekten ein, welche es erlaubt, zweidimensionale Objektformen auf Pixelebene zu verfolgen und damit objektspezifische Merkmalskorrespondenzen zu bestimmen. Um die Umgebung zu rekonstruieren, werden Merkmale verwendet, die nicht

einem Objekt zugeordnet sind. Die vorgeschlagene MSfM Methode erlaubt dreidimensionale Objektformen und Objekttrajektorien zu rekonstruieren.
Wir nutzen Kameraposen der Objektrekonstruktion und zugehörige instanzbewusste semantische Segmentierungen, um Objektpunkte zu bestimmen, die konsistent zu Bildbeobachtungen sind. Die so generierte Punktwolke ist geeignet, um ein Dreiecksnetz des Objektes zu berechnen.
Um dreidimensionale Objekttrajektorien zu berechnen, kombinieren wir zusammengehörige Kameraposen in den Objekt- und Hintergrundrekonstruktionen. Wir präsentieren verschiedene Algorithmen, um Objekttrajektorien in monokularen und binokularen Bildsequenzen zu rekonstruieren. Im monokularen Fall sind die dreidimensionalen Objekttrajektorien bis auf die Skalierung eindeutig definiert. Um diese Mehrdeutigkeit aufzulösen schlagen wir zwei verschiedene Nebenbedingungen vor, welche es erlauben, das Skalenverhältnis zwischen Objekt- und Umgebungsrekonstruktionen zu bestimmen.
Im Rahmen dieser Arbeit wurden zwei öffentlich verfügbare Datensätze geschaffen, welche es ermöglichen, neue und existierende Algorithmen zur Rekonstruktion bewegter Objekte zu evaluieren. Der erste Datensatz umfasst Bildsequenzen eines sich bewegenden Fahrzeugs und eines Laserscans des Fahrzeugs, welcher sich dazu eignet, die Rekonstruktionen der Objektform zu evaluieren. Der zweite Datensatz enthält synthetische Sequenzen von verschiedenen Fahrzeugen in einer urbanen Umgebung. Die Grundwahrheit schließt die Fahrzeugformen als auch die Fahrzeug- und Kamerapose für jeden Zeitpunkt mit ein. Der Datensatz ermöglicht die quantitative Auswertung von rekonstruierten Formen und Trajektorien bewegter Objekte.
Mit Hilfe dieser Datensätze werten wir die entwickelten Algorithmen auf Szenarien von fahrenden Fahrzeugen aus. Die Bildsequenzen weisen herausfordernde Eigenschaften wie kleine Objektgrößen, reflektierende Oberflächen als auch beleuchtungs- und blickwinkelabhängige Erscheinungsänderungen auf. Wir demonstrieren, dass das vorgeschlagene Verfahren zur Rekonstruktion von Objektformen es ermöglicht, Dreiecksnetze zu bestimmen, welche robust bzgl. Reflexionen und Erscheinungsänderungen sind. Die quantitative Auswertung der Algorithmen der Trajektorienrekonstruktion zeigt, dass die Skalenmehrdeutigkeit von (monokularen) bildbasierten Rekonstruktionen ein anspruchsvolles Problem darstellt. Die Verwendung von binokularen Bildsequenzen löst diese Mehrdeutigkeit auf und resultiert in genaueren und robusteren Rekonstruktionen. Durch die quantitative Evaluierung der vorgestellten Algorithmen auf den präsentieren Datensätzen stellen wir eine Referenz für

zukünftige Arbeiten in dem Gebiet der Rekonstruktion bewegter Objekte zur Verfügung.

Acknowledgement

Pursuing a PhD is a challenging task. As with many things in life, it becomes easier with a supportive environment. Fortunately, I experienced great guidance and assistance of various people throughout this dissertation.
First of all, I would like to thank my dissertation advisor Prof. Dr.-Ing. Rainer Stiefelhagen for his guidance and encouragement, which were invaluable to complete this dissertation. His advices substantially improved the structure of this thesis.
Furthermore, I want to thank Dr. Christoph Bodensteiner for his valuable expertise. Our meetings have been crucial to determine the different research topics covered in this thesis.
I am grateful to Prof. Dr.-Ing. Jürgen Beyerer, Prof. Dr. Mehdi B. Tahoori, Prof. Dr. Tamim Asfour and Prof. Dr. Dennis Hofheinz for serving in my examination committee. In particular, I would like to thank Prof. Dr.-Ing. Jürgen Beyerer for his thorough revision of this thesis.
I want to acknowledge my colleagues of the *Object Recognition Department* at the *Fraunhofer Institute of Optronics, System Technologies and Image Exploitation* for their constant assistance. Our conversations were essential to solve various technical challenges.
Also, I would like to thank the members of the *Computer Vision for Human-Computer Interaction Lab* at the *Karlsruhe Institute of Technology*. Our meetings have been very helpful to keep track of the latest developments in computer vision and machine learning.
Lastly, I want to thank my family and friends for their counsel. Our common time is always a good possibility to refresh my mind for upcoming tasks.

Contents

Nomenclature

Terms and Definitions

AP	Assignment Problem
BA	Bundle Adjustment
BP	Boundary Point
CFS	Coordinate Frame System
ConvNet	Convolutional Neural Network
DoF	Degree of Freedom
MAD	Mean Absolute Deviation
ML	Maximum Likelihood
MSfM	Multibody Structure from Motion
MOT	Multiple Object Tracking
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
NRSfM	Non-Rigid Structure from Motion
OTE	Object Trajectory Error
PBP	Possible Boundary Point
RSR	Reference Scale Ratio
RSRD	Reference Scale Ratio Deviation
SfM	Structure from Motion
SLAM	Simultaneous Localization and Mapping
SVD	Singular Value Decomposition

Notation

Scalars	Italic Roman and Greek lowercase letters, e.g. x, α.
Sets	Calligraphic Roman uppercase letters, e.g. $\mathcal{S}$.
Vectors	Bold Roman lowercase letters, e.g. $\mathbf{v}$.
Matrices	Bold Roman uppercase letters, e.g. $\mathbf{M}$.

Variables, Symbols and Operators

General Geometry

$\mathbf{I}$	Identity matrix.
$\mathbf{R}$	Rotation matrix.
$\mathbf{T}$	Transformation matrix.
SO	The *special orthogonal group* consists of orthogonal matrices with determinant one.
$[\mathbf{R}\vert\mathbf{t}]$	Operator that appends a new column vector $\mathbf{t}$ to a matrix $\mathbf{R}$.
$[\mathbf{e}]_x$	Skew-symmetric matrix corresponding to the vector $\mathbf{e}$.

Camera Modeling

x, y, z	Space coordinates.
u, v	Image coordinates.
f	Focal length of a pinhole camera.
b	Baseline of a stereo camera.
$\mathbf{x}$	Point in space.
$\mathbf{m}$	Measurement in image space corresponding to $\mathbf{x}$.
$\mathbf{c}$	Center of a pinhole camera.
$\mathbf{p}$	Principal point of a pinhole camera.
$\mathbf{P}$	Projection matrix of a pinhole camera.
$\mathbf{K}$	Calibration matrix corresponding to $\mathbf{P}$.
$\mathbf{m} \leftrightarrow \mathbf{m}'$	Feature correspondence of two features $\mathbf{m}$ and $\mathbf{m}'$.
$\mathbf{e}$	Epipole.
$\mathbf{l}$	Epipolar line.
$\mathbf{F}$	Fundamental matrix.
$\mathbf{E}$	Essential matrix.

Factor Graphs

$f_k(\cdot)$	The k-th local function.
f_k	The k-th factor node corresponding to the local function $f_k(\cdot)$.
Θ_k	Set of variable nodes adjacent to f_k.
θ_l	The l-th variable node.

$h_k(\cdot)$	The k-th measurement function.
z_k	The k-th measurement.
$\boldsymbol{\theta}_\mathbf{s}$	Variable node representing a stereo camera pose.
$\boldsymbol{\theta}_\mathbf{p}$	Variable node representing a triangulated point.

Multiple Object Tracking

x, y	Image coordinates.
$\mathbf{I}_i$	The i-th frame of a monocular image sequence.
$\mathbf{S}_i$	(Instance-aware) semantic segmentation corresponding to $\mathbf{I}_i$.
$\mathbf{F}_{i \rightarrow i'}$	Optical flow between frame i and i' in a monocular image sequence.
$\mathcal{V}_i$	Set of pixel positions with valid optical flow information at time i.
$\mathcal{S}_{i,u}$	Set of pixels corresponding to the object with index u at time i.
$\mathcal{V}_{i,u}$	Valid pixel positions with optical flow information of an object with index u at time i.
$\mathcal{P}_{i \rightarrow i',u}$	Prediction of the shape of an object with index u from time i to i'.
$\mathcal{A}_{i \rightarrow i'}$	Affinity matrix between predictions $\mathcal{P}_{i \rightarrow i',u}$ and segmentations $\mathcal{S}_{i',v}$.
$\mathcal{T}_i$	Tracker state at the frame with index i.
md	Number of missing object detections of a tracklet.

Multibody Structure from Motion

$\mathbf{x}^{(c)}$	Vector $\mathbf{x}$ in coordinate frame system c.
$sfm^{(b)}$	Background reconstruction.
$\mathbf{c}_i^{(b)}$	Center of the i-th camera in $sfm^{(b)}$.
$\mathbf{R}_i^{(b)}$	Rotation matrix of the i-th camera in $sfm^{(b)}$.
$\mathcal{P}^{(b)}$	3D points contained in $sfm^{(b)}$.
$\mathbf{b}_k^{(b)}$	The k-th point in $sfm^{(b)}$.
$sfm^{(o)}$	Object reconstruction.
$\mathbf{c}_i^{(o)}$	Center of the i-th camera in $sfm^{(o)}$.
$\mathbf{R}_i^{(o)}$	Rotation matrix of the i-th camera in $sfm^{(o)}$.
$\mathcal{P}^{(o)}$	3D points contained in $sfm^{(o)}$.

$\mathbf{o}_j^{(o)}$	The j-th point in $sfm^{(o)}$.
$\mathbf{o}_j^{(i)}$	The j-th point in $sfm^{(o)}$ expressed in the coordinate frame system of the i-th camera.
$\mathbf{o}_{j,i}^{(b)}(r)$	The j-th point in $sfm^{(o)}$ expressed in the coordinate frame system of $sfm^{(b)}$ at time i depending on the corresponding scale ratio r.
$\mathbf{v}_{j,i}^{(b)}$	Vector pointing from $\mathbf{c}_i^{(b)}$ to $\mathbf{o}_{j,i}^{(b)}(r)$ expressed in the coordinate frame system of $sfm^{(b)}$.
$\mathbf{T}^{(c2c')}$	Transformation from coordinate frame system c to coordinate frame system c'.
$\mathbf{T}^{(c2c')}(r)$	Transformation from coordinate frame system c to coordinate frame system c' depending on the scale ratio r between c and c'.
SE	The *special euclidean group* defines transformations consisting of a rotation $\mathbf{R} \in SO$ and a translation $\mathbf{v} \in \mathbb{R}^n$.
$\mathbf{K}_i$	Calibration matrix of the i-th camera.
$\mathbf{p}_{j,i}$	Projection of $\mathbf{o}_j^{(o)}$ in the camera with index i.
$\theta_i(\cdot)$	Function that determines if a given pixel corresponds to the object in image i.
$\nu_i(\cdot)$	Function that determines if a given pixel corresponds to the ground category in image i.
$\sigma_i(\cdot)$	Function that determines if a given projection is visible in image i.

1 Introduction

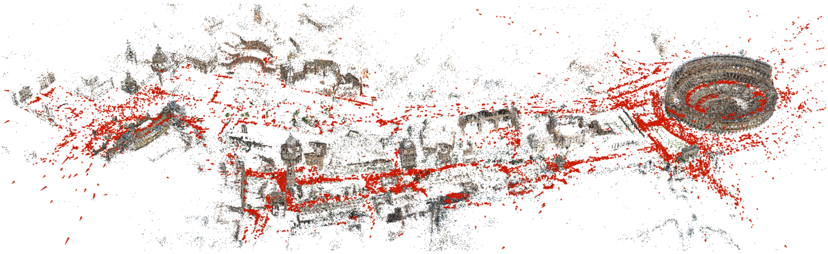
This chapter gives a brief introduction to dynamic scene reconstruction using Multibody Structure from Motion based algorithms.

1.1 Motivation

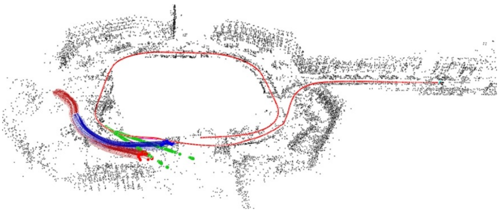
Many elementary human everyday activities such as orientation or interaction with objects rely on information about the three-dimensional structure of our environment. Similarly, various computer-based applications benefit from 3D world models. *Image reconstruction algorithms* leverage image or video data to compute three-dimensional scene models. This task is one of the fundamental challenges in computer vision, because metric scene properties are lost during image acquisition.
Recent progress in image-based modeling shows that such algorithms are able to reconstruct entire city districts. Large-scale 3D reconstructions require an efficient and accurate three-dimensional representation of scene structures. For example, *Structure from Motion (SfM)* or *Visual Simultaneous Location and Mapping (Visual SLAM)* allows to reconstruct geometric properties of the scene given suitable conditions like sufficiently textured surfaces and non-degenerated camera motions.
However, many available methods neither reconstruct *semantic information* nor *dynamic components*. Such knowledge is essential to leverage these models in different application scenarios like autonomous transportation systems, robotics, augmented reality, visualization or visual editing. For example, autonomous systems rely in particular in uncontrolled scenarios on a spatial and a semantic interpretation of dynamic environments to avoid collisions or to perform path planning. Other domains like augmented reality also require three-dimensional object shape and motion information to present adequate user information or to determine interaction between reality and the virtual

world. Cinematic visual effect pipelines leverage three-dimensional object tracking for different tasks including color correction, object replacement or texturing. We present a method that computes *dynamic* reconstructions, which are inherently annotated with *semantic* instance and category information to cover such application scenarios.
Generally, there is a variety of sensor types to capture the required three-dimensional information of dynamic environments. Sensors can be categorized in passive sensors (*e.g.*, *RGB* cameras), active sensors (*e.g.*, *Lidar* or *Radar*) and hybrid devices (*e.g.*, *RGB-D* sensors). The methods in this work only use cameras, *i.e.*, passive sensors, to compute three-dimensional models of dynamic environments. Cameras show several advantages over other sensing modalities. Compared to active sensors, cameras require less energy and lower production costs. Because of their small weight and size, cameras are even suitable to be integrated in body-worn and drone-mounted systems. Another advantage of cameras (or passive sensors in general) is the absence of cross talk effects, *i.e.*, there is no interference of signals of simultaneously operating sensors. Finally, in the context of creating semantically annotated reconstructions, image-based methods are especially useful due to the huge amount of publicly available (annotated) image data, which allows a reliable *semantic segmentation* of scene structures.
Since images contain only two-dimensional projections of the captured scene, the three-dimensional reconstruction with a single monocular sensor is in general an underconstrained problem. Additional assumptions are required to recover the 3D properties of a scene, which may be categorized depending on underlying scene dynamics as follows:

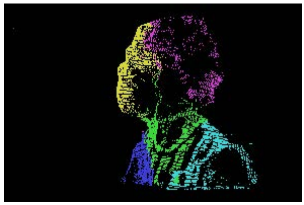
- Structure from Motion (SfM)
  - Assumes that the images show a static scene, *i.e.*, a single component.
  - Decomposes the reconstruction process into more controllable subproblems. Detects salient features in each image and determines inter-image feature correspondences. Corresponding features in different images are considered as projections of the same three-dimensional point and allow to estimate corresponding camera parameters such as focal length or camera pose. Combining the reconstructed camera parameters enables the triangulation of scene structures, *i.e.*, to compute the three-dimensional coordinates of the scene points.
  - Allows to reconstruct large scenes - see Fig. 1.1a.



(a) SfM result of a city district in rome, from Schönberger and Frahm (2016).



(b) MSfM result of two vehicles trajectories (blue and red) near Versailles, from Kundu et al. (2011).



(c) NRSfM result of a person (Russell et al., 2014). Consistently moving points show the same color.

Figure 1.1: Reconstruction results of using SfM, MSfM and NRSfM. The camera poses estimated by SfM and MSfM are shown in red. In the case of NRSfM, relative camera poses can not be determined, since the reference system, *i.e.*, the scene points, are allowed to undergo arbitrary motions.

- Multibody Structure from Motion (MSfM)
  - Is a generalization of SfM and assumes that the scene may be described by a *multibody system*, *i.e.*, the scene consists of multiple rigidly structured components, which may undergo independent translational as well as rotational displacements.
  - Determines component specific features, which allows to leverage SfM techniques to reconstruct the individual components.
  - See Fig. 1.1b for an example.
- Non-Rigid Structure from Motion (NRSfM)
  - Allows to reconstruct deformable objects in contrast to SfM and MSfM, but is underconstrained for each pixel.

– Domain specific shape priors are required to compensate for under-constrained properties. These properties restrict NRSfM to a limited set of application scenarios, *e.g.*, face or body pose reconstruction.
– Reconstruction of arbitrary huge environments with NRSfM is infeasible, because of the corresponding scene variety - see Fig. 1.1c for an example.

This work focuses on MSfM, since it provides a suitable trade-off between scene dynamics and scalability.

1.2 Challenges and Problem Statement

Image-based reconstruction of dynamic scenes is a challenging problem, since the three-dimensional information is lost during image acquisition, *i.e.*, only the two-dimensional projection of scene structures is captured. Structure from Motion is suitable to compute three-dimensional models of static scenes. As we will see in Chapter 2 and Chapter 3, the reconstruction of multiple independently moving scene components with SfM requires the determination of consistently moving groups of visual features, *i.e.*, sets of key points corresponding to specific objects or to static environment structures.
Because of the requirements mentioned above, a SfM based pipeline for dynamic scene reconstruction (*i.e.*, a MSfM pipeline) must comprise the following steps:

- Determination of consistently moving groups of visual features, *e.g.*, object detection and tracking.
- Reconstruction of independent components, *e.g.*, object and environment. Intrinsic camera parameters may be shared during reconstruction.
- Estimation of the scale ratios between components, *e.g.*, scale ratio between objects and environment reconstruction.

Most existing MSfM approaches use motion segmentation or epipolar constraints to identify consistently moving groups of visual features. As such methods are agnostic to semantics, they fail in certain scenarios like stationary or parallel moving objects. Recent advances in instance-aware semantic segmentation detect and describe the two-dimensional shape of objects in a given image on pixel level. This work proposes a MSfM approach that uses

instance-aware semantic segmentations to compute component specific features. It allows us to reconstruct situations where traditional methods fail.
One important application domain of MSfM is the reconstruction of dynamic objects in a mainly static environment. Many of the previously mentioned scenarios like autonomous transportation systems, robotics, augmented reality, visualization and visual editing potentially fall into this category. Throughout this thesis we consider the reconstruction of dynamic vehicles as the predominant use case.
We tackle the problem of *object shape* as well as *object trajectory* reconstruction. Specific object properties such as small object sizes, reflecting surfaces, illumination and view dependent appearance changes hamper the reconstruction of accurate object shapes. The computation of three-dimensional object motion trajectories is particularly difficult, because of the scale ambiguity of image-based reconstructions. Even observations of subsequent images are not sufficient to determine consistent trajectories. We propose several novel motion and geometric constraints to tackle this problem.
The following **research question** summarizes the main aspects of this thesis:
Does semantic segmentation based Multibody Structure from Motion allow to accurately reconstruct real-world scenarios of moving objects?

1.3 Research Context

One of the first works in the context of computer vision addressing the question how the three-dimensional structure and motion of objects may be inferred from two-dimensional image projections was examined in Ullman (1979). In addition, Ullman (1979) proposes the *Structure from Motion Theorem*, which states that three (distinct) orthographic views of four non-coplanar points allow to compute the structure of a non-deformable object. Two years later, Longuet-Higgins (1981) proposed an algorithm to reconstruct static scene structures using only two projections. Motivated by these ideas, Adiv (1985) presents a fully automated pipeline to reconstruct the three-dimensional motion and structure of several moving objects, which may be considered as the first Multibody Structure from Motion pipeline. In contrast to Ullman (1979) and Longuet-Higgins (1981), Adiv (1985) determines pixel correspondences automatically by assuming that scene components are roughly planar surfaces. In the next decades, many others (Debrunner and

Ahuja, 1992; Debrunner and Ahuja, 1998; Fitzgibbon and Zisserman, 2000; Gear, 1998; Kundu et al., 2011; Ozden et al., 2004, 2010) proposed alternative algorithms to improve reconstruction quality and efficiency. However, the question how to achieve a predefined robustness, accuracy, completeness and scalability in complex scenarios is still unanswered. One limitation of previous methods is the usage of motion segmentation or epipolar constraints to group scene components. There are many situations where both methods show a fragile behavior. We show that recent advances in instance-aware semantic segmentation (Dai et al., 2016; He et al., 2017; Li et al., 2017) as well as optical flow computation (Hu et al., 2016; Ilg et al., 2017; Sun et al., 2018) offer an appealing alternative to determine object specific points on pixel level. This allows us to leverage mature state-of-the-art Structure from Motion pipelines such as OpenMVG (Moulon et al., 2012) or Colmap (Schönberger and Frahm, 2016) to compute MSfM reconstructions of complex real world scenarios.

1.4 System Overview

Fig. 1.2 shows an overview of the proposed approach for dynamic scene reconstruction. The pipeline takes a monocular or stereo image sequence as input. In contrast to previous works, our approach exploits recent advances in instance-aware semantic segmentations to cluster visual features. To compute video object segmentations we perform Multiple Object Tracking using similarity scores based on optical flow and object segmentations of adjacent frames (Bullinger et al., 2017, 2019b). The proposed similarity scores reflect locality and visual similarity.

Determining object specific features with semantic segmentation allows us to leverage publicly available state-of-the-art Structure from Motion pipelines to tackle the problem of image-based three-dimensional scene modeling. We apply SfM to video object segmentations and images of background structures to compute separate object and background reconstructions (Bullinger et al., 2018b). The reconstruction results contain a set of three-dimensional points representing object or background structures and camera poses of different time steps with respect to the corresponding point cloud. The computed MSfM reconstructions are inherently connected with corresponding semantic information, which eases the usage of these models for many application scenarios.

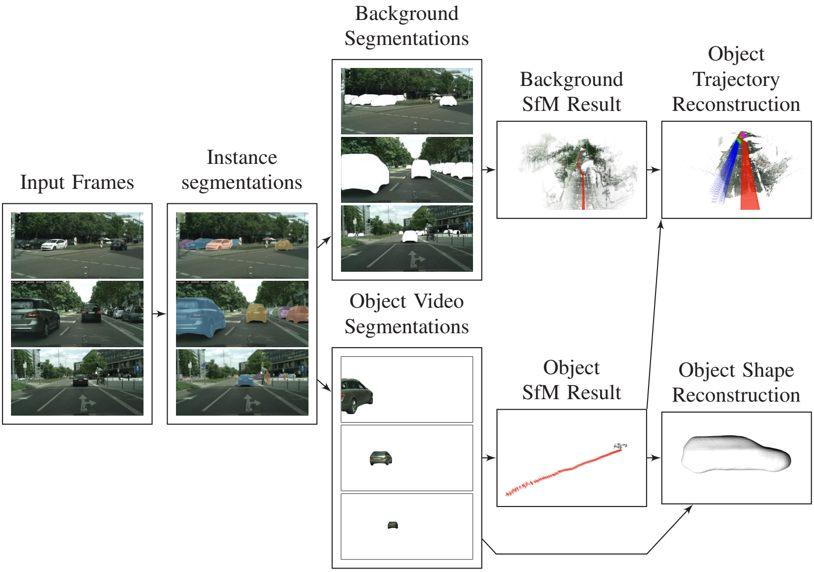


Figure 1.2: Overview of the dynamic object reconstruction pipeline. We track objects on pixel level with instance-aware semantic segmentations and optical flow features. Corresponding three-dimensional object and environment models as well as camera poses are computed with Structure from Motion. We exploit semantic projection constraints to compute three-dimensional watertight object meshes. Combining object and environment reconstructions allow us to compute three-dimensional object trajectories. The reconstructed camera trajectories are shown in red - the object trajectory in blue.

Integrating the object reconstructions into the static environment SfM result allows us to compute three-dimensional object trajectories (Bullinger et al., 2018a,b, 2019b). Due to the scale ambiguity of image-based reconstructions, the three-dimensional object motion trajectory is only defined up to an unknown scale factor. We require additional constraints like assumptions about the type of object motion to determine consistent three-dimensional object trajectories (Bullinger et al., 2018a,b). Capturing the scene with stereo video data allows us to solve the scale ambiguity using the stereo camera baseline (Bullinger et al., 2019b).
In addition, the pipeline allows to use semantic projection constraints (given appropriate camera-object-poses) to determine object point clouds convenient

for mesh computations (Bullinger et al., 2016). That is, the points are uniformly distributed and corresponding normal vectors are consistent, which is particularly difficult to achieve for dynamic objects with reflecting surfaces using Multi-View stereo algorithms.

1.5 Datasets

Prior to this work, there was a lack of publicly available benchmark datasets for image-based reconstruction of dynamic objects, *i.e.*, existing datasets did not provide essential ground truth data. However, such datasets are crucial for validation to establish efficient research and development cycles.

In order to evaluate algorithms for dynamic object reconstruction, accurate object and environment models as well as synchronized object and camera poses of different time steps are required. Capturing the corresponding ground truth is difficult due to synchronization and registration errors of object and camera poses. This is presumably the main reason why current real world datasets do not provide this kind of ground truth information. In Chapter 4 this thesis presents two datasets for *shape* and *trajectory* reconstruction of moving objects.

In order to evaluate object *shape* reconstruction approaches, it is sufficient to create three-dimensional object shapes as ground truth data. We present a dataset of driving vehicle sequences. Multiple registered laser scans, which capture the object from different views serve as shape ground truth. Registering the image-based object shape reconstructions to the set of laser scans allows us to determine metric reconstruction errors.

Evaluation of object *trajectory* reconstructions requires different types of ground truth data such as object and environment models as well as synchronized object and camera poses of different time steps. To circumvent problems of ground truth data acquisition, we create a virtual world that allows to render sequences of driving vehicles in urban environments. We apply skeletal animation to automatically determine steering, wheel rotation and consistent vehicle placement on uneven ground surfaces. The ground truth geometry as well as the camera and object poses are free of noise and show no spatial registration or temporal synchronization inaccuracies. We exploit procedural generation of textures to avoid artificial repetitions. This makes our dataset suitable for evaluation of image-based reconstruction algorithms.

Previous works do not show quantitative comparisons because of the lack of publicly available implementations and benchmark datasets with suitable ground truth data. This makes it difficult to perform a quantitative evaluation of corresponding algorithms. We address this issue with the datasets described above.

1.6 Contribution

This thesis presents a framework for Multibody Structure from Motion to compute three-dimensional shapes and motion trajectories of dynamic objects. The core contributions of this work are as follows.
1) An instance-aware semantic segmentation based **online Multiple Object Tracking approach** (Bullinger et al., 2017, 2019b). The method combines object segmentations and optical flow information to track two-dimensional object shapes on pixel level in monocular as well as stereo image sequences. Converting the segmentation masks to bounding boxes allows us to compare the tracking with publicly available bounding box based tracking approaches.
2) **A Multibody Structure from Motion algorithm** based on the proposed Multiple Object Tracking approach. The method allows to leverage state-of-the-art standard Structure from Motion for multibody reconstruction. We provide a detailed description of the tracking as well as the reconstruction components.
3) **An algorithm to compute three-dimensional object shapes** consistent to constraints derived from semantic segmentations (Bullinger et al., 2016). The resulting point clouds show a high point density making them suitable for computation of watertight meshes.
4) Creation of a publicly available multi-view benchmark **dataset to evaluate image-based algorithms for moving object shape reconstruction** (Bullinger et al., 2016). The dataset consists of videos capturing a vehicle performing several maneuvers and provides registered object laser scans as three-dimensional shape ground truth.
5) **A new framework to reconstruct the three-dimensional trajectory of vehicles/objects** in monocular and stereo image sequences using the proposed instance-aware semantic segmentation based Multibody Structure from Motion approach. We propose several novel methods to tackle the scale ambiguity of Structure from Motion reconstructions. This allows us to compute

vehicle/object motion trajectories consistent to image observations and environment structures (Bullinger et al., 2018a,b, 2019b).
6) **A new stereo-matching based algorithm for the three-dimensional object trajectory reconstruction** in stereo image sequences (Bullinger et al., 2019a). In contrast to previous works, the method avoids the triangulation of environment points during object reconstruction.
7) Creation of a publicly available **synthetic vehicle trajectory benchmark dataset** due to the lack of publicly available video data of vehicles with suitable ground truth data (Bullinger et al., 2018b, 2019b). The dataset consists of photo-realistic rendered videos of animated vehicles in urban environments. 3D vehicle and environmental models used for rendering serve as ground truth. The dataset and evaluation scripts are publicly available to foster future object motion reconstruction related research. The dataset allows for the first time to evaluate reconstructions of the three-dimensional motion of the vehicles visible in the image sequences.
8) Previous works do not show quantitative comparisons because of the lack of publicly available benchmark datasets for dynamic object reconstruction. We adress this issue with the dataset described in 4) and 7) and provide a **thorough qualitative and quantitative evaluation** of the proposed shape and trajectory reconstruction methods.

In the context of this thesis the following peer-reviewed papers have been published. Where appropriate the corresponding parts of this work reference these publications.

C. Bodensteiner, S. Bullinger, S. Lemaire, and M. Arens. Single frame based video geo-localisation using structure projection. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.

C. Bodensteiner, S. Bullinger, and M. Arens. Multispectral matching using conditional generative appearance modeling. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018.

S. Bullinger, C. Bodensteiner, S. Wuttke, and M. Arens. Moving object reconstruction in monocular video data using boundary generation. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2016.

S. Bullinger, C. Bodensteiner, and M. Arens. Instance flow based online multiple object tracking. In *IEEE International Conference on Image Processing (ICIP)*, 2017.

S. Bullinger, C. Bodensteiner, and M. Arens. Monocular 3D vehicle trajectory reconstruction using terrain shape constraints. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018a.

S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen. 3D vehicle trajectory reconstruction in monocular video data using environment structure constraints. In *European Conference on Computer Vision (ECCV)*, 2018b.

S. Bullinger, C. Bodensteiner, and M. Arens. 3D object trajectory reconstruction using stereo matching and instance flow based multiple object tracking. In *IAPR International Conference on Machine Vision Applications (MVA)*, 2019a.

S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen. 3D object trajectory reconstruction using instance-aware multibody structure from motion and stereo sequence constraints. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019b.

1.7 Thesis Outline

In Chapter 1 we give an introduction into image based reconstruction of dynamic objects. The chapter motivates the usage of Multibody Structure from Motion and discusses important properties. Chapter 2 presents well known methods relevant for static environment reconstruction such as local image description, Multiple View Geometry, Structure from Motion and factor graphs. Many of the presented concepts are used by the instance-aware Multibody Structure from Motion approach proposed in Chapter 3. The algorithm in Chapter 3 is an essential component of the methods for shape and trajectory reconstruction of dynamic objects presented in Chapter 5 and Chapter 6. Chapter 4 presents two novel benchmark datasets for shape and trajectory reconstruction of dynamic objects. We use the corresponding ground truth to quantitatively evaluate the algorithms in Chapter 5 and Chapter 6. The method in Chapter 5 focuses on the reconstruction of three-dimensional object shapes using semantic volumetric constraints. Chapter 6 uses the instance-aware Multibody Structure from Motion approach to reconstruct three-dimensional vehicle/object trajectories using monocular and stereo image sequences of a single device. Finally, Chapter 7 concludes the thesis by providing a summary and a future work section.

2 Fundamentals of Image-Based Reconstruction Techniques

This chapter describes well known image-based methods to reconstruct static environments. The underlying principles are important for the instance-aware Multibody Structure from Motion approach proposed in Chapter 3. In the first Section 2.1, we describe the process of determining salient and distinct local features. Section 2.2 summarizes the key aspects of multi-view geometry and shows how image observations allow to reconstruct three-dimensional scene structures with a single pair of images. Section 2.3 presents the structure of a typical SfM pipeline building on top of the basics described in Section 2.2. Further, Section 2.3 highlights the requirements of Multibody Structure from Motion. Section 2.4 gives an introduction to factor graphs allowing to refine Structure from Motion results by modeling additional constraints.

2.1 Local Image Description

Local image description methods determine local image structures, so called *local features* or *keypoints*, which are salient, distinct and identifiable across a set of images.

2.1.1 Problem Statement

In order to identify local features across different images, they must show suitable properties such as invariance w.r.t. translation, rotation and scaling as well as robustness w.r.t. changing illumination and affine projections.
Most previously published (hand-crafted) feature methods tackle this task using the following steps: 1) detection of salient features such as corners or edges, 2) determination of a predominant orientation and 3) description of a

Figure 2.1: Detected features using SIFT (Lowe, 1999).

local area around the feature. Salient features may be determined as maxima in scale-space to achieve invariance w.r.t. translation and scaling. Orientation assignment aims to ensure rotation invariant features. The appearance descriptors typically reflect normalized gradient information instead of absolute color values to achieve robustness w.r.t. illumination changes.
The feature descriptors as well as the feature geometry like their position, orientation and scale allow to determine correspondences in different images. This information is useful to re-identify scene components and allows to compute geometry relations between these images as shown in Section 2.2. Fig. 2.1 shows a visualization of detected local features.

2.1.2 Related work and State-of-the-Art

Local image description methods may be categorized in hand-crafted local features and learned local features. Hand-crafted features such as Scale Invariant Feature Transform (SIFT) (Lowe, 1999, 2004), Speeded Up Robust Features (SURF) (Bay et al., 2006, 2008) or AKAZE (Alcantarilla et al., 2012) follow the standard pipeline described above. Other detectors like GFTT (Jianbo Shi and Tomasi, 1994) and FAST (Rosten and Drummond, 2006) do not consider scale while determining salient points to speed up processing. Binary descriptors like ORB (Rublee et al., 2011) and BRISK (Leutenegger et al., 2011) provide a trade-off between lower computational costs and reduced robustness. Recently, different works (Simo-Serra et al., 2015; Simonyan et al., 2014; Vassileios Balntas and Mikolajczyk, 2016; Yi et al.,

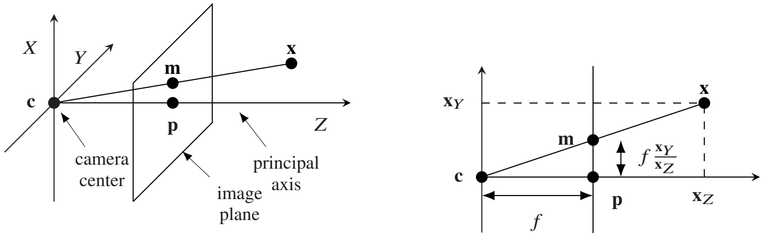


Figure 2.2: Model of a pinhole camera depicting the projection of scene structures. The properties of a pinhole camera are described by the camera center **c**, the principal point **p** and the focal length f. The principal axis is orthogonal w.r.t. the image plane. The focal length f denotes the corresponding distance. A three-dimensional point **x** given by $(\mathbf{x}_X, \mathbf{x}_Y, \mathbf{x}_Z)^\top$ is projected onto the image point $(\frac{f\mathbf{x}_X}{\mathbf{x}_Z}, \frac{f\mathbf{x}_Y}{\mathbf{x}_Z})^\top$.

2016) proposed automatically learned local features. Schönberger et al. (2017) show that advanced hand-crafted features achieve comparable results to recent learned features in the domain of image-based reconstruction. Further, the evaluation emphasizes that learned features show higher variances across different datasets than traditional hand crafted features.

2.2 Multiple View Geometry

This section describes elementary components of current Structure from Motion pipelines such as camera models, camera calibration, epipolar geometry and triangulation. These concepts allow to reconstruct scene structures and camera poses simultaneously without prior knowledge about the camera motion or scene geometry. The theory of epipolar geometry is crucial to understand the limitations of standard Structure from Motion w.r.t. dynamic scenes. The subject is discussed in more detail in Hartley and Zisserman (2004).

2.2.1 Pinhole Camera Model

The pinhole camera model describes the central projection of a point in space onto an image plane. The geometry of the image capturing process is shown in Fig. 2.2. The center of the projection is called the camera center **c**. The line

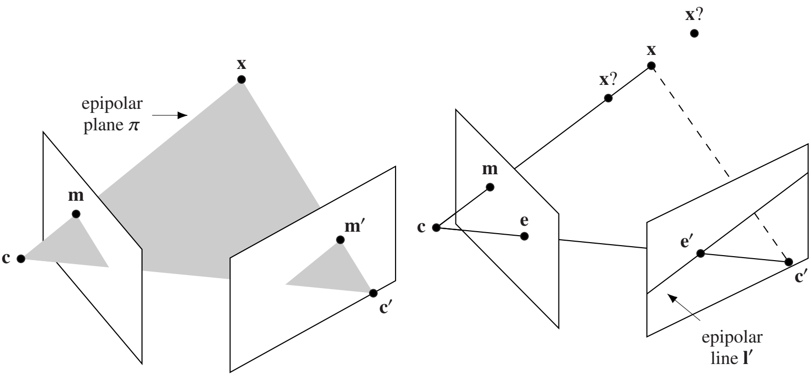


Figure 2.3: Left: The geometric relation of two cameras (defined by their camera centers **c** and **c′**) and a three-dimensional point **x** including the corresponding projections **m** and **m′** is described by the epipolar plane. Right: The back-projection of **m** is imaged in the other view as the epipolar line **l′**. All epipolar lines **l′** intersect the epipole **e′**.

from the camera center perpendicular to the image plane is called the principal axis.
Let $\mathbf{m} = (m_1, m_2, m_3)^\mathsf{T}$ denote the homogenous vector corresponding to the image point $(\frac{m_1}{m_3}, \frac{m_2}{m_3})$. The pinhole camera model allows us to project homogeneous 3-space point $\mathbf{x} = (x, y, z, 1)^\mathsf{T}$ with the projection matrix $\mathbf{P}$ to a point on the image plane according to equation (2.1)

$$\mathbf{m} = \mathbf{P}\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{x} = \mathbf{K}\mathbf{R}[\mathbf{I}|-\mathbf{c}]\mathbf{x}. \tag{2.1}$$

$\mathbf{R} \in SO(3)$ and $\mathbf{c}$ denote the rotation and the center of the camera. The calibration matrix $\mathbf{K}$ defined in (2.2) describes intrinsic camera parameters such as focal length f, principal point (p_u, p_v) and shearing factor s.

$$\mathbf{K} = \begin{bmatrix} f & s & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

2.2.2 Epipolar Geometry

This section provides an overview of the *epipolar geometry*, which describes the geometric relation between two different views without explicitly reconstructing the structure of the scene. The epipolar geometry is represented by a 3x3 matrix - the fundamental matrix (Hartley, 1992).
Let $\mathbf{m} \leftrightarrow \mathbf{m'}$ denote corresponding pixel observations of the same 3-space point $\mathbf{x}$ in the first and the second view. Given $\mathbf{m}$ the epipolar geometry allows to constrain the position of $\mathbf{m'}$ in the second image. Furthermore, the camera matrices $\mathbf{P}$ and $\mathbf{P'}$ may be computed from the fundamental matrix $\mathbf{F}$.
Let us consider in the following two views with associated camera matrices $\mathbf{P}$ and $\mathbf{P'}$. According to Section 2.2.1 a 3-space point $\mathbf{x}$ is imaged as image point $\mathbf{m} = \mathbf{Px}$ and $\mathbf{m'} = \mathbf{P'x}$ in the first and second view, respectively. The back-projection of an image point $\mathbf{m}$ is imaged as a line $\mathbf{l'}$ in the second view - the epipolar line of $\mathbf{m}$. Thus, the point $\mathbf{m'}$ corresponding to $\mathbf{m}$ must lie on $\mathbf{l'}$. All epipolar lines intersect at the epipole, which is the intersection of the camera baseline and the corresponding image plane, *i.e.*, $\mathbf{e} = \mathbf{Pc'}$ and $\mathbf{e'} = \mathbf{P'c}$. Fig. 2.3 illustrates the relations described above.
In the following, we present the algebraic derivation of the fundamental matrix proposed by Xu and Zhang (1996). Let $\mathbf{P^+}$ be the pseudo-inverse of $\mathbf{P}$. This allows us to define two 3-space points on the back-projection of $\mathbf{m}$: the camera center $\mathbf{c}$ and $\mathbf{P^+m}$. Both points are imaged in the second view with $\mathbf{P'c}$ and $\mathbf{P'P^+m}$. The corresponding epipolar line is defined by

$$\mathbf{l'} = (\mathbf{P'c}) \times (\mathbf{P'P^+m}) = \mathbf{e'} \times (\mathbf{P'P^+m}) = [\mathbf{e'}]_\times(\mathbf{P'P^+})\mathbf{m} = \mathbf{Fm}, \tag{2.3}$$

where $[\mathbf{x}]_\times$ defines the skew-symmetric matrix of a vector $\mathbf{x}$ according to equation (2.4).

$$[\mathbf{e}]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{2.4}$$

In (2.3) the line in the image plane is defined by a vector $\mathbf{l} = (a, b, c)$ corresponding to the equation $ax + by + c = 0$. Two vectors $\mathbf{l}$ and $k\mathbf{l}$ represent the same line for any constant $k \neq 0$. A line going through two points $\mathbf{m_1}$ and $\mathbf{m_2}$ may be represented with the cross product $\mathbf{l} = \mathbf{m_1} \times \mathbf{m_2}$.
Let $\mathbf{P}$ and $\mathbf{P'}$ be decomposed according to $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ and $\mathbf{P'} = \mathbf{K'}[\mathbf{R'}|\mathbf{t'}]$. Further, let $\mathbf{R^\star} = \mathbf{R'R}^\mathsf{T}$ and $\mathbf{t^\star} = \mathbf{t'} - \mathbf{R^\star t}$ denote the relative pose between the

first and the second view. According to equation (2.3) the fundamental matrix $\mathbf{F}$ describes the mapping of a point $\mathbf{m}$ onto the corresponding epipolar line $\mathbf{l'}$ and shows the following relation to the cameras $\mathbf{P}$ and $\mathbf{P'}$.

$$\mathbf{F} = [\mathbf{e'}]_{\times}\mathbf{P'}\mathbf{P}^{+} = \mathbf{K'}^{-\mathsf{T}}[\mathbf{t}^{\star}]_{\times}\mathbf{R}^{\star}\mathbf{K}^{-1} \tag{2.5}$$

For two corresponding points $\mathbf{m} \leftrightarrow \mathbf{m'}$ the fundamental matrix shows the following property

$$\mathbf{m'}^{\mathsf{T}}\mathbf{F}\mathbf{m} = \mathbf{m'}^{\mathsf{T}}\mathbf{l'} = 0. \tag{2.6}$$

Note that a point $\mathbf{m}$ lies on a line $\mathbf{l}$ if and only if $\mathbf{m}^{\mathsf{T}}\mathbf{l} = 0$.
Equation (2.6) shows that $\mathbf{F}$ can be purely computed from image correspondences $\mathbf{m} \leftrightarrow \mathbf{m'}$, *i.e.*, there is no knowledge about the corresponding camera matrices necessary. However, the formulation in (2.5) breaks down, when both views share the same camera centers, i.e. it is only valid for non-coincident camera centers. Thus, pure rotations are degenerated reconstruction cases.
In contrast to the camera matrices $\mathbf{P}$ and $\mathbf{P'}$ the fundamental matrix $\mathbf{F}$ is independent of the chosen world coordinate frame. Multiple pairs $\mathbf{P}$ and $\mathbf{P'}$ correspond to the same fundamental matrix. Thus, the fundamental matrix $\mathbf{F}$ determines a camera pair $\mathbf{P}$ and $\mathbf{P'}$ up to a projective transformation.
The essential matrix (Longuet-Higgins, 1981) is a specialization of the fundamental matrix, which can be applied in cases where the camera calibration is known. The essential matrix is the correspondence of the fundamental matrix for normalized image coordinates $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}'$ as shown in equation (2.7),

$$\hat{\mathbf{m}}'^{\mathsf{T}}\mathbf{E}\hat{\mathbf{m}} = 0 \tag{2.7}$$

with $\hat{\mathbf{m}} = \mathbf{K}^{-1}\mathbf{m}$ and $\hat{\mathbf{m}}' = \mathbf{K'}^{-1}\mathbf{m'}$. By combining equation (2.5), (2.6) and (2.7) we obtain the following relation of the essential and the fundamental matrix (2.8).

$$\mathbf{E} = \mathbf{K'}^{\mathsf{T}}\mathbf{F}\mathbf{K} = [\mathbf{t}^{\star}]_{\times}\mathbf{R}^{\star} \tag{2.8}$$

The possible camera matrices defined by the essential matrix are ambiguous w.r.t. a scale and a four-fold ambiguity. Only one of the four possible solutions is geometrically consistent and leads to triangulated points in front of both cameras $\mathbf{P}$ and $\mathbf{P'}$.
Let be $\mathrm{SVD}(\mathbf{E}) = \mathbf{U\Sigma V}^{\mathsf{T}} = \mathbf{U}\,\mathrm{diag}(1,1,0)\mathbf{V}^{\mathsf{T}}$ the *singular value decomposition* of the essential matrix. Note, that a 3x3 matrix is an essential matrix if

and only if the first two singular values are equal and the last is zero. The possible factorizations of $\mathbf{E} = [\mathbf{t}^\star]_\times \mathbf{R}^\star$ are shown in (2.9) and (2.10).

$$[\mathbf{t}^\star]_\times = \mathbf{U}\mathbf{Z}\mathbf{U}^\mathsf{T} \quad \text{with} \quad \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

The factorization of $[\mathbf{t}^\star]_\times$ determines the translation $\mathbf{t}^\star$ up to scale. Because $0 = [\mathbf{t}^\star]_\times \mathbf{t}^\star$ it follows that $\mathbf{U}\mathbf{Z}\mathbf{U}^\mathsf{T}\mathbf{t}^\star = 0$ and $\mathbf{t}^\star = \mathbf{U}(0,0,1)^\mathsf{T} = \mathbf{u}_3$.

$$\mathbf{R}^\star = \mathbf{U}\mathbf{W}\mathbf{V}^\mathsf{T} \quad \text{or} \quad \mathbf{R}^\star = \mathbf{U}\mathbf{W}^\mathsf{T}\mathbf{V}^\mathsf{T} \quad \text{with} \quad \mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

2.2.3 Computation of the Fundamental Matrix

As shown in Section 2.2.2 the fundamental and essential matrix can be computed from image correspondences alone. In the following, we present well known techniques to estimate the fundamental matrix. The described concepts apply for the computation of the essential matrix as well. Because the essential matrix has a lower degree of freedom, the estimation of the essential matrix requires less matching point pairs. For more information we refer the reader to Hartley and Zisserman (2004).

Computation of the Fundamental Matrix

As shown in Section 2.2.2, observations and the fundamental matrix share the following relation $\mathbf{m}_i\mathbf{F}\mathbf{m}_i = \mathbf{0}$ with matching keypoints $\mathbf{m}_i \leftrightarrow \mathbf{m'}_i$. With $\mathbf{m} = (x, y, 1)^\mathsf{T}$ and $\mathbf{m'} = (x', y', 1)^\mathsf{T}$ this is equivalent to (2.11).

$$x'xf_{1,1} + x'yf_{1,2} + x'f_{1,3} + y'xf_{2,1} + y'yf_{2,2} + y'f_{2,3} + xf_{3,1} + yf_{3,2} + f_{3,3} = 0 \tag{2.11}$$

With **f** denoting the 9-vector of the entries of **F** in row row-major order (2.11) can be simplified to (2.12)

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\,\mathbf{f} = 0. \tag{2.12}$$

For n different matches this translates to (2.13).

$$\mathbf{A}\mathbf{f} = \underbrace{\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{f} = 0 \tag{2.13}$$

If the matrix **A** has a rank of eight, the solution for **f** is unique. Because of noisy point measurements, the rank of **A** is usually nine. In this case we determine a solution for **f** by applying a linear least-squares optimization to (2.13). The resulting estimation for **F** that corresponds to **f** will in general be of rank three. However, true fundamental matrices **F** have a rank of two. The *normalized 8-point algorithm* (*q.v.* Algorithm 1) is designed to tackle this issue by enforcing a singularity constraint. While the normalized 8-point algorithm performs well in many cases, a more robust estimation of the fundamental matrix **F** can be achieved by iteratively minimizing the algebraic error or by iteratively minimizing a geometric image distance like the reprojection error (2.14).

$$\sum_i d(\mathbf{m}_i, \hat{\mathbf{m}}_i)^2 + d(\mathbf{m}'_i, \hat{\mathbf{m}}'_i)^2 \tag{2.14}$$

In both cases, the 8-point algorithm may be used to compute the initial solution.

Automatic Computation of the Fundamental Matrix

In the beginning of this section, we assumed that the matches $\mathbf{m}_i \leftrightarrow \mathbf{m'}_i$ are given. Correspondences based on keypoint detectors and descriptors usually contain mismatches (*e.g.*, because of ambiguous keypoint descriptors) and correspond potentially to inconsistently moving scene structures. Using such correspondences directly, results in degenerated fundamental matrix estimations. Algorithm 2 shows a scheme to compute the fundamental matrix given

Algorithm 1: Normalized 8-point algorithm.

Normalization

Compute normalized image coordinates $\hat{\mathbf{m}}_i = \mathbf{T}\mathbf{m}_i$ and $\hat{\mathbf{m}}'_i = \mathbf{T}'\mathbf{m}'_i$, where $\mathbf{T}$ and $\mathbf{T}'$ denote transformations consisting of translation and scaling

Linear Solution

Let $\hat{\mathbf{A}}$ and $\hat{\mathbf{f}}$ denote the counterpart of $\mathbf{A}$ and $\mathbf{f}$ corresponding to $\hat{\mathbf{m}}_i \leftrightarrow \hat{\mathbf{m}}'_i$

Determine $\hat{\mathbf{F}}$ from the singular vector $\hat{\mathbf{f}}$ corresponding to the smallest singular value of $\hat{\mathbf{A}}$

Constraint enforcement

Let $\hat{\mathbf{F}} = UDV^{\mathsf{T}}$ be the SVD of $\hat{\mathbf{F}}$ with $D = \mathrm{diag}(r, s, t)$ and $r \geq s \geq t$

$\hat{\mathbf{F}}' = U\mathrm{diag}(\mathrm{r,s,0})V^{\mathsf{T}}$

Replace $\hat{\mathbf{F}}$ with $\hat{\mathbf{F}}'$, the closest singular matrix to $\hat{\mathbf{F}}$ under a Frobenius norm

Denormalization

Compute $\mathbf{F} = \mathbf{T}'\hat{\mathbf{F}}'\mathbf{T}$ corresponding to $\mathbf{m}_i \leftrightarrow \mathbf{m}'_i$

a pair of images. The detected correspondences are considered as putative matches, *i.e.*, correspondences with noisy feature point positions and incorrect feature matches.

2.2.4 Point Triangulation

In the previous section we have seen how camera parameters can be determined from image observations alone. In this section we combine camera parameters and image observations to infer three-dimensional scene structures using a *linear* triangulation method. Image-based measurements are potentially noisy. The back-projections of corresponding viewing rays are not intersecting, *i.e.*, the projections $\mathbf{m} = \mathbf{P}\mathbf{x}$ and $\mathbf{m}' = \mathbf{P}'\mathbf{x}$ as well as the epipolar constraint $\mathbf{m}'^{\mathsf{T}}\mathbf{F}\mathbf{m} = 0$ of two measurements $\mathbf{m}$ and $\mathbf{m}'$ are not exactly satisfied.

Following the direct linear transformation (DLT) algorithm (Sutherland, 1974), we rewrite $\mathbf{m} = \mathbf{P}\mathbf{x}$ as $\mathbf{m} \times \mathbf{P}\mathbf{x} = 0$. This allows us to eliminate the homo-

Algorithm 2: Automatic computation of the fundamental matrix.

Compute Interest Points
Compute putative correspondences $\mathbf{p}_i \leftrightarrow \mathbf{p'}_i$
RANSAC robust estimation *(Repeat for N samples)*
- Select 8 random correspondences
- Compute the fundamental matrix $\mathbf{p}_i \mathbf{F} \mathbf{p'}_i = 0$ using the 8-point algorithm
- Calculate the reprojection error for all $\mathbf{p}_i \leftrightarrow \mathbf{p'}_i$
- Compute the number of inliers consistent to $\mathbf{F}$

Re-estimate $\mathbf{F}$ from all correspondences classified as inliers by minimizing the cost function (2.14) using the Levenberg-Marquardt algorithm
Determine further interest point correspondences along the epipolar lines.

geneous scale factor λ. With $\mathbf{m} = (u^\star, v^\star, \lambda) \simeq (u, v, 1)$, the cross product is explicitly defined according to (2.15).

$$\mathbf{m} \times \mathbf{P}\mathbf{x} = \mathbf{m} \times \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} v^\star \mathbf{p}_3^\top \mathbf{x} - \lambda \mathbf{p}_2^\top \mathbf{x} \\ \lambda \mathbf{p}_1^\top \mathbf{x} - u^\star \mathbf{p}_3^\top \mathbf{x} \\ u^\star \mathbf{p}_2^\top \mathbf{x} - v^\star \mathbf{p}_1^\top \mathbf{x} \end{bmatrix} = \mathbf{0} \Leftrightarrow \underbrace{\begin{bmatrix} v \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_2^\top \mathbf{x} \\ \mathbf{p}_1^\top \mathbf{x} - u \mathbf{p}_3^\top \mathbf{x} \\ u \mathbf{p}_2^\top \mathbf{x} - v \mathbf{p}_1^\top \mathbf{x} \end{bmatrix}}_{\mathbf{M}} = \mathbf{0} \quad (2.15)$$

Let $\mathbf{M'}$ denote the counter part of $\mathbf{M}$ corresponding to $\mathbf{m'}$. Because only two components of $\mathbf{M}$ and $\mathbf{M'}$ are linear independent, we use the first two rows of $\mathbf{M}$ and $\mathbf{M'}$ to create (2.16).

$$\begin{bmatrix} v \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_2^\top \mathbf{x} \\ \mathbf{p}_1^\top \mathbf{x} - u \mathbf{p}_3^\top \mathbf{x} \\ v' \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_2^\top \mathbf{x} \\ \mathbf{p}_1^\top \mathbf{x} - u' \mathbf{p}_3^\top \mathbf{x} \end{bmatrix} = \mathbf{0} \Leftrightarrow \underbrace{\begin{bmatrix} v \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_2^\top \mathbf{x} \\ u \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_1^\top \mathbf{x} \\ v' \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_2^\top \mathbf{x} \\ u' \mathbf{p}_3^\top \mathbf{x} - \mathbf{p}_1^\top \mathbf{x} \end{bmatrix}}_{\mathbf{A}} = \mathbf{0} \qquad (2.16)$$

This equation system is overdetermined, since the position of the point $\mathbf{x}$ corresponding to $\mathbf{m}$ and $\mathbf{m'}$ is a 3-space vector. The solution of (2.16) is given by the unit singular vector of the smallest singular value of $\mathbf{A}$.

In contrast to the method described above, different algorithms for point triangulation have been proposed, which leverage the redundancy in *multiple* views (Aholt et al., 2012; Hartley and Sturm, 1995; Kang et al., 2014; Lu and Hartley, 2007; Schönberger and Frahm, 2016).

2.2.5 Bundle Adjustment

Bundle adjustment (BA) (Triggs et al., 2000) is a non-linear method commonly used to refine image-based reconstructions. Given a set of 3D points $\mathbf{x}_j$ and camera matrices $\mathbf{P}_i$ as well as noisy image observations $\mathbf{m}_{j,i}$ that are approximately described by $\mathbf{m}_{j,i} = \mathbf{P}_i \mathbf{x}_j$. Bundle adjustment attempts to determine the Maximum Likelihood estimate of $\mathbf{x}_j$ and $\mathbf{P}_i$ (*i.e.*, $\hat{\mathbf{x}}_j$ and $\hat{\mathbf{P}}_i$) assuming Gaussian measurement noise. Concretely, BA computes $\hat{\mathbf{x}}_j$ and $\hat{\mathbf{P}}_i$ that minimize the reprojection error of all points for each view according to (2.17).

$$\underset{\mathbf{P}_i, \mathbf{x}_j}{\operatorname{argmin}} \sum_{i,j} d(\mathbf{P}_i \mathbf{x}_j, \mathbf{m}_{j,i})^2 \tag{2.17}$$

Here, $d(\mathbf{u}, \mathbf{v})$ represents the geometric image distance between two points $\mathbf{u}$ and $\mathbf{v}$. Minimizing the reprojection error corresponds to the *adjustment of the bundle* of rays between the camera center and visible scene points.
A common method to minimize (2.17) is the Levenberg-Marquardt algorithm (Levenberg, 1944). BA requires a good initialization to converge to the actual optimum, which is why it is used as a refinement. Hartley and Zisserman (2004) recommend to use BA as final step of any image-based reconstruction.

2.3 Structure from Motion

There are two categories of SfM approaches: *incremental* and *global* SfM. Incremental SfM is currently the prevalent state-of-the-art method (Schönberger and Frahm, 2016) and used throughout this thesis. This section provides an overview of typical components of incremental Structure from Motion pipelines.

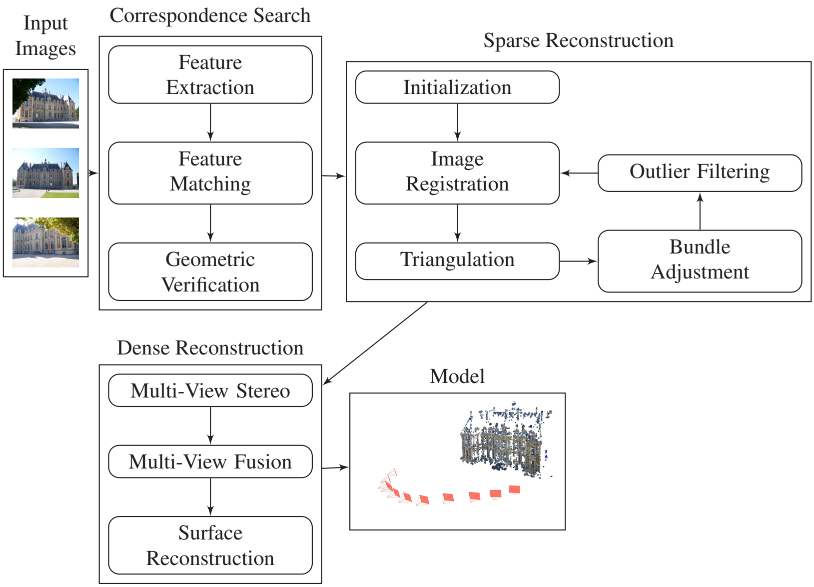


Figure 2.4: Building blocks of a state-of-the-art incremental SfM pipeline. The input images are part of the Sceaux Castle dataset (Moulon, 2012).

2.3.1 Problem Statement and Algorithm Outline

Creating three-dimensional reconstructions using a single camera is an under-constrained problem, since images are only two-dimensional projections of the corresponding environment. Structure from Motion tackles this ambiguity by combining information from different time steps.

In order to manage the complexity of the problem, SfM decomposes the reconstruction process into more controllable subproblems. Fig. 2.4 illustrates the dependencies of typical building blocks of a state-of-the-art incremental SfM pipeline.

Correspondence Search

The pipeline takes a set of images as input to perform feature correspondence search. The correspondences are considered as projections of the same three-dimensional point and allow to implicitly determine input images with overlapping field of views.
First, local features (Alcantarilla et al., 2012; Bay et al., 2006; Lowe, 1999; Simo-Serra et al., 2015; Simonyan et al., 2014; Vassileios Balntas and Mikolajczyk, 2016; Yi et al., 2016) are detected - *q.v.* Section 2.1.
Next, corresponding descriptors of features detected in different images are matched. The output of this building block are potentially overlapping image pairs defined by so called *putative feature matches*. The computational effort of a naive method determining all potential feature correspondences increases quadratically with the number of images N_I and the number of corresponding features N_{F_i}, *i.e.*, $O({N_I}^2 {N_{F_i}}^2)$, making it unsuitable for large image sets. A common approach is to use a vocabulary tree to hierarchical index features and/or to compute a *global image description* (Nister and Stewenius, 2006). This allows to determine visual similar images in linear time w.r.t. number of images. For each pair of visual similar images, the vocabulary tree allows to determine matching local features in linear time w.r.t. to the number of features. This results in an amortized computational effort of $O(N_I N_{F_i})$.
In the next step, SfM attempts to compute a mapping between putative feature correspondences to determine *geometrically verified* matches. The fundamental or the essential matrix (*q.v.* Section 2.2.2) describe such transformations between two images for a moving camera. Pure sensor rotations and planar scenes may be described by homographies (Hartley and Zisserman, 2004). If the transformation is supported by a sufficient number of inliers the corresponding images are considered geometrically verified. The output of this step is a data structure called *scene graph* (Raguram et al., 2011). The nodes in the graph represent images and scene points.

Sparse Reconstruction

The incremental sparse reconstruction step, works solely on the scene graph - no other image information is required. SfM initializes the model with the result of a two-view reconstruction and incrementally performs image registra-

tion, triangulation (*q.v.* Section 2.2.4), bundle adjustment (*q.v.* Section 2.2.5) and outlier filtering.
Initialization of the incremental reconstruction is crucial (Beder and Steffen, 2006), since it affects all subsequent reconstruction steps. A bad initialization may lead to degenerated reconstructions or incomplete results.
Gao et al. (2003) and Lepetit et al. (2008) provide efficient solutions for the *Perspective-n-Point* (Fischler and Bolles, 1981) problem using 2D-3D correspondences to register new images to the set of three-dimensional scene points. The triangulation step leverages feature correspondences of the last registered camera to compute additional three-dimensional scene points. Extending the set of scene points with triangulation allows to potentially register new cameras to the model.
Deviations in the camera pose result in incorrectly triangulated points and vice versa. BA allows to leverage observation redundancies (*e.g.*, scene points that are observed by more than two views) to jointly optimize scene points and camera matrices (*q.v.* Section 2.2.5). The application of BA reduces the accumulation of errors during the reconstruction process. Without refinement of the intermediate reconstruction results, iterative SfM usually drifts into a non-recoverable state. Because of the computational costs of BA, it is not applied in each iteration.

Multi-View Stereo

The SfM pipeline may be extended optionally by an additional Multi-View Stereo building block, which computes a **dense reconstruction** representing the scene with a dense point cloud or a textured mesh.
Multi-View Stereo (MVS) estimates multi-view depth maps potentially including surface normals for registered images. Using the camera parameters computed in the previous sparse reconstruction step allows to leverage the (known) epipolar geometry to determine pixel correspondences, *i.e.*, for each pixel potential matches in other views are constrained by the corresponding epipolar line (*q.v.* Section 2.2.2). The dense reconstruction allows to recover surfaces that have no correspondence in the sparse point cloud, *e.g.*, areas without salient features.
Multiple pixels along the epipolar line may appear similar because of repetitive elements. Occlusions and reflections result potentially in no similar values at all. To tackle these issues, MVS determines dense correspondences with

similar appearance by simultaneously evaluating the epipolar lines of multiple views.
During Multi-View Fusion the previously computed depth maps (and corresponding normal vectors) of each camera are fused into a single dense point cloud leveraging the redundancy of overlapping views. Redundant points and normal vectors allow to filter remaining outliers.
The surface reconstruction step, leverages the dense point cloud and corresponding normals of the previous step to compute a watertight textured mesh, which shows beneficial properties for many application scenarios.

2.3.2 Ambiguity of Reconstruction Results

Without information about calibration or the pose of the cameras, the reconstruction shows a projective transformation ambiguity. In the calibrated case the reconstruction can be determined up to a similarity transformation. For example, scaling the reconstructed scene by a factor k and the camera matrices by a factor $1/k$ according to (2.18) does not change the projections of the scene points. This shows that SfM reconstructions are scale ambiguous.

$$\mathbf{m} = \mathbf{P}\mathbf{x} = (\frac{1}{k}\mathbf{P})(k\mathbf{x}) \tag{2.18}$$

Similarly, applying a transformation to the scene and the corresponding inverse transformation to the camera projection matrices from the right side (see (2.19)), preserves the measurements.

$$\mathbf{m} = \mathbf{P}\mathbf{x} = (\mathbf{P}\mathbf{T}^{-1})(\mathbf{T}\mathbf{x}) \tag{2.19}$$

2.3.3 Related Work and State-of-the-Art

Longuet-Higgins (1981) proposed one of the first algorithms (using two images) to reconstruct scene structures. Subsequent works leverage the redundancy of multiple images (Hartley, 1994; Shashua and Werman, 1995; Szeliski and Kang, 1994; Tomasi and Kanade, 1992). Modern SfM pipelines are able to reconstruct more than hundred thousand (Agarwal et al., 2009), millions (Frahm et al., 2010; Schönberger et al., 2015b), and even several tens of mil-

lions of images unstructured images (Heinly et al., 2015). These methods may be divided into iterative and global approaches. Iterative or sequential SfM methods (Moulon et al., 2012; Schönberger and Frahm, 2016; Snavely et al., 2006; Wu, 2011) are more likely to find reasonable solutions than global SfM approaches (Moulon et al., 2013; Sweeney et al., 2015; Wilson and Snavely, 2014). However, the latter are less prone to drift.
Incremental and global SfM pipelines rely on an efficient feature matching (Agarwal et al., 2009; Frahm et al., 2010; Havlena and Schindler, 2014; Heinly et al., 2015; Lou et al., 2012; Schönberger et al., 2015a) to handle large scale reconstruction problems. For incremental SfM methods the selection of the intitial image pair is especially important (Beder and Steffen, 2006), since all subsequent computation steps depend on it. Bao and Savarese (2011) propose to leverage semantic information of static objects, *i.e.*, object detections, as an additional constraint to estimate the scene structure and the camera motion.
The advances in image-based modeling led to a number of open source SfM and MVS tools and libraries: Bundler (Snavely et al., 2006), OpenMVG (Moulon et al., 2012), TheiaSfM (Sweeney et al., 2015), MVE (Fuhrmann et al., 2015), Colmap (Schönberger and Frahm, 2016; Schönberger et al., 2016) and Meshroom (Jancosek and Pajdla, 2011; Moulon et al., 2012). But also commercial products such as Pix4D (Pix4D, 2019), Metashape (Agisoft, 2019) and RealityCapture (Capturing Reality, 2019) are available. Li et al. (2010), Crandall et al. (2011) and Wilson and Snavely (2014) present different datasets with unordered image collections to evaluate large scale Structure from Motion algorithms.

2.4 Factor Graphs

Factor graphs (Kschischang et al., 2001) are a family of probabilistic graphical models representing factorizations of functions and are applicable when a (difficult) problem can be expressed as a product of (simple) *local* functions, which depend only on a subset of the problem variables. The factorization of a probability distribution function with factor graphs and the so called sum-product algorithm (Kschischang et al., 2001) allows to compute marginal distributions making it suitable for modeling and solving inference problems. In computer vision, factor graphs have a widespread use to determine camera poses and to estimate landmark positions. Similar to bundle adjustment (*q.v.*

Section 2.2.5) factor graphs may be used to minimize the reprojection error of existing reconstructions. However, the flexibility of the framework allows to model more complex constraints such as stereo or odometry constraints. We leverage these capabilities in Section 6.5.3 to compute consistent object trajectories in stereo image sequences.

2.4.1 Function Factorization

We use (2.20) to define the factorization of a function f.

$$f(\Theta) = \prod_k f_k(\Theta_k) \tag{2.20}$$

The local functions $f_k(\cdot)$ only depend on a subset of variables Θ_k. Using a Gaussian measurement model allows to compute the factors $f_k(\Theta_k)$ according to (2.21).

$$f_k(\Theta_k) \propto \exp\left(-\frac{1}{2}\|h_k(\Theta_k) - z_k\|^2_{\Sigma_k}\right) \tag{2.21}$$

Here, $\|\mathbf{e}\|^2_{\Sigma_k} = \mathbf{e}^\mathsf{T}\Sigma_k{}^{-1}\mathbf{e}$ denotes the squared Mahalanobis distance with covariance matrix Σ_k. $h_k(\Theta_k)$ and z_k denote the measurement function and measurement corresponding to the factor f_k. For more details see Dellaert and Kaess (2017).
Let be Θ^* the optimal variable assignment that maximizes (2.20), *i.e.*,

$$\Theta^* = \operatorname*{argmax}_{\Theta} f(\Theta) = \operatorname*{argmax}_{\Theta} \log f(\Theta). \tag{2.22}$$

Plugging (2.20) and (2.21) in (2.22) results in an optimization problem of *nonlinear* least-squares as shown in (2.23). We drop the factor $\frac{1}{2}$, since it does not affect the solution. In the case of probability distributions all variable values f_k are between zero and one. Using the logarithm avoids numerical effects, *e.g.*, instabilities, when computing (2.23) for large k.

$$\Theta^* = \operatorname*{argmin}_{\Theta}(-\log f(\Theta)) = \operatorname*{argmin}_{\Theta} \sum_k \|h_k(\Theta_k) - z_k\|^2_{\Sigma_k} \tag{2.23}$$

We minimize the nonlinear least-squares problem shown in (2.23) to find the optimal variable assignment Θ^*. To determine the maximum a posteriori

(MAP) estimate, one may apply the Levenberg-Marquardt algorithm (Levenberg, 1944) to (2.23), which solves the nonlinear least-squares problem iteratively. In each iteration, the measurement functions $h_k(\cdot)$ are linearized using a Taylor expansion according to (2.24).

$$\begin{aligned} h_k(\Theta_k) &= h_k(\Theta_k^0 + \Delta_k) \\ &\simeq h_k(\Theta_k^0) + \left.\frac{\partial h_k(\Theta_k)}{\partial \Theta_k}\right|_{\Theta_k^0} \Delta_k \\ &:= h_k(\Theta_k^0) + H_k \Delta_k \end{aligned} \tag{2.24}$$

Here, $\Delta_k = \Theta_k - \Theta_k^0$ denotes the state update vector. To find a solution Δ^* for the locally linearized problem, we plug (2.24) into (2.23) according to (2.25), which is a *linear* least-squares problem.

$$\begin{aligned} \Delta^* &= \underset{\Delta}{\operatorname{argmin}} \sum_k \|h_k(\Theta_k^0) + H_k \Delta_k - z_k\|^2_{\Sigma_k} \\ &= \underset{\Delta}{\operatorname{argmin}} \sum_k \|H_k \Delta_k - (z_k - h_k(\Theta_k^0))\|^2_{\Sigma_k} \\ &= \underset{\Delta}{\operatorname{argmin}} \sum_k \|\Sigma_k^{-1/2} H_k \Delta_k - \Sigma_k^{-1/2}(z_k - h_k(\Theta_k^0))\|^2_2 \\ &:= \underset{\Theta}{\operatorname{argmin}} \sum_k \|A_k \Delta_k - b_k\|^2_2 := \underset{\Theta}{\operatorname{argmin}} \|\mathbf{A}\Delta - \mathbf{b}\|^2_2 \end{aligned} \tag{2.25}$$

Note that $\mathbf{A}$ in (2.25) is a sparse block matrix.

2.4.2 Factor Graphs, Function Factorization and Image-Based Reconstructions

A factor graph (Kschischang et al., 2001) is a bipartite graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ with two node types: *factor* nodes $f_k \in \mathcal{F}$ and *variable* nodes $\theta_l \in \Theta$. The edges $e_{k,l} \in \mathcal{E}$ connect factor and variable nodes. In the context of factor graphs, Θ_k in (2.20) represents the set of variables θ_l adjacent to f_k, *i.e.*, each $\theta_l \in \Theta_k$ is connected with an edge to f_k. Variable nodes represent quantities we want to estimate. In contrast, factor nodes define constraints on variable nodes and represent prior knowledge or information of measurements.

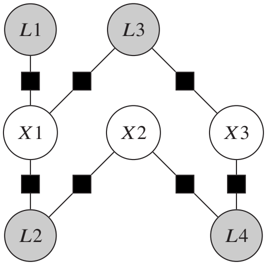


Figure 2.5: Example of a factor graph representing a Structure from Motion problem. The gray and white circles denote landmarks and camera poses, respectively. The image observations (*i.e.*, the keypoint measurements) impose constraints on possible camera poses and landmark positions. The black squares represent such constraints.

Examples for variable and factor nodes are camera poses and keypoint correspondences, respectively. Fig. 2.5 shows a factor graph representing a typical Structure from Motion problem.

Dellaert and Kaess (2006) analyse the connection of Visual SLAM problems to factorization and factor graphs. The paper shows that the block-structure of $\mathbf{A}$ in (2.25) corresponds to the adjacency matrix of a Gaussian factor graph representing the same problem. Additionally, they point out that factorization and variable elimination are equivalent. Kaess et al. (2008) propose an approach to tackle SLAM problems with incremental matrix factorizations. SLAM algorithms add new observations incrementally, $\mathbf{A}$ and (2.25) must be recomputed for each time step. Since $\mathbf{A}$ is a potentially huge (sparse) matrix, the repetitive computation of (2.25) becomes computationally expensive with increasing problem sizes. Kaess et al. (2011) propose a graph-based counterpart to Kaess et al. (2008). This algorithm benefits from the compact representation and the spatial structure of factor graphs, *i.e.*, only a local part of the factor graph must be updated when new data is inserted. Recently, Dellaert and Kaess (2017) present a detailed summary of Dellaert and Kaess (2006) and Kaess et al. (2011).

In Section 6.5.3 we define stereo constraints leveraging the GTSAM library (Daellert, 2012), which provides implementations of the algorithms presented in Kaess et al. (2011).

2.5 Summary

In this chapter we have seen how three-dimensional models of rigid scenes may be reconstructed from image observations. We emphasized important properties of local image description methods to determine salient and distinct features suitable for matching across different images. One of the central insights in image-based modeling is that plain point correspondences are sufficient to compute camera parameters and scene structures. The fundamental matrix (uncalibrated case) or essential matrix (calibrated case) represent the corresponding geometric constraints for feature observations in two views. We present important building blocks of current state-of-the-art incremental Structure from Motion approaches. In addition, we give a brief introduction into factor graphs - a general frame work allowing to model image-based reconstruction constraints, which are not part of most SfM pipelines. A combination of factor graphs with different image-based modeling methods is recommended, since factor graphs require a reasonable initialization and are not suitable to perform data association of many real-world problems.

3 Instance-Aware Multibody Structure from Motion

This chapter presents a novel *Multibody* Structure from Motion approach that allows to reconstruct multiple dynamic objects in mainly static environments using image sequences. Parts of this chapter have been published in Bullinger et al. (2017), Bullinger et al. (2018b) and Bullinger et al. (2019b). The proposed concepts are the foundation of the object shape and object trajectory reconstruction algorithms presented in Chapter 5 and Chapter 6.

Multibody Structure from Motion is an extension of standard Structure from Motion (*q.v.* Section 2.3) that allows to reconstruct independently moving non-deformable objects. As we have seen in Algorithm 2 in Section 2.2.3, Standard SfM is usually limited to static scenes, since the determination of the fundamental matrix between two cameras depends only on the largest inlier set of putative matches. Other correspondences are treated as outliers and are not reconstructed.

In contrast to existing Multibody Structure from Motion approaches, we use a combination of instance-aware semantic segmentation and optical flow methods to determine object specific keypoints in image sequences. The approach is robust to occlusion and handles stationary as well as parallel moving objects, which represent challenging cases for many previously proposed algorithms.

The remaining part of this chapter is organized as follows. We give a short description of the problem statement in Section 3.1 and discuss related work of Multibody Structure from Motion methods in Section 3.2. In Section 3.3 we give a brief overview of the proposed pipeline. Section 3.4 presents the association of object detections in monocular (Section 3.4.4) and binocular (Section 3.4.5) image sequences, which is used in Section 3.5 to determine object and backround specific point clouds as well as corresponding camera poses. Section 3.6 highlights important implementation details. In Section 3.7 we quantitatively evaluate the proposed tracking algorithm using the MOT

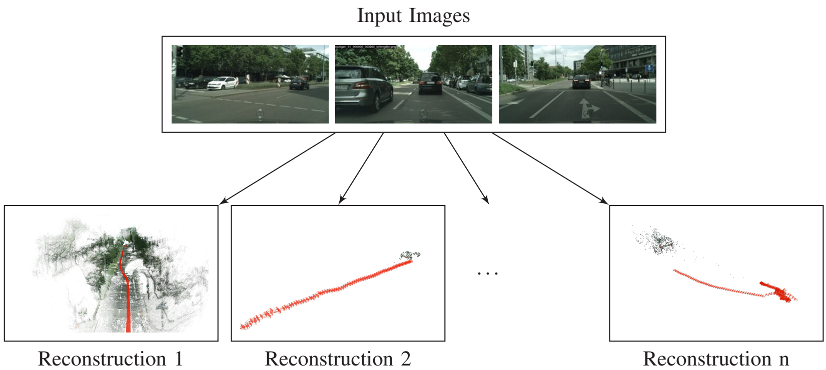


Figure 3.1: Visualization of the MSfM problem. Independently moving scene components are reconstructed separately. Each reconstruction is represented by a point cloud and a corresponding set of camera poses (red).

dataset (Leal-Taixé et al., 2015). Section 3.8 discusses important properties of our Multibody Structure from Motion approach.

3.1 Problem Statement

The reconstruction of dynamic scenes with visual information is an under-constrained task because of possible object deformations. To mitigate this issue, Multibody Structure from Motion (MSfM) assumes that the scene can be modeled as a *multibody system*, *i.e.*, the scene consists of non-deformable components moving independently. This allows to leverage Multiple View Geometry techniques (*q.v.* Section 2.2) to reconstruct independent scene components simultaneously.
As shown in Algorithm 2 in Section 2.2.3, standard Structure from Motion methods sample putative matches, *e.g.*, feature correspondences, randomly to determine the fundamental matrix of two images with the largest inlier set. This elementary step of current state-of-the-art SfM pipelines limits their application to static environments. In scenes with multiple independently moving components, random sampling of putative matches leads (for non-degenerated cases) to fundamental or essential matrices with inlier matches of

one element - features corresponding to other scene components are considered as outliers. The resulting reconstruction contains only three-dimensional scene structures of a single scene element. To reconstruct individually moving scene components we need to determine object specific image correspondences.
We consider monocular and stereo image sequences of dynamic objects in mainly static environments captured by a single camera. Fig. 3.1 shows a visualization of the problem statement.
SfM based reconstructions define restrictions on the poses of the reconstructed scene elements. In the scenario of moving objects in a mainly static environment, it is suitable to express the relative object poses w.r.t. the coordinate frame system of the environment. Such pose constraints are essential to reconstruct object trajectories as demonstrated in Chapter 6.

3.2 Related Work

Many Multibody Structure from Motion methods use epipolar constraints (with motion segmentation) to determine corresponding feature matches to reconstruct multiple objects simultaneously.
Epipolar constraint based approaches (Fitzgibbon and Zisserman, 2000; Kundu et al., 2009; Kundu et al., 2011; Lebeda et al., 2014; Ozden et al., 2010; Sabzevari and Scaramuzza, 2016) determine, if there exists a valid fundamental or essential matrix for each potential object pair in a pair of images. This task is highly non-trivial, since the random selection of putative matches in scenes with many independently moving components has a high probability to result in inconsistent inlier sets, *i.e.*, the selected matches correspond to different components. Let w describe the probability that an arbitrary feature match is an inlier w.r.t. to an specific object. Further, let p denote the probability that out of n different samples at least one sample consisting of s different feature matches contains only inliers. Then, w, p, n and s show the relation in (3.1).

$$(1 - w^s)^n = 1 - p \Leftrightarrow n = \frac{log(1 - p)}{log(1 - w^s)} \tag{3.1}$$

A common choice for p is $p = 0.99$ (Hartley and Zisserman, 2004). As we have seen in Section 2.2.3 the fundamental matrix may be computed with the normalized 8-point algorithm from $s = 8$ feature observations. Let us consider a pair of images showing 2, 5 and 10 different components with an equal number of object specific feature observations, *i.e.*, $w_2 = 0.5$, $w_5 = 0.2$, $w_{10} = 0.1$. To achieve a probability of $p = 0.99$ we need $n_2 \approx 1177$, $n_5 \approx 1.8 \cdot 10^6$ and $n_{10} \approx 4.6 \cdot 10^8$ sampling steps, respectively. In addition, it is difficult to identify adequate thresholds to correctly determine object specific observations without semantic contextual information. For instance, it is unclear how to define the minimum size of valid inlier sets.

Different attempts have been made to determine object specific feature correspondences with a reduced computational effort. For example, Rubino et al. (2015) present an approach that uses higher semantics, *i.e.*, the output of an object detector, to determine consistently moving feature matches with a reduced number of samples n. Unfortunately, Rubino et al. (2015) do not provide any reconstruction results, which show the efficiency of the proposed algorithm.

Another approach to reduce the computational effort is the exploitation of inherent characteristics of the image data. For example, motion segmentation and keypoint tracking based methods (Kundu et al., 2009; Kundu et al., 2011; Lebeda et al., 2014; Yuan and Medioni, 2006) allow to find feature correspondences in image sequences. These methods use visual information of subsequent images to determine corresponding object specific feature observations. This simplifies the problem of associating corresponding matches and reduces the computational complexity. However, motion segmentation shows limitations in certain situations such as consistently moving and partly stationary objects. Furthermore, these methods are vulnerable to occlusion. Specific approaches such as Kundu et al. (2011) or Grinberg (2018) are required to merge different feature sets, which are separated by occlusions.

MSfM reconstructions are inherently scale ambiguous. Additional constraints are required to solve the scale ambiguity. Song and Chandraker (2015), Lee et al. (2015) and Chhaya et al. (2016) assume that the camera is mounted on a driving vehicle, *i.e.*, the camera has specific height and a known pose. Ozden et al. (2004) propose the *non-accidental motion* principle, which allows to solve the scale ambiguity by making assumptions about object and camera motion trajectories. Yuan and Medioni (2006), Namdev et al. (2013) and Park et al. (2015) follow this principle introducing complementary motion constraints. Özden (2007) provides an extended analysis of the *non-accidental motion* principle.

Other works tackling the problem of NRSfM such as Russell et al. (2014) and Kumar et al. (2016) are beyond the scope of this work.

3.3 Pipeline Overview

The proposed Multibody Structure from Motion method uses instance-aware semantic segmentations and optical flow correspondences to compute video object segmentations on pixel level. This allows us to determine object specific feature observations throughout video sequences. In contrast to epipolar constraint and motion segmentation based methods, our approach handles stationary and parallel moving objects naturally.
Fig. 3.2 shows the pipeline of the proposed approach allowing us to process monocular as well as stereo image sequences. The algorithm tracks multiple two-dimensional object shapes on pixel level throughout video sequences. Details of the multiple object tracking (MOT) approach are described in Section 3.4. The method uses instance-aware semantic segmentations (Li et al., 2017) to identify object shapes and optical flow features (Ilg et al., 2017) to associate extracted object shapes in subsequent frames. The tracking of object shapes on pixel level allows us to compute for each object a set of images containing only color information corresponding to this object instance - *q.v.* Fig. 3.2. We use the complement of all detected objects to create a set of environment images. The sets of object and background images allow us to determine object and background specific feature points. We apply SfM (Moulon et al., 2012; Schönberger and Frahm, 2016) as shown in Fig. 3.2 to compute object and background specific camera poses as well as corresponding point clouds. In scenarios with dynamic objects in mainly static environments, it is suitable to express object poses w.r.t. to the background coordinate frame system. We describe details about the reconstruction and relative object poses in Section 3.5.

3.4 Multiple Object Tracking for Multibody Structure from Motion

Most current Multiple Object Tracking (MOT) methods use two-dimensional bounding boxes to represent object detections. Bounding boxes contain not only detected objects but also background structures, which makes them unsuitable to determine object specific feature points. This section describes an online MOT approach that allows to track objects on pixel level in monocular

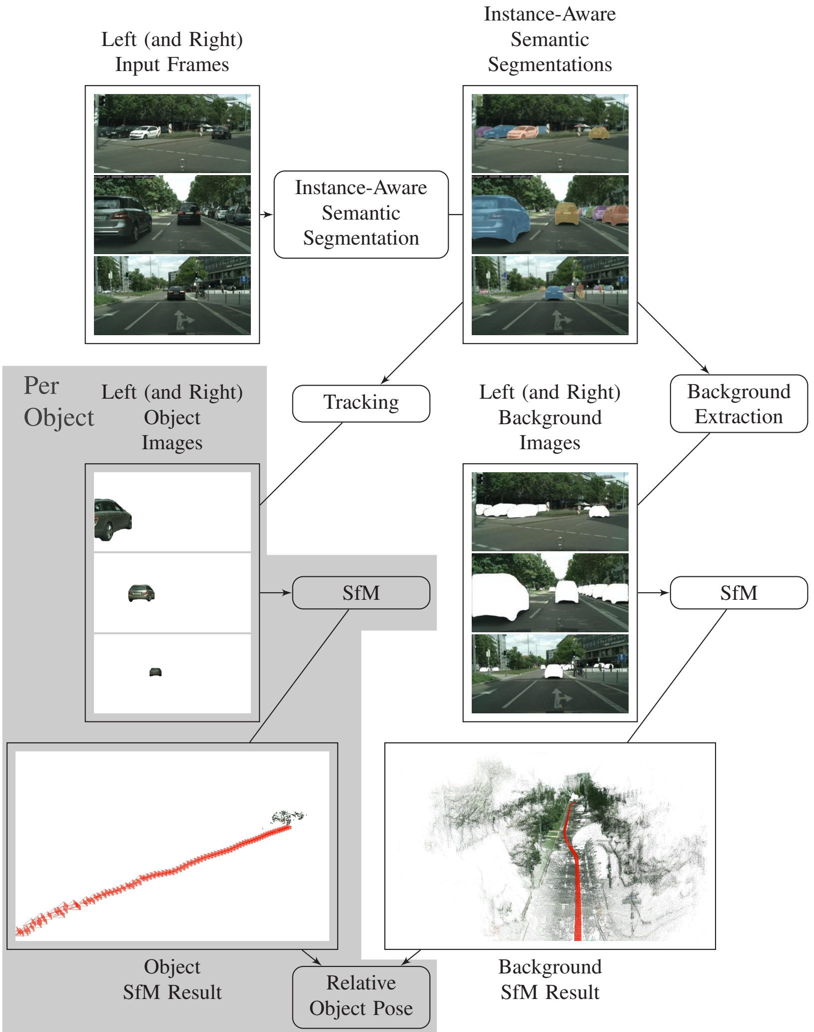


Figure 3.2: Overview of the MSfM pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps. We use instance-aware semantic segmentations as well as optical flow features to compute object and environment specific images. Corresponding three-dimensional models and camera poses (red rectangles) are computed with well known SfM techniques.

and stereo image sequences using instance-aware semantic segmentation (see Section 3.4.1) and optical flow (see Section 3.4.1).

3.4.1 Fundamentals and Terminology

This section describes briefly the fundamentals of the proposed Multiple Object Tracking approach, *i.e.*, instance-aware semantic segmentation as well as optical flow. The section presents corresponding formal definitions and important related work. We use the presented notation throughout the following sections.

Instance-Aware Semantic Segmentation

Semantic segmentation or *scene parsing* is the task of determining a class or a category label for each pixel of a given color image. Note that class labels are agnostic to instance information. There is typically an additional class to label unknown image areas. In addition to class labels, *Instance-aware semantic segmentation* assigns unique object identifiers for a subset of pixels. Not all pixels represent objects, but background categories like street or sand. Instance-aware semantic segmentation allows to determine the two dimensional shape on pixel level for each object.

Formal Definition Let $\mathbf{I}_i$ denote the i-th image of an ordered sequence with height h and width w. Furthermore, let $\mathbf{I}_i(x, y)$ denote the color of pixel position (x, y) in $\mathbf{I}_i$ with $(x, y) \in \{1, \cdots, w\} \times \{1, \cdots, h\}$.
Instance-aware segmentation systems like Dai et al. (2016), Li et al. (2017) or He et al. (2017), predict for each pixel position of an input image $\mathbf{I}_i$ a semantic category label c and a corresponding instance index u according to (3.2),

$$\mathbf{S}_i(x, y) = (c, u) \tag{3.2}$$

where $\mathbf{S}_i$ denotes the instance-aware semantic segmentation of image $\mathbf{I}_i$.
Fig. 3.3 and Fig. 3.4 show an example for semantic segmentation and instance-aware semantic segmentation, respectively.

Figure 3.3: Semantic segmentation examples with images of the Pascal VOC dataset (Everingham et al., 2010) using Long et al. (2015). Each pixel is labeled with a semantic category. The colors highlight different classes. Semantic labels are agnostic to instance information.

Related Work and State-of-the-Art Semantic segmentation or scene parsing is the task of providing semantic information at pixel level. Early works, like Rother et al. (2004), require rough human annotated fore- and background information to compute an exact fore- and background segmentation. In contrast, semantic segmentation approaches using pre-trained ConvNets do not require any supervision at run time. Early semantic segmentation approaches using ConvNets, *e.g.*, Farabet et al. (2013), exploit patchwise training.
Long et al. (2015) propose a new architectural style of ConvNets, so called Fully Convolutional Networks (FCNs), which allow for end-to-end training. The training of FCNs from scratch is not feasible, because the annotation of training data for semantic segmentation is quite expensive (*e.g.*, 60 minutes per image for the CamVid dataset (Brostow et al., 2009) or 90 minutes per image for the Cityscapes dataset (Cordts et al., 2016)). Instead, a common approach is to modify a ConvNet trained on another domain with simpler ground truth annotations. Usually a ConvNet trained for classification like Simonyan and Zisserman (2015) or He et al. (2016) serves as backbone, *i.e.*, the last layers

Figure 3.4: Instance-aware semantic segmentation examples with images of the Pascal VOC dataset (Everingham et al., 2010) using Mask-RCNN He et al. (2017). Note that different instances of the same category, *e.g.*, car and person, show individual colors indicating different object identifiers.

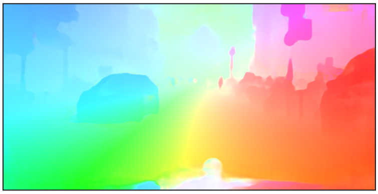
in the original network are replaced with layers designed specific for semantic segmentation. During fine-tuning, *i.e.*, (re-)training of the new ConvNet for semantic segmentation, mainly the new layers defining the purpose of the network are updated.

Complementary, several synthetic datasets for semantic segmentation have been proposed (Richter et al., 2016; Ros et al., 2016) to mitigate the lack of real-world training data.

The concepts proposed in Long et al. (2015) inspired many state-of-the-art semantic segmentation approaches. Different works (Chen et al., 2014; Lin et al., 2016; Zheng et al., 2015) combine Convolutional Networks with Conditional Random Fields to refine the segmentation at boundaries of scene components with different category labels.

Dai et al. (2016) propose a novel approach called *Multi-task Network Cascades* (MNC), which tackles the task of instance-aware semantic segmentation. In contrast to semantic segmentation, instance-aware semantic segmentation label only pixels corresponding to object-like classes, *e.g.*, persons or

(a) Overlay of two subsequent images.



(b) Visualization of the optical flow.

Figure 3.5: Optical flow (Sun et al., 2018) examples with two images of the Cityscapes dataset (Cordts et al., 2016) using the color coding defined in Fig. 3.6.

cars. Dai et al. (2016) rely on Region Proposal Networks presented in Ren et al. (2015). This concept has been improved in Li et al. (2017) and He et al. (2017).

Kirillov et al. (2018) propose the task of *panoptic segmentation*, *i.e.*, the joint consideration of semantic segmentation and instance-aware semantic segmentation (*i.e.*, countable and non-countable classes). Kirillov et al. (2019) present a generalization of the Mask R-CNN (He et al., 2017) for panoptic segmentation.

During this thesis, several networks, have been evaluated for semantic segmentation (Shelhamer et al., 2017) as well as instance-aware semantic segmentation (Dai et al., 2016; He et al., 2017; Li et al., 2017). Common datasets to train and to evaluate ConvNets are PASCAL VOC (Everingham et al., 2010), Microsoft COCO (Lin et al., 2014), Cityscapes (Cordts et al., 2016) and ADE20K (Zhou et al., 2017).

Optical Flow

The concept of *Optical Flow* was proposed by Gibson (1950) and describes the apparent motion of structures caused by the relative motion between observer and scene. Optical flow estimation techniques allow to determine pixel correspondences in ordered image sequences. Usually, it is not possible to determine a correspondence for each pixel due to occlusions and field of view limitations.

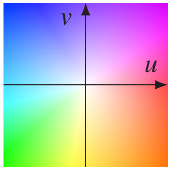


Figure 3.6: Optical flow color coding scheme proposed in Baker et al. (2011). Each optical flow vector $\mathbf{F}_{i \to i'}(x, y) = (u, v)$ is represented with a color according to the direction and the length of (u, v).

Formal Definition Optical Flow or quasi-dense matching methods like Revaud et al. (2016), Hu et al. (2016) or Ilg et al. (2017) compute for a pair of images denoted as $\mathbf{I}_i$ and $\mathbf{I}_{i'}$ a two-dimensional pixel offset field. The optical flow $\mathbf{F}_{i \to i'}(x, y)$ of a pair of images $\mathbf{I}_i$ and $\mathbf{I}_{i'}$ at a non-occluded pixel position (x, y) shows the relation in (3.3).

$$\mathbf{I}_i(x, y) \simeq \mathbf{I}_{i'}(x + \mathbf{F}_{h,i \to i'}(x, y), y + \mathbf{F}_{v,i \to i'}(x, y)) \tag{3.3}$$

Here, $\mathbf{F}_{h,i \to i'}$ and $\mathbf{F}_{v,i \to i'}$ denote the horizontal and the vertical component of $\mathbf{F}_{i \to i'}$. Fig. 3.5 shows two example results of Ilg et al. (2017) on the KITTI dataset (Geiger et al., 2013) - the corresponding optical flow color coding scheme is explained in Fig. 3.6.
Some optical flow algorithms estimate the optical flow only for a subset of pixels. In this case, there are pixel positions where no optical flow information is available. We will denote the set of pixel positions with valid flow information at time i with $\mathcal{V}_i$.

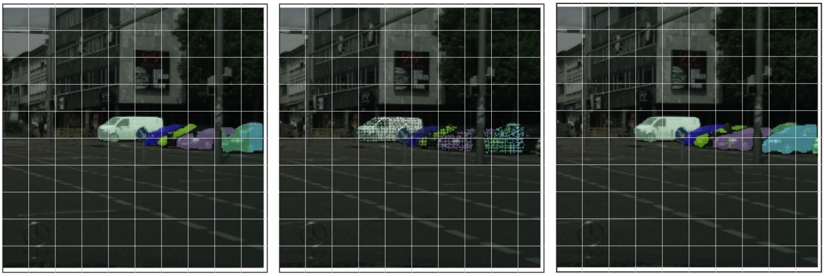
Related Work and State-of-the-Art The field of optical flow may be subdivided in sparse, quasi-dense and dense estimation approaches. The computation of optical flow information depends on scene structure and surface textures. For example, homogeneous areas hamper the estimation of consistent vectors. Sparse methods determine optical flow information only for (small) pixel sets with suitable local neighborhoods. This allows to compute robust optical flow vectors, which may be used to to track keypoints. A common approach is for example the combination of *Good Features to Track* (Jianbo Shi and Tomasi, 1994) and the Lucas-Kanade optical flow estimation approach (Lucas and Kanade, 1981).

In contrast to sparse methods, (quasi-)dense approaches aim to estimate optical flow vectors for all image regions. We observed that early dense optical flow methods like Farnebäck (2003) fail to compute correct optical flow vectors of videos captured by moving cameras. Already small camera motions result in inconsistent optical flow fields. *Epic Flow* (Revaud et al., 2015) computes considerably more robust dense optical flow vectors leveraging quasi-dense matches (Weinzaepfel et al., 2013). Revaud et al. (2015) use the *Sintel* (Butler et al., 2012), the *KITTI* (Geiger et al., 2013) and the *Middlebury* (Baker et al., 2011) dataset for evaluation.
Dosovitskiy et al. (2015) introduced a paradigm shift by proposing the first end-to-end trained ConvNet estimating optical flow vectors directly from raw image data. The authors trained the network on the *Flying Chair* dataset. Leveraging the *FlyingThings3D* dataset (Mayer et al., 2016) led to state-of-the-art optical flow methods using this paradigm (*cf*. FlowNet2 (Ilg et al., 2017)). However, FlowNet2 must be trained sequentially to avoid over-fitting. Integrating pyramidal processing, warping, and the use of a cost volume into a ConvNet architecture, PWC-Net (Sun et al., 2018) offers a end-to-end trainable pipeline outperforming previously published methods.
Other works like Sevilla-Lara et al. (2016) and Bai et al. (2016) propose methods that leverage semantic information for optical flow computations. Such information is presumably beneficial for the computation of consistent optical flow vectors close to object boundaries. However, there overall results are outperformed by current state-of-art approaches such as PWC-Net (Sun et al., 2018).
In our experiments, we observed that *Coarse to fine patch match* (CPM) (Hu et al., 2016) outperforms Sun et al. (2018) in the case of large object displacements. This is an important property of optical flow methods used for multiple object tracking (*q.v*. Section 3.7).

3.4.2 Prediction of Segmentation Instances

We use the ConvNet presented in He et al. (2017) to compute the instance-aware semantic segmentation $\mathbf{S}_i$ for image $\mathbf{I}_i$. For an instance with index u of the target category c we use $\mathbf{S}_i$ to extract the corresponding set of occupied pixel positions $\mathcal{S}_{i,u}$. More formally, we compute $\mathcal{S}_{i,u}$ according to (3.4).

$$\mathcal{S}_{i,u} = \{(x,y)|(x,y) \in \{1,\cdots,w\} \times \{1,\cdots,h\} \wedge \mathbf{S}_i(x,y) = (c,u)\} \tag{3.4}$$



(a) Instance-aware segmentations at time i.
(b) Predicted instance-aware segmentations from frame i to $i + 1$.
(c) Instance-aware segmentations at time $i + 1$.

Figure 3.7: Instance prediction using optical flow. Different colors indicate different object segmentations.

For a pair pair of images $\mathbf{I}_i$ and $\mathbf{I}_{i'}$ we compute the optical flow $\mathbf{F}_{i\rightarrow i'}$. This allows us to predict the pixel positions (x, y) contained in $\mathcal{S}_{i,u}$ to image $\mathbf{I}_{i'}$. We denote the set of predicted pixel positions as $\mathcal{P}_{i\rightarrow i',u}$ and compute it according to (3.5),

$$\mathcal{P}_{i\rightarrow i',u} = \{(x, y) + \mathbf{F}_{i\rightarrow i'}(x, y) | (x, y) \in \mathcal{V}_{i,u}\} \tag{3.5}$$

where $\mathcal{V}_{i,u} = \mathcal{S}_{i,u} \cap \mathcal{V}_i$ is the set of valid optical flow positions of instance u. Fig. 3.7 shows the prediction of several instance segmentations using He et al. (2017) and Hu et al. (2016).

If the optical flow algorithm does not provide flow information for each pixel, we interpolate the optical flow at positions where no flow information is available. This allows us to compute dense predictions of instance segmentations. We interpolate the optical flow for each instance separately to avoid the influence of optical flow vectors corresponding to other objects and background structures, *i.e.*, we consider only vectors at pixel positions $\mathcal{V}_{i,u}$. We use a linear interpolation of points inside the convex hull of $\mathcal{V}_{i,u}$. The optical flow of points lying outside the convex hull is interpolated using the corresponding nearest neighbor.

The interpolation of optical flow vectors pointing in opposite directions generates holes and overlaps in the predicted segmentation instance. Consider the following one-dimensional example with four adjacent pixel positions and two optical flow values at the first and fourth position: $[-3, \_, \_, 3]$. The linear interpolation of the missing optical flow values yields $[-3,-1,1,3]$. Shifting the second and the third point according to the corresponding optical flow values, *i.e.*, -1 and 1, moves the second pixel to the left as well as the third pixel to the right and leaves a hole in the corresponding segmentation mask. We close these holes by performing a morphological closing operation.

3.4.3 Affinity of Objects in Pairs of Images

To associate objects in image $\mathbf{I}_i$ with objects in frame $\mathbf{I}_{i'}$ we compute an affinity score between corresponding instance segmentations. We define the similarity of an object with index u in frame $\mathbf{I}_i$ and an object with index v in frame $\mathbf{I}_{i'}$ as the overlap of the intersection of the predicted pixel set $\mathcal{P}_{i\to i',u}$ and the pixel set of instance segmentation $\mathcal{S}_{i',v}$. Note that the number of objects and the order of the corresponding indices may differ. This formulation of the affinity measure reflects locality and visual similarity. Let $o_{u,v}$ denote the overlap of the prediction $\mathcal{P}_{i\to i',u}$ and segmentation $\mathcal{S}_{i',v}$, *i.e.*, $o_{u,v} = \#(\mathcal{P}_{i\to i',u} \cap \mathcal{S}_{i',v})$. Furthermore, let n_u and n_v denote the number of segmentation instances in image $\mathbf{I}_i$ and $\mathbf{I}_{i'}$, respectively. We build an affinity matrix $\mathbf{A}$ using the pairwise overlaps $o_{u,v}$ according to (3.6)

$$\mathbf{A}_{i\to i'} = \begin{bmatrix} o_{1,1} & \cdots & o_{1,v} & \cdots & o_{1,n_v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ o_{u,1} & \cdots & o_{u,v} & \cdots & o_{u,n_v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ o_{n_u,1} & \cdots & o_{n_u,v} & \cdots & o_{n_u,n_v} \end{bmatrix} \tag{3.6}$$

The rows and columns may contain multiple non-zero entries because of incorrect instance segmentations and optical flow computations. The Hungarian method (Kuhn, 1955) is a common algorithm to resolve such ambiguities in affinity matrices.

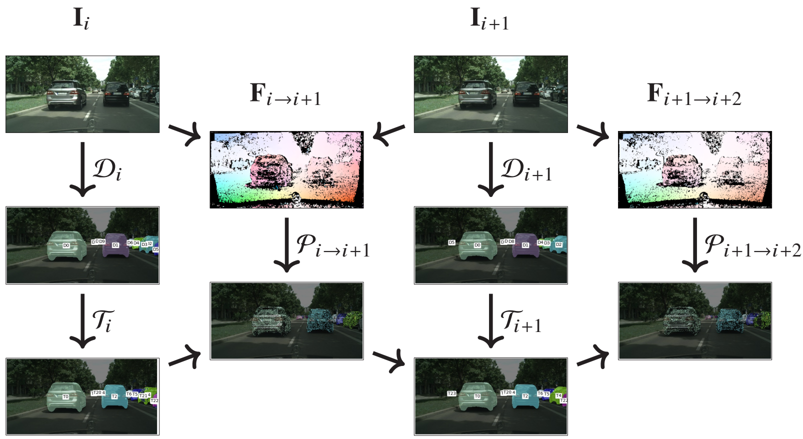


Figure 3.8: Monocular object tracking scheme. The variables have the following meaning. **I**: image, **F**: optical flow, $\mathcal{D}$: detection, $\mathcal{P}$: Prediction, $\mathcal{T}$: Tracker State, i: image index. Arrows show the relation of computation steps. A computation step depends on the results connected with incoming arrows. The tracked objects $\mathcal{T}_i$ at time i are predicted to the next image using the optical flow $\mathbf{F}_{i\to i+1}$ of image $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$. The predictions $\mathcal{P}_{i\to i+1}$ are associated with the detections $\mathcal{D}_{i+1}$ to update the tracker state. The used optical flow color coding is defined in Baker et al. (2011) (*q.v.* Fig. 3.6).

3.4.4 Online Monocular Multiple Object Tracking on Pixel Level

This section presents an approach that allows to track multiple objects on pixel level in monocular image sequences. The main ideas have been already presented in Bullinger et al. (2017). Fig. 3.8 shows an overview of the tracking scheme. Let $\mathbf{I}_i$ denote the previous image and $\mathbf{I}_{i+1}$ the current image. The state of the proposed object tracker $\mathcal{T}_i$ at time i consists of a set of segmentation instances $\mathcal{S}_{i,k}$ with unique identifiers $id_{i,k}$ and a counter for the number of consecutive missing detections $m_{i,k}$, *i.e.*, $\mathcal{T}_i = \{(\mathcal{S}_{i,k}, id_{i,k}, m_{i,k}) | k \in \{1, \cdots, n_i\}\}$, where n_i is the number of tracks at time i. We initialize this state with the segmentation instances in the first frame (if any). The tracker state segmentations $\mathcal{S}_{i,k}$ are predicted to subsequent frames using (3.5). Let $\mathcal{P}_{i\to i+1,k}$ denote the corresponding predictions. In order to solve the association of predicted segmentation instances in the tracker state $\mathcal{P}_{i\to i+1,k}$ and segmentations instances

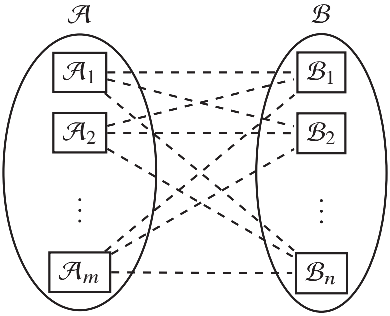
Algorithm 3: Object tracking in monocular video data.

Initialize tracker state with segmentation instances detected in frame 0.
for *Subsequent Frames* **do**
    Compute optical flow $\mathbf{F}_{i \to i+1}$ between the previous and the current image.
    Compute predictions $\mathcal{P}_{i \to i+1}$ using tracker state $\mathcal{T}_i$.
    Compute instance-aware semantic segmentations $\mathcal{D}_{i+1}$.
    Build affinity matrix corresponding to $\mathcal{P}_{i \to i+1,k}$ and $\mathcal{D}_{i+1,v}$.
    Solve associations using the hungarian method.
    Update matching $\mathcal{S}_{i,k}$ with corresponding $\mathcal{D}_{i+1,v}$.
    Update non-matching $\mathcal{S}_{i,k}$ with corresponding predictions $\mathcal{P}_{i \to i+1,k}$.
    Remove dead tracks using $m_{i,k}$.
    Create new tracks for all unmatched segmentations $\mathcal{D}_{i+1,v}$.
end

$\mathcal{D}_{i+1,v}$ found in current image we compute the affinity matrix $\mathbf{A}_{i \to i+1}$. We apply the Hungarian Method (Kuhn, 1955) on $\mathbf{A}_{i \to i+1}$, which results in a set of matching index pairs $\mathcal{M}_{i \to i+1}$. We ensure the validity of each index pair $(k, v) \in \mathcal{M}_{i \to i+1}$ by verifying that $\mathbf{A}_{i \to i+1}(k, v) > 0$.
For all valid index pairs $(k, v) \in \mathcal{M}_{i \to i+1}$ we update the segmentation instances maintained by the tracker, *i.e.*, we set $\mathcal{S}_{i,k} = \mathcal{D}_{i+1,v}$, but keep the unique tracklet identifier $id_{i,k}$. We add all non-matching segmentation instances found in image $\mathbf{I}_{i+1}$ with a new unique identifier to the set of segmentation instances maintained by the tracker. In addition, we remove all non-matching segmentation instances contained in the tracker state, if $m_{i,k} > md$, where md is the number of allowed missing detections. Otherwise, we replace the instance segmentation with a dense prediction of the corresponding pixel positions. Algorithm 3 summarizes the steps of the proposed online Multiple Object Tracking algorithm.

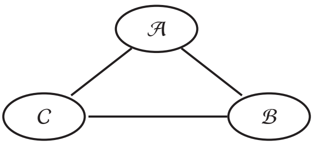
3.4.5 Online Stereo Multiple Object Tracking on Pixel Level

We extend the monocular Multiple Object Tracking approach described in the previous section to stereo image sequences following the algorithm proposed in Bullinger et al. (2019b).

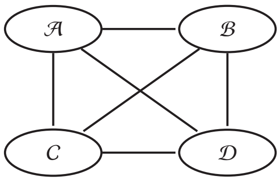


(a) *Detailed* visualization of the two-dimensional assignment problem as 2-partite graph.

(b) *Simplified* visualization of the two-dimensional assignment problem as 2-partite graph.



(c) *Simplified* visualization of the three-dimensional assignment problem as 3-partite graph.



(d) *Simplified* visualization of the four-dimensional assignment problem as 4-partite graph.

Figure 3.9: Comparison of the two-dimensional, the three-dimensional and the four-dimensional assignment problem (AP). The rectangles and the dashed lines in Fig. 3.9a denote the nodes and the corresponding weights of the graph. The circles represent the partition of the (multipartite) graph. There exist no edges between nodes of the same partition. The edge between two partitions denote that the corresponding nodes are also connected with edges - *q.v.* Fig. 3.9b. Fig. 3.9c and 3.9d show that the four-dimensional AP comprises the three-dimensional AP, *i.e.*, the edges in the graph representing the three-dimensional AP are a true subset of the edges in the graph representing the three-dimensional AP.

Stereo Multiple Object Tracking Complexity

Bullinger et al. (2017) use the Kuhn-Munkres algorithm (Kuhn, 1955) to assign corresponding objects in a pair of images. This task is an instance of the two-dimensional assignment problem (AP). The two-dimensional AP consists of finding a maximum weight matching in a weighted bipartite (or 2-partite) graph - *q.v.* Fig. 3.9a. In the context of object tracking, each partition of the graph represents the objects of one image. An improved version of the Kuhn-Munkres algorithm (Wong, 1979) solves the two-dimensional AP in $\mathcal{O}(n^3)$, where n is the number of elements to be assigned. Higher dimensional extensions of the two-dimensional AP, *i.e.*, higher multidimensional assignment problems (Pierskalla, 1969) like the three-dimensional or the four-dimensional AP, are NP-hard (Frieze, 1983; Gilbert and Hofstra, 1988). Fig. 3.9 shows a comparison of the two-dimensional, the three-dimensional and the four-dimensional AP. In the stereo MOT case, object instances in the left image $\mathbf{I}_{i,l}$ and the right image $\mathbf{I}_{i,r}$ at time i as well as the object instances in the left image $\mathbf{I}_{i+1,l}$ and the right image $\mathbf{I}_{i+1,r}$ at time $i+1$ must be associated. Therefore, the stereo MOT AP is an instance of the four-dimensional AP and is NP-hard as well.

Online Stereo Multiple Object Tracking Algorithm

The proposed stereo Multiple Object Tracking method extends the monocular tracking algorithm presented in Bullinger et al. (2017). Fig. 3.10 shows a scheme of the tracking aproach. The prediction and association of object detections described in Section 3.4.4 allows to associate objects not only in subsequent frames, but also in the left and right image of a stereo camera. We do not solve the associations of $\mathbf{I}_{i,l}$, $\mathbf{I}_{i+1,l}$, $\mathbf{I}_{i,r}$ and $\mathbf{I}_{i+1,r}$ simultaneously, since (a) the brute force search for a solution of the stereo MOT AP is in certain scenarios infeasible and (b) the simultaneous determination of two subsequent stereo image pairs requires the computation of three optical flow fields in addition to $\mathbf{F}_{i,l\to i,r}$ and $\mathbf{F}_{i,l\to i+1,l}$. Here, $\mathbf{F}_{i,l\to i,r}$ and $\mathbf{F}_{i,l\to i+1,l}$ denote the optical flow between image $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i,r}$ as well as $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i+1,l}$. Instead, we apply the following greedy approximation of the stereo MOT AP by solving two different two-dimensional assignment problems. This allows us to determine object correspondences in $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i,r}$ as well as $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i+1,l}$ in $\mathcal{O}(n^3)$. We associate object instances in the left images $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i+1,l}$ using the object affinity matrix presented in Bullinger et al. (2017) as input for the Kuhn-Munkres algorithm to compute the tracker state $\mathcal{T}_{i+1,l}$. In this case the affinity matrix is defined according to (3.7). Here, $o_{p,d}$ denotes the overlap of the prediction with index p in $P_{i,l\to i+1,l}$ and the detection with index d in $\mathcal{D}_{i+1,l}$.

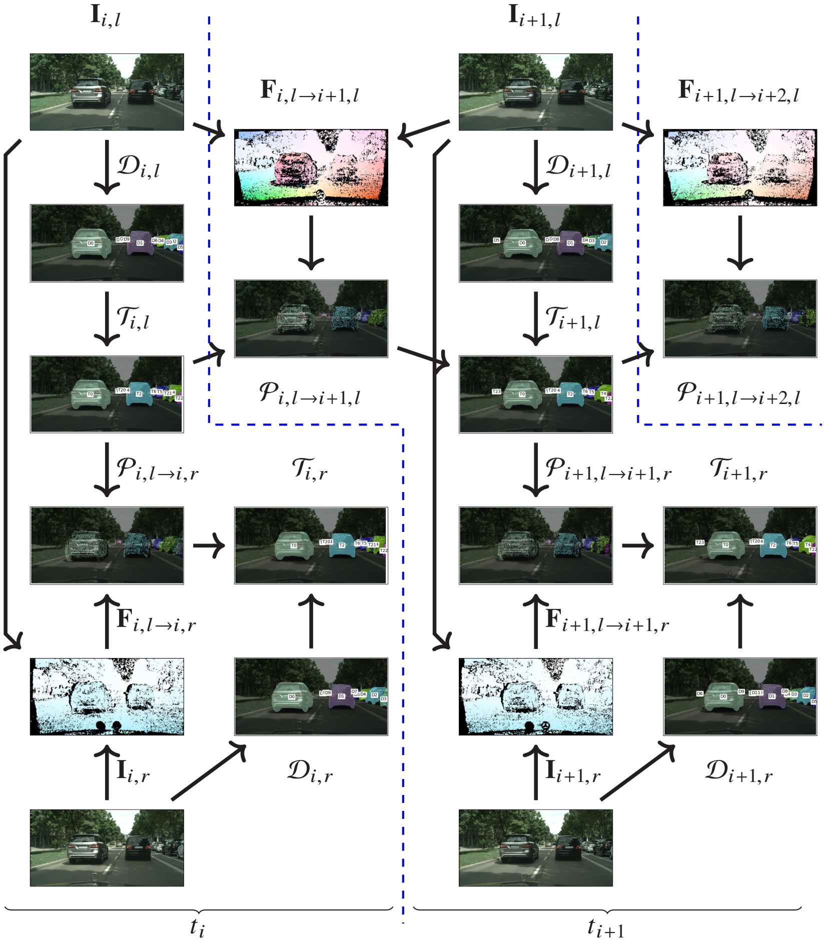


Figure 3.10: Scheme of the stereo object tracking algorithm. The variables have the following meaning. $\mathbf{I}$: image, $\mathbf{F}$: optical flow, $\mathcal{D}$: detection, $\mathcal{P}$: prediction, $\mathcal{T}$: tracker state, i: image index, l: left, r: right. Arrows show the relation of computation steps. A computation step depends on the results connected with incoming arrows. The tracked objects $\mathcal{T}_{i,l}$ at time i in the left image are predicted to the next image using the optical flow $\mathbf{F}_{i,l\to i+1,l}$ of image $\mathbf{I}_{i,l}$ and $\mathbf{I}_{i+1,l}$. The predictions $\mathcal{P}_{i,l\to i+1,l}$ are associated with the detections $\mathcal{D}_{i+1,l}$ to update the left tracker state. Simultaneously, the tracked object objects $\mathcal{T}_{i,l}$ are predicted to the corresponding right image of the same time step using the optical flow $\mathbf{F}_{i,l\to i,r}$.

Figure 3.10 (cont.): The predictions $\mathcal{P}_{i,l \to i,r}$ are associated with the detections $\mathcal{D}_{i,r}$ to compute the tracker state $\mathcal{T}_{i,r}$ of the objects in right image at time step i. Corresponding objects in the left and the right tracker state $\mathcal{T}_{i,l}$ and $\mathcal{T}_{i,r}$ share the same identifier, which is not necessarily the case for detections in $\mathcal{D}_{i,l}$ and $\mathcal{D}_{i,r}$. The used optical flow color coding is defined in Baker et al. (2011).

Let n_p and n_d denote the number of predictions in $\mathcal{P}_{i,l \to i+1,l}$ and the number of detections in $\mathcal{D}_{i+1,l}$.

$$\mathbf{A}_i = \begin{bmatrix} o_{1,1} & \cdots & o_{1,d} & \cdots & o_{1,n_d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ o_{p,1} & \cdots & o_{p,d} & \cdots & o_{p,n_d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ o_{n_p,1} & \cdots & o_{n_p,d} & \cdots & o_{n_p,n_d} \end{bmatrix} \tag{3.7}$$

Fig. 3.10 shows examples of $\mathcal{P}_{i,l \to i+1,l}$ and $\mathcal{D}_{i+1,l}$. The tracker state $\mathcal{T}_{i+1,l}$ contains only tracks of object instances in images corresponding to the left camera. We use the optical flow between left and right images $\mathbf{F}_{i+1,l \to i+1,r}$ to associate the tracker state of left images $\mathcal{T}_{i+1,l}$ with objects visible in the corresponding right image. The association between predictions $\mathcal{P}_{i+1,l \to i+1,r}$ and detections $\mathcal{D}_{i+1,r}$ in the right images are also computed using an affinity matrix and the Kuhn-Munkres algorithm. In this case $o_{p,d}$ denotes the overlap of prediction p in $\mathcal{P}_{i+1,l \to i+1,r}$ and detection d in $\mathcal{D}_{i+1,r}$. n_p denotes the number of predictions in $\mathcal{P}_{i+1,l \to i+1,r}$ and n_d denotes the number of detections in $\mathcal{D}_{i+1,r}$. The overlap $o_{p,d}$ is an affinity measure that reflects locality and visual similarity.

3.5 Instance-Aware Multibody Structure from Motion for Dynamic Object Reconstruction

As shown in Fig. 3.2 we track objects on pixel level and apply SfM simultaneously to object and background images. Without loss of generality, we describe the reconstruction of a single object in a static background. We denote the corresponding SfM results with $sfm^{(o)}$ and $sfm^{(b)}$. Let $\mathbf{o}_j^{(o)} \in \mathcal{P}^{(o)}$ and $\mathbf{b}_k^{(b)} \in \mathcal{P}^{(b)}$ denote the 3D points contained in $sfm^{(o)}$ or $sfm^{(b)}$, respectively. The superscripts o and b in $\mathbf{o}_j^{(o)} \in \mathbb{R}^3$ and $\mathbf{b}_k^{(b)} \in \mathbb{R}^3$ describe the corresponding coordinate frame system (CFS). The variables j and k are the indices of points in the object or the background point cloud, respectively. We denote the reconstructed intrinsic and extrinsic parameters of each registered input image as virtual camera. Each virtual camera in $sfm^{(o)}$ and $sfm^{(b)}$ corresponds to a certain frame from which object and background images are extracted. In the following, we consider only virtual cameras in $sfm^{(o)}$ with a corresponding virtual camera in $sfm^{(b)}$. Because of missing image registrations this may not be the case for all virtual cameras.

We determine the object pose relative to the reconstructed environment by combining information of corresponding virtual cameras. For any virtual camera pair of an image with index i, the object SfM result $sfm^{(o)}$ contains information of object point positions $\mathbf{o}_j^{(o)}$ relative to virtual cameras with camera centers $\mathbf{c}_i^{(o)} \in \mathbb{R}^3$ and rotations $\mathbf{R}_i^{(o)} \in SO(3)$.

Two coordinate frames are related via a rotation and a translation. To transform object points $\mathbf{o}_j^{(o)}$ in camera coordinates $\mathbf{o}_j^{(i)}$ of camera i we use (3.8).

$$\mathbf{o}_j^{(i)} = \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) = \mathbf{R}_i^{(o)} \cdot \mathbf{o}_j^{(o)} - \mathbf{R}_i^{(o)} \cdot \mathbf{c}_i^{(o)} \tag{3.8}$$

Rewriting (3.8) with homogeneous coordinates allows us to express this operation with a single transformation matrix $\mathbf{T}_i^{(o2c)} \in SE(3)$.

$$\begin{bmatrix} \mathbf{o}_j^{(i)} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}_i^{(o)} & -\mathbf{R}_i^{(o)} \cdot c_i^{(o)} \\ 0 & 1 \end{bmatrix}}_{:=\mathbf{T}_i^{(o2c)}} \begin{bmatrix} \mathbf{o}_j^{(o)} \\ 1 \end{bmatrix} \tag{3.9}$$

In contrast to (3.8), $\mathbf{T}_i^{(o2c)}$ allows us to transform position vectors and arbitrary transformation matrices alike. Inverting $\mathbf{T}_i^{(o2c)}$ results in $\mathbf{T}_i^{(c2o)}$ (3.10), which transforms the CFS of the camera with index i in $sfm^{(o)}$ to the CFS of $sfm^{(o)}$.

$$\mathbf{T}_i^{(c2o)} = \begin{bmatrix} \mathbf{R}_i^{(o)\top} & \mathbf{c}_i^{(o)} \\ 0 & 1 \end{bmatrix} \tag{3.10}$$

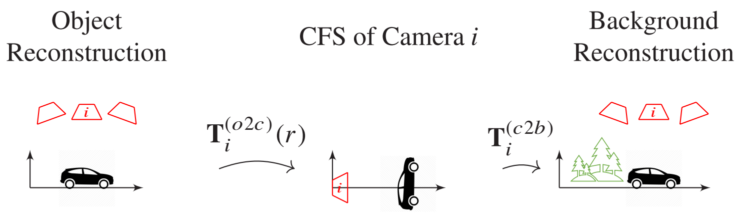


Figure 3.11: Relation between object reconstruction, camera coordinate frame system of camera i and background reconstruction.

Similarly to the object reconstruction, the background SfM result $sfm^{(b)}$ contains the camera centers $\mathbf{c}_i^{(b)} \in \mathbb{R}^3$ and the corresponding rotations $\mathbf{R}_i^{(b)} \in SO(3)$, which provide pose information of the cameras with respect to the reconstructed background. The counterpart of (3.10) is shown in (3.11).

$$\mathbf{T}_i^{(c2b)} = \begin{bmatrix} \mathbf{R}_i^{(b)\top} & \mathbf{c}_i^{(b)} \\ 0 & 1 \end{bmatrix} \tag{3.11}$$

Combining $\mathbf{T}_i^{(o2c)}$ and $\mathbf{T}_i^{(c2b)}$ allows to compute a transformation $\mathbf{T}_i^{(o2b)}$ from object to world coordinates. Fig. 3.11 visualizes the relation between object and background reconstruction. Note that the camera CFS of virtual cameras in $sfm^{(o)}$ and $sfm^{(b)}$ are equivalent up to scale, since in general the scale ratio of object and background reconstruction does not match due to the scale ambiguity of SfM reconstructions (Hartley and Zisserman, 2004). We tackle this problem by treating the scale of the background as reference scale and by introducing a scale ratio factor r to adjust the scale of the object CFS. This approach leads to a matrix $\mathbf{T}_i^{(o2c)}(r)$ that performs a transformation according to $\mathbf{T}_i^{(o2c)}$ as well as the scale adjustment using the scale ratio r. $\mathbf{T}_i^{(o2c)}(r)$ is given in (3.12).

$$\mathbf{T}_i^{(o2c)}(r) = \begin{bmatrix} r\mathbf{R}_i^{(o)} & -r\mathbf{R}_i^{(o)} \cdot \mathbf{c}_i^{(o)} \\ 0 & 1 \end{bmatrix} \tag{3.12}$$

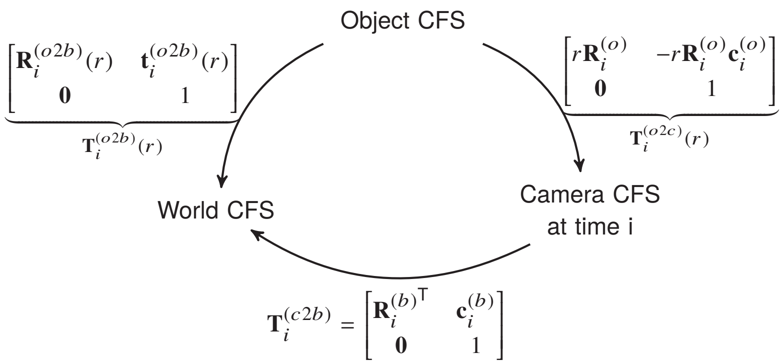


Figure 3.12: Transformations between object, camera and background coordinate frame systems. We use the world scale as reference. Here, $\mathbf{R}_i^{(o2b)}(r)$ and $\mathbf{t}_i^{(o2b)}(r)$ denote the correctly scaled matrix components.

The transformation between object and background CFS is defined according to (3.13).

$$\begin{aligned} \mathbf{T}_i^{(o2b)}(r) = \mathbf{T}_i^{(c2b)} \cdot \mathbf{T}_i^{(o2c)}(r) &= \begin{bmatrix} \mathbf{R}_i^{(b)\top} & \mathbf{c}_i^{(b)} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r\mathbf{R}_i^{(o)} & -r\mathbf{R}_i^{(o)} \cdot \mathbf{c}_i^{(o)} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} r\mathbf{R}_i^{(b)\top}\mathbf{R}_i^{(o)} & -r\mathbf{R}_i^{(b)\top}\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} + \mathbf{c}_i^{(b)} \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.13}$$

The relation between the different coordinate frame systems (camera, object and world) is depicted in Fig. 3.12. We will use these pose constraints in Chapter 6 to determine three-dimensional trajectories of dynamic objects. Using stereo image data to reconstruct a scene allows to infer r from the baseline of the virtual stereo cameras in $sfm^{(o)}$ and $sfm^{(b)}$.

3.6 Implementation Details

The proposed Multibody Structure from Motion approach builds upon different methods including instance-aware semantic segmentation, optical flow and standard Structure from Motion. Our current implementation uses He et al. (2017) to perform instance-aware semantic segmentation. This ConvNet computes stable object detections requiring a reasonable amount of computation time. He et al. (2017) defined the state-of-the-art in instance-aware semantic segmentation at the time of its publication. The network uses Xie et al. (2017) and Lin et al. (2017) as backbone and has been trained on the Microsoft COCO dataset (Lin et al., 2014).

We use Hu et al. (2016) to determine optical flow between pairs of images. The architecture of Hu et al. (2016) is designed to compute optical flow for large displacements. This property is important to leverage optical flow correspondences for object tracking. With regard to large displacements, Hu et al. (2016) outperform state-of-the-art method like Ilg et al. (2017) or Sun et al. (2018). At the time of publication, Hu et al. (2016) reported state-of-the-art results on the KITTI (Geiger et al., 2013) and the MPI-Sintel (Butler et al., 2012) datasets.

We observe that the reconstruction quality of current state-of-the-art Structure from Motion approaches depends strongly on viewing angles and object sizes. For example, the algorithm in Schönberger and Frahm (2016) computes reasonable reconstructions of small objects. Other methods reconstruct the same objects only partially. On the downside, the algorithm in Schönberger and Frahm (2016) computes frequently degenerated environment reconstructions for straight camera motions. Moulon et al. (2012) reconstruct such cases successfully. Since current state-of-the-art SfM pipelines depend on many different parameters, it is difficult to identify a parameter configuration that produces reliably results for objects as well as environment structures. We use Schönberger and Frahm (2016) for object and Moulon et al. (2012) for environment reconstruction to achieve robust results.

3.7 Online Multiple Object Tracking Evaluation

In this section, we evaluate the proposed MOT tracking algorithm on monocular image sequences.

3.7.1 Multiple Object Tracking Measures

There are different measures to evaluate the quality of multiple object tracking algorithms. Simple tracker properties are defined by *true positives (TP)*, *false positives (FP)*, *i.e.*, false alarms of the tracker, *true negatives (TN)*, *false negatives (FN)*, *i.e.*, missing detections and *id switches (IDsw)*, *i.e.*, tracker ID mismatch errors.
Bernardin and Stiefelhagen (2008) propose the *Multiple Object Tracking Accuracy (MOTA)* and *Multiple Object Tracking Precision (MOTP)* measures, which allow to evaluate MOT trackers in various domains and for different modalities.
The Multiple Object Tracking Accuracy (Bernardin and Stiefelhagen, 2008) is defined according to (3.14) and reflects the total overall error of the tracking algorithm.

$$MOTA = 1 - \frac{\sum_i (fn_i + fp_i + IDsw_i)}{\sum_i g_i} \tag{3.14}$$

Let fn_i, fp_i and $IDsw_i$ denote the false negative, the false positives and the ID switches in frame $\mathbf{I}_i$. Further, let g_i represent the number of ground truth objects in the corresponding image.
In contrast, the Multiple Object Tracking Precision (Bernardin and Stiefelhagen, 2008) is defined according to (3.15) and represents the average (location) dissimilarity between the true positives and their corresponding ground truth position.

$$MOTP = \frac{\sum_{i,u} d_{i,u}}{\sum_i c_i} \tag{3.15}$$

Here, $d_{i,u}$ represents the distance of the object hypothesis with index u to the true object position in image $\mathbf{I}_i$ and c_i the number of object-hypothesis-correspondences in frame $\mathbf{I}_i$.

A common measure for $d_{i,u}$ is given by the *Intersection over Union (IoU)* (3.16) distance.

$$IoU = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \tag{3.16}$$

Here, $\mathcal{A}$ and $\mathcal{B}$ denote two sets representing the object hypothesis as well as the object ground truth. The Generalized Intersection over Union (GIoU) (Rezatofighi et al., 2019) is an extension of the standard IoU, designed to handle cases with non-overlapping bounding boxes. Equation (3.17) shows the corresponding definition, where C represents the smallest convex hull that encloses both $\mathcal{A}$ and $\mathcal{B}$.

$$GIoU = IoU - \frac{|C \setminus (\mathcal{A} \cap \mathcal{B})|}{|\mathcal{A} \cup \mathcal{B}|} \tag{3.17}$$

In addition to MOTA and MOTP, the MOT challenge (Leal-Taixé et al., 2015) introduces the following measures: mostly tracked (MT), partially tracked (PT) and mostly lost (ML). These values reflect how much of the trajectories are recovered by the tracking algorithm.
Precision and *Recall* reflect specific properties of a MOT algorithm. The corresponding definitions are given in (3.18) and (3.19).

$$Precision = \frac{TP}{TP + FP} \tag{3.18}$$

Recall is also referred to as the true positive rate.

$$Recall = \frac{TP}{TP + FN} \tag{3.19}$$

3.7.2 Multiple Object Tracking Challenge

We evaluate the proposed monocular Multiple Object Tracking approach on a popular MOT dataset (Leal-Taixé et al., 2015). We use instance-aware semantic segmentations computed by Dai et al. (2016) and optical flow / matching algorithms presented in Farnebäck (2003), Revaud et al. (2016) and Hu et al. (2016). Let *<DetectionMethod>+<TrackingMethod>* denote a specific MOT algorithm that uses the *<DetectionMethod>* for object detection and the *<TrackingMethod>* to track object detections in a given image sequence.

We compare the proposed MOT approach with *FasterRNN+SORT*. The open source online MOT algorithm *SORT* (Bewley et al., 2016) shows competitive results using *FasterRNN* (Ren et al., 2015) detections. SORT follows the Tracking-by-Detection pipeline, *i.e.*, it uses Bounding Box detections, a Kalman filter for motion prediction and the Hungarian method for object association.
The performance of Tracking-By-Detection approaches strongly depends on the quality of corresponding detections. Applying SORT on detections derived from instance-aware semantic segmentations computed with *Multi-task Network Cascades (MNC)* (Dai et al., 2016) allows us to compare the MOT algorithms without the influence of different detector performances. Concretely, we consider *MNC+SORT* in our evaluation, which uses Bounding Box detections extracted from MNC instance segmentations.
To demonstrate the effectiveness of the proposed MOT algorithm we leverage different approaches to compute correspondences in subsequent images: *Coarse-To-Fine PatchMatch (CPM)* (Hu et al., 2016), *DeepMatching (DeepMatch)* (Revaud et al., 2016) and *Polynomial Expansion (PolyExp)* (Farnebäck, 2003). This results in the following instances of the proposed MOT pipeline: *MNC+CPM*, *MNC+DeepMatch* and *MNC+PolyExp*.
We also analyze the effect of varying the maximum number of allowed missing detections *md* - *q.v.* Section 3.4.4. The parameter *md* defines how the MOT algorithm handles tracklets without corresponding segmentation instances in the current frame. If $md > 0$, the MOT algorithms performs a dense prediction of non-matching tracklets using the corresponding optical flow information. This allows to compensate missing detections of the instance-aware segmentation method. *md* defines the maximum number of subsequent dense prediction steps. If $md = 0$, non-matching tracklets are immediately removed from the tracker state.
MNC+CPM, *MNC+DeepMatch* and *MNC+PolyExp* achieve similar results on the MOT 2015 training set. A reason for this is the slow motion of camera and pedestrians in most MOT 2015 sequences. In such cases, the quality of object associations is mainly dependent on the segmentation quality. The results of *MNC+CPM* for the test set is shown in Table 3.1 and Table 3.2.
The biggest difference of the evaluated algorithms in the train dataset is observed in the KITTI-13 sequence, which is the only video captured from a driving platform. In this case, the positions of the objects in image coordinates are strongly affected by the motion of the vehicle, *i.e.*, object positions show remarkable shifts between subsequent images. The corresponding re-

Method	md	**MOTA**	MOTP	MT	ML	FP	FN
FasterRNN+SORT	-	**33.4**	72.1	11.7%	30.9%	7,318	32,615
MNC+SORT	-	**27.5**	70.5	7.5%	50.9%	2,972	40,924
MNC+CPM (ours)	0	**30.6**	71.3	10.5%	34.0%	4,863	35,325
MNC+CPM (ours)	1	**32.1**	70.9	13.2%	30.1%	6,551	33,473

Table 3.1: MOT 2D 2015 benchmark test set evaluation - part I. The variable md represents the number of missing detections. The evaluation measures are defined as follows: MOTA: Multiple Object Tracking Accuracy, MOTP: Multiple Object Tracking Precision, MT: mostly tracked, ML: mostly lost, FP: False Positive and FN: False Negative. A target is mostly tracked if it is successfully tracked for at least 80% of the corresponding frames. If a track is only recovered for less than 20%, it is considered to be mostly lost (ML).

Method	md	FAF	IDsw	Frag
FasterRNN+SORT	-	1.3	1,001	1,764
MNC+SORT	-	0.5	661	1,292
MNC+CPM (ours)	0	0.8	2,459	2,953
MNC+CPM (ours)	1	1.1	1,687	2,471

Table 3.2: MOT 2D 2015 benchmark test set evaluation - part II. The evaluation measures are defined as follows: FAF: false alarms per frame, IDsw: ID switches, Frag: number fragmentations.

sults are shown in Table 3.3 and Table 3.4. In terms of MOTA, *MNC+CPM* (with $md = 1$) outperforms *MNC+DeepMatch* as well as *MNC+PolyExp*. This shows the importance of the quality, *e.g.*, density and reliability, of the selected optical flow / matching algorithm. With the default parameter configuration *MNC+DeepMatch* computes sparse results and *MNC+PolyExp* can not handle big object shifts as shown in Fig. 3.13.
In the KITTI-13 sequence *MNC+CPM* and *MNC+DeepMatch* show a higher MOTA score than *MNC+SORT*. This demonstrates the strength of optical flow based approaches in videos with high relative motions of objects. It also shows the difficulty to describe a superposition of motions with a single motion model. We observe, that the number of id switches (IDs) of *MNC+SORT* is significantly lower than the ones of the evaluated optical flow based approaches. This confirms our impression that the used semantic instance seg-

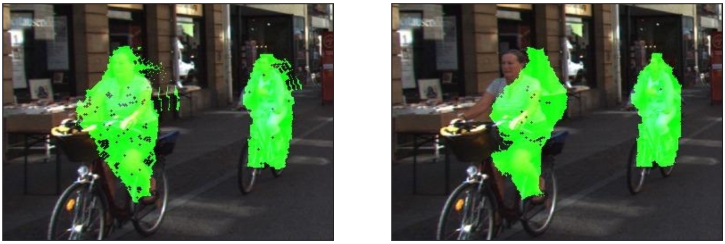
Method	*md*	**MOTA**	MOTP	MOTAL	GT	MT	PT	ML
MNC+SORT	-	**12.9**	65.2	13.2	42	0	14	28
MNC+CPM (ours)	0	**18.6**	67.2	21.7	42	0	32	10
MNC+CPM (ours)	1	**19.2**	66.7	20.8	42	4	32	6
MNC+DeepMatch (ours)	0	**18.6**	67.2	21.7	42	0	32	10
MNC+DeepMatch (ours)	1	**16.9**	66.8	18.6	42	3	31	8
MNC+PolyExp (ours)	0	**16.8**	67.3	21.7	42	0	32	10
MNC+PolyExp (ours)	1	**11.7**	66.8	15.4	42	3	31	8

Table 3.3: MOT 2015 benchmark KITTI-13 evaluation - part I. The variable md represents the number of missing detections. The evaluation measures are defined as follows: MOTA: Multiple Object Tracking Accuracy, MOTP: Multiple Object Tracking Precision, MOTAL: Multiple Object Tracking Accuracy with logarithmic ID switches, GT: number ground truth tracks, MT: mostly tracked tracks, PT: partially tracked tracks, ML: mostly lost tracks.

Method	*md*	Rcll	Prcn	FAR	FP	FN	IDsw	FM
MNC+SORT	-	18.8	77.3	0.12	42	619	3	6
MNC+CPM (ours)	0	38.7	69.7	0.38	128	467	25	38
MNC+CPM (ours)	1	43.8	65.7	0.51	174	428	14	30
MNC+DeepMatch (ours)	0	38.7	69.7	0.38	128	467	25	38
MNC+DeepMatch (ours)	1	43.8	63.8	0.55	188	431	14	29
MNC+PolyExp (ours)	0	38.7	69.7	0.38	128	467	39	40
MNC+PolyExp (ours)	1	42.7	61.2	0.61	206	437	30	33

Table 3.4: MOT 2015 benchmark KITTI-13 evaluation - part II. The evaluation measures are defined as follows: Rcll: Recall, Prcn: Precision, FAR: false alarm rate (*i.e.*, false alarm per frame), FP: false positives, FN: false negatives, IDsw: number ID switches, FM: number of track fragmentations

mentation (Dai et al., 2016) is unstable. However, we are able to decrease the number of id switches by using dense predictions as instance segmentations in the subsequent frame (*e.g.*, $md = 1$).



(a) Prediction using CPM (Hu et al., 2016).

(b) Prediction using PolyExp (Farnebäck, 2003).

Figure 3.13: Importance of the quality of the optical flow algorithm. The prediction using PolyExp is not correctly shifted.

3.8 Discussion

We proposed a novel Multibody Structure from Motion approach, which leverages instance-aware semantic segmentations to identify two-dimensional object shapes. We present an online Multiple Object Tracking algorithm that tracks objects on pixel level allowing us to determine object specific features correspondences throughout monocular and stereo image sequences. The usage of semantic information enables our algorithm to compute three-dimensional reconstructions of dynamic environments in which many existing motion segmentation or epipolar geometry based methods fail - such as scenes with partly stationary or parallel moving objects. The presented MOT approach leverages inherent properties of sequential image data and proposes an affinity measure reflecting locality and visual similarity. As we will see in Chapter 5 and Chapter 6, the presented Multibody Structure from Motion approach is suitable to reconstruct three-dimensional shapes and motion trajectories of moving objects. We demonstrated the effectiveness of the MOT algorithm using the dataset of the MOT challenge. Extending the proposed MOT algorithm with a Kalman Filter or a Particle Filter could improve the robustness of the method in situations with fully occluded objects. Chapter 6 presents quantitative and qualitative three-dimensional reconstruction results of the full MSfM pipeline.

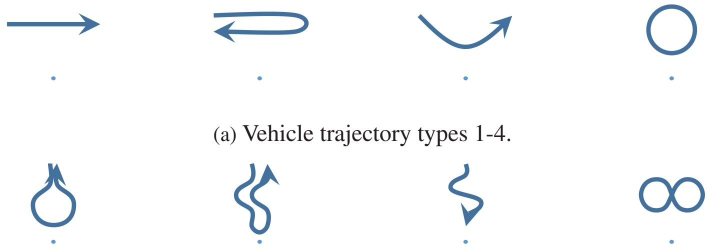
4 Datasets for Imaged-Based Moving Object Reconstruction

This chapter describes two datasets, which allow to evaluate moving object reconstruction algorithms. We use the corresponding ground truth to quantitatively evaluate the methods proposed in Chapter 5 and Chapter 6. The datasets have been first presented in Bullinger et al. (2016) and Bullinger et al. (2018b). Other datasets in the vehicle domain such as CityScapes (Cordts et al., 2016) or KITTI (Geiger et al., 2013) do not provide the required ground truth.
The first dataset[1] presented in Section 4.1 comprises real-world image sequences of a moving vehicle and a corresponding vehicle laser scan suitable for evaluation of three-dimensional object shape reconstructions. The second dataset[2] described in Section 4.2 contains synthetic sequences of different vehicles in an urban environment. The ground truth includes vehicle shapes as well as vehicle and camera poses for each frame. This dataset allows to quantitatively evaluate shape and trajectory reconstructions of moving objects. Both datasets and corresponding evaluation scripts are publicly available to foster future analysis of moving object reconstruction.

4.1 Object Shape Dataset

The dataset consists of 25 video sequences capturing a car moving on eight different trajectories. Fig. 4.1 depicts the shapes of the trajectory types. The lines denote the motion trajectory and the dots represent the position of the camera. As illustrated the video sequences cover a high variety of object-specific viewing angles. Fig. 4.2 shows an example of a corresponding image

[1] Project page: `http://s.fhg.de/boundarygeneration`
[2] Project page: `http://s.fhg.de/trajectory`



(a) Vehicle trajectory types 1-4.

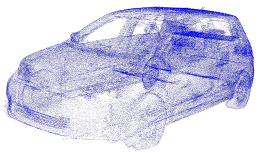
(b) Vehicle trajectory types 5-8.

Figure 4.1: Vehicle trajectory types of the dataset contained in the shape reconstruction dataset. The blue dot denotes the position of the camera capturing the vehicle.

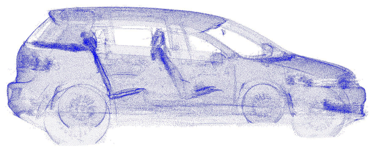
Figure 4.2: Example video sequence for trajectory type 6.

sequence. A laser scan (*q.v.* Fig. 4.3) of the vehicle serves as ground truth. We acquire the laser scan using a *Zoller+Fröhlich* scanner, which estimates the distance to the reflecting object on the phase shift between received and emitted signal. We created the vehicle laser scan indoors to reduce measurement noise. The scanning head was operated on a rigid tripod which results in ranging accuracies of a few millimeters. The laser scans from different views are automatically registered using a set of salient and distinct markers. Noise artifacts in the measurement and points corresponding to the environment are manually removed.

The usage of the *Iterative Closest Point* (Chen and Medioni, 1991) allows to register reconstructed object point clouds to the laser scan ground truth and to perform corresponding scale adjustments. We use the following steps to find reasonable correspondences between laser scan and reconstructed object points. First, we compute for each object point the nearest neighbor in the laser scan. We determine the distance between each reconstruction-laser-scan-point-pair. If multiple reconstructed points share the same nearest neighbor we keep only the reconstructed point with the smallest distance. For evaluation



(a) Laser scan (front view).



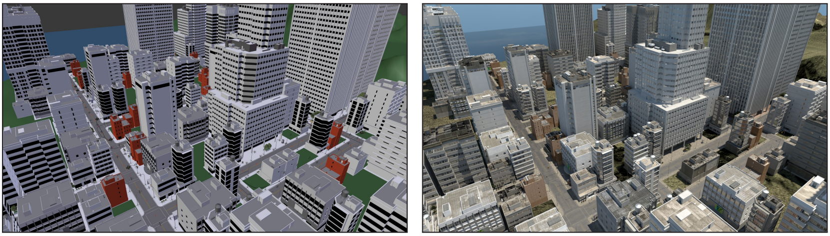
(b) Laser scan (side view).

Figure 4.3: Laser scan of the object shape reconstruction dataset.

we define the average distance and the standard deviation of the remaining reconstruction-laser-scan-point-pairs as evaluation measure.

4.2 Virtual Object Trajectory Dataset

To quantitatively evaluate the quality of reconstructed object motion trajectories we require accurate object and environment models as well as object and camera poses for each time step. The simultaneous capturing of corresponding ground truth data with sufficient quality is difficult to achieve. For example, one could capture the environment geometry with *Lidar* sensors and the camera as well as object pose with an additional system. However, the registration and synchronization of all these different modalities is a complex and cumbersome process. The result will contain noise and other artifacts like drift. This is probably the main reason why publicly available real-world datasets such as Geiger et al. (2013), Cordts et al. (2016) and Huang et al. (2018) do not provide the corresponding ground truth information.
To tackle these issues we exploit virtual models. Previously published virtually generated and virtually augmented datasets (Gaidon et al., 2016; Richter et al., 2016; Ros et al., 2016; Tsirikoglou et al., 2017) provide data for different application domains and do not include three-dimensional ground truth information. Other planned datasets such as Ruf (2018) are not yet publicly available. We build a virtual world specifically for SfM applications including an urban environment, animated vehicles as well as predefined vehicle and camera motion trajectories. This allows us to compute spatial and temporal



(a) Environment model without textures in Blender.

(b) Rendered environment model with Cycles.

Figure 4.4: Virtual Environment in Blender.

(a) Environment model with repetitive textures. A few examples are emphasized with green and blue.

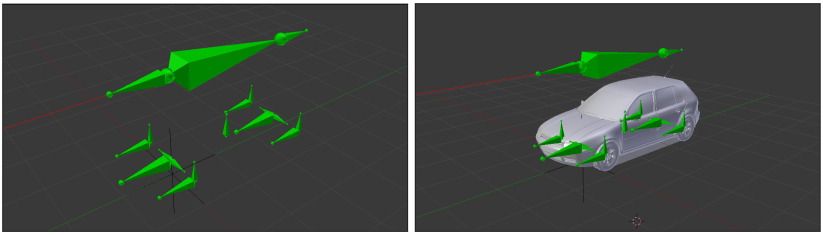
(b) Environment model with procedurally textures.

Figure 4.5: Comparison of repetitive and procedurally generated textures.

error free ground truth data. Our dataset is suitable for evaluating SfM algorithms, since we exploit procedural generation of textures to avoid artificial repetitions.
While creating this virtual world two novel platforms *Airsim* (Shah et al., 2017) and *Carla* (Dosovitskiy et al., 2017) have been made publicly available, which could potentially provide the same type of ground truth.

4.2.1 Virtual World

We use *Blender* (Blender Foundation, 2019) to create a virtual world (*q.v.* Fig. 4.4) consisting of an urban environment surrounded by a countryside and



(a) Vehicle Rig in stationary pose with and without attached vehicle meshes.

(b) Vehicle rig adjusted to the motion trajectory. The rig controls the correct placement (rear left wheel) and the correct steering (front left wheel).

Figure 4.6: Skeletal animation (rigging) of vehicle models. The rig consists of a set of bones and is shown in green. The pose of each bone is defined by the position of the corresponding head and tail. We define a set of bone-specific constraints, which determine the motion of each bone.

a set of camera-object-trajectory pairs used to render sequences with moving cameras and driving vehicles. The camera and vehicle trajectories are defined as curves in 3D space. The virtual world includes different assets like trees, traffic lights, streetlights, phone booths, bus stops and benches.
Texture mapping is a common method in computer graphics to define surface textures of 3D models and allows to reduce the number of polygons and lighting computations needed to render realistic scenes. Texture repetition is a common technique to handle cases where the model size exceeds the spatial extent of the texture. This is typically the case for large objects like streets. Feature matching applied to 3D models with repeated textures results in many incorrect correspondences between distinct surfaces points due to their visual similarity. We exploit procedural generation to compute textures of large surfaces, like streets and sidewalks, to avoid degenerated Structure from Motion results caused by artificial texture repetitions (*q.v.* Fig. 4.5).

Figure 4.7: Frames from sequences contained in the presented virtual vehicle trajectory dataset.

We collected a set of publicly available vehicle assets to populate the scenes. We used emphSkeletal Animation, also referred to as *rigging*, to animate the vehicle motion, which allows us to define the vehicle trajectories as simple curves in 3D space. A rig consists of a set of *bones*, which follow specific user defined constraints such as relative translations and rotations, deformations or tail constraints. In our case the rig controls wheel rotation and steering w.r.t. the motion trajectory as well as consistent vehicle placement on uneven ground surfaces. The animation of wheels is important to avoid unrealistic wheel point triangulations. Overall, we defined more than 300 rig constraints to animate all vehicles. Fig. 4.6 shows an example of a rigged vehicle model. To control the camera pose we use Blenders built-in *Follow-Path* and *Track-To* constraints.
We adjusted the scale of vehicles and virtual environment using Blender's unit system. This allows us to set the virtual space in relation to the real world. The extent of the generated virtual world corresponds to one square kilometer. We combined environment mapping with raytracing to achieve a realistic scene illumination. With Blender's built-in tools, we defined a set of camera and object motion trajectories. This allows us to determine the exact 3D pose of cameras and vehicles for each time step.

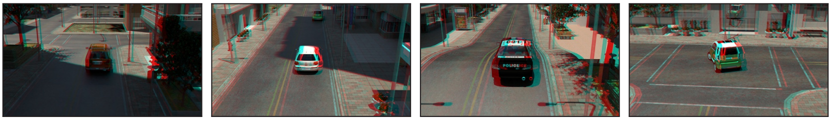


Figure 4.8: Anaglyph images representing stereo information of the sequences contained in the presented virtual vehicle trajectory dataset. Information of left and right images are highlighted with green and red, respectively.

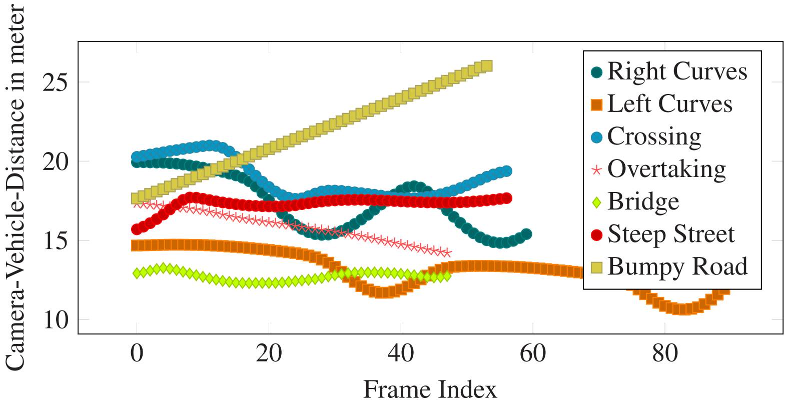


Figure 4.9: Distance between camera and vehicle per frame in meter for each trajectory type in the dataset.

4.2.2 Trajectory Dataset

We use the previously created virtual world to build a new vehicle trajectory dataset. The dataset consists of 35 sequences capturing five vehicles in different urban scenes.

For each sequence we rendered monocular (see Fig. 4.7) and binocular (see Fig. 4.8) image sequences with a resolution of 1920 px × 1080 px. We use a focal length of 35 mm, which corresponds to 2100 px and a stereo camera baseline of 0.3 m, which lies between the stereo baselines used in common real word datasets like CityScapes (Cordts et al., 2016) and KITTI (Geiger et al., 2013). The anaglyph images in Fig. 4.8 reflect the properties of the corresponding stereo camera. Fig. 4.9 shows the distance between the camera and the vehicle per sequence per frame for each trajectory type.

The virtual video sequences cover a high variety of vehicle and camera poses. The vehicle trajectories reflect common vehicle motions and include vehicle acceleration, different curve types and motion on changing slopes. The camera trajectory mimics the motion of a drone and captures the scene from a bird's-eye perspective. We use the path-tracing render engine Cycles (Blender Foundation, 2019) to achieve photo realistic rendering results. We observed that the removal of artificial path-tracing artifacts using denoising improves feature matching.

In addition to the rendered imagery, the dataset contains depth maps as well as vehicles, ground and background segmentations allowing to separate the reconstruction task from specific semantic segmentation and tracking approaches. This aims to simplify future trajectory reconstruction evaluations. The dataset includes 6D vehicle and (stereo) camera poses for each frame as well as ground truth meshes of corresponding vehicle models. The provided virtual ground truth is free of noise and shows no spatial registration or temporal synchronization inaccuracies. In addition to the virtual data, the dataset includes scripts for automatic registration and evaluation of trajectory reconstructions.

The combination of scene illumination leveraging environment mapping and path-tracing using a state-of-the art render engine like Cycles (Blender Foundation, 2019) results in naturally reflecting surfaces and realistic shadow computations. This makes the dataset challenging for visual reconstruction problems.

5 Shape Reconstruction of Dynamic Objects using Semantic Volumetric Constraints

This chapter is partly published in Bullinger et al. (2016) and tackles the problem of computing three-dimensional appearance models of moving objects. Dense reconstructions (*q.v.* Section 2.3.1) of dynamic objects obtained with keypoint based methods are affected by shadows, reflecting surfaces and illumination changes, which lead to point clouds with high outlier ratios and varying point densities. Meshes built on top of these point clouds show irregular surface properties. To mitigate the aforementioned effects, we present an algorithm that combines semantic segmentations and object specific camera poses to compute three-dimensional object boundaries consistent to image observations. The resulting point cloud consists of uniformly distributed points with consistent normal vectors. Given suitable camera poses we compute clean object representations superior to Structure from Motion and Multi-View Stereo based meshes. We determine appropriate object textures by projecting visual information of corresponding object images onto the computed 3D model.

This chapter is structured as follows. We define the problem statement in Section 5.1. Section 5.2 highlights relevant related work, which present Multi-View Stereo and model based approaches for three-dimensional shape reconstruction. Section 5.3 gives a short overview of the proposed pipeline. Our approach includes filtering of virtual cameras (Section 5.4), outlier removal of 3D object points using an objectness score (Section 5.5) and computation of the final object shape using the algorithm in Section 5.6. Our approach is motivated by limitations of previously presented algorithms. Section 5.7 presents qualitative and quantitative results using drone footage and sequences from the moving object reconstruction dataset described in Chapter 4.

5.1 Problem Statement

Given a set of images showing a moving object, we want to compute a three-dimensional model consistent to the object appearance including not only shape, but also color information. More formally, we want to minimize the visual error of the model projection w.r.t. the set of input images - see equation (5.1).

We represent our model with a set of vertex positions and corresponding color information $\mathbf{m} \in \mathcal{M}$. Let $\mathbf{m}_p$ and $\mathbf{m}_c$ denote the corresponding position and color vector. $\mathbf{K}_i$, $\mathbf{R}_i^{(o)}$ and $\mathbf{c}_i^{(o)}$ denote the calibration, the object specific camera rotation matrix as well as the center of camera i. Let $\mathcal{V}_i$ denote the set of visible model vertices in image i. $\mathcal{P}_i$ represents the set of pixels occupied by the object in image i.

$$
\begin{aligned}
&\underset{\mathcal{M}}{\operatorname{argmin}} && \sum_i \sum_{\mathbf{m} \in \mathcal{V}_i} \|\mathbf{I}_i(\mathbf{K}_i \mathbf{R}_i^{(o)}(\mathbf{m}_p - \mathbf{c}_i^{(o)})) - \mathbf{m}_c\| \\
&\text{such that} && \forall \mathbf{p} \in \mathcal{P}_i : \exists \mathbf{m} \in \mathcal{V}_i \text{ with } \mathbf{p} \simeq \mathbf{p}' = \mathbf{K}_i \mathbf{R}_i^{(o)}(\mathbf{m}_p - \mathbf{c}_i^{(o)})
\end{aligned}
\tag{5.1}
$$

$\mathbf{I}_i(\mathbf{p}')$ represents the image color of pixel position $\mathbf{p}$ corresponding to the homogeneous vector $\mathbf{p}'$.

Compared to (dense) point clouds, meshes require less memory to achieve a consistent object model appearance and represent surface normals naturally, which are important to render suitable object model images with reasonable computational effort. Thus, we use meshes to represent the three-dimensional appearance of objects. Fig. 5.1 shows a visualization of the problem statement.

5.2 Related Work

Many state-of-the-art image-based methods designed to compute accurate dense *scene* models consist of the following algorithm scheme. First, determination of sparse scene points and camera poses with Structure from Motion (Moulon et al., 2012; Schönberger and Frahm, 2016; Wu, 2011). Second, Multi-View Stereo (Fuhrmann et al., 2015; Furukawa and Ponce, 2010; Schönberger et al., 2016) to compute dense point clouds including normal vectors. Third, mesh-triangulation methods (Fuhrmann et al., 2015; Jancosek

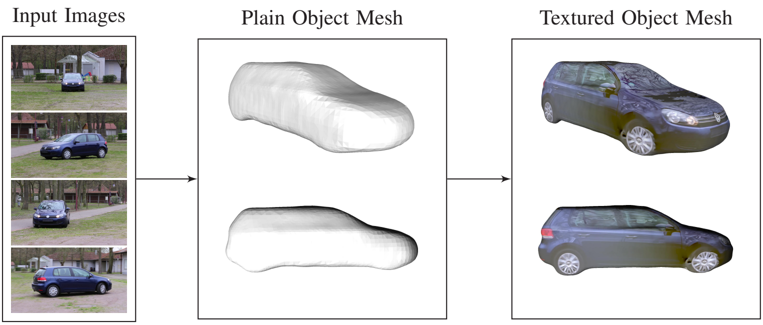


Figure 5.1: Visualization of the shape reconstruction problem statement.

and Pajdla, 2011; Kazhdan and Hoppe, 2013; Kazhdan et al., 2006) to derive watertight meshes. For more details see Section 2.3.
In contrast, previously published approaches focussing on the reconstruction of *moving objects* provide only sparse object representations (Chhaya et al., 2016; Feng et al., 2012; Kundu et al., 2011; Yuan and Medioni, 2006) or computed a dense object models leveraging specific object priors (Lebeda et al., 2014). One reason is that the shape reconstruction is more challenging because of stronger visual appearance changes.
Previous moving object reconstruction approaches using video data usually exploit color or motion detection. Feng et al. (2012) present a color-based segmentation to achieve 3D monocular tracking. Yuan and Medioni (2006) and Kundu et al. (2011) use motion segmentation to distinguish objects and background. Yuan and Medioni (2006) use this information to apply SfM to single objects, where as Kundu et al. (2011) perform Multibody Visual SLAM. In contrast to previous methods, Lebeda et al. (2014) use feature tracking in order to extract moving objects in unstructured video data. The object shape is visualized using watertight meshes.
In contrast to previously mentioned reconstruction methods, a new type of model based approaches have been proposed. Such methods like Kundu et al. (2018) rely on an initially provided database of three-dimensional object models to infer the object shape using a single image.
Our method builds on top of recent 3D reconstruction as well as semantic segmentation techniques. See Section 3.4.1 and Section 2.3 for the corresponding related work.

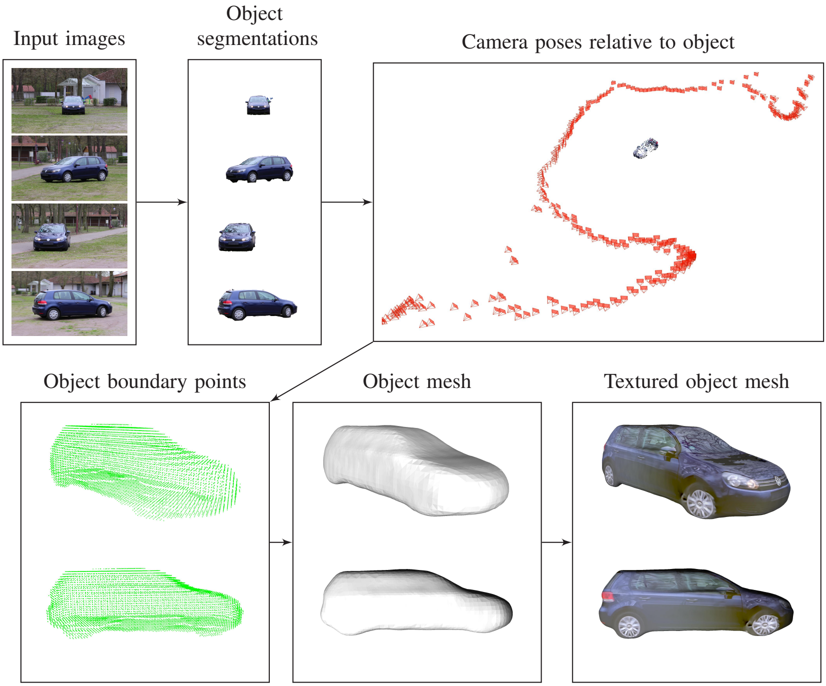


Figure 5.2: Overview of the shape reconstruction pipeline. In the first step, only object specific areas are considered to compute a sparse SfM reconstruction as explained in Chapter 3. The camera poses are shown in red.

5.3 Pipeline Overview

Fig. 5.2 shows an overview of the proposed shape reconstruction pipeline. As described in Section 3.5 we apply SfM to all object images, which allows us to leverage the full available information to compute camera pose w.r.t. the object model. We use the two-dimensional object boundaries in combination with the previously computed camera parameters to generate a set of three-dimensional object boundary points. We iteratively refine this point cloud by creating points in the space spanned by boundary points. The uniform distribution of the point cloud allows us to compute consistent normal vectors. By combining the boundary points and corresponding surface normals we compute a watertight object mesh. We determine an object texture by projecting visual information of corresponding object images onto the computed 3D model.

5.4 3D Object Reconstruction and Virtual Camera Filtering

In order to reconstruct objects in video sequences with varying (unknown) focal lengths, we follow the approach in Wu (2011) and initialize the focal length according to (5.2).

$$f = \frac{1.2 \cdot \max(w_I[px], h_I[px])}{w_S[mm]} \cdot f_S[mm] \tag{5.2}$$

Here, f and f_S represent the focal length in pixel and the focal length of the sensor in millimeter. w_I and h_I denote the width and the height of the input image in pixel. The factor 1.2 is empirically determined by Wu (2011) and corresponds to a medium field of view. w_S describes the width of the sensor in millimeter.

In contrast to the original frames, object images contain only camera pose information relative to the object. The scene is equivalent to one, where the virtual camera is moving and the object is stationary. The SfM computation produces a point cloud representing the object and parameters of corresponding virtual cameras. Let n be the number of virtual cameras. n may be smaller than the number of input frames due to failed image registrations.

SfM reconstructions contain sometimes outliers, *i.e.*, single isolated virtual cameras. However, valid virtual camera positions extracted from a single video sequence show usually similar distances to their respective closest virtual camera, since the camera as well as the object move and rotate gradually from frame to frame. In order to detect isolated cameras, we compute for each camera i the distance d_i to the respective nearest neighbor. We assume that there are less than 25 % isolated cameras. If there are more than 25 % isolated cameras it is likely that the reconstruction is degenerated and not useful at all. Thus, we consider the 75th percentile p_{75} of all nearest neighbor distances $\{d_i | i = 1, \ldots, n\}$ as a valid nearest neighbor distance. In video data the distance of valid virtual cameras to their nearest neighbor is limited due to the

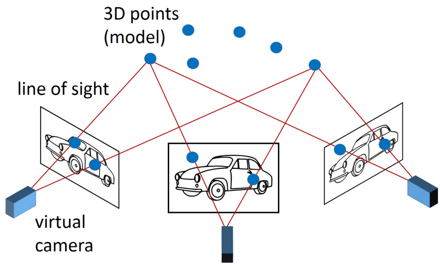


Figure 5.3: Projection of 3D points onto the image plane for each virtual camera in order to compute the objectness for each object point.

gradual movement of object and capturing device. Thus, we define a validity threshold t_{val} which describes the accepted exceeding of p_{75}. t_{val} should be several orders of magnitude greater than p_{75}. We compute the validity of a given camera by testing if $d_i < t_{val}$ holds. The removal of isolated cameras is important for the algorithms presented in Section 5.5 and Section 5.6.

5.5 Objectness and Outlier Removal

Misclassified pixels in the semantic segmentation potentially lead to noise in the resulting three-dimensional reconstruction. We determine outliers in the object point cloud by computing the objectness for each three-dimensional object point $\mathbf{o}_j^{(o)}$. We use the camera calibrations $\mathbf{K}_i$, rotations $\mathbf{R}_i^{(o)}$ and centers $\mathbf{c}_i^{(o)}$ estimated during the Structure from Motion process to project the object points $\mathbf{o}_j^{(o)}$ onto the focal plane of each virtual camera cam_i. Let $\mathbf{p}_{j,i}$ denote the homogeneous image projection of a point $\mathbf{o}_j^{(o)}$ given in object coordinates w.r.t. to camera cam_i. The projection is defined according to (5.3). The mapping of (5.3) is visualized in Fig. 5.3.

$$\mathbf{p}_{j,i} = \mathbf{K}_i \mathbf{R}_i^{(o)} (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) \tag{5.3}$$

By analyzing the projections $\mathbf{p}_{j,i}$ of $\mathbf{o}_j^{(o)}$ we determine a measure for the *objectness* of $\mathbf{o}_j^{(o)}$. For each visible projection $\mathbf{p}_{j,i}$ of $\mathbf{o}_j^{(o)}$ we use the corre-

	3D Object Point	3D Background Point
Visible	Projected on O. (TP)	Projected on B. (TN)
Occluded by O.	Projected on O. (TP)	Projected on O. (FP)
Occluded by B.	Projected on B. (FN)	Projected on B. (TN)

Projection possibilities of 3D points on object (O.) and background (B.) image areas. In the context of point classification, the different possibilities correspond to True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

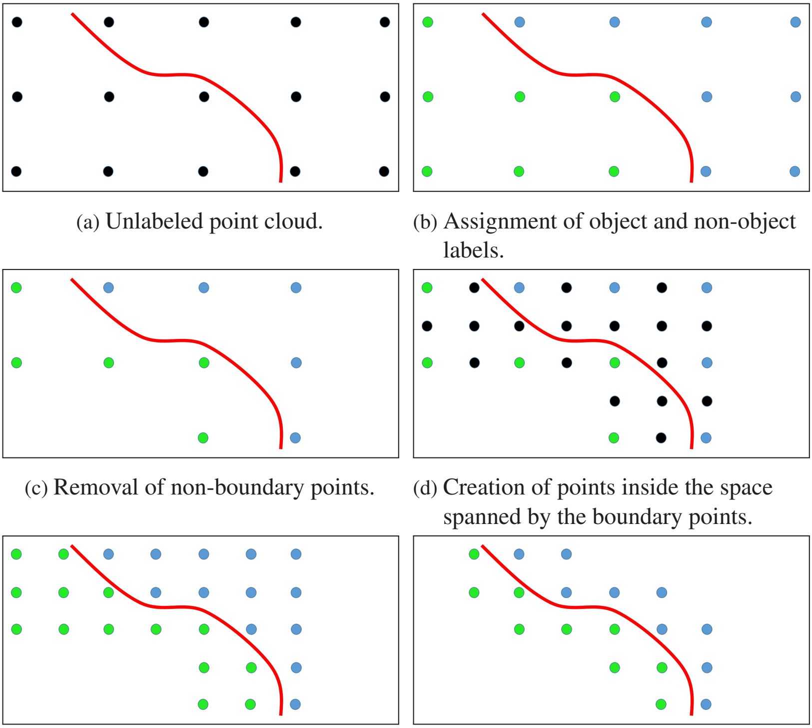
Table 5.1: Projection possibilities.

sponding segmentation information in order to determine if the point belongs to the object or background category. This allows us to count the number of projections projected onto object category pixels. Concretely, we define the object affinity o_j of a three-dimensional point $\mathbf{o}_j^{(o)}$ in the object point cloud according to (5.4).

$$o_j = \sum_i \theta_i(\mathbf{p}_{j,i}) \;/\; \sum_i \sigma_i(\mathbf{p}_{j,i}) \tag{5.4}$$

The pixel classification function $\theta_i(\mathbf{p}) = 1$, if $\mathbf{p}$ corresponds to the object in image i and $\theta_i(\mathbf{p}) = 0$, otherwise. $\sigma_i(\mathbf{p})$ takes the visibility into account, *i.e.*, $\sigma_i(\mathbf{p}) = 1$, if $\mathbf{p}$ is visible in image i and $\sigma_i(\mathbf{p}) = 0$, otherwise. Defining a threshold ratio r_o allows us to filter the reconstructed object points, *i.e.*, we keep only points for which $o_j > r_o$ holds. By weighting each camera equally and without any prior knowledge the optimal decision is achieved using $r_o = 0.5$. However, it is reasonable to adjust r_o according to the video content. For instance, video data with low object occlusions allow us to use higher r_o values.

Let us assume for simplicity that our segmentation is perfect for a certain picture. Table 5.1 shows the corresponding projection cases of a 3D point. The evaluation of 3D point projections can be understood as a 3D object point classification task. A *False Positive* (FP) describes the case where a background point is being considered as part of the object and a *False Negative* (FN) represents the complementary situation. The cases FP and FN may lead to an incorrect filtering of 3D points. In order to handle FPs resulting from 3D points close to the object surface we give background projections more



(a) Unlabeled point cloud.

(b) Assignment of object and non-object labels.

(c) Removal of non-boundary points.

(d) Creation of points inside the space spanned by the boundary points.

(e) Assignment of point labels.

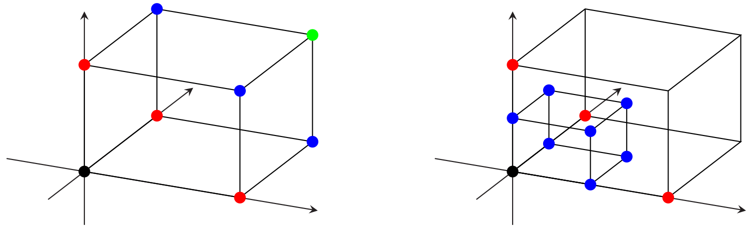
(f) Removal of non-boundary points.

Figure 5.4: Two dimensional example of the boundary generation algorithm. The red line denotes the true but unknown object shape. Blue and green represent object and non-object labels.

emphasis. This can be achieved by selecting a high r_o value. However, increasing r_o will also increase the influence of FNs. However, video data with low object occlusion and stable object segmentations contain only few FNs. We used $r_o = 0.98$ in our experiments.

5.6 3D Boundary Generation

Applying SfM to moving objects with reflecting and textureless surfaces, *e.g.*, cars, usually reconstructs few stable points and produces outliers or points with incorrect normals. Both properties potentially lead to meshes with incorrect shapes. We compute clean object meshes by replacing the original object point cloud with virtually generated points. We exploit the object segmentations and the virtual camera poses computed during the SfM reconstruction process to create 3D points consistent to the two-dimensional shapes of the object in the corresponding images. The generated points are uniformly distributed and show consistent normal vectors.
First, we compute a three-dimensional bounding box corresponding to the original sparse point cloud. Next, we divide the space of this bounding box in $O(k^3)$ equal subspaces and represent each cell with one point at the center. Here k is the number of subdivisions in each dimension.
By applying the method described in Section 5.5 we assign an object or a non-object flag to each point in the grid - *q.v.* Fig. 5.4b. This divides the space of the bounding box in an object and a background volume. We compute the corresponding object boundary by removing all non-boundary points - see Fig. 5.4c. A point is considered as a boundary point if and only if one of the corresponding neighbors has a complementary flag.
Unfortunately, the computation of an accurate surface reconstruction using this approach is not reasonable as it requires $O(k^3)$ (non-)object flag computations. Therefore, we iteratively apply this approach as follows. First, we use a coarse subdivision of the bounding box space to compute an initial set of boundary points, *i.e.*, points lying on the boundary of the object or the background volume. Next, we create points on a more fine-grained level inside the space spanned by the current boundary points, which are possibly closer to the true object boundary - see Fig. 5.4d. Then, we assign (non-)object flags to the newly generated points - see Fig. 5.4e. This allows us to update the set of real boundary points and to adjust the shape of the object boundary represented by these points - see Fig. 5.4f. We iteratively repeat these computations.
We initialize the set of boundary points (BPs_i) in iteration 1 with boundary points computed at a coarse division (*e.g.*, 1000 cells) of the bounding box space (BPs_{Coarse}). In each iteration i we build a kd-tree containing all points of BPs_i to efficiently determine the nearest neighbors of each boundary point. It is important to note that the neighbors of a boundary point differ in terms



(a) Neighbor types of a boundary point. (b) Creation of possible boundary points.

Figure 5.5: Neighbor types and creation of new boundary points. For reasons of clarity, this figures shows only one of eight grid cells that determine the neighbors of a boundary point (black). A boundary point has maximal 26 neighbor points. Fig. 5.5a shows that a boundary point has six neighbors with distance d (red), twelve neighbors with distance $\sqrt{2}d$ (blue), and eight neighbors with $\sqrt{3}d$ (green), where d is the length of the edges of the cells in the grid. Fig. 5.5b visualizes the creation of seven new possible boundary points (blue) using three neighbors with distance d (red).

of their distance. The different neighbor types are illustrated in Fig. 5.5a. The varying distances must be considered while using the kd-tree to determine the nearest grid neighbors of a boundary point.

We iterate over the current boundary points (BPs_i) and determine for each BP the set of neighbors with distance d to create a set of possible boundary points ($PBPs_i$). We use the points $(x+d,y,z)$, $(x,y+d,z)$ and $(x,y,z+d)$ to compute new points on a more fine-grained level according to Fig. 5.5b. Here, (x,y,z) represents the three-dimensional coordinate of the current boundary point. We generate only a subset of points or no new points at all, if there are less or no points meeting the criteria above. The creation of new points on a more fine-grained level is equivalent to a division of the cell into eight cuboids as well as increasing the point density by a factor of two.

Next, we compute (non-)object flags for all newly generated $PBPs_i$. By removing all non-boundary points in $PBPs_i \cup BPs_i$ we adjust the boundary. To determine if a point is a boundary point we analyze the corresponding 26 grid neighbors. After several iterations we cover the space on a fine-grained level. The essential steps of the proposed algorithm are depicted in Algorithm 4.

The number of required iterations depends on the granularity of the points before the first iteration d_0 as well as the desired point density d_i. In each iteration the distance between points is halved. To reach a point density of d_d the algorithm requires $\lceil \log \frac{d_0}{d_i} \rceil$ iterations.

Algorithm 4: Outline of the boundary generation algorithm.

```
BPs_1 = BPs_Coarse                         // BP: Boundary Point
for i = 1...k do
    PBPs_i ← ∅              // PBP: Possible Boundary Point
    // Create points inside the space spanned by BPs_i
     (Fig. 5.4d)
    for p ∈ BPs_i do
        NNs ← getNearestNeighborsKDTree(6, p, BPs_i)
        // GNs correspond to the red points in Fig. 5.5a
        GNs ← getGridNeighbors(p, NNs)
        // PBPs_i correspond to the blue points in
         Fig. 5.5b
        PBPs_i ← PBPs_i ∪ createPossibleBPs(p, GNs)
    end
    PBPs_i = assignFlagsToPoints(PBPs_i) // See Fig. 5.4e
    PBPs_i = PBPs_i ∪ BPs_i
    BPs_{i+1} ← ∅
    // Compute refined boundary points (Fig. 5.4f)
    for p ∈ PBPs_i do
        NNs ← getNearestNeighborsKDTree(26, p, BPs_{i+1})
        GNs ← getGridNeighbors(p, NNs)
        // We must consider the flags of the
         corresponding
        // neighbors to decide if a point is a boundary
         point
        if isBoundaryPoint(p, GNs) then
            BPs_{i+1} ← BPs_{i+1} ∪ {p}
        end
    end
end
```

The generated boundary point set is used to generate a mesh describing the object contour.

5.7 Experimental Evaluation

This section shows qualitative and quantitative results using drone footage and the dataset described in Section 4.1.

5.7.1 Qualitative Evaluation

All evaluations presented in the following use the semantic segmentation ConvNet proposed by Zheng et al. (2015). We choose this one over Long et al. (2015) since the latter creates a less accurate silhouette, produces sometimes false positives as well as disconnected components. In order to compute the dilation and erosion we select an ellipse as structuring element and set its radius to ten pixels, since our investigation of different segmentation samples showed that the boundary inaccuracies of the ConvNet are usually smaller than five pixels.

Comparison of Boundary Generation and Multi-View Stereo

We use a video sequence of 510 images viewing a vehicle from multiple sides to emphasize the different reconstruction results obtained by Structure from Motion, Multi-View Stereo and Boundary Generation. We use the approach in Section 5.5 to remove outliers in the object point cloud. Fig. 5.6 shows an example of a Structure from Motion reconstruction result before and after the semantic outlier filtering step. The removal of outliers reduces the number of incorrect polygons in the meshes computed during the dense reconstruction step.
Fig. 5.7 compares the boundary generation results with sparse and dense reconstructions as well as the corresponding polygon mehses. Fig. 5.7d shows the sparse point cloud of the car using Wu (2011). The dense model representation shown in Fig. 5.7e is computed applying the Multi-View Stereo algorithm by Goesele et al. (2007). The stereo matching technique uses a

(a) Original reconstruction. (b) Reconstruction after outlier removal.

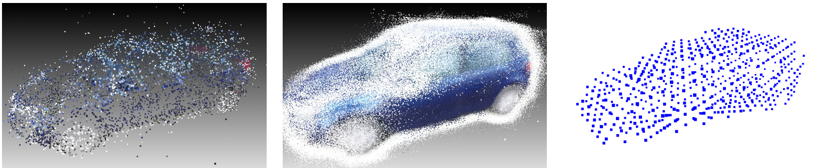
Figure 5.6: SfM reconstruction (top view) before and after semantic outlier removal. The reconstructed cameras are represented with red triangles. In the corresponding scene the vehicle as well as the camera is moving.

previously computed SfM result to build a depth map for each virtual camera. The dense model is created by projecting the depth values of each virtual camera into the world coordinate system. For both, the sparse and the dense point clouds, we remove outliers using the method described in Section 5.5. The corresponding results are shown in Fig. 5.7g and Fig. 5.7h. Also the boundary generation uses the virtual camera poses estimated during the SfM computation. Fig. 5.7f and 5.7i show the results after the first and third iteration, respectively. All meshes (see Fig. 5.7j, 5.7k and 5.7l) are computed with the Poisson surface reconstruction algorithm by Kazhdan et al. (2006). We leverage Waechter et al. (2014) to determine a texture for each mesh.

Comparison of Boundary Generation to Lebeda *et al*.

We compare our boundary generation method visually to the approach presented in Lebeda et al. (2015) on publicly available video data. The video sequence consists of 76 input images showing a rally car. Fig. 5.8 contains two example input pictures, the result computed by Lebeda et al. (2015) and a mesh based on the output of our boundary generation algorithm. The shape of our result is more accurate, *i.e.*, closer to the real shape of the car. Due to missing ground truth data a quantitative comparison is not possible.

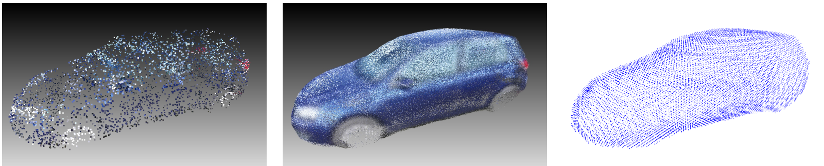
(a) Input image. (b) Input image. (c) Input image.



(d) Sparse reconstruction (Wu, 2011).

(e) Dense reconstruction (Goesele et al., 2007).

(f) Boundary generation: iteration 0.



(g) Sparse reconstruction (Wu, 2011) of (d) after outlier removal.

(h) Dense reconstruction (Goesele et al., 2007) of (e) after outlier removal.

(i) Boundary generation: iteration 2.

(j) Textured mesh (Kazhdan et al., 2006; Waechter et al., 2014) using the points of (g).

(k) Textured mesh (Fuhrmann and Goesele, 2014; Waechter et al., 2014) using the points of (h).

(l) Textured mesh (Kazhdan et al., 2006; Waechter et al., 2014) using the points of (i).

Figure 5.7: Object shape reconstruction results using a single sequence of 510 images.

(a) Example input image. (b) Textured object mesh using Lebeda et al. (2014). (c) Textured mesh using the proposed boundary generation method (ours).

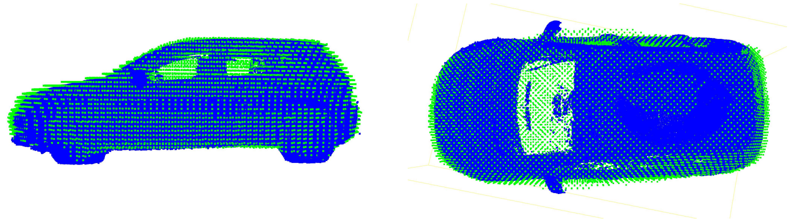
Figure 5.8: Comparison of our boundary generation method using the video sequence presented in Lebeda et al. (2014).

5.7.2 Quantitative Evaluation

In this section we evaluate the accuracy of the proposed boundary generation algorithm quantitatively using the dataset described in Section 4.1. We applied our pipeline to all 25 sequences contained in the dataset.

In nine sequences the reconstruction fails, *i.e.*, the SfM process performs incorrect image registrations or produces (multiple) partial models. Incorrect image registrations are caused by ambiguous feature matches due to object symmetries and repetitive elements (*e.g.*, feature matches between opposite wheels). The lack of consistent feature matches results in (multiple) partial models (*e.g.*, only the front or the back side are reconstructed). The main reasons for this is that a) only few salient features on the vehicle surface are detected and b) the corresponding descriptors are corrupted by reflections and illumination changes.

In seven of the eight trajectory-types the car is captured from at least three sides. We select for each of these seven trajectories one sequence and compute the distance between the boundary and the laser scan point cloud. We automatically scale and register the boundary point cloud to the ground truth using the Iterative Closest Point implementation of CloudCompare (Girardeau-Montaut, 2016). It is important to note that in contrast to the generated object boundaries the laser scan data contains no points at windows and at the bottom side of the vehicle. Since there is no correspondence information between laser scan and boundary generation points, we use the following steps to



(a) Overlay of boundary (green) and laser scan (blue) points - side view.

(b) Overlay of boundary (green) and laser scan (blue) points - top view.

Figure 5.9: Overlay of a generated boundary point cloud and the vehicle laser scan ground truth.

Trajectory Type							
Average Distance (cm)	4.4	3.0	2.6	3.1	3.2	3.3	3.4
Standard Deviation (cm)	5.8	3.0	2.6	2.9	3.1	3.3	3.4

Table 5.2: Evaluation boundary accuracy.

find reasonable correspondences. First, we compute for each boundary point the nearest neighbor in the laser scan. We determine the distance between each boundary-laser-scan-point-pair. If multiple boundary points share the same nearest neighbor we keep only the boundary point with the smallest distance. We use the average distance and the standard deviation of the remaining boundary-laser-scan-point-pairs as evaluation measure. Since the videos in the dataset contain no object occlusions we use an object ratio of $r_o = 0.98$. Fig. 5.9 shows the laser scan and an overlay of a generated boundary and the ground truth.

Table 5.2 shows the evaluation of the seven trajectories using the output of the 4th iteration and roughly 1000 cells as initial subdivision. Fig. 5.10 shows an overlay of the reconstructed 3D boundary mesh and the original input image sequence.

(a) Input Frames.

(b) Input images overlayed with the object meshes rendered from the corresponding camera poses.

Figure 5.10: Comparison of input images and object meshes computed by the proposed boundary generation method allows to assess color and shape consistency of the result. The images show frames of the 5th trajectory of the presented dataset.

5.8 Discussion

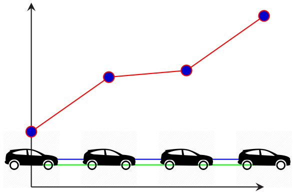
We presented a pipeline to reconstruct the three-dimensional structure of moving objects in video data. We observe that SfM based point clouds of moving objects with reflecting surfaces often result in crumbled meshes due to outliers, irregular point densities and incorrect normal vectors. We tackled this problem by introducing an algorithm combining the information contained in virtual camera poses and semantic segmentations. The proposed approach constrains surfaces of the object not directly seen by the camera. We applied our algorithm on publicly available video data and on 25 sequences from our dataset. The algorithm achieves an average point distance of 3.3 cm evaluating seven trajectories contained in the dataset using a laser scan as ground truth.
At the moment, we initialize the focal length with (5.2). The estimation of correct focal length values is especially difficult for moving objects because of limited object sizes - objects usually cover only minor parts of the image. In the case of a moving camera, one could use object and background images to compute a joint estimation of the focal length values, since the corresponding cameras in the object and the background reconstruction share the same parameters.

One limitation of the presented approach is that object occlusions may cause incorrect object point classifications, *i.e.*, actual object points are being considered as non-object structures. One way to tackle this issue may be the detection of such cases in the input images. Also, the geometric information of different scene components contained in the Multibody Structure from Motion reconstruction (*q.v.* Section 3.4 and Chapter 6) are presumably useful to detect object occlusions.
Given suitable camera-object-trajectories we have demonstrated that semantic segmentations provide useful cues to infer three-dimensional object shapes. A tight coupling of semantic boundary information and SfM/MVS may allow to reconstruct moving objects in more constrained scenarios. For example, by leveraging semantic constraints during point triangulation. Such a combination could improve the accuracy and the consistency of the computed point cloud.

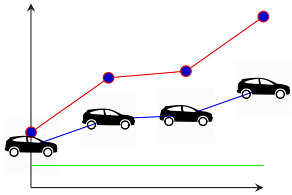
6 Object Trajectory Reconstruction using Instance-Aware Multibody Structure from Motion

This chapter presents several methods to reconstruct *trajectories of dynamic objects* in monocular and stereo image sequences. Parts of this chapter have been published in Bullinger et al. (2018a), Bullinger et al. (2018b), Bullinger et al. (2019a) and Bullinger et al. (2019b). The described methods rely on the Instance-aware Multibody Structure from Motion approach presented in Chapter 3 to reconstruct moving objects and environment structures. Because of the scale ambiguity of MSfM we will analyze motion as well as stereo constraints to determine consistent vehicle trajectories.

The remaining part of this chapter is organized as follows. We describe the problem statement in Section 6.1. As shown in Section 3.5 the different components of a multibody reconstruction are defined up to scale. We derive a formal representation of an object trajectory in Section 6.2 that reflects the scale ambiguities of MSfM reconstructions. Section 6.3 shows that any reconstructed object trajectory may be considered a superposition of the camera motion and the true object trajectory. Because of the scale ambiguity of MSfM, we require additional constraints to compute consistent object trajectories. In the monocular case, we apply object motion constraints to determine the scale ratio of object and environment reconstruction (*q.v.* Section 6.4). In Section 6.5 we leverage the basesline of the stereo camera to resolve the scale ambiguity in stereo image sequences. In Section 6.6 and Section 6.7 we show qualitative and quantitative results of the proposed trajectory reconstruction methods. Section 6.8 concludes this chapter.



(a) Consistent object trajectory. Object and background reconstruction are scaled accordingly.



(b) Inconsistent object trajectory. The scale ratio between object and background reconstruction is 0.5.

Figure 6.1: Visualization of the visual trajectory reconstruction problem statement. The camera trajectory is shown in red, the object trajectory in blue and the ground in green.

6.1 Problem Statement

Reconstructing trajectories of dynamic objects consists of estimating the corresponding object pose (6 degrees of freedom) for each time step. As seen in Section 3.5 Multibody Structure from Motion allows to determine object poses up to scale, *i.e.*, MSfM restricts the object trajectory to a one-parameter family of possible solutions parameterized by the unknown scale ratio between object and background reconstruction. Thus, the computation of consistent object trajectories is equivalent to the determination of the corresponding scale ratio. The computation of consistent scale ratios is important, since incorrect scale ratios do not only change the *extent* of the reconstructed trajectories but also the corresponding *shape*. Fig. 6.1 shows an example, which illustrates this effect. The reason for this is that the reconstructed trajectory is a superposition of the true object trajectory as well as the camera trajectory. More details are given in Section 6.3.
Because of the scale ambiguity of image based reconstructions, it is impossible to compute the scale ratio directly. One way to tackle this problem is the exploitation of object motion constraints, which are typically category-specific. For example, vehicles move on the ground and specific subcategories like cars additionally rotate around the center of the back axles. In the case of stereo image sequences, the scale ratio between object and background reconstruction may be determined using the baseline of the stereo cameras. High attention must be paid to the accuracy of the reconstructed camera pose, since small deviations typically have a strong impact on scale ratio.

With the correct scale ratios we are able to compute consistent object trajectories.

6.2 Scale Ambiguous Trajectory Representation

Without loss of generality, we describe the motion trajectory reconstruction of a single object. In Section 3.5 we have seen that the transformation between the object and the background CFS using a camera with index i may be expressed according to (6.1),

$$\mathbf{T}_i^{(o2b)}(r) := \begin{bmatrix} r\mathbf{R}_i^{(b)^\mathsf{T}}\mathbf{R}_i^{(o)} & -r\mathbf{R}_i^{(b)^\mathsf{T}}\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} + \mathbf{c}_i^{(b)} \\ 0 & 1 \end{bmatrix} \tag{6.1}$$

where $\mathbf{R}_i^{(o)}$ and $\mathbf{c}_i^{(o)}$ denote the camera rotation and position corresponding to frame i in the object reconstruction $sfm^{(o)}$. In contrast, $\mathbf{R}_i^{(b)}$ and $\mathbf{c}_i^{(b)}$ represent the camera pose in the background reconstruction $sfm^{(b)}$.
Applying $\mathbf{T}_i^{(o2b)}$ to a point $\mathbf{o}_j^{(o)}$ in *object* coordinates according to (6.2) yields the corresponding point in *background* coordinates $\mathbf{o}_{j,i}^{(b)}(r)$ at time i.

$$\mathbf{T}_i^{(o2b)}(r) \begin{bmatrix} \mathbf{o}_j^{(o)} \\ 1 \end{bmatrix} = \begin{bmatrix} r\mathbf{R}_i^{(b)^\mathsf{T}}\mathbf{R}_i^{(o)}(\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) + \mathbf{c}_i^{(b)} \\ 1 \end{bmatrix} =: \begin{bmatrix} \mathbf{o}_{j,i}^{(b)}(r) \\ 1 \end{bmatrix} \tag{6.2}$$

We compute the position of $\mathbf{o}_{j,i}^{(b)}(r)$ according to (6.3).

$$\mathbf{o}_{j,i}^{(b)}(r) = \mathbf{c}_i^{(b)} + r \cdot \mathbf{R}_i^{(b)^\mathsf{T}} \cdot \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) := \mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)} \tag{6.3}$$

with

$$\mathbf{v}_{j,i}^{(b)} = \mathbf{R}_i^{(b)^T} \cdot \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}). \tag{6.4}$$

Given the scale ratio r, we can determine the true object point positions for each time step i using (6.3). We use $\mathbf{o}_{j,i}^{(b)}(r)$ of all cameras and object points as object motion trajectory representation, *i.e.*, the trajectory is represented by a one-parameter family of possible solutions. The scale ambiguity is expressed by the unknown scale ratio r. Fig. 6.2 shows a visualization of (6.3).

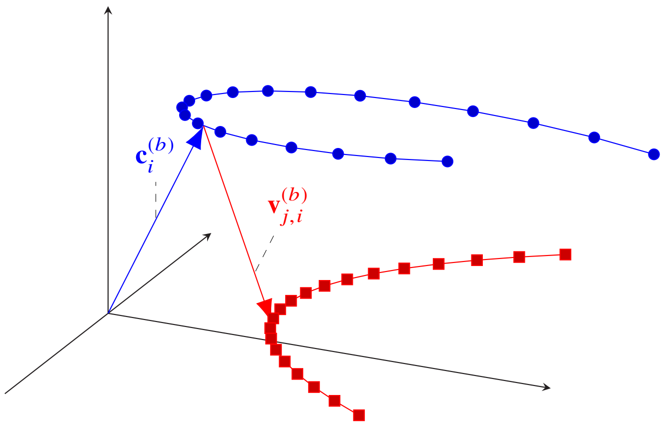


Figure 6.2: Concept of the trajectory computation. The vector $\mathbf{c}_i^{(b)}$ is the vector from the origin to the camera center. $\mathbf{v}_{j,i}^{(b)}$ is the rotated vector pointing from the camera center $\mathbf{c}_i^{(b)}$ to object point $\mathbf{o}_j^{(b)}$.

6.3 Scale Effects and Object Trajectory Shape

As shown in Ozden et al. (2004) any incorrectly scaled trajectory is a linear combination of the camera and the true object motion. According to Fig. 3.12 the transformations between the different coordinate frame systems (object, camera and world) are subject to (6.5).

$$\begin{bmatrix} \mathbf{R}_i^{(o2b)}(r) & \mathbf{t}_i^{(o2b)}(r) \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^{(b)^\mathsf{T}} & \mathbf{c}_i^{(b)} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} r\mathbf{R}_i^{(o)} & -r\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} \\ \mathbf{0} & 1 \end{bmatrix} \tag{6.5}$$

$\mathbf{R}_i^{(o2b)}(r)$ and $\mathbf{t}_i^{(o2b)}(r)$ describe rotation and scaling as well as the translation of a vector given in the object CFS at time i to the CFS of the background reconstruction in dependence of the scale ratio r. Considering only the translation components in (6.5) yields (6.6).

$$\mathbf{t}_i^{(o2b)}(r) = \mathbf{c}_i^{(b)} - r\mathbf{R}_i^{(b)^\mathsf{T}}\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} \tag{6.6}$$

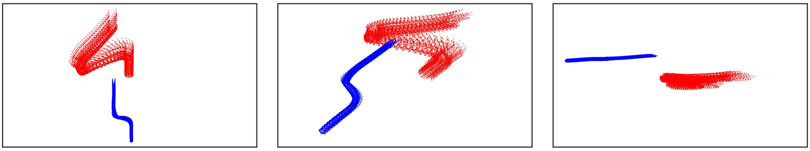


Figure 6.3: Example of the influence of an inconsistent scale ratio on the shape of the reconstructed object trajectory. The consistent object trajectory is shown in blue. The red trajectory is the result of a direct combination of an object and a background reconstruction without computing the corresponding scale ratio. The images show different views of the same trajectory pair.

Multiplying $\mathbf{R}_i^{(b)}$ from left to (6.6) yields (6.7).

$$\begin{aligned} \mathbf{R}_i^{(b)}\mathbf{t}_i^{(o2b)}(r) = \mathbf{R}_i^{(b)}\mathbf{c}_i^{(b)} - r \cdot \mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} \Leftrightarrow \\ r \cdot \mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} = \mathbf{R}_i^{(b)}\mathbf{c}_i^{(b)} - \mathbf{R}_i^{(b)}\mathbf{t}_i^{(o2b)}(r) \end{aligned} \tag{6.7}$$

Let us assume that the object and background reconstructions are correctly scaled. In this case the scale ratio r in (6.7) must be equal to one. Other scale ratios (*i.e.*, $r \neq 1$), will result in incorrect trajectories. In this case (6.7) can be adjusted to (6.8).

$$\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} = \mathbf{R}_i^{(b)}\mathbf{c}_i^{(b)} - \mathbf{R}_i^{(b)}\mathbf{t}_i^{(o2b)}(1) \tag{6.8}$$

Substituting (6.8) in (6.6) yields (6.9).

$$\begin{aligned} \mathbf{t}_i^{(o2b)}(r) = \mathbf{c}_i^{(b)} + r \cdot \mathbf{R}_i^{(b)^\mathsf{T}}\mathbf{R}_i^{(o)}\mathbf{c}_i^{(o)} \Leftrightarrow \\ \mathbf{t}_i^{(o2b)}(r) = \mathbf{c}_i^{(b)} - r \cdot \mathbf{R}_i^{(b)^\mathsf{T}}(\mathbf{R}_i^{(b)}\mathbf{c}_i^{(b)} - \mathbf{R}_i^{(b)}\mathbf{t}_i^{(o2b)}(1)) \Leftrightarrow \\ \mathbf{t}_i^{(o2b)}(r) = \mathbf{c}_i^{(b)} - r \cdot (\mathbf{c}_i^{(b)} - \mathbf{t}_i^{(o2b)}(1)) \Leftrightarrow \\ \mathbf{t}_i^{(o2b)}(r) = (1-r)\mathbf{c}_i^{(b)} + r\mathbf{t}_i^{(o2b)}(1) \end{aligned} \tag{6.9}$$

Equation (6.9) shows that the scaled object motion is a linear combination of the true object trajectory $\mathbf{t}_i^{(o2b)}(1)$ and the camera trajectory $\mathbf{c}_i^{(b)}$. Further, (6.9) describes how the scale ratio influences extent and shape of an object trajectory. For $r = 1$ (6.9) shows that the scaled trajectory is equal to the true trajectory. Fig. 6.3 shows an example how different scale ratios change the extent as well as the shape of the object trajectory.

6.4 Monocular Trajectory Reconstruction

The reconstruction of object motion trajectories in monocular video data captured by moving cameras is a challenging task, since in general it cannot be solely solved leveraging image observations. Because observed object motion trajectories are scale ambiguous, additional constraints such as motion priors are required to identify motion trajectories consistent to environment structures. Presumably, there is no universal motion constraint providing satisfiable results for all types of object motion. Previous works (*q.v.* Section 6.4.1) proposed category specific trajectory constraints. In this work we consider the domain of vehicles, which is of a broad interest for many applications. We define two types of object trajectory constraints that apply directly to the reconstructed object points. Both methods leverage geometric information of environment reconstructions to determine consistent object trajectories. Section 6.4.2 presents an approach, which assumes that the object of interest moves on a locally planar surface, *i.e.*, each object point in the background CFS shows for different time steps a constant distance to the corresponding local approximation of the terrain surface. In contrast, Section 6.4.3 leverages vehicle-environment-projections to solve the scale ambiguity.

6.4.1 Related Work

The determination of the correct three-dimensional object trajectory, *i.e.*, the computation of the correct scale ratio between object and background reconstruction, requires additional priors or constraints.
Lee et al. (2015), Song and Chandraker (2015) and Chhaya et al. (2016) focus on vehicle mounted cameras where the sensor pose shows specific properties, *e.g.*, a fixed height and angle. These approaches are not applicable to other scenarios in which the camera undergoes less controlled motions such as cameras mounted on drones or motorcycles.
Ozden et al. (2004) propose the *non-accidentalness* and the *independence* principle to reconstruct 3D object trajectories. The first states that the motion of moving objects is not coincidental whereas the latter assumes that consistent object motions and camera trajectories are linearly independent. In contrast to Lee et al. (2015), Song and Chandraker (2015) and Chhaya et al. (2016), these approaches are also applicable to non-driving scenarios.

Several previously proposed methods (Kundu et al., 2011; Namdev et al., 2013; Ozden et al., 2004; Park et al., 2015; Yuan and Medioni, 2006) leverage the *non-accidentalness* principle to determine consistent object motions. Yuan and Medioni (2006) propose to reconstruct the 3D object trajectory by assuming that the object motion is parallel to a single ground plane. Kundu et al. (2011) exploit motion segmentation with multibody Visual SLAM to reconstruct the trajectory of moving cars. Kundu et al. (2011) use an instantaneous constant velocity model in combination with a *Bearing only Tracker* to estimate consistent object scales. Namdev et al. (2013) assume that the vehicles show motions according to non-holonomic curves and straight lines. Park et al. (2015) propose an approach to reconstruct the trajectory of a single 3D point tracked over time by approximating the motion using a linear combination of trajectory basis vectors. This approach is suitable to reconstruct independently moving point sets.
Ozden et al. (2010) propose an approach that leverages splitting and merging of points of different scene components to determine the corresponding scale ratios, which is conceptually different to the previously mentioned methods.
In contrast to previous work (Chhaya et al., 2016; Kundu et al., 2011; Lee et al., 2015; Namdev et al., 2013; Ozden et al., 2004, 2010; Park et al., 2015; Song and Chandraker, 2015; Yuan and Medioni, 2006), we show quantitative results and use the environment geometry to determine consistent three-dimensional object trajectories.

6.4.2 Vehicle Trajectory Reconstruction using Constant Distance Constraints

This section bases on the monocular vehicle trajectory reconstruction approach proposed in Bullinger et al. (2018b). The core idea of the presented motion constraint is that each object point in the CFS of the background reconstruction shows a constant distance to the ground for all time steps i. In contrast to Yuan and Medioni (2006), our method uses information of temporal distant time steps and assumes that the object of interest moves on a *locally* planar surface, *i.e.*, the terrain may contain slopes and different elevations. The reconstructed vehicle trajectory shows this property only for the true scale ratio and a non-degenerated camera motion.

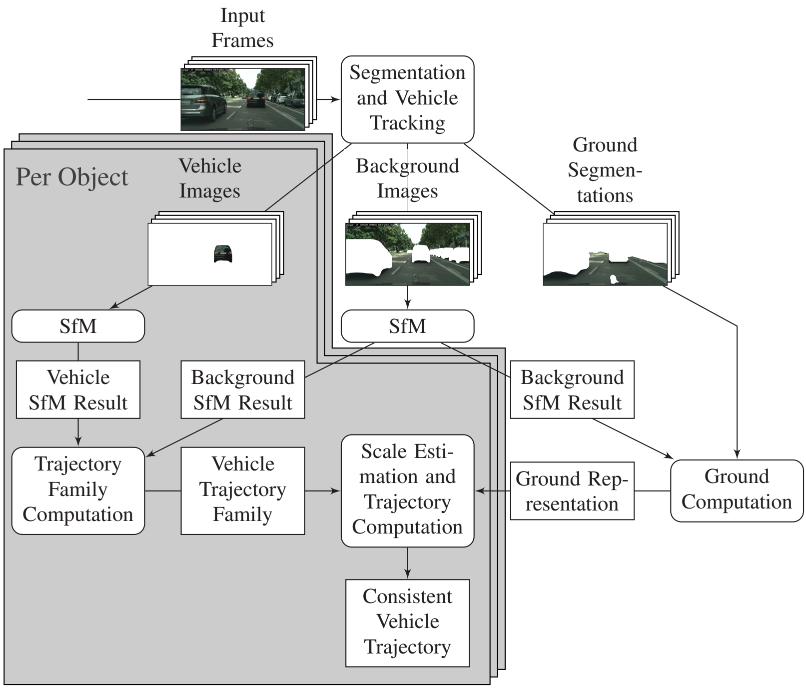


Figure 6.4: Overview of the trajectory reconstruction pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps, respectively. The steps in the gray area are performed for each object.

Pipeline Overview

Fig. 6.4 shows the elements of the proposed pipeline. We use the Multibody Structure from Motion approach presented in Chapter 3 to track two-dimensional object shapes in monocular image sequences on pixel level and to reconstruct corresponding three-dimensional object/environment points as well as camera poses. Without loss of generality, we describe the reconstruction of single object motion trajectories. We apply SfM (Moulon et al., 2012; Schönberger and Frahm, 2016) simultaneously to object and background images as shown in Fig. 6.4. Object images denote images containing only color information of a single object instance. Similarly, background images show only background structures. We combine object and background reconstructions to compute a one-parameter-family of possible, visually identical, object motion trajectories. We determine the correct scale ratio by exploiting constraints derived from the reconstructed terrain geometry.

Terrain Ground Approximation

We combine Structure from Motion results and semantic segmentations to estimate locally planar approximations of the ground surface. We apply the ConvNet presented in Shelhamer et al. (2017) to determine ground categories like street and grass for all input images on pixel level. We determine for each 3D point in the background reconstruction a ground or non-ground label by accumulating the semantic labels of the corresponding keypoint measurement pixel positions. This allows us to determine a subset of background points, which represent the ground of the scene. Let $\mu(j)$ be the set of image indices used to triangulate a background point j. Further, let $\boldsymbol{\nu}_{j,i}$ denote the pixel position of the corresponding observation in image i. We define the ground affinity according to (6.10).

$$g_j = \frac{1}{|\mu(j)|} \sum_{i \in \mu(j)} \phi_i(\boldsymbol{\nu}_{j,i}) \tag{6.10}$$

The pixel classification function $\phi_i(\boldsymbol{\nu}) = 1$, if $\boldsymbol{\nu}$ corresponds to ground in image i and $\theta_i(\boldsymbol{\nu}) = 0$, otherwise. We use the points $\mathbf{b}_j^{(b)}$ in the background point cloud with $g_j > 0.5$ to represent the ground. We consider only stable background points, *i.e.*, 3D points that are observed at least four times.
We approximate the ground surface locally using plane representations. For each frame i we use the corresponding estimated camera parameters and object point observations to determine a set of ground points $\mathcal{P}_i$ close to the object. We build a kd-tree containing all ground measurement positions of the current frame. For each object point observation, we determine the num_b closest background measurements. Let $card_i$ be the cardinality of $\mathcal{P}_i$. While $card_i$ is less than num_b, we add the next background observation of each point measurement. This results in an equal distribution of local ground points around the vehicle. We apply RANSAC (Fischler and Bolles, 1981) to compute a local approximation of the ground surface using $\mathcal{P}_i$. Each plane is defined by a corresponding normal vector $\mathbf{n}_i$ and an arbitrary point $\mathbf{p}_i$ lying on the plane.

Scale Estimation using Constant Distance Constraints

This approach exploits priors of object motion to improve the robustness of the reconstructed object trajectory. We assume that the object of interest moves on a locally planar surface. In this case the distance of each object point $\mathbf{o}_{j,i}^{(b)}(r)$ to the ground is constant for all cameras i. The reconstructed trajectory shows this property only for the true scale ratio and non-degenerated camera motion.

Scale Ratio Estimation using a Single View Pair We use the term *view* to denote cameras and corresponding local ground planes. The signed distance of an object point $\mathbf{o}_{j,i}^{(b)}(r)$ to the ground plane can be computed according to $d_{j,i} = \mathbf{n}_i \cdot (\mathbf{o}_{j,i}^{(b)}(r) - \mathbf{p}_i)$, where $\mathbf{p}_i$ is an arbitrary point on the local ground plane and $\mathbf{n}_i$ is the corresponding normal vector. If the object moves on top of the approximated terrain ground the distance $d_{j,i}$ is independent of a specific camera i. Thus, for a specific point with index j and two different cameras with index i and i' the relation shown in (6.11) holds.

$$\mathbf{n}_i \cdot (\mathbf{o}_{j,i}^{(b)}(r) - \mathbf{p}_i) = \mathbf{n}_{i'} \cdot (\mathbf{o}_{j,i'}^{(b)}(r) - \mathbf{p}_{i'}). \tag{6.11}$$

Substituting (6.1) in (6.11) results in (6.12). Here, $r_{j,i,i'}$ highlights that the computed scale ratio depends on the object point with index j and the cameras with index i and i'.

$$\mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} + r_{j,i,i'} \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{p}_i) = \mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} + r_{j,i,i'} \cdot \mathbf{v}_{j,i'}^{(b)} - \mathbf{p}_{i'}) \tag{6.12}$$

Solving (6.12) for $r_{j,i,i'}$ yields (6.13).

$$r_{j,i,i'} = \frac{\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)}{(\mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)})} \tag{6.13}$$

Equation (6.13) allows us to determine the scale ratio $r_{j,i,i'}$ between object and background reconstruction using the extrinsic parameters of two cameras and corresponding ground approximations.

Scale Ratio Estimation using View Pair Ranking The accuracy of the estimated scale ratio $r_{j,i,i'}$ in (6.13) is subject to the condition of the parameters of the particular view pair. For instance, incorrectly estimated local plane position and normal vectors may disturb camera-plane distances. In addition, if the numerator or denominator is close to zero, small errors in the camera poses or ground approximations may result in negative scale ratios. Because of the effects mentioned previously, not all view pairs are suitable to estimate reasonable scale ratios. A least squares solution of all camera pairs and all points is inadvisable, since the corresponding scale ratios $r_{j,i,i'}$ do not necessarily follow a Gaussian distribution. Instead, we tackle these problems by combining two different view pair rankings. The first ranking uses for each view pair the difference of the camera-plane distances (6.14).

$$\delta_{i,i',cam} = |\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)| \tag{6.14}$$

The second ranking reflects the quality of the local ground approximation w.r.t. the object reconstruction. A single view pair allows to determine $\left|\mathcal{P}^{(o)}\right|$ different scale ratios, where $\mathcal{P}^{(o)}$ represents the number of object points. For a view pair with stable camera registrations and well reconstructed local planes the variance of the corresponding scale ratios must be small. This allows us to determine ill conditioned view pairs. The second ranking uses the scale ratio difference to order the view pairs. We sort the view pairs by weighting both ranks equally.

This ranking is crucial to deal with motion trajectories close to degenerated cases. In contrast to other methods, this ranking allows to estimate consistent vehicle motion trajectories, even if the majority of local ground planes are incorrectly reconstructed. Concretely, this approach allows to determine a consistent trajectory using a single suitable view pair.

Let v denote the view pair with the lowest overall rank. The final scale ratio is determined by using a least squares method w.r.t. all equations of v according to (6.15). Let i and i' denote the image indices corresponding to v.

$$\underbrace{\begin{bmatrix} \cdots \\ \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)} \\ \cdots \\ \mathbf{n}_i \cdot \mathbf{v}_{j+1,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j+1,i'}^{(b)} \\ \cdots \end{bmatrix}}_{\mathbf{A}} \cdot r = \underbrace{\begin{bmatrix} \cdots \\ \mathbf{n}_{i'}(\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i) \\ \cdots \\ \mathbf{n}_{i'}(\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i) \\ \cdots \end{bmatrix}}_{\mathbf{b}} \tag{6.15}$$

Algorithm 5: View pair selection algorithm.

Let $\mathcal{V}$ denote the set of all possible view pairs.
Let $v_{i,i'} \in \mathcal{V}$ denote the view pair corresponding to the frame indices i and i'.
for $v_{i,i'} \in \mathcal{V}$ **do**
  $r_{min} \leftarrow \infty$
  $r_{max} \leftarrow -\infty$
  for $\mathbf{o}_j^{(o)} \in \mathcal{P}^{(o)}$ **do**
    $r_{j,i,i'} = \frac{\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)}{(\mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)})}$ // q.v. (6.13)
    $r_{min} \leftarrow \min(r_{min}, r_{j,i,i'})$
    $r_{max} \leftarrow \max(r_{max}, r_{j,i,i'})$
  end
  $\delta_{i,i',var} \leftarrow r_{max} - r_{min}$
  $\delta_{i,i',cam} \leftarrow |\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)|$ // q.v. (6.14)
end
Compute ranking R_1 of $v_{i,i'} \in \mathcal{V}$ using $\delta_{i,i',var}$.
Compute ranking R_2 of $v_{i,i'} \in \mathcal{V}$ using $\delta_{i,i',cam}$.
Weight the ranks of R_1 and R_2 equally to determine a final ranking R_f.
Select the view pair with the lowest rank in R_f.

The equation system in (6.15) contains $|\mathcal{P}^{(o)}|$ rows, *i.e.*, one row for each point $\mathbf{o}_j^{(o)}$ in the object reconstruction. The solution may be computed using the normal equation $r = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. Algorithm 5 summarizes the steps of the proposed algorithm.

Degenerated Motion Case

The proposed method shows similar to previous published motion constraint based scale ratio estimation methods a degenerated reconstruction case. Concretely, it is impossible to reconstruct the vehicle trajectory, when the camera shows a constant orthogonal distance to the local ground approximation around the vehicle.

Section 6.2 shows that object points in the background CFS are constrained by (6.16).

$$\mathbf{o}_{j,i}^{(b)}(r) = \mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)} \tag{6.16}$$

The orthogonal distance $d_{j,i}$ of an object point $\mathbf{o}_{j,i}^{(b)}(r)$ in background coordinates to the i-th plane is given by (6.17).

$$d_{j,i} = \mathbf{n}_i \cdot (\mathbf{o}_{j,i}^{(b)}(r) - \mathbf{p}_i) \tag{6.17}$$

Substituting $\mathbf{o}_{j,i}^{(b)}(r)$ in (6.17) yields (6.18).

$$d_{j,i} = \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{p}_i) = \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i) + r \cdot \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} \tag{6.18}$$

If the camera shows for each time step i the same distance c to the corresponding local ground plane $(\mathbf{n}_i, \mathbf{p}_i)$ then (6.18) may be simplified to (6.19).

$$d_{j,i} = c + r \cdot \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} \tag{6.19}$$

The approach proposed in Section 6.4.2 assumes that the distance of each object point is constant for any time steps i and i', *i.e.*, $d_{j,i} = d_{j,i'}$. Combining both assumptions results in (6.20).

$$d_{j,i} = d_{j,i'} \Leftrightarrow c + r \cdot \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} = c + r \cdot \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)} \Leftrightarrow \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} = \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)} \tag{6.20}$$

Equation (6.20) shows that in this case, $d_{j,i} = d_{j,i'}$ is true for any selection of r and therefore not suitable to determine r.
For a more general discussion of degenerated camera motions see Ozden et al. (2004).

6.4.3 Vehicle Trajectory Reconstruction using Projection Constraints

In Section 6.4.2 we determined consistent object trajectories using motion constraints. As other previously published methods (Kundu et al., 2011; Namdev et al., 2013; Ozden et al., 2004; Park et al., 2015; Yuan and Medioni, 2006), the algorithm in Section 6.4.2 shows a degenerated reconstruction case. In this section we present a vehicle trajectory reconstruction approach with-

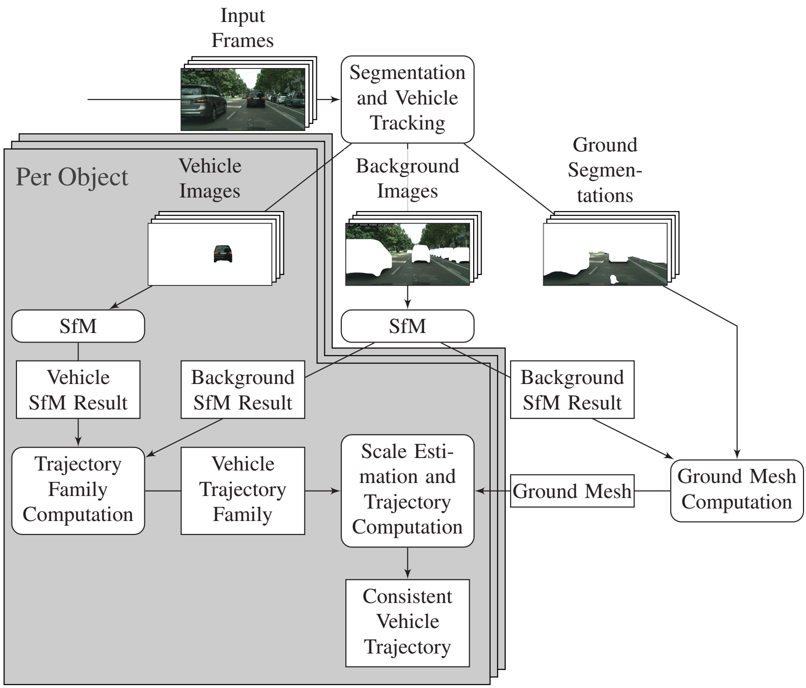


Figure 6.5: Pipeline of the vehicle trajectory reconstruction approach. Computation results are represented by boxes with corners and computation steps by boxes with rounded corners. Arrows show computational dependencies.

out ill-posed camera-object-trajectories. The proposed algorithm is originally described in Bullinger et al. (2018a) and uses vehicle projection constraints to determine the scale ratio between object and background reconstruction, *i.e.*, the algorithm computes consistent object trajectories by projecting dense vehicle reconstructions on the terrain surface. We present an efficient projection implementation leveraging depth buffer values of mesh renderings.

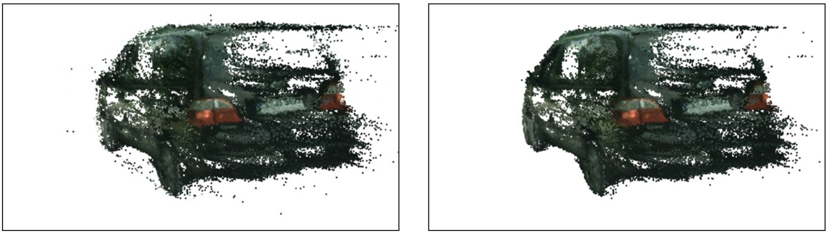
Pipeline Overview

Fig. 6.5 outlines the pipeline of our approach. The input is an ordered image sequence. We track two-dimensional object shapes on pixel level across video sequences following the scheme proposed in Chapter 3. As described in Chapter 3 we apply SfM (Moulon et al., 2012; Schönberger and Frahm, 2016) to reconstruct object and background images as shown in Fig. 6.5. In contrast to the pipeline described in Section 6.4.2, we represent the environment with a watertight terrain mesh. We project dense vehicle point clouds onto the reconstructed mesh to determine consistent vehicle environment scale ratios.

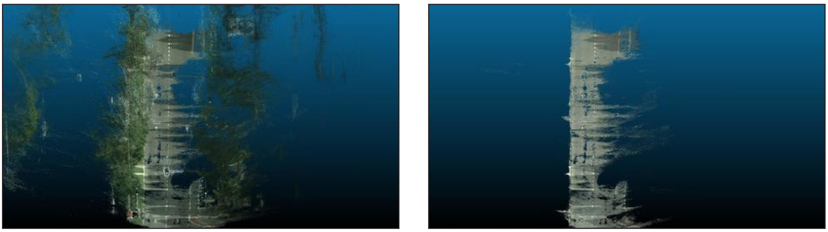
Scale Ratio Estimation using Projection Constraints

We tackle the problem of determining consistent object-background-scale-ratios by exploiting geometric consistency constraints applicable to ground restricted object categories, like vehicles. In contrast to previous works, our approach does neither rely on restrictions of camera and object motions nor specific camera poses. The proposed scale-ratio estimation approach shows no degenerated cases, in which a consistent object trajectory computation is impossible.
Our method exploits the fact that some vehicle points should touch the terrain surface, like 3D points corresponding to the wheels of a car. To ensure the presence of suitable 3D points we enhance the points in $sfm^{(o)}$ by leveraging the Multi-View Stereo (MVS) algorithm presented in Schönberger et al. (2016). In contrast to sparse SfM algorithms, the MVS library by Schönberger et al. (2016) reliably triangulates points at wheels of driving vehicles.
We exploit the previously computed instance-aware object segmentations to determine outliers in the dense object point cloud by following the outlier removal method in Section 5.5 - see (5.4). We classify an object point j as outlier, if $o_j < 0.9$. This threshold is empirically determined and takes the robustness of the instance-aware segmentation computed with He et al. (2017) into account.
We apply statistical outlier removal to the previously computed object points using the standard deviation of the mean distance as outlier criterion. The mean distance is computed considering the five next neighbors. Fig. 6.6b shows a dense object reconstruction before and after outlier removal.

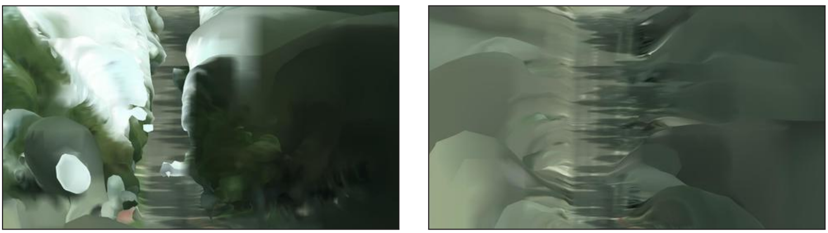
(a) Example input frames.



(b) Dense object reconstruction before (left) and after (right) outlier removal.



(c) Dense point cloud representing background (left) and ground (right).



(d) Mesh representing background (left) and ground (right).

Figure 6.6: Intermediate results used for the scale ratio computation. Results are computed with data from the Cityscape dataset (Cordts et al., 2016).

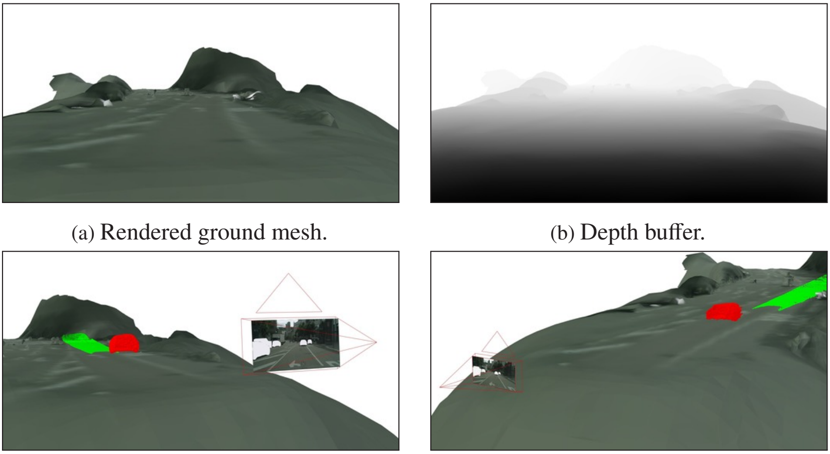
We apply the MVS algorithm presented in Schönberger et al. (2016) to the sparse background reconstruction $sfm^{(b)}$ to compute a dense background representation. We exploit ground segmentations to classify 3D points in the background point cloud as ground or non-ground points following the approach in Section 6.4.2 - see (6.10). We use the points $\mathbf{b}_j^{(b)}$ in the dense background point cloud with $g_j > 0.5$ to compute a dense ground point cloud. Fig. 6.6c compares the dense background reconstruction with the points classified as ground.

We use the algorithm described in Kazhdan and Hoppe (2013) to compute watertight ground meshes. This allows us to inter- and extrapolate ground surface areas occluded by moving objects. We determine connected components in the ground mesh and remove isolated mesh parts. Fig. 6.6d shows an example of a computed ground mesh. The removal of non-ground points before computing the mesh speeds up the computation and leads to a more precise representation of the ground geometry.

To determine a consistent object-background-reconstruction scale ratio we use (6.3) to create for each camera i a set of vectors $\mathbf{v}_{j,i}^{(b)}$ pointing from the camera center $\mathbf{c}_i^{(b)}$ to the position $\mathbf{o}_{j,i}^{(b)}$ of point j. Let $\mathcal{F}$ denote the set of faces contained in the ground mesh and $h_{j,i}$ the ray defined by $\mathbf{c}_i^{(b)}$ and $\mathbf{v}_{j,i}^{(b)}$.

A naive approach to determine the closest ray-ground-mesh-intersection of a ray $h_{j,i}$ requires the computation of the intersection of the ray with each face $f \in \mathcal{F}$ and the corresponding intersection parameters. This includes intersection tests with occluded faces and faces not visible in the field of view of the current background camera i. This makes the object-ground-ray intersection computation for all rays $h_{j,i}$ computationally expensive.

Instead of computing object-ground-ray intersections, we use the visualization toolkit (VTK) (Schroeder et al., 2006) to render the ground mesh from the perspective of camera i. We exploit the information stored in the depth buffer to determine 3D-3D object-ground-correspondences. We determine for each point $\mathbf{o}_j^{(o)}$ the corresponding point $\mathbf{o}_j^{(i)} = \mathbf{R}_i^{(o)}(\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)})$ in the camera coordinate system of camera i as well as the corresponding image projection $\mathbf{x}_{i,j} = \mathbf{K}_i \mathbf{o}_j^{(i)}$. For $\mathbf{x}_{i,j}$ we use the corresponding depth buffer value to determine a point $\mathbf{p}_j^{(i)}$ lying on the ground mesh surface with the same projection than $\mathbf{o}_j^{(i)}$ w.r.t. camera i. We apply bilinear interpolation while accessing depth buffer values. Fig. 6.7 shows an example of a terrain mesh and corresponding depth buffer values as well as object projections.



(a) Rendered ground mesh.

(b) Depth buffer.

(c) Object point cloud and corresponding projected points.

(d) Object point cloud and corresponding projected points.

Figure 6.7: Projection of the object point cloud (red) onto the ground mesh using the depth buffer. The projected points are shown in green. The inconsistent initial scale ratio becomes apparent by examining the distance between object points and corresponding projections. Results are computed using the Stuttgart01 sequence in the Cityscape dataset (Cordts et al., 2016).

To determine a consistent scale ratio, we must find the smallest r, which satisfies $\|\mathbf{o}_j^{(i)}\| = r \cdot \|\mathbf{p}_j^{(i)}\|$ for an arbitrary point $\mathbf{o}_j^{(i)}$ in the object point cloud. We separately compute r according to equation (6.21) for each image i.

$$r_i = \min(\{\|\mathbf{p}_j^{(i)}\| \cdot (\|\mathbf{o}_j^{(i)}\|)^{-1} | j \in \{1, \ldots, |\mathcal{P}^{(o)}|\}\}) \tag{6.21}$$

The scale ratio and intersection parameter r_i corresponds to the point being closest to the ground surface, *i.e.*, a point at the bottom of the vehicle. Plugging r_i in (6.3) for camera i places the object point cloud on top of the ground surface. Thus, the smallest ray-plane-intersection-parameter r_i represents the object-to-background-scale-ratio. We reconstruct the three-dimensional vehicle trajectory as defined in equation (6.22).

$$r = med(\{r_i | i \in \{1, \ldots, n_I\}\}) \tag{6.22}$$

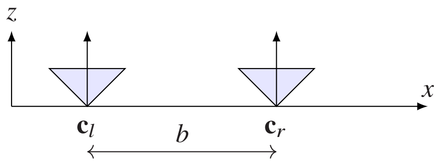


Figure 6.8: Coordinate frame system of the stereo camera. $\mathbf{c}_l$ and $\mathbf{c}_r$ denote the centers of the left and right camera and b denotes the corresponding baseline. x and z are the coordinate axis of the stereo camera system.

Here, med denotes the median and n_I the number of images. Using the median of the scale ratios r_i improves the robustness of the proposed approach w.r.t. to outlier scale ratios caused by incorrectly triangulated parts of the environment mesh. For the median computation we do not consider invalid image scale ratios r_i, *i.e.*, cameras which have no camera-object-point-rays intersecting the ground representation.
To compute the final object trajectory we compute (6.3) for each point j at all time steps i. The removal of outliers in the object reconstruction greatly improves the object trajectory visualization, since a single outlier in the object reconstruction results in multiple outliers in the final object trajectory.

6.5 Stereo Trajectory Reconstruction

In Section 6.4, we used *monocular* image sequences to reconstruct three-dimensional vehicle trajectories. The scale ambiguity of image based reconstructions represents one of the main challenges using monocular image data for three-dimensional object trajectory reconstruction. In the case of *stereo* image data, the baseline of the stereo camera induces an unambigous scale of the corresponding reconstruction result.
The stereo camera set up used in the following sections is depicted in Fig. 6.8. The coordinate axis of the left and right image plane are coplanar and shifted along the x axis. The baseline b represents the distance between the camera centers.
In the following, we propose two methods for three-dimensional object trajectory reconstruction. The approach in Section 6.5.2 uses stereo matching to triangulate object points and standard SfM of background images to compute

the three-dimensional camera motion. The algorithm uses the tracking presented in Chapter 3 to determine object specific disparity values. Section 6.5.3 presents a method that leverages the full stereo MSfM described in Chapter 3 to compute object and environment reconstructions. The pipeline uses factor graphs to model stereo projection constraints, which allow to refine the reconstructed stereo camera poses.

6.5.1 Related Work

The majority of previously published methods for three-dimensional object trajectory reconstruction such as Ošep et al. (2017), Engelmann et al. (2017) or Coenen et al. (2018) use stereo matching (Scharstein and Szeliski, 2002) for object point triangulation. Such methods are limited by the stereo camera baseline, since stereo matching uses pixel disparities of rectified stereo image pairs to infer corresponding depth values.
Liang et al. (2018) and Chang and Chen (2018) presented two ConvNet based stereo matching approaches outperforming the previous state-of-the-art on the Stereo Robust Vision Challenge (Rob, 2018). The usage of Liang et al. (2018) as well as Chang and Chen (2018) is limited because of the lack of pre-trained models and the required fine-tuning in the target domain. Fine-tuning is necessary to use these methods in arbitrary scenarios, since the corresponding ConvNet models are trained on image data with specific sensor properties. Thus, we considered different widely used off-the-shelf stereo matching methods (Geiger et al., 2010; Hirschmuller, 2008; Yamaguchi et al., 2014) to compute disparity values. We observe that Geiger et al. (2010) compute more stable object specific disparities than Hirschmuller (2008) and Yamaguchi et al. (2014).
Ladický et al. (2012) compute stereo matching and class segmentation jointly using Conditional Random Fields. Bleyer et al. (2011) leverage Markov Random Fields to perform joint optimization of stereo matching and class-agnostic object segmentation. In contrast, our methods described in Section 6.5.2 and Section 6.5.3 allow to compute stereo matching results associated with instance information as well as class labels.
Recently, several works determined object models including object shape and pose using stereo matching based point triangulations. Ošep et al. (2017) present a combination of object proposals, stereo matching, visual odometry and scene flow to compute three-dimensional vehicle tracks in traffic scenes.

Ošep et al. (2017) combine 2D object bounding box detections and 3D stereo depth measurements, which results in background structures being considered as object points. The detections and measurements are tracked with a 2D-3D Kalman filter to compute three-dimensional bounding box proposals for each object. Coenen et al. (2018) leverage a deformable vehicle shape prior to reconstruct 3D pose and shape. Engelmann et al. (2017) use off-the-shelf ego-motion and stereo matching methods for vehicle trajectory reconstruction. Engelmann et al. (2017) track objects in 3D using Chen et al. (2015) and impose a common shape and motion model by combining the information acquired by multiple frames corresponding to the same track.

6.5.2 3D Object Trajectory Reconstruction using Stereo Matching

This section bases on Bullinger et al. (2019a) and presents a method for object trajectory reconstruction using stereo matching. Stereo matching does not allow to differentiate between static and dynamic scene structures, since corresponding points are triangulated using a single stereo image pair. To tackle this issue our pipeline uses the tracking approach described in Chapter 3 to track two-dimensional objects and corresponding disparities on pixel level. This allows us to triangulate object specific points using stereo matching.

Pipeline Overview

Fig. 6.9 shows an overview of the proposed object trajectory reconstruction pipeline. The input images of the stereo camera are rectified to simplify subsequent processing steps. We apply stereo matching to compute corresponding pixel disparity values for each image pair of the stereo image sequence. Following the Multiple Object Tracking (MOT) approach presented in Chapter 3 we track objects on pixel level in the images captured by the left sensor of the stereo camera. For each object we leverage corresponding segmentation masks to determine object specific disparity values, which allow us to triangulate dense object points for each frame, *i.e.*, the baseline of the stereo camera is known. We use the background images as input for SfM to compute a background model and associated stereo camera poses for each time

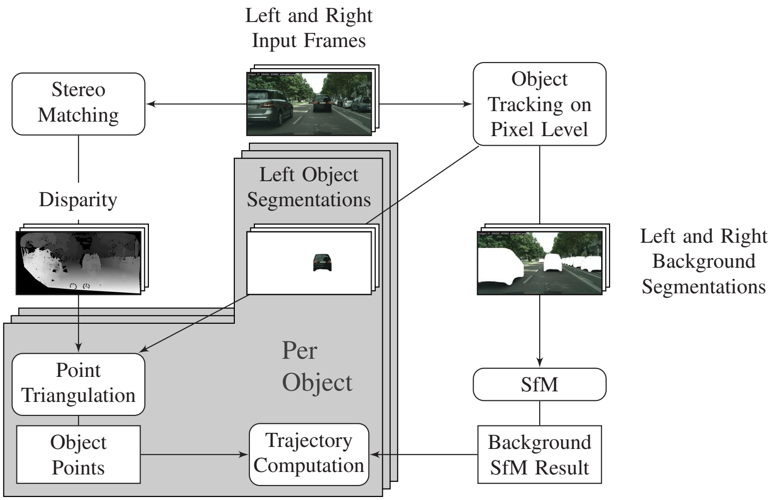


Figure 6.9: Overview of the trajectory reconstruction pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps.

step. We transform the stereo matching based triangulated object points in the background CFS using the camera poses of the background reconstruction. Transforming the object points into the environment reconstruction CFS step allows us to determine the three-dimensional object motion trajectory. The usage of the left and right camera images for SfM allows to compute metric reconstructions, *i.e.*, the results are not scale ambiguous.

Trajectory Reconstruction and Outlier Removal

Stereo matching (Scharstein and Szeliski, 2002) based point triangulation exploits the relative poses of the left and the right sensor of a stereo camera to determine three-dimensional scene points. Corresponding matches are determined along so called scan lines and allow to define pixelwise disparity functions $d_i(\cdot)$ for each time step.
Without loss of generality, we describe the trajectory reconstruction for a single object. In the following, $(u, v) \in \mathcal{P}_i$ denotes the set of pixels representing the current object in image i. Fig. 6.8 shows the setup of the stereo cam-

era system and the corresponding coordinate frame systems, *i.e.*, the x axis is pointing to the right, the y axis downwards and the z axis forward. We use the disparity-to-depth mapping matrix $\mathbf{Q}$ according to (6.23) to determine homogeneous points $(x_u, y_v, z, w_{u,v,i})$ corresponding to the pixel disparity triplets $(u, v, d_i(u, v))$ of the left image.

$$\begin{bmatrix} x_u \\ y_v \\ z \\ w_{u,v,i} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & -c_u \\ 0 & 1 & 0 & -c_v \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{b} & \frac{c_u - c'_u}{b} \end{bmatrix}}_{\mathbf{Q}} \cdot \begin{bmatrix} u \\ v \\ d_i(u,v) \\ 1 \end{bmatrix} \tag{6.23}$$

Here, (c_u, c_v) and f denote the principal point and the focal length in pixels. b is the extent of the stereo camera baseline in the background SfM coordinate frame system. This ensures, that object points and camera poses are correctly scaled. Normalizing $(x_u, y_v, z, w_{u,v,i})^T$ yields the actual three-dimensional object point $\mathbf{o}_{u,v}^{(i)} = (\frac{x_u}{w_{u,v,i}}, \frac{y_v}{w_{u,v,i}}, \frac{z}{w_{u,v,i}})^T$ in camera coordinates. We decrease computation time and memory consumption using only every second object pixel for triangulation.
We observe that incorrectly estimated disparity values lead to distant, isolated object points - usually close to the object boundary. We assume that each object consists of a single connected component, *i.e.*, each object point has neighbor points with similar depth values. Equation (6.24) shows the depth error δz of a stereo camera system with parallel optical axes depending on the stereo camera baseline b, the focal length f and the disparity error δd. For more details see Chang and Chatterjee (1992).

$$\delta z = \frac{z^2 \cdot \delta d}{b \cdot f} \tag{6.24}$$

Equation (6.24) shows a) the estimated depth error δz increases quadratic with the distance z and b) the estimation of close points is more reliable than the computation of distant points. Defining a threshold for disparity variation between adjacent pixels allows us to compute dynamic depth intervals of valid object points, which take the corresponding depth value into account.
For each object pixel $(u, v) \in \mathcal{P}_i$, we consider a local $l \times l$ neighborhood of object points $\mathcal{N} = \{\mathbf{o}_{u+m,v+n}^{(i)} \mid m,n \in \{-\lfloor \frac{l}{2} \rfloor, \ldots, \lfloor \frac{l}{2} \rfloor\} \wedge (u+m, v+n) \in$

$\mathcal{P}_i\}$ around (u, v). Let $z_{u,v,i}$ denote the depth value corresponding to $\mathbf{o}_{u,v}^{(i)}$. We consider $\mathbf{o}_{u,v}^{(i)}$ as outlier, if there is a point $\mathbf{o}_{u+m,v+n}^{(i)} \in \mathcal{N}$ with $z_{u,v,i} > z_{u+m,v+n,i} + \delta z_{u+m,v+n,i}$. In this case $\mathbf{o}_{u+m,v+n}^{(i)}$ lies closer to the camera and according to equation (6.23) the corresponding depth can be estimated more reliably.
To compute the full object trajectory we transform the object point cloud for each time step i into world coordinates with $\mathbf{o}_{u,v,i}^{(b)} = \mathbf{c}_i + \mathbf{R}_i{}^T \cdot \mathbf{o}_{u,v}^{(i)}$.

6.5.3 3D Object Trajectory Reconstruction Stereo Sequence Constraints

Stereo matching is a common approach to determine three-dimensional scene information from images taken by a stereo camera. The stereo camera baseline limits corresponding point triangulations (Pinggera et al., 2014). In contrast, this section describes an approach that leverages information of subsequent frames for object point triangulation to reconstruct three-dimensional object trajectories. Already small object rotations may result in big virtual camera baseline changes. In contrast to stereo matching methods, the proposed approach builds object models reflecting the information of each frame. To build a holistic object model with stereo matching requires additional steps to fuse triangulated points of subsequent frames.
The proposed approach has been first published in Bullinger et al. (2019b) and uses the *stereo* MSfM described in Chapter 3 to compute object and background reconstructions. The initial SfM results are refined with stereo projection constraints using factor graphs. We compute object trajectories using stereo sequence constraints of object and background reconstructions.

Pipeline Overview

Fig. 6.10 shows the pipeline of the presented object trajectory reconstruction approach. The input is an ordered sequence captured by a stereo camera. The images of the stereo camera are rectified to simplify subsequent processing steps. We compute object and background reconstructions with the *stereo* MSfM described in Chapter 3. We represent the SfM reconstruction results (camera poses, three-dimensional structure points, keypoint observations) as

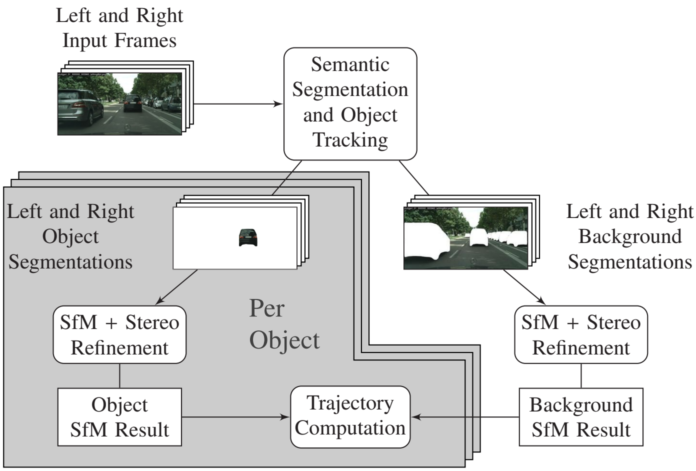


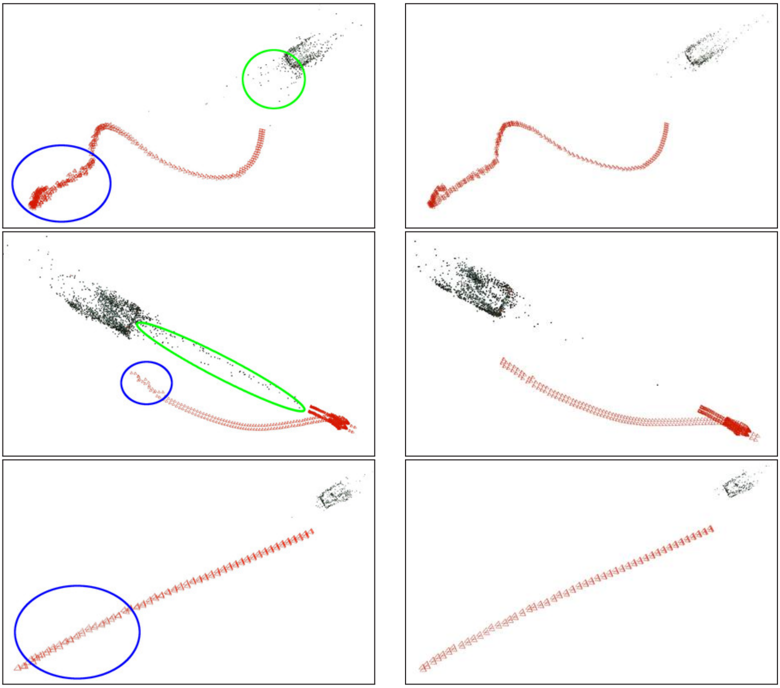
Figure 6.10: Overview of the trajectory reconstruction pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps.

factor graphs (Kschischang et al., 2001). The integration of corresponding stereo projection constraints allows us to refine the initial reconstructions. We obtain SfM results with consistent camera baselines that allow us to use the stereo camera baseline to a) resolve the scale ambiguity between object and background reconstruction and b) compute consistent object trajectories.

Structure from Motion Refinement and Outlier Removal using Factor Graphs

Reconstructions of dynamic objects using state-of-the-art SfM tools occasionally contain incorrectly registered cameras as well as incorrectly triangulated object points due to small object sizes, changing illumination and reflecting surfaces. Fig. 6.11a shows a few examples. Incorrect camera baselines hamper the correct estimation of the scale ratio between object and background reconstruction.

We model stereo projection constraints to refine the previously computed SfM reconstructions by leveraging factor graphs (Kschischang et al., 2001). As described in Section 2.4 a factor graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ consists of factor nodes



(a) Initial object SfM result. (b) Refined object SfM result.

Figure 6.11: Comparison of initial SfM object reconstructions and corresponding refinements using stereo constraints. The cameras are shown in red. The blue and green circle emphasizes incorrectly registered cameras and triangulated points.

$f_k \in \mathcal{F}$ and variable nodes $\theta_l \in \Theta$, which allow to model several reconstruction constraints. The *variable* nodes represent quantities to be estimated, *i.e.*, entities that can not be directly measured, such as camera poses or three-dimensional scene points. *Factor* nodes represent constraints on possible, valid variable nodes. In the following, we describe the factor graph based refinement for the object reconstruction. The refinement of the background reconstruction is performed analogously.

In our case the variable nodes Θ represent stereo camera poses $\boldsymbol{\theta}_\mathbf{s}$ and triangulated object points $\boldsymbol{\theta}_\mathbf{p}$. We use a set of stereo factors f_k to reflect the relation of triangulated object points projected into specific stereo cameras and their cor-

responding observations. For many real world problems including most SfM reconstruction problems it is necessary to model data association, *e.g.*, feature correspondences, explicitly to achieve reasonable reconstruction results. In order to map the observation constraints in the SfM result onto the stereo factors, we determine for each triangulated object point in the SfM reconstruction all pairs of corresponding feature observations $\mathbf{m}$ and $\mathbf{m'}$ of the left and the right image of the same time step. We add stereo projection factors of the form $f_k(\boldsymbol{\theta}_\mathbf{p}, \boldsymbol{\theta}_\mathbf{s}; m_h, m'_h, m^\star_v, \mathbf{K}, b)$, where m_h and m'_h denote the horizontal pixel positions of the measurements $\mathbf{m}$ and $\mathbf{m'}$. $m^\star_v$ denotes the averaged vertical pixel position of corresponding left and right observations. $\mathbf{K}$ and b represent the calibration matrix and the stereo camera baseline. Note that in $f_k(\boldsymbol{\theta}_\mathbf{p}, \boldsymbol{\theta}_\mathbf{s}; m_h, m'_h, m^\star_v, \mathbf{K}, b)$ the parameters $\boldsymbol{\theta}_\mathbf{p}$ and $\boldsymbol{\theta}_\mathbf{s}$ are variable nodes, whereas $m_h, m'_h, m^\star_v, \mathbf{K}$ and b are fixed (measured) values. Fig. 6.12 shows an example of a mapping between a SfM reconstruction result and the corresponding factor graph.
We use the GTSAM library (Daellert, 2012) to model the SfM problem with corresponding stereo constraints as a factor graph. To determine the maximum a posteriori estimate, we apply the Levenberg-Marquardt algorithm to (2.23), which solves the nonlinear least-squares problem iteratively. We initialize the stereo camera variable nodes $\boldsymbol{\theta}_\mathbf{s}$ with the pose of the left cameras $[\mathbf{R}^{(o)}_i | \mathbf{t}^{(o)}_i]$ with $\mathbf{t}_i = -\mathbf{R}^{(o)}_i \mathbf{c}_i$ and the landmark variables nodes $\boldsymbol{\theta}_\mathbf{p}$ with the triangulated points $\mathbf{o}^{(o)}_j$ to start the optimization with reasonable values. In order to fix the reference system we add an additional variable node representing the corresponding pose prior. The resulting reconstructions show consistent camera stereo baselines. Fig. 6.11 shows a comparison of several MSfM results and corresponding factor graph refinements.
Monocular projection factors and odometry factors (between each left and right camera pose) provide an alternative to stereo projection factors. We do not consider this approach, since an increase of variable and factor nodes results in a higher computation time.
We determine for all 3D object points in the stereo-refined reconstruction result an objectness score by projecting each point onto the tracked object segmentation for all cameras. This allows us to remove outliers using the semantic outlier filtering presented in Bullinger et al. (2018a).

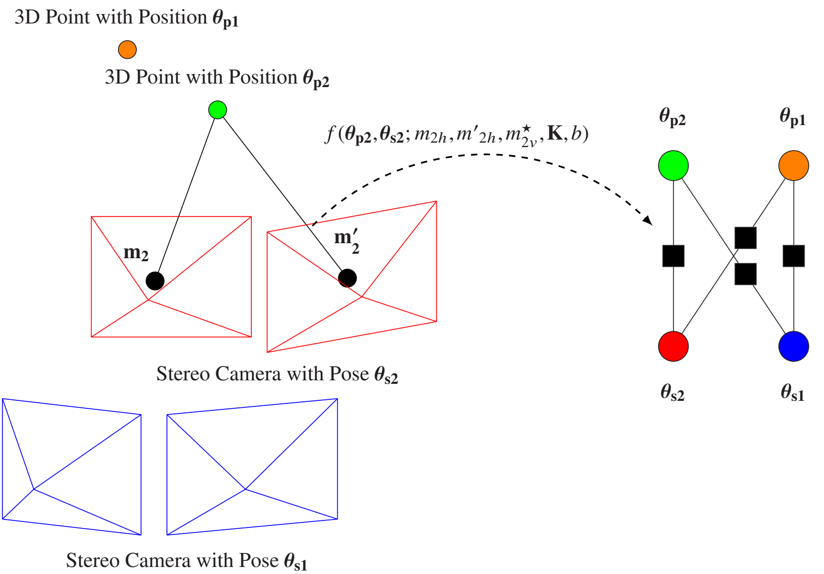
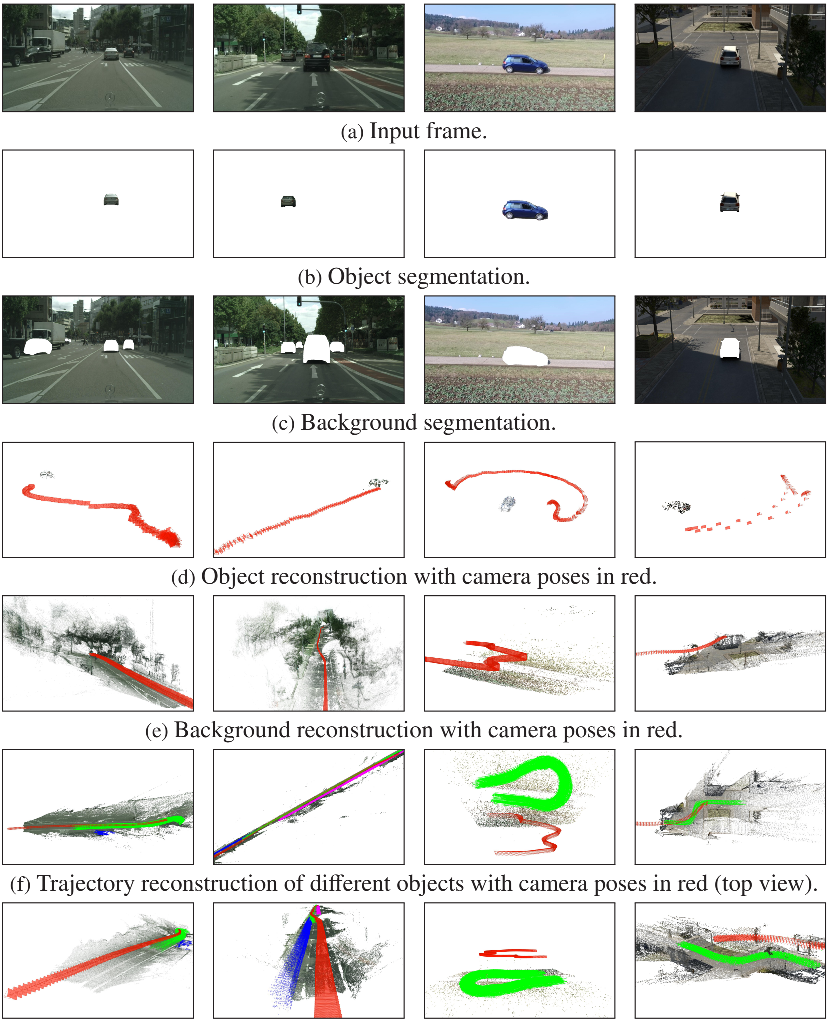


Figure 6.12: Example of a mapping of a SfM reconstruction result (left side) onto the corresponding factor graph (right side). The variable nodes of the factor graph are represented with the colors of the corresponding SfM elements. The constraint defined by the factor node $f(\boldsymbol{\theta}_{\mathbf{p2}}, \boldsymbol{\theta}_{\mathbf{s2}}; m_{2h}, m'_{2h}, m^{\star}_{2v}, \mathbf{K}, b)$ uses the variable node $\boldsymbol{\theta}_{\mathbf{p2}}$ and $\boldsymbol{\theta}_{\mathbf{s2}}$, the measurement results m_{2h}, m'_{2h} and $m^{\star}_{2v} = \frac{m_{2v}+m'_{2v}}{2}$, the calibration matrix $\mathbf{K}$ as well as the stereo baseline b. The factor nodes are represented with black squares.

6.6 Qualitative Evaluation

This section shows qualitative results (*q.v.* Fig. 6.13, Fig. 6.14, Fig. 6.15 and Fig. 6.16) of the object trajectory reconstruction methods presented in Section 6.4.2, Section 6.4.3, Section 6.5.2 and Section 6.5.3. The columns in each figure show the (intermediate) results of the corresponding method of a single input image sequence. The video data consists of custom drone footage with a resolution of 1920 px × 1080 px and sequences contained in the CityScapes dataset (Cordts et al., 2016) of 2048 px × 1024 px, in the KITTI dataset (Geiger et al., 2013) of 1242 px × 375 px and in the virtual dataset described in Section 4.2.2 of 1920 px × 1080 px. All input images are rectified.



(a) Input frame.

(b) Object segmentation.

(c) Background segmentation.

(d) Object reconstruction with camera poses in red.

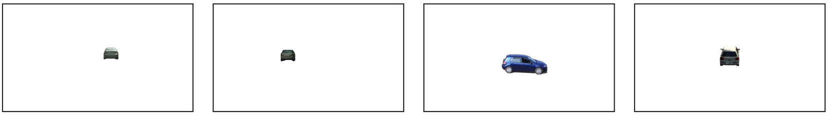
(e) Background reconstruction with camera poses in red.

(f) Trajectory reconstruction of different objects with camera poses in red (top view).

(g) Trajectory reconstruction of different objects with camera poses in red (side view).

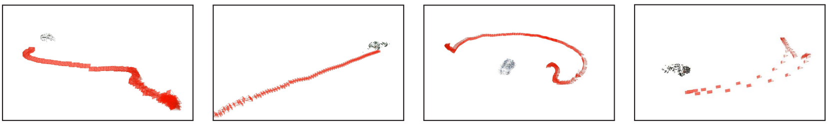
Figure 6.13: Vehicle trajectory reconstruction using the method presented in Section 6.4.2. The first two columns show sequences (stuttgart01 and stuttgart03) from the Cityscape dataset (Cordts et al., 2016), the third column represents a video captured by a drone and the sequence in the last column is part of our virtual dataset.

(a) Input frame.

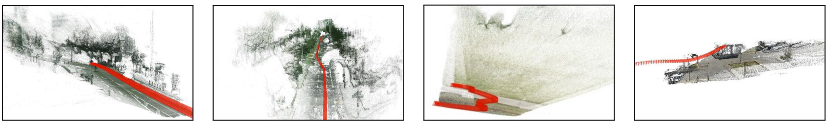


(b) Object segmentation.

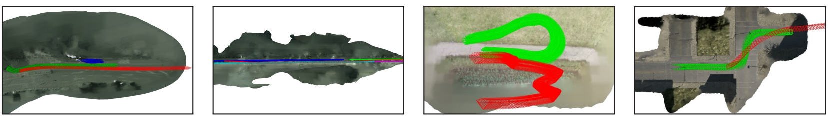
(c) Background segmentation.



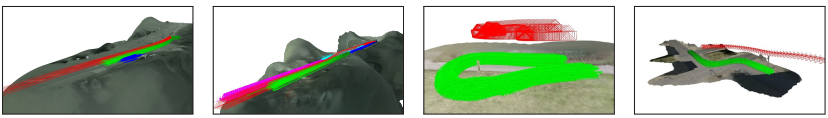
(d) Object reconstruction with camera poses in red.



(e) Background reconstruction with camera poses in red.



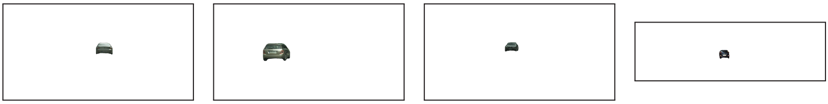
(f) Trajectory reconstruction of different objects with camera poses in red (top view).



(g) Trajectory reconstruction of different objects with camera poses in red (side view).

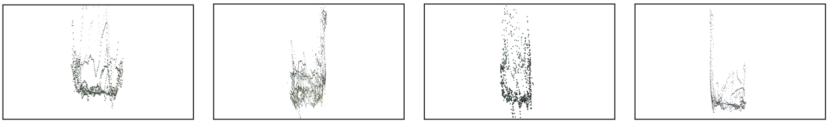
Figure 6.14: Vehicle trajectory reconstruction using the method presented in Section 6.4.3. The first two columns show sequences (stuttgart01 and stuttgart03) from the Cityscape dataset (Cordts et al., 2016), the third column represents a video captured by a drone and the sequence in the last column is part of our virtual dataset.

(a) Left input frame.

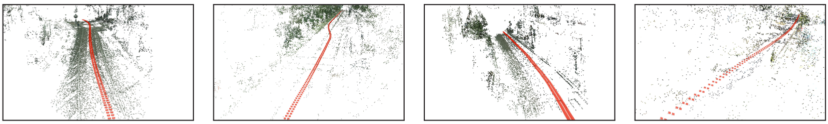


(b) Left object segmentation.

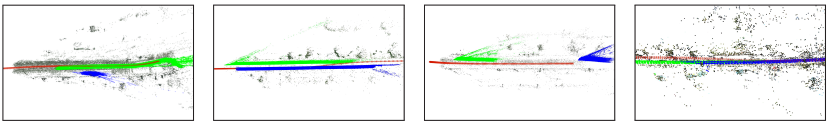
(c) Left background segmentation.



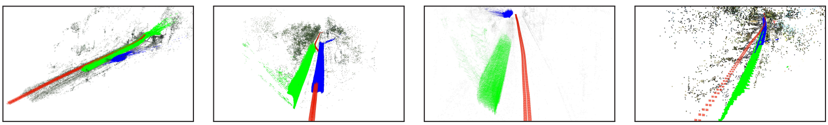
(d) Object point cloud (top view).



(e) Background reconstruction with camera poses in red.



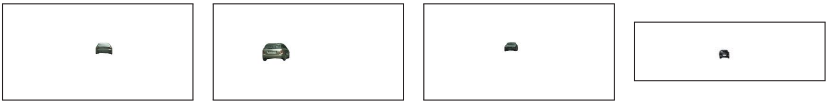
(f) Trajectory reconstruction of different objects with camera poses in red (top view).



(g) Trajectory reconstruction of different objects with camera poses in red (side view).

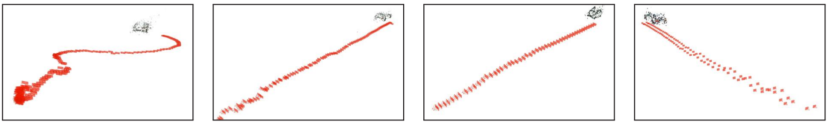
Figure 6.15: Vehicle trajectory reconstruction using the method presented in Section 6.5.2. The first three columns represent stereo sequences (stuttgart01-stuttgart03) included in the Cityscapes dataset (Cordts et al., 2016). The last column is part of the KITTI dataset (Geiger et al., 2013) - sequence (2011_09_26_drive_0013).

(a) Left input frame.

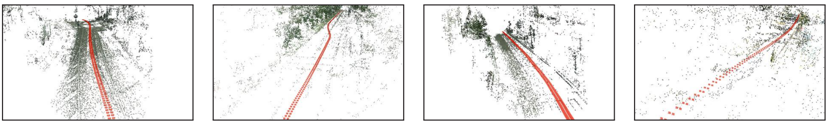


(b) Left object segmentation.

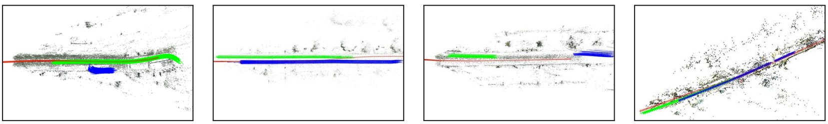
(c) Left background segmentation.



(d) Object reconstruction with camera poses in red.



(e) Background reconstruction with camera poses in red.



(f) Trajectory reconstruction of different objects with camera poses in red (top view).

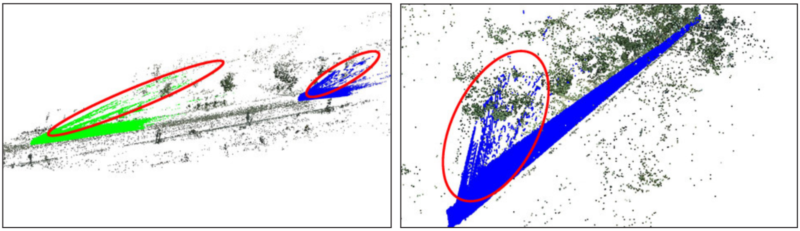
(g) Trajectory reconstruction of different objects with camera poses in red (side view).

Figure 6.16: Vehicle trajectory reconstruction using the method presented in Section 6.5.3. The first three sequences (stuttgart01-stuttgart03) are contained in the Cityscape dataset (Cordts et al., 2016). The last sequence (2011_09_26_drive_0013) is part of the KITTI dataset (Geiger et al., 2013).

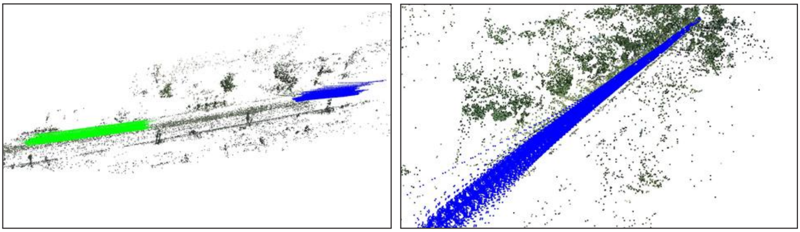
For reconstruction we used a fixed (known) camera calibration including the focal length, the principal point and potentially the stereo camera baseline.
Row (a) in Fig. 6.13, Fig. 6.14, Fig. 6.15 and Fig. 6.16 show example images of the monocular / stereo input image sequence. Row (b) and (c) visualize object and background segmentations, which are used to compute corresponding object and background reconstructions (*q.v.* row (d) and (e)). Following the pipeline presented in Chapter 3 we use the instance-aware semantic segmentation by He et al. (2017) and the optical flow features by Hu et al. (2016) to track the objects on pixel level throughout the sequences. We use Schönberger and Frahm (2016) / Geiger et al. (2010) to compute object SfM / disparity values and Moulon et al. (2012) for background reconstruction. The last two rows ((f) and (g)) show the reconstructed environment point cloud / mesh with the object trajectory / trajectories in green, blue, pink and teal from different perspectives to emphasize the three-dimensional motion in space. The reconstructed cameras are shown in red.
Although row (b) and (d) show only segmentation and reconstruction results of a single object, we perform tracking and reconstruction for multiple objects. Row (f) and (g) show the trajectory reconstruction results corresponding to multiple objects (if present).
Fig. 6.17 shows a comparison of the stereo matching based method (*q.v.* Section 6.5.2) and the stereo MSfM approach (*q.v.* Section 6.5.3). According to the Fresnel equations (Born et al., 1999) the reflection intensity depends on the angle between camera and object surface. Since the method in Section 6.5.3 uses temporal adjacent views to triangulate object points it is less prone to reflection based point correspondences. The red circle denote outliers of triangulated object points.
Fig. A.1 and Fig. A.2 in the appendix show trajectory reconstructions for individual frames of a monocular image sequences using the method proposed in Section 6.4.3.

6.7 Quantitative Evaluation

We use the virtual dataset described in Section 4.2.2 to quantitatively evaluate the proposed algorithms for object trajectory reconstruction (*q.v.* Section 6.4.2, Section 6.4.3, Section 6.5.2 and Section 6.5.3). For evaluation we must compute a registration of the reconstructed three-dimensional object tra-



(a) Stereo matching based object trajectory reconstruction (*q.v.* Section 6.5.2).



(b) Stereo MSfM based object trajectory reconstruction (*q.v.* Section 6.5.3).

Figure 6.17: Trajectory reconstruction examples using sequences of the CityScapes dataset. The red circles emphasize incorrectly triangulated trajectory points.

jectories w.r.t. the virtual environment (*q.v.* Section 6.7.1). In Section 6.7.2 we define a metric between a reconstructed trajectory (represented by multiple point sets) and a ground truth trajectory (defined by multiple watertight meshes). We perform a quantitative evaluation in Section 6.7.3 leveraging the proposed metric as well as the presented registration approach. The evaluation of the reconstructed object trajectories in the CFS of the virtual environment allows us to express the reconstruction errors in meter.

6.7.1 Registration of Background Reconstruction and Virtual Environment

Each reconstructed object trajectory consists of multiple point sets - one point cloud per frame. The virtual dataset (*q.v.* Section 4.2.2) uses object meshes to render the vehicle sequences, *i.e.*, the ground truth object trajectories are represented with a mesh per frame. We compute the *distance* between the

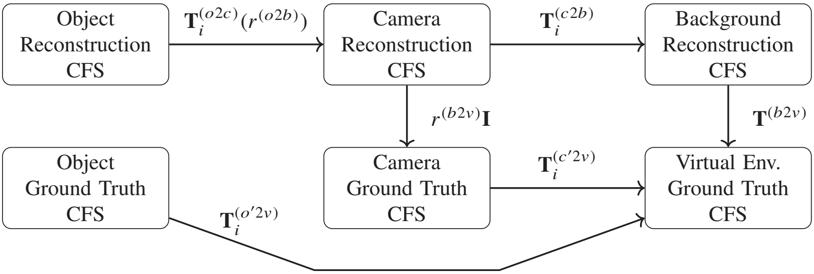
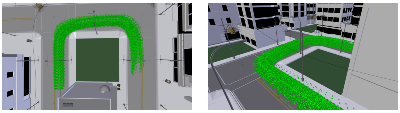


Figure 6.18: Relation of the coordinate frame systems of the reconstructed object and the coordinate frame system of the virtual corresponding virtual model. $\mathbf{T}_i^{(o2c)}$ and $\mathbf{T}_i^{(c2b)}$ are the transformations between the object, camera and background CFS defined in Section 3.5. Here, $r^{(o2b)}$ denotes the scale ratio between object and background reconstruction. Ignoring reconstruction errors the background reconstruction and the virtual environment CFS are related by a similarity transformation $\mathbf{T}^{(b2v)}$. Thus, the CFS of corresponding reconstruction and virtual cameras are (ideally) identical up to the scale ratio $r^{(b2v)}$ between background reconstruction and virtual environment. The object and the camera poses in the virtual environment are defined by $\mathbf{T}_i^{(o'2v)}$ and $\mathbf{T}_i^{(c'2v)}$, respectively.

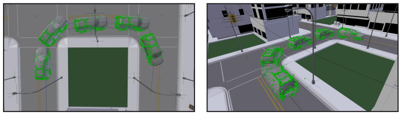
point clouds and the meshes for each frame to quantitatively evaluate the reconstructed trajectories. This requires a registration of the object point cloud w.r.t. the virtual environment.

Fig. 6.18 shows the relation of the different CFSs. The transformations $\mathbf{T}_i^{(o'2v)} \in SE(3)$ between the *object ground truth CFS* and the *virtual environment ground truth CFS* of each time step are part of the virtual dataset. The proposed object trajectory reconstruction methods (*q.v.* Section 6.4.2, Section 6.4.3, Section 6.5.2 and Section 6.5.3) compute the transformations $\mathbf{T}_i^{(c2b)}\mathbf{T}_i^{(o2c)}(r^{(o2b)}) \in SE(3)$ between the object and the background reconstruction for each frame. In order to register the reconstructed trajectories w.r.t. the virtual environment we must determine the similarity transformation $\mathbf{T}^{(b2v)} \in SE(3)$ between the background reconstruction and the virtual environment.

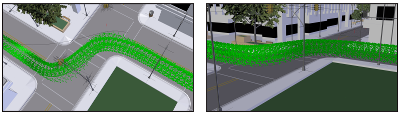
A common approach to register different coordinate systems is to exploit 3D-3D correspondences. To determine points in the virtual environment corresponding to background reconstruction points one could create a set of rays from each camera center to all visible reconstructed background points. The corresponding environment points are defined by the intersection of these rays



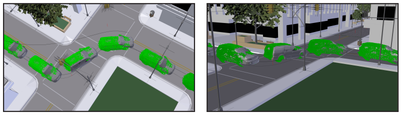
(a) Registered vehicle trajectory in the ground truth CFS.



(b) Vehicle trajectory with ground truth vehicle model at selected frames.



(c) Registered vehicle trajectory in the ground truth CFS.



(d) Vehicle trajectory with ground truth vehicle model at selected frames.

Figure 6.19: Object trajectory registration for quantitative evaluation.

with the mesh of the virtual environment. Due to the complexity of our environment model this computation is in terms of memory and computational effort quite expensive. Instead, we use the algorithm presented in Umeyama (1991) to estimate the similarity transformation $\mathbf{T}^{(b2v)}$ between the cameras contained in the background reconstruction and the virtual cameras used to render the corresponding video sequence. This allows us to perform 3D-3D-registrations of background reconstructions and the virtual environment as well as to quantitatively evaluate the quality of the reconstructed object motion trajectory. We use the camera centers as input for Umeyama (1991) to compute an initial reconstruction-to-virtual-environment transformation. Depending on the shape of the camera trajectory there may be multiple valid similarity transformations using camera center positions. In order to find the semantically correct solution we enhance the original point set with camera pose information, *i.e.*, we add points reflecting up vectors $\mathbf{u}_i^{(b)} = \mathbf{R}_i^{(b)T} \cdot (0,1,0)^T$ and forward vectors $\mathbf{f}_i^{(b)} = \mathbf{R}_i^{(b)T} \cdot (0,0,1)^T$. For the reconstructed cameras, we adjust the magnitude of these vectors using the scale computed during the initial similarity transformation. We add the corresponding end points of up $\mathbf{c}_i^{(b)} + m \cdot \mathbf{u}_i^{(b)}$ as well as viewing vectors $\mathbf{c}_i^{(b)} + m \cdot \mathbf{f}_i^{(b)}$ to the camera center point set. Here, m denotes the corresponding magnitude. Fig. 6.19 shows the results of two trajectory registrations.

6.7.2 Trajectory Reconstruction Metrics

In this work, we use two different metrics to evaluate the quality of reconstructed object trajectories.

Object Trajectory Error

The *Absolute Trajectory Error* proposed by Sturm et al. (2012) is a common evaluation measure to compare an estimated trajectory with the corresponding ground truth trajectory - both represented by a sequence of poses. In our case, the reconstructed object trajectory consists not only of a pose per frame but also of a point cloud describing the object shape. Similarly, the ground truth trajectory consists of several vehicle meshes per sequence. We require an *object* trajectory error that reflects these properties.
For each frame we transform the ground truth object mesh (using $\mathbf{T}_i^{(o'2v)}$) and the reconstructed object point cloud (using $\mathbf{T}_i^{(b2v)}\mathbf{T}_i^{(c2b)}\mathbf{T}_i^{(o2c)}(r^{(o2b)})$) into the virtual environment CFS. We compute the shortest distance of each vehicle trajectory point to the vehicle mesh. For each sequence we define the mean / median *Object Trajectory Error (OTE)* as the mean / median trajectory-point-mesh distance of all frames. We use the *mean absolute deviation (MAD)* of the median OTE to measure the distribution of the triangulated points.

Reference Scale Ratio Deviation

The proposed object trajectory error (OTE) of (monocular) reconstruction methods is subject to four different error sources, *i.e.*, deviations of camera poses w.r.t. vehicle and background point clouds, incorrect triangulated vehicle points as well as scale ratio discrepancies.
To independently evaluate the quality of the estimated scale ratios $r^{(o2b)}$ of the methods in Section 6.4.2 and Section 6.4.3 we compute the deviation of the estimated scale ratio w.r.t. a reference scale ratio. The scale ratios between object reconstruction, background reconstruction and virtual environment are linked via the relation shown in (6.25),

$$r^{(o2v)} = r^{(o2b)} \cdot r^{(b2v)} \Leftrightarrow r^{(o2b)} = r^{(o2v)} \cdot {r^{(b2v)}}^{-1} \tag{6.25}$$

where $r^{(o2v)}$ and $r^{(b2v)}$ are the scale ratios between object and background reconstructions and virtual environment, respectively.
The similarity transformation $\mathbf{T}^{(b2v)}$ (see Fig. 6.18 in Section 6.7.1) implicitly contains a reference value $r_{ref}^{(b2v)}$ for the scale ratio $r^{(b2v)}$ between background reconstruction and virtual environment.
To compute a reference value $r_{ref}^{(o2v)}$ for $r^{(o2v)}$ we use corresponding pairs of object reconstruction and virtual cameras. We use the extrinsic parameters of the object reconstruction camera to transform all 3D points in the object reconstruction into camera coordinates. Similarly, the object mesh with the pose of the corresponding frame is transformed into the camera coordinates leveraging the extrinsic camera parameters of the corresponding virtual camera. The ground truth pose and shape of the object mesh is part of the dataset. In camera coordinates we generate rays from the camera center (*i.e.*, the origin) to each 3D point $\mathbf{o}_j^{(i)}$ in the object reconstruction. We determine the shortest intersection $\mathbf{m}_j^{(i)}$ of each ray with the object mesh in camera coordinates. This allows us to compute $r_{ref}^{(o2v)}$ according to (6.26)

$$r_{ref}^{(o2v)} = \mathrm{med}(\{\mathrm{med}(\{\|\mathbf{m}_j^{(i)}\| \cdot \|\mathbf{o}_j^{(i)}\|^{-1} | j \in \{1, \ldots, n_J\}\}) | i \in \{1, \ldots, n_I\}\}) \tag{6.26}$$

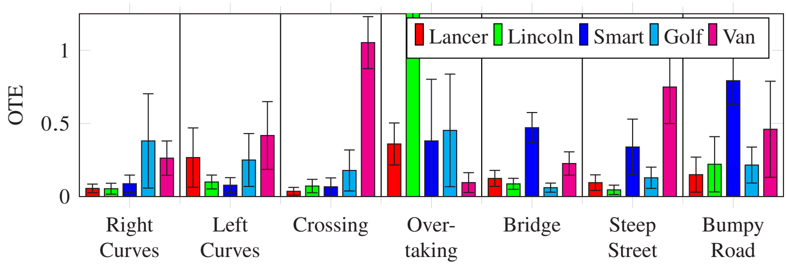
and the *Reference Scale Ratio (RSR)* $r_{ref}^{(o2b)}$ according to (6.27).

$$r_{ref}^{(o2b)} = r_{ref}^{(o2v)} \cdot r_{ref}^{(b2v)^{-1}} \tag{6.27}$$

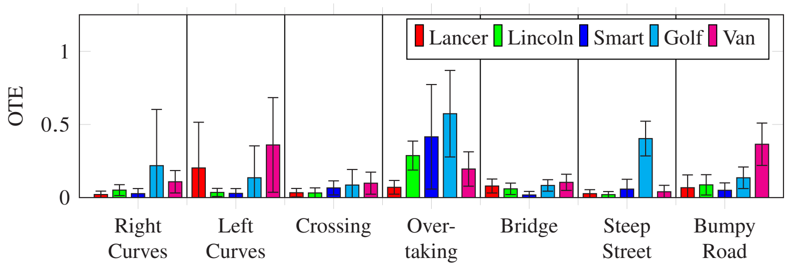
The Reference Scale Ratio $r_{ref}^{(o2b)}$ depends on the quality of the estimated camera poses in the background reconstruction, *i.e.*, $r_{ref}^{(b2v)}$, and may slightly differ from the true scale ratio. We define the *Reference Scale Ratio Deviation (RSRD)* as the deviation of the estimated scale ratios w.r.t. the corresponding reference values $r_{ref}^{(o2b)}$.

6.7.3 Trajectory Evaluation

We use the dataset presented in Section 4.2 to quantitatively evaluate the proposed object motion trajectory reconstruction approaches (*q.v.* Section 6.4.2, Section 6.4.3, Section 6.5.2 and Section 6.5.3). The evaluation is based on vehicle, background and ground segmentations included in the dataset. This al-



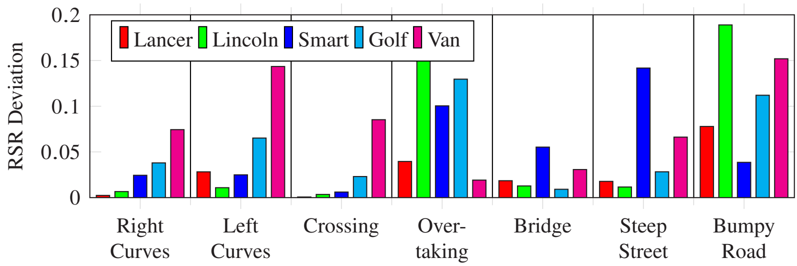
(a) MAD of OTE median of the method introduced in Section 6.4.2.



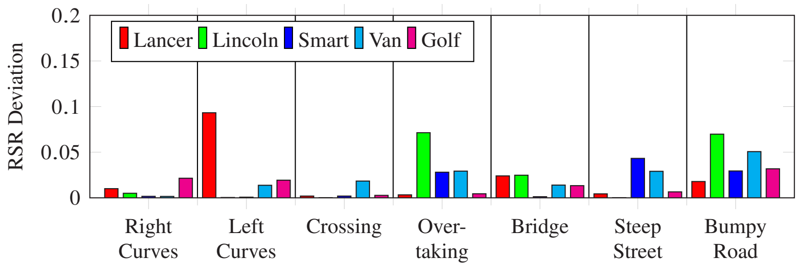
(b) MAD of OTE median of the method introduced in Section 6.4.3.

Figure 6.20: OTE of the *monocular* trajectory reconstruction methods presented in Section 6.4.2 and Section 6.4.3. The intervals show the corresponding standard deviations.

lows us to show results independent from the performance of specific instance segmentation and tracking approaches. The dataset contains seven different vehicle trajectories (*Right Curves*, *Left Curves*, *Crossing*, *Overtaking*, *Bridge*, *Steep Street* and *Bumpy Road*) and five different vehicle models (*Lancer*, *Lincoln*, *Smart*, *Golf* and *Van*). We compare the introduced object trajectory reconstruction algorithms using all 35 sequences contained in the dataset.
We use a fixed camera calibration model with known focal length, principal point and radial distortion to compute the object and background reconstructions. We automatically register the reconstructed vehicle trajectories to the ground truth using the method described in Section 6.7.1.
Fig. 6.20 shows the OTE (*q.v.* Section 6.7.2) in meter for the *monocular* trajectory reconstruction algorithms presented in Section 6.4.2 and Section 6.4.3. The results are itemized for each trajectory and vehicle type. Both methods use the same object and background reconstructions to improve comparability.



(a) RSRD of the method introduced in Section 6.4.2.



(b) RSRD of the method introduced in Section 6.4.3.

Figure 6.21: RSRD of the *monocular* trajectory reconstruction methods presented in Section 6.4.2 and Section 6.4.3.

The mean (ground truth) distance of the camera to the vehicle in the dataset is as follows: *Right Curves*: 17.37 m, *Left Curves*: 13.10 m, *Crossing*: 18.91 m, *Overtaking*: 15.90 m, *Bridge*: 12.71 m, *Steep Street*: 17.30 m, *Bumpy Road*: 21.81 m.

We observe that in some cases (*Right Curves*: *Golf*; *Left Curves*: *Lancer*, *Van* and *Golf*; *Steep Street*: *Smart*; *Bumpy Road*: *Lancer*, *Lincoln*, *Smart* and *Golf*) the *object* reconstructions are *mirror-inverted*. This is caused by nearly affine object views, *i.e.*, the observed object depth values show a small variation compared to the object-camera-distance. In this case, the projection of the actual three-dimensional object shape is (almost) identical to an object shape reflected at a plane parallel to the image plane (Ozden et al., 2010). Such situations can lead to two different reconstructions. The incorrect result usually causes a high OTE.

Method	Mean OTE in meter					
	Lancer	Lincoln	Smart	Van	Golf	Overall
Section 6.4.2 (monocular)	0.16	0.28	0.32	0.24	0.47	0.29
Section 6.4.3 (monocular)	0.11	0.09	0.14	0.21	0.30	0.17
Section 6.5.2 (stereo)	0.06	**0.06**	0.07	0.10	0.27	0.11
Section 6.5.3 (stereo)	**0.05**	0.13	**0.06**	**0.09**	**0.13**	**0.09**

Table 6.1: Mean OTE per vehicle using the introduced vehicle trajectory benchmark dataset. The best (stereo) approach achieves an average OTE of 0.09 m considering all sequences and outperforms the monocular methods.

OTE values of the *monocular* trajectory reconstructions reflects four types of computational inaccuracies: deviations of camera poses w.r.t. vehicle and background point clouds, incorrect triangulated vehicle points as well as scale ratio discrepancies. Therefore, Fig. 6.21 compares the estimated scale ratios of both methods w.r.t. the RSR. The RSR computation is described in Section 6.7.2. The provided RSRs are subject to the registration described in Section 6.7.1. Incorrectly reconstructed background camera poses may influence the RSR. The RSR values allow to independently evaluate the scale ratio estimation constraints.

The average OTE of both methods is shown per vehicle and per trajectory in Table 6.1 and Table 6.2. The approach proposed in Section 6.4.3 computes lower OTEs than the algorithm described in Section 6.4.2. Fig. A.3 and Fig. A.4 in the appendix show trajectory reconstructions for individual frames of a monocular image sequence using the method proposed in Section 6.4.2.

Fig. 6.22 shows OTE values for the *stereo* trajectory reconstruction methods presented in Section 6.5.2 and Section 6.5.3. We observe that the stereo methods outperform the monocular approaches. In case of the algorithm presented in Section 6.5.3 there are only four cases (*Right Curves*: *Lincoln*; *Left Curves*: *Van*; *Overtaking*: *Lincoln*; *Bumpy Road*: *Lincoln*) with *mirror-inverted* vehicle reconstructions. The average OTE per vehicle and per trajectory using the full dataset is shown in Table 6.1 and Table 6.2.

Method	Mean OTE in meter							
	Right Curves	Left Curves	Crossing	Over-taking	Bridge	Steep Street	Bumpy Road	Overall
Section 6.4.2	0.17	0.22	0.28	0.54	0.19	0.27	0.37	0.29
Section 6.4.3	0.15	0.25	0.09	0.34	0.08	0.12	0.16	0.17
Section 6.5.2	**0.11**	**0.10**	0.12	0.10	0.09	0.10	0.16	0.11
Section 6.5.3	0.13	0.11	**0.08**	**0.09**	**0.05**	**0.09**	**0.09**	**0.09**

Table 6.2: Mean OTE per trajectory using the introduced vehicle trajectory benchmark dataset. The mean camera-vehicle-distance in the ground truth data is as follows: Right Curves: 17.37 m, Left Curves: 13.10 m, Crossing: 18.91 m, Overtaking: 15.90 m, Bridge: 12.71 m, Steep Street: 17.30 m, Bumpy Road: 21.81 m.

6.8 Discussion

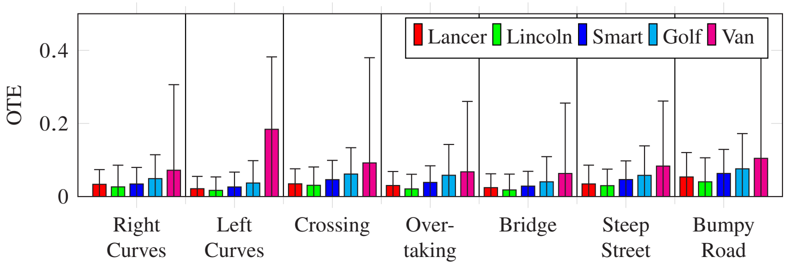
This chapter focuses on image-based reconstructions of three-dimensional object trajectories. We use transformations between object, camera and background CFSs of the MSfM reconstruction to derive a mathematical representation of an object trajectory that reflects the corresponding scale ambiguity. The trajectory is represented by a one-parameter family of possible solutions. We show that possible object trajectories are superpositions of the camera and the true object trajectory. Object and background reconstructions with incorrect scale ratios lead to object trajectories with incorrect shapes.

We propose four methods for object trajectory reconstruction leveraging the instance-aware MSfM approach introduced in Chapter 3.

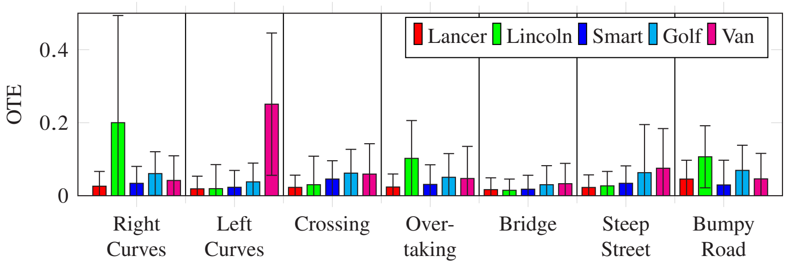
The methods in Section 6.4.2 and Section 6.4.3 exploit geometric relations of object points and environment structures to determine consistent three-dimensional vehicle trajectories in *monocular* image sequences.

The algorithm described in Section 6.4.2 uses a distance constraint of object and environment points to compute consistent three-dimensional object motions. As existing constraints, the approach shows a degenerated reconstruction case. Our constraint represents a generalization of Yuan and Medioni (2006) and the degenerated reconstruction cases are disjoint from the approaches presented in Ozden et al. (2004), Kundu et al. (2011) and Namdev et al. (2013).

The algorithm in Section 6.4.3 avoids degenerated reconstruction cases by leveraging projection constraints of object points and the environment sur-



(a) MAD of OTE median of the method presented in Section 6.5.2.



(b) MAD of OTE median of the method presented in Section 6.5.3.

Figure 6.22: OTE of the *stereo* trajectory reconstruction methods presented in Section 6.5.2 and Section 6.5.3. The intervals show the corresponding standard deviations.

face. This approach is computational more expensive, since it triangulates a dense object point cloud and computes a watertight mesh of the terrain.
Section 6.5.2 and 6.5.3 propose two algorithms for object trajectory reconstruction using *stereo* image sequences. In the case of stereo image data, the scale ambiguity of object and background reconstructions can be resolved using the baseline of the stereo camera. The method in Section 6.5.2 uses stereo matching to triangulate object points and is therefore not hampered by incorrectly registered camera poses caused by small object sizes, reflecting surfaces and changing illumination. At the same time stereo matching based point triangulations are limited by the baseline of the stereo camera and do not allow to reconstruct an object model, *i.e.*, no relative object-camera poses. The algorithm in Section 6.5.3 overcomes the baseline limitation leveraging SfM for object reconstruction. The method uses stereo projection constraints to determine stable camera poses w.r.t. the object reconstruction.

We observe in our evaluation that the proposed stereo methods achieve lower OTEs than the introduced monocular approaches (*q.v.* Table 6.1 and Table 6.2). However, in some scenarios such as (monocular) Internet video data it is only possible to apply monocular reconstructions methods. The different trajectory reconstruction accuracies between monocular and stereo based methods are partly caused by incorrect scale ratio estimations between object and background reconstruction (*q.v.* Fig. 6.21). As discussed in Özden (2007) motion constraints for the estimation of the scale ratio are category and situation-specific and therefore limited to certain scenarios. Another approach to deal with the scale ambiguity of monocular reconstruction results are category-specific scale priors. For instance, a database with geometric properties of vehicle models and typical environment assets such as street signs or traffic lights could allow to determine consistent scales. For future work, we intend to integrate such constraints in our reconstruction pipeline to increase the robustness of the scale ratio estimation.

In general, it is reasonable to use the stereo trajectory reconstruction algorithms whenever possible. Both presented stereo based methods show different advantages. The approach in Section 6.5.2 is not hampered by cameras in the object reconstruction with incorrect baselines whereas the method in Section 6.5.3 leverages information of temporal adjacent frames. For future work, we intend to combine both approaches (*i.e.*, the usage of (known) stereo camera poses to determine feature correspondences) to improve the reconstruction results.

7 Conclusion

7.1 Summary

This work tackles the problem of reconstructing dynamic objects in real-world environments by introducing a novel semantic segmentation based Multibody Structure from Motion approach.
In Chapter 2 we extensively review previously published literature covering image-based methods for three-dimensional scene reconstruction from multiple images. We describe the major building blocks of modern Structure from Motion pipelines. The thesis emphasizes corresponding key concepts that allow to reconstruct scene structures from feature correspondences alone. By analyzing epipolar geometry properties we identify limitations of classical Structure from Motion when applied to the reconstruction of dynamic environments.
To reconstruct scenes with multiple components (*i.e.*, several moving objects in a static environment), we develop a novel Multibody Structure from Motion algorithm in Chapter 3. Our approach exploits instance-aware semantic segmentation and optical flow methods to track objects on pixel level throughout video sequences, which allows to determine consistently moving groups of visual features. Our method is able to compute MSfM reconstructions from monocular as well as stereo image sequences. An evaluation of the Multiple Object Tracking algorithm on a publicly available benchmark dataset shows competitive results.
In Chapter 4 we present two datasets to quantitatively evaluate moving object reconstructions. The first dataset comprises real-world image sequences of a moving vehicle and a corresponding vehicle laser scan suitable for evaluation of three-dimensional object shape reconstructions. The second dataset contains synthetic sequences of different vehicles in an urban environment. We provide vehicle shapes as well as vehicle and camera poses for each frame as ground truth. This dataset allows the quantitative evaluation of shape and

trajectory reconstructions of moving objects. We made both datasets and corresponding evaluation scripts publicly available to foster future analysis of moving object reconstructions.
In Chapter 5 we leverage the proposed MSfM approach to reconstruct textured object models. We combine sparse object reconstructions (*i.e.*, camera poses and triangulated object points) with two-dimensional object segmentations to derive three-dimensional object meshes consistent to image observations. We show that our approach produces meshes that are robust w.r.t. to reflections and appearance changes.
Chapter 6 focuses on the reconstruction of three-dimensional object trajectories in monocular and stereo image sequences using the proposed MSfM algorithm. We observe that the coordinate frame systems of object and background reconstructions are geometrically related by commonly registered images. The corresponding transformations allow to formulate a trajectory representation that reflects the scale ambiguity of Structure from Motion results, *i.e.*, we define potential object trajectories as a one-parameter family of possible solutions. In this formulation we observe that each potential object trajectory can be considered as a superposition of the camera and the corresponding true object trajectory. Therefore, the scale ratio between object and background reconstruction does not only change the extent but also the shape of the object trajectory.
For monocular image sequences we resolve this ambiguity by introducing two different vehicle motion constraints to estimate the scale ratio between object and environment reconstructions. Both constraints exploit geometric relations of object points and environment structures to determine consistent vehicle motion trajectories. For stereo image sequences we exploit the baseline of the stereo camera to determine the true (unique) extent of the object and the environment reconstruction. We leverage stereo projection constraints to compute consistent stereo camera baselines. The quantitative evaluation of the trajectory reconstruction algorithms shows that inconsistent scale ratio estimations lead to significant trajectory errors. The usage of stereo image sequences results in more accurate and robust reconstructions.

7.2 Discussion and Future Work

To the best of our knowledge, this work proposes the first instance-aware semantic segmentation based MSfM approach. We are convinced that semantic information offers a superior alternative to existing techniques that allow to compute object specific feature correspondence such as epipolar constraints or motion segmentation. The usage of semantic information improves the robustness of object-specific feature correspondence computations and inherently assigns category information to the obtained reconstructions. Robustness of dynamic reconstructions and corresponding semantic annotations are crucial properties for many applications such as environment perception for autonomous driving.

We focus in this work on SfM instead of Visual SLAM, since it simplifies the reconstruction problem, *i.e.*, SfM is more robust (*e.g.*, w.r.t. scale drift) and requires less video specific parameter adjustments.

The proposed MSfM approach includes an online MOT algorithm that allows to track objects on pixel level. The majority of existing MOT methods only allow to reliably track objects on bounding box level. Thus, our approach enables new applications such as the direct determination of object specific disparity values.

This work considers two applications of the proposed MSfM approach, *i.e.*, three-dimensional shape and trajectory reconstruction of dynamic objects. Related works do not show quantitative evaluations of their algorithms, because of the lack of publicly available implementations and benchmark datasets for moving object reconstruction. To improve the comparability of dynamic object reconstruction methods we make our datasets and associated evaluation scripts publicly available.

Given suitable camera-object-poses, the object shape reconstruction algorithm shows that semantic projection constraints allow to determine textured meshes consistent to image observation for driving vehicles in real-world scenarios. Such results are difficult to achieve with modern Multi-View Stereo algorithms, since shadows, reflecting surfaces and illumination changes cause frequently object points with inconsistent normal vectors. Our results suggest to integrate semantic projection constraints into existing Multi-View Stereo algorithms.

The evaluation of the trajectory reconstruction algorithms confirms the expectation that the proposed stereo approaches outperform the presented monocu-

lar methods. In general, it is therefore reasonable to integrate one of the stereo methods into applications that rely on the reconstruction of object motion trajectories. However, in certain application domains such as (monocular) Internet video data it is only possible to apply monocular reconstructions methods. The different trajectory reconstruction accuracies observed in our experiments are partly caused by the estimation of the scale ratio between object and background reconstruction. Object motion constraints are category specific and only apply for certain situations. Therefore, it is reasonable to compare in future work the proposed constraints with other non-motion methods such as category specific scale priors. For example, the usage of a database with geometric properties of different vehicle models and typical environment assets such as street signs or traffic lights could allow to tackle the scale ambiguity.
Recently, different algorithms have been proposed to learn class-specific shape priors from single and multiple images. Such methods represent a reasonable alternative to correspondence-based object reconstruction techniques. Learning unconstrained (dynamic) environment representations with methods based on deep learning is currently infeasible, because the high complexity of real-world scenes can not be represented with a reasonable number of model parameters. We consider the combination of MSfM and object reconstruction approaches based on deep learning as an important step towards robust systems for the reconstruction of dynamic objects in challenging real world scenarios.
The topic of this thesis has been originally motivated by the following research question: *Does semantic segmentation based Multibody Structure from Motion allow to accurately reconstruct real-world scenarios of moving objects?* We have shown that the combination of semantic information and modern image-based reconstruction techniques allows to compute consistent MSfM models of real world scenes. The computation of accurate MSfM reconstructions must deal with several difficult problems such as the scale ambiguity of monocular MSfM reconstructions. Additional investigations are required to solve these issues - possibly based on the approaches mentioned above.

Own Publications

C. Bodensteiner, S. Bullinger, S. Lemaire, and M. Arens. Single frame based video geo-localisation using structure projection. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015. URL `https://doi.org/10.1109/ICCVW.2015.136`.

C. Bodensteiner, S. Bullinger, and M. Arens. Multispectral matching using conditional generative appearance modeling. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018. URL `https://doi.org/10.1109/AVSS.2018.8639403`.

S. Bullinger, C. Bodensteiner, S. Wuttke, and M. Arens. Moving object reconstruction in monocular video data using boundary generation. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2016. URL `https://doi.org/10.1109/ICPR.2016.7899640`.

S. Bullinger, C. Bodensteiner, and M. Arens. Instance flow based online multiple object tracking. In *IEEE International Conference on Image Processing (ICIP)*, 2017. URL `https://doi.org/10.1109/ICIP.2017.8296388`.

S. Bullinger, C. Bodensteiner, and M. Arens. Monocular 3D vehicle trajectory reconstruction using terrain shape constraints. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018a. URL `https://doi.org/10.1109/ITSC.2018.8569508`.

S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen. 3D vehicle trajectory reconstruction in monocular video data using environment structure constraints. In *European Conference on Computer Vision (ECCV)*, 2018b. URL `https://doi.org/10.1007/978-3-030-01249-6_3`.

S. Bullinger, C. Bodensteiner, and M. Arens. 3D object trajectory reconstruction using stereo matching and instance flow based multiple object tracking. In *IAPR International Conference on Machine Vision Applications (MVA)*, 2019a. URL `https://doi.org/10.23919/MVA.2019.8757921`.

S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen. 3D object trajectory reconstruction using instance-aware multibody structure from motion and stereo sequence constraints. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019b. URL `https://doi.org/10.1109/IVS.2019.8814118`.

References

Robust Vision Challenge, 2018. URL http://www.robustvision.net/. [Accessed December 2018].

G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1985. URL https://doi.org/10.1109/TPAMI.1985.4767678.

S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. URL https://doi.org/10.1109/ICCV.2009.5459148.

Agisoft. Agisoft Metashape, 2019. URL https://www.agisoft.com/. [Accessed August 2019].

C. Aholt, S. Agarwal, and R. Thomas. A QCQP approach to triangulation. In *European Conference on Computer Vision (ECCV)*, 2012. URL https://doi.org/10.1007/978-3-642-33718-5_47.

P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *IEEE European Conference on Computer Vision (ECCV)*, 2012. URL http://doi.org/10.1007/978-3-642-33783-3_16.

M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision (ECCV)*, 2016. URL https://doi.org/10.1007/978-3-319-46466-4_10.

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 2011. URL https://doi.org/10.1007/s11263-010-0390-2.

S. Y. Bao and S. Savarese. Semantic structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. URL `https://doi.org/10.1109/CVPR.2011.5995462`.

H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, 2006. URL `https://doi.org/10.1007/11744023_32`.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 2008. URL `http://dx.doi.org/10.1016/j.cviu.2007.09.014`.

C. Beder and R. Steffen. Determining an initial image pair for fixing the scale of a 3D reconstruction from an image sequence. In *Pattern Recognition*, 2006. URL `https://doi.org/10.1007/11861898_66`.

K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008. URL `https://doi.org/10.1155/2008/246309`.

A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing (ICIP)*, 2016. URL `https://doi.org/10.1109/ICIP.2016.7533003`.

Blender Foundation. Blender - a 3D modelling and rendering package, 2019. URL `http://www.blender.org`. [Accessed August 2019].

M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo - joint stereo matching and object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. URL `https://doi.org/10.1109/CVPR.2011.5995581`.

M. Born, E. Wolf, A. B. Bhatia, P. C. Clemmow, D. Gabor, A. R. Stokes, A. M. Taylor, P. A. Wayman, and W. L. Wilcock. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. 1999. URL `https://doi.org/10.1017/CBO9781139644181`.

G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009. URL `http://doi.org/10.1016/j.patrec.2008.04.005`.

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012. URL https://doi.org/10.1007/978-3-642-33783-3_44.

Capturing Reality. RealityCapture, 2019. URL https://www.capturingreality.com/. [Accessed August 2019].

C. Chang and S. Chatterjee. Quantization error analysis in stereo vision. In *IEEE Asilomar Conference on Signals, Systems Computers*, 1992. URL https://doi.org/10.1109/ACSSC.1992.269140.

J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://doi.org/10.1109/CVPR.2018.00567.

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *CoRR*, 2014. URL http://arxiv.org/abs/1412.7062.

X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems (NIPS)*. 2015. URL https://papers.nips.cc/paper/5644-3d-object-proposals-for-accurate-object-class-detection.

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1991. URL https://doi.org/10.1109/ROBOT.1991.132043.

F. Chhaya, N. D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna. Monocular reconstruction of vehicles: Combining SLAM with shape priors. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016. URL http://doi.org/10.1109/ICRA.2016.7487799.

M. Coenen, F. Rottensteiner, and C. Heipke. Recovering the 3D pose and shape of vehicles from stereo images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018. URL https://doi.org/10.5194/isprs-annals-IV-2-73-2018.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.350`.

D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. URL `https://doi.org/10.1109/CVPR.2011.5995626`.

F. Daellert. Factor graphs and GTSAM: A hands-on introduction. Technical report, GT-RIM-CP&R-2012-002, 2012. URL `http://hdl.handle.net/1853/45226`.

J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.343`.

C. Debrunner and N. Ahuja. Motion and structure factorization and segmentation of long multiple motion image sequences. In *European Conference on Computer Vision (ECCV)*, 1992. URL `https://doi.org/10.1007/3-540-55426-2_24`.

C. Debrunner and N. Ahuja. Segmentation and factorization-based motion and structure estimation for long image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1998. URL `https://doi.org/10.1109/34.659941`.

F. Dellaert and M. Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research (IJRR)*, 2006. URL `https://doi.org/10.1177/0278364906072768`.

F. Dellaert and M. Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 2017. URL `http://doi.org/10.1561/2300000043`.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15.

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Annual Conference on Robot Learning*, 2017. URL http://proceedings.mlr.press/v78/dosovitskiy17a.html.

F. Engelmann, J. Stückler, and B. Leibe. SAMP: shape and motion priors for 4d vehicle reconstruction. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017. URL https://doi.org/10.1109/WACV.2017.51.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 2010. URL https://doi.org/10.1007/s11263-009-0275-4.

C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. URL https://doi.org/10.1109/TITS.2017.2726546.

G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis (SCIA)*, 2003. URL http://dl.acm.org/citation.cfm?id=1763974.1764031.

Y. Feng, Y. Wu, and L. Fan. On-line object reconstruction and tracking for 3D interaction. In *International Conference on Multimedia and Expo (ICME)*, 2012. URL https://doi.org/10.1109/ICME.2012.144.

M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. URL http://doi.acm.org/10.1145/358669.358692.

A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-D reconstruction of independently moving objects. In *European Conference on Computer Vision (ECCV)*, 2000. URL https://doi.org/10.1007/3-540-45054-8_58.

J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, 2010. URL `http://dl.acm.org/citation.cfm?id=1888089.1888117`.

A. Frieze. Complexity of a 3-dimensional assignment problem. *European Journal of Operational Research*, 1983. URL `https://doi.org/10.1016/0377-2217(83)90078-4`.

S. Fuhrmann and M. Goesele. Floating scale surface reconstruction. *ACM Transactions on Graphics (TOG)*, 2014. URL `http://doi.acm.org/10.1145/2601097.2601163`.

S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele. MVE – an image-based reconstruction environment. *Computer and Graphics*, 2015. URL `https://doi.org/10.1016/j.cag.2015.09.003`.

Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. URL `https://doi.org/10.1109/TPAMI.2009.161`.

A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.470`.

X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2003. URL `https://doi.org/10.1109/TPAMI.2003.1217599`.

C. Gear. Multibody grouping from motion images. *International Journal of Computer Vision (IJCV)*, 1998. URL `https://doi.org/10.1023/A:1008026310903`.

A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010. URL `https://doi.org/10.1007/978-3-642-19315-6_3`.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. URL https://doi.org/10.1177/0278364913491297.

J. J. Gibson. *The perception of the visual world*. Houghton Mifflin, 1950. URL https://nla.gov.au/nla.cat-vn2273892.

K. C. Gilbert and R. B. Hofstra. Multidimensional assignment problems. *Decision Sciences*, 1988. URL https://doi.org/10.1111/j.1540-5915.1988.tb00269.x.

D. Girardeau-Montaut. Cloudcompare, 2016. URL http://www.cloudcompare.org/. [Accessed August 2019].

M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. URL https://doi.org/10.1109/ICCV.2007.4408933.

M. Grinberg. *Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking*. KIT Scientific Publishing, 2018. URL http://doi.org/10.5445/KSP/1000081430.

R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision (ECCV)*, 1992. URL http://dl.acm.org/citation.cfm?id=645305.648678.

R. I. Hartley. Lines and point in three views - an integrated approach. In *ARPA Image Understanding Workshop*, 1994. URL http://citeseer.ist.psu.edu/viewdoc/citations;jsessionid=D139BFA2E3A244720B79D02660176765?doi=10.1.1.2.5917.

R. I. Hartley and P. Sturm. Triangulation. In *Computer Analysis of Images and Patterns (CAIP)*, 1995. URL https://doi.org/10.1007/3-540-60268-2_296.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. URL https://doi.org/10.1017/CBO9780511811685.

M. Havlena and K. Schindler. Vocmatch: Efficient multiview correspondence for structure from motion. In *European Conference on Computer Vision (ECCV)*, 2014. URL `https://doi.org/10.1007/978-3-319-10578-9_4`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.90`.

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. URL `https://doi.org/10.1109/ICCV.2017.322`.

J. Heinly, J. L. Schönberger, E. Dunn, and J. Frahm. Reconstructing the world* in six days. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `https://doi.org/10.1109/CVPR.2015.7298949`.

H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008. URL `http://doi.org/10.1109/TPAMI.2007.1166`.

Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.615`.

X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The ApolloScape dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. URL `https://doi.org/10.1109/CVPRW.2018.00141`.

E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL `http://lmb.informatik.uni-freiburg.de/Publications/2017/IMKDB17`.

M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. URL `https://doi.org/10.1109/cvpr.2011.5995693`.

Jianbo Shi and Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994. URL `https://doi.org/10.1109/CVPR.1994.323794`.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics (TRO)*, 2008. URL `https://doi.org/10.1109/TRO.2008.2006706`.

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. URL `https://doi.org/10.1109/ICRA.2011.5979641`.

L. Kang, L. Wu, and Y.-H. Yang. Robust multi-view L2 triangulation via optimal inlier selection and 3D structure refinement. *Pattern Recognition*, 2014. URL `http://www.sciencedirect.com/science/article/pii/S0031320314001204`.

M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 2013. URL `http://doi.acm.org/10.1145/2487228.2487237`.

M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, 2006. URL `http://dl.acm.org/citation.cfm?id=1281957.1281965`.

A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *CoRR*, 2018. URL `http://arxiv.org/abs/1801.00868`.

A. Kirillov, R. B. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. *CoRR*, 2019. URL `http://arxiv.org/abs/1901.02446`.

F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001. URL `https://doi.org/10.1109/18.910572`.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955. URL `https://doi.org/10.1007/978-3-540-68279-0_2`.

S. Kumar, Y. Dai, and H. Li. Multi-body non-rigid structure-from-motion. In *International Conference on 3D Vision (3DV)*, 2016. URL https://doi.org/10.1109/3DV.2016.23.

A. Kundu, K. M. Krishna, and J. Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009. URL https://doi.org/10.1109/IROS.2009.5354227.

A. Kundu, K. M. Krishna, and C. V. Jawahar. Realtime multibody visual SLAM with a smoothly moving monocular camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. URL https://doi.org/10.1109/ICCV.2011.6126482.

A. Kundu, Y. Li, and J. M. Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://doi.org/10.1109/CVPR.2018.00375.

L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision (IJCV)*, 2012. URL https://doi.org/10.1007/s11263-011-0489-0.

L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*, 2015. URL http://arxiv.org/abs/1504.01942.

K. Lebeda, S. Hadfield, and R. Bowden. 2D or not 2D: Bridging the gap between tracking and structure from motion. In *Asian Conference on Computer Vision (ACCV)*, 2014. URL http://doi.org/10.1007/978-3-319-16817-3_42.

K. Lebeda, S. Hadfield, and R. Bowden. Dense rigid reconstruction from unstructured discontinuous video. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2015. URL https://doi.org/10.1109/ICCVW.2015.110.

B. Lee, K. Daniilidis, and D. D. Lee. Online self-supervised monocular visual odometry for ground vehicles. In *IEEE International Conference on*

Robotics and Automation (ICRA), 2015. URL https://doi.org/10.1109/ICRA.2015.7139928.

V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 2008. URL https://doi.org/10.1007/s11263-008-0152-6.

S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. URL http://doi.org/10.1109/ICCV.2011.6126542.

K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 1944. URL https://doi.org/10.1090%2Fqam%2F10666.

Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision (ECCV)*, 2010. URL https://doi.org/10.1007/978-3-642-15552-9_57.

Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://doi.org/10.1109/CVPR.2017.472.

Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang. Learning for disparity estimation through feature constancy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://doi.org/10.1109/CVPR.2018.00297.

G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://doi.org/10.1109/CVPR.2016.348.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. URL https://doi.org/10.1007/978-3-319-10602-1_48.

T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL `https://doi.org/10.1109/CVPR.2017.106`.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `https://doi.org/10.1109/TPAMI.2016.2572683`.

H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. URL `https://doi.org/10.1038/293133a0`.

Y. Lou, N. Snavely, and J. Gehrke. Matchminer: Efficient spanning structure mining in large image collections. In *European Conference on Computer Vision (ECCV)*, 2012. URL `https://doi.org/10.1007/978-3-642-33709-3_4`.

D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, 1999. URL `http://dl.acm.org/citation.cfm?id=850924.851523`.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004. URL `http://doi.org/10.1023/B:VISI.0000029664.99615.94`.

F. Lu and R. Hartley. A fast optimal algorithm for L2 triangulation. In *Asian Conference on Computer Vision (ACCV)*, 2007. URL `https://doi.org/10.1007/978-3-540-76390-1_28`.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981. URL `http://dl.acm.org/citation.cfm?id=1623264.1623280`.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://doi.org/10.1109/CVPR.2016.438`.

P. Moulon. Sceaux Castle dataset, 2012. URL `https://github.com/openMVG/ImageDataset_SceauxCastle`. [Accessed August 2019].

P. Moulon, P. Monasse, and R. Marlet. Adaptive structure from motion with a contrario model estimation. In *Asian Conference on Computer Vision (ACCV)*, 2012. URL `http://doi.org/10.1007/978-3-642-37447-0_20`.

P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. URL `https://doi.org/10.1109/ICCV.2013.403`.

R. K. Namdev, K. M. Krishna, and C. V. Jawahar. Multibody VSLAM with relative scale solution for curvilinear motion reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013. URL `https://doi.org/10.1109/ICRA.2013.6631401`.

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. URL `http://doi.org/10.1109/CVPR.2006.264`.

A. Ošep, W. Mehner, M. Mathias, and B. Leibe. Combined image- and world-space tracking in traffic scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. URL `https://doi.org/10.1109/ICRA.2017.7989230`.

K. E. Ozden, K. Cornelis, L. V. Eycken, and L. J. V. Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *Computer Vision and Image Understanding (CVIU)*, 2004. URL `https://doi.org/10.1016/j.cviu.2004.03.015`.

K. E. Ozden, K. Schindler, and L. V. Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. URL `https://doi.org/10.1109/TPAMI.2010.23`.

H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D trajectory reconstruction under perspective projection. *International Journal of Computer Vision (IJCV)*, 2015. URL `https://doi.org/10.1007/s11263-015-0804-2`.

W. P. Pierskalla. Erratum: The multidimensional assignment problem. *Operations Research*, 1969. URL http://www.jstor.org/stable/168841.

P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester. Know your limits: Accuracy of long range stereoscopic object measurements in practice. In *European Conference on Computer Vision (ECCV)*, 2014. URL https://doi.org/10.1007/978-3-319-10605-2_7.

Pix4D. Pix4Dmapper, 2019. URL https://www.pix4d.com. [Accessed August 2019].

R. Raguram, C. Wu, J.-M. Frahm, and S. Lazebnik. Modeling and recognition of landmark image collections using iconic scene graphs. *International Journal of Computer Vision (IJCV)*, 2011. URL https://doi.org/10.1007/s11263-011-0445-z.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *International Conference on Neural Information Processing Systems (NIPS)*, 2015. URL http://dl.acm.org/citation.cfm?id=2969239.2969250.

J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015. URL https://doi.org/10.1109/CVPR.2015.7298720.

J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision (IJCV)*, 2016. URL http://doi.org/10.1007/s11263-016-0908-3.

H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Rezatofighi_Generalized_Intersection_Over_Union_A_Metric_and_a_Loss_for_CVPR_2019_paper.html.

S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*

(ECCV), 2016. URL https://doi.org/10.1007/978-3-319-46475-6_7.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://doi.org/10.1109/CVPR.2016.352.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, 2006. URL http://doi.org/10.1007/11744023_34.

C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 2004. URL http://doi.acm.org/10.1145/1015706.1015720.

C. Rubino, M. Crocco, V. Murino, and A. D. Bue. Semantic multi-body motion segmentation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. URL https://doi.org/10.1109/WACV.2015.157.

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. URL http://doi.org/10.1109/ICCV.2011.6126544.

M. Ruf. OCTANE - a flexible simulation platform for automotive applications, 2018. URL https://octane.org. [Accessed August 2019].

C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *European Conference on Computer Vision (ECCV)*, 2014. URL https://doi.org/10.1007/978-3-319-10584-0_38.

R. Sabzevari and D. Scaramuzza. Multi-body motion estimation from monocular vehicle-mounted cameras. *IEEE Transactions on Robotics (TRO)*, 2016. URL https://doi.org/10.1109/TRO.2016.2552548.

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 2002. URL https://doi.org/10.1109/SMBV.2001.988771.

J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://doi.org/10.1109/CVPR.2016.445.

J. L. Schönberger, A. C. Berg, and J. Frahm. PAIGE: Pairwise image geometry encoding for improved efficiency in structure-from-motion. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015a. URL https://doi.org/10.1109/CVPR.2015.7298703.

J. L. Schönberger, F. Radenović, O. Chum, and J. Frahm. From single image query to detailed 3D reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015b. URL https://doi.org/10.1109/CVPR.2015.7299148.

J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. URL https://doi.org/10.1007/978-3-319-46487-9_31.

J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://doi.org/10.1109/CVPR.2017.736.

W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit*. 2006. URL https://www.kitware.com/products/books/VTKTextbook.pdf.

L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://doi.org/10.1109/CVPR.2016.422.

S. Shah, D. Dey, C. Lovett, and A. Kapoor. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. URL https://doi.org/10.1007/978-3-319-67361-5_40.

A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *IEEE International Conference on Computer Vision (ICCV)*, 1995. URL `https://doi.org/10.1109/ICCV.1995.466837`.

E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. URL `http://doi.org/10.1109/TPAMI.2016.2572683`.

E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL `http://doi.org/10.1109/ICCV.2015.22`.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. URL `https://arxiv.org/abs/1409.1556`.

K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014. URL `https://doi.org/10.1109/TPAMI.2014.2301163`.

N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 2006. URL `http://doi.acm.org/10.1145/1141911.1141964`.

S. Song and M. Chandraker. Joint SFM and detection cues for monocular 3D localization in road scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `https://doi.org/10.1109/CVPR.2015.7298997`.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE International Conference on Intelligent Robot Systems (IROS)*, 2012. URL `https://doi.org/10.1109/IROS.2012.6385773`.

D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL `https://doi.org/10.1109/CVPR.2018.00931`.

I. E. Sutherland. Three-dimensional data input by tablet. *Proceedings of the IEEE*, 1974. URL https://doi.org/10.1109/PROC.1974.9449.

C. Sweeney, T. Sattler, T. Höllerer, M. Turk, and M. Pollefeys. Optimizing the viewing graph for structure-from-motion. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL https://doi.org/10.1109/ICCV.2015.98.

R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 1994. URL https://doi.org/10.1006/jvci.1994.1002.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision (IJCV)*, 1992. URL https://doi.org/10.1007/BF00129684.

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice*, 2000. URL https://doi.org/10.1007/3-540-44480-7_21.

A. Tsirikoglou, J. Kronander, M. Wrenninge, and J. Unger. Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *CoRR*, 2017. URL http://arxiv.org/abs/1710.06270.

S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 1979. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1979.0006.

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1991. URL https://doi.org/10.1109/34.88573.

D. P. Vassileios Balntas, Edgar Riba and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference (BMVC)*, 2016. URL https://doi.org/10.5244/C.30.119.

M. Waechter, N. Moehrle, and M. Goesele. Let there be color! Large-scale texturing of 3D reconstructions. In *European Conference on Computer Vision (ECCV)*, 2014. URL `https://doi.org/10.1007/978-3-319-10602-1_54`.

P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, 2013. URL `http://hal.inria.fr/hal-00873592`.

K. Wilson and N. Snavely. Robust global translations with 1DSfM. In *European Conference on Computer Vision (ECCV)*, 2014. URL `https://doi.org/10.1007/978-3-319-10578-9_5`.

J. K. Wong. A new implementation of an algorithm for the optimal assignment problem: An improved version of Munkres' algorithm. *BIT Numerical Mathematics*, 1979. URL `https://doi.org/10.1007/BF01930994`.

C. Wu. VisualSFM: A visual structure from motion system, 2011. URL `http://ccwu.me/vsfm/`. [Accessed August 2019].

S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL `https://doi.org/10.1109/CVPR.2017.634`.

G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. 1996. URL `https://doi.org/10.1007/978-94-015-8668-9`.

K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision (ECCV)*, 2014. URL `https://doi.org/10.1007/978-3-319-10602-1_49`.

K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. In *European Conference on Computer Vision (ECCV)*, 2016. URL `https://doi.org/10.1007/978-3-319-46466-4_28`.

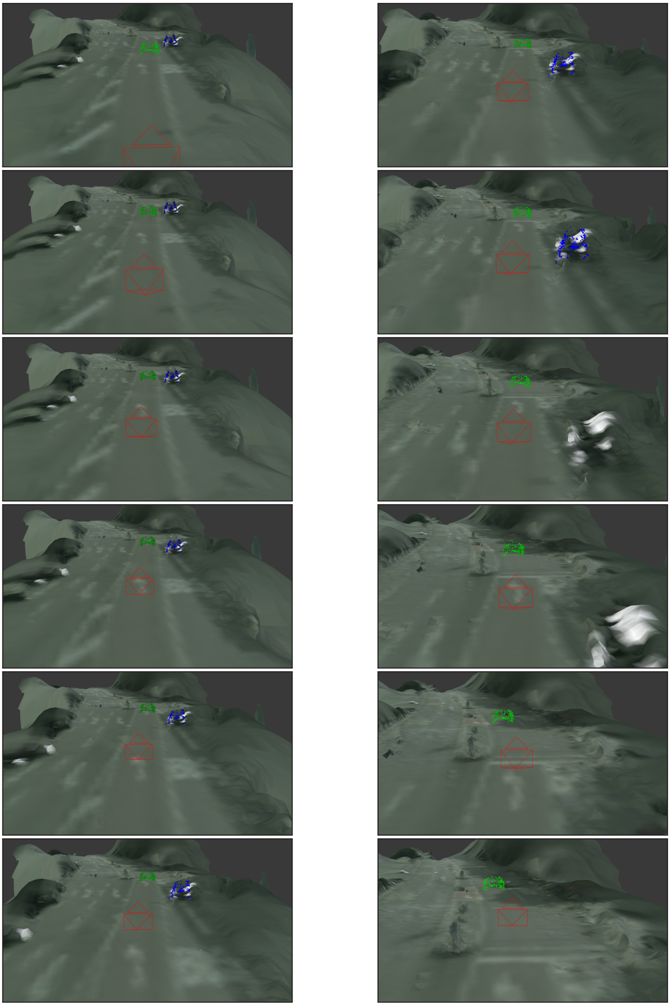
C. Yuan and G. G. Medioni. 3D reconstruction of background and objects moving on ground plane viewed from a moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. URL http://dx.doi.org/10.1109/CVPR.2006.16.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL https://doi.org/10.1109/ICCV.2015.179.

B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://doi.org/10.1109/CVPR.2017.544.

K. Özden. *3D reconstruction of dynamic scenes*. PhD thesis, KU Leuven, 2007. URL https://limo.libis.be/primo-explore/fulldisplay?docid=LIRIAS1669872&context=L&vid=Lirias&lang=en_US.

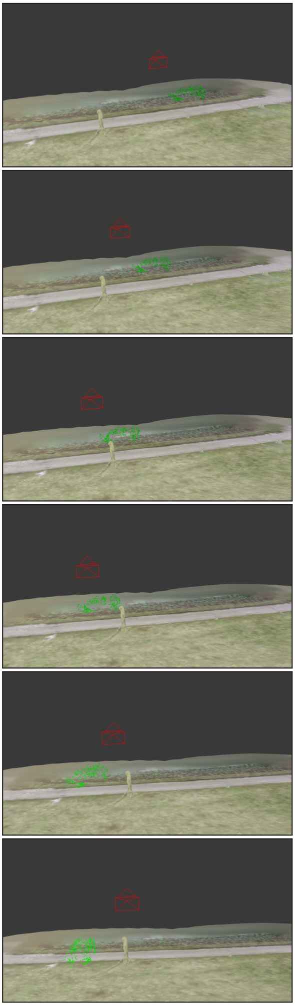
A Appendix



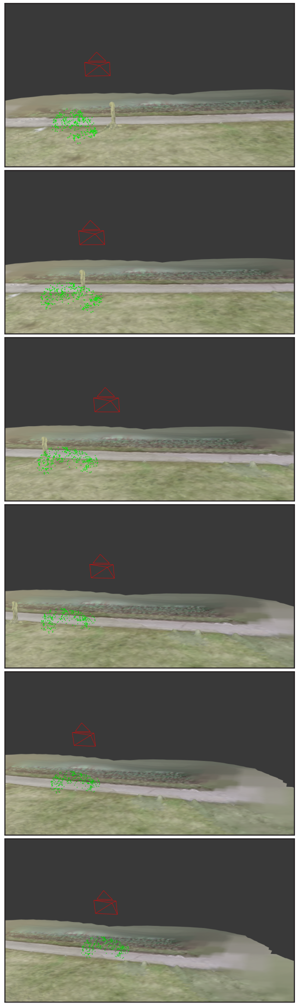
(a) Frame 10, 25, 40, 55, 70 and 85 (from top to bottom).

(b) Frame 100, 115, 130, 145, 160 and 175 (from top to bottom).

Figure A.1: Trajectory reconstruction results per frame. The vehicle points are shown in green and blue. The camera pose are represented by the red camera symbol.



(a) Frame 0, 22, 44, 66, 88 and 110 (from top to bottom).



(b) Frame 132, 154, 176, 198, 220 and 242 (from top to bottom).

Figure A.2: Trajectory reconstruction results per frame. The vehicle points are shown in green. The camera pose are represented by the red camera symbol.

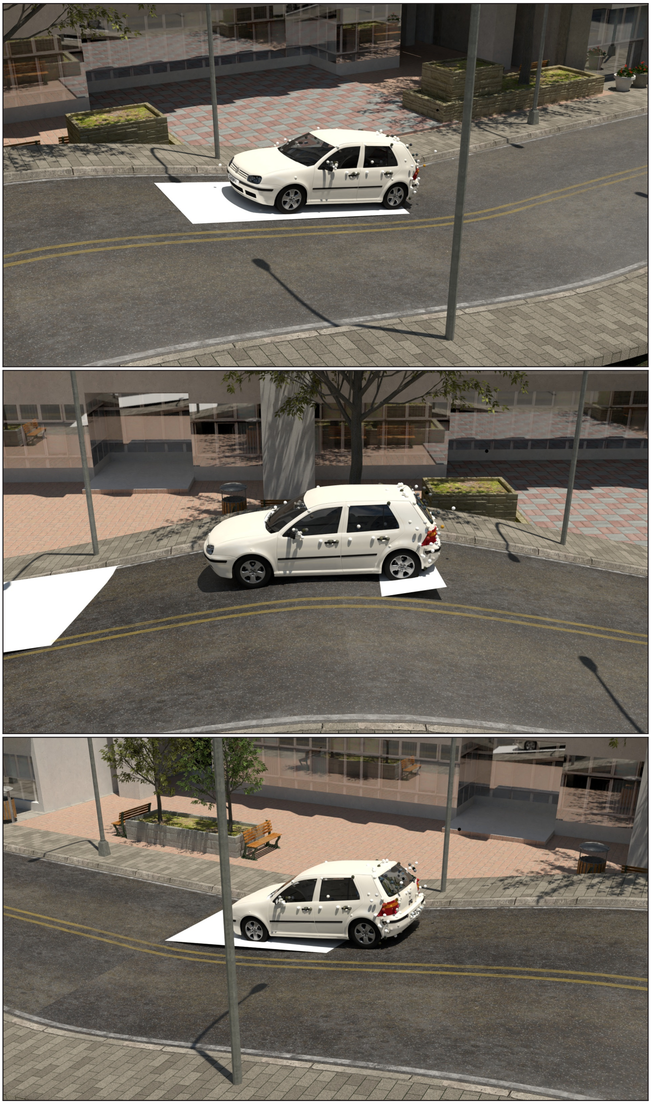


Figure A.3: Reconstruction results using the trajectory estimation method described in section 6.4.2. The small spheres represent triangulated object points and the white planes the local ground approximations.

Figure A.4: Reconstruction results using the trajectory estimation method described in section 6.4.2. The small spheres represent triangulated object points and the white planes the local ground approximations.

Karlsruher Schriftenreihe zur Anthropomatik
(ISSN 1863-6489)

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Band 1 Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz. 2006
ISBN 3-86644-070-7

Band 2 Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse. 2007
ISBN 978-3-86644-149-1

Band 3 Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme. 2010
ISBN 978-3-86644-457-7

Band 4 Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2010
ISBN 978-3-86644-469-0

Band 5 Thomas Usländer
Service-oriented design of environmental information systems. 2010
ISBN 978-3-86644-499-7

Band 6 Giulio Milighetti
Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter. 2010
ISBN 978-3-86644-568-0

Band 7 Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2011
ISBN 978-3-86644-609-0

Band 8 Eduardo Monari
Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken. 2011
ISBN 978-3-86644-729-5

Band 9 Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen. 2011
ISBN 3-86644-760-8

Band 10 Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile. 2012
ISBN 978-3-86644-798-1

Band 11 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2012
ISBN 978-3-86644-855-1

Band 12 Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES). 2013
ISBN 978-3-86644-955-8

Band 13 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2013
ISBN 978-3-86644-988-6

Band 14 Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design" Prinzip. 2013
ISBN 978-3-7315-0041-4

Band 15 Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8

Band 16 Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung. 2014
ISBN 978-3-7315-0167-1

Band 17 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0212-8

Band 18 Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance. 2015
ISBN 978-3-7315-0320-0

Band 19 Marco Huber
Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications. 2015
ISBN 978-3-7315-0338-5

Band 20 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0401-6

Band 21 Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen. 2015
ISBN 978-3-7315-0426-9

Band 22 Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration. 2016
ISBN 978-3-7315-0457-3

Band 23 Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung. 2016
ISBN 978-3-7315-0460-3

Band 24 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2016
ISBN 978-3-7315-0519-8

Band 25 Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement. 2016
ISBN 978-3-7315-0538-9

Band 26 Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug. 2016
ISBN 978-3-7315-0541-9

Band 27 Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments. 2017
ISBN 978-3-7315-0580-8

Band 28 Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen. 2017
ISBN 978-3-7315-0598-3

Band 29 Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6

Band 30 Andrey Belkin
World Modeling for Intelligent Autonomous Systems. 2017
ISBN 978-3-7315-0641-6

Band 31 Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos. 2017
ISBN 978-3-7315-0642-3

Band 32 David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information. 2017
ISBN 978-3-7315-0644-7

Band 33 Jürgen Beyerer, Alexey Pak (Eds.)
Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2017
ISBN 978-3-7315-0678-2

Band 34 Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)
Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2018
ISBN 978-3-7315-0779-6

Band 35 Michael Grinberg
Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking. 2018
ISBN 978-3-7315-0781-9

Band 36 Christian Herrmann
Video-to-Video Face Recognition for Low-Quality Surveillance Data. 2018
ISBN 978-3-7315-0799-4

Band 37 Chengchao Qu
Facial Texture Super-Resolution by Fitting 3D Face Models. 2018
ISBN 978-3-7315-0828-1

Band 38 Miriam Ruf
Geometrie und Topologie von Trajektorienoptimierung für vollautomatisches Fahren. 2018
ISBN 978-3-7315-0832-8

Band 39 Angelika Zube
Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung. 2018
ISBN 978-3-7315-0855-7

Band 40 Jürgen Beyerer and Miro Taphanel (Eds.)
Proceedings of the 2018 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2019
ISBN 978-3-7315-0936-3

Band 41 Marco Thomas Gewohn
Ein methodischer Beitrag zur hybriden Regelung der Produktionsqualität in der Fahrzeugmontage. 2019
ISBN 978-3-7315-0893-9

Band 42 Tianyi Guan
Predictive energy-efficient motion trajectory optimization of electric vehicles. 2019
ISBN 978-3-7315-0978-3

Band 43 Jürgen Metzler
Robuste Detektion, Verfolgung und Wiedererkennung von Personen in Videodaten mit niedriger Auflösung. 2019
ISBN 978-3-7315-0968-4

Band 44 Sebastian Bullinger
Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion. 2020
ISBN 978-3-7315-1012-3

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

This work proposes a Multibody Structure from Motion (MSfM) algorithm for moving object reconstruction that incorporates instance-aware semantic segmentation and multiple view geometry methods. The MSfM pipeline tracks two-dimensional object shapes on pixel level to determine object specific feature correspondences, which create a foundation to reconstruct 3D object shapes as well as 3D object motion trajectories. This work proposes several algorithms to reconstruct object motion trajectories in stereo and monocular image sequences including constraints to estimate scale ratios between object and environment reconstructions, which allow resolving scale ambiguities in monocular image data. An evaluation on video data of driving vehicles shows that meshes computed with the proposed algorithm for object shape reconstruction are robust to reflections and appearance changes. Additional experiments demonstrate that trajectory reconstructions of monocular image sequences are less accurate and robust than reconstructions of stereo imagery, because of the corresponding scale ambiguities.

ISBN 978-3-7315-1012-3

9 783731 510123 >

ISSN 1863-6489
ISBN 978-3-7315-1012-3

Gedruckt auf FSC-zertifiziertem Papier