

Anomaly detection in 3D space for autonomous driving

Master Thesis

Marcus Schilling

Department of Computer Science
Institute for Anthropomatics
and
FZI Research Center for Information Technology

Reviewer: Prof. Dr.–Ing. J. M. Zöllner
Second reviewer: Prof. Dr. rer. nat. R. Reussner
Advisor: Daniel Bogdoll, M. Sc.

Research Period: November 12, 2021 – June 16, 2022

Affirmation

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Karlsruhe,
June 2022

Marcus Schilling



Abstract

Current state-of-the-art perception models do not always detect all objects in an image. Therefore, they cannot currently be relied upon in safety critical applications such as autonomous driving. Objects that cannot be detected are called anomalies. Current work on anomaly detection is primarily based on camera data. This work evaluates to what extent it is possible today to do anomaly detection in 3D on pseudo-lidar data. A pseudo-lidar is a model that estimates 3D depth for each pixel of an image. Currently, there is no approach available that performs anomaly detection on pseudo-lidar data.

Research Question 1 (RQ1) considers whether dissimilarities between lidar and pseudo-lidar are an indicator of anomalies. For this purpose, it is evaluated whether there are larger deviations between pseudo-lidar and lidar point clouds for anomalies compared to non-anomalies. There is no multi-modal dataset for anomaly detection available which could be directly used. Therefore, in the multi-modal KITTI-360 dataset each instance that was not correctly segmented by a panoptic segmentation model, was labeled as anomaly. It is shown how the anomaly definition depends on the used segmentation criterion. The dataset contains 2D images with instance labels and corresponding 3D lidar point clouds with corresponding instance labels. With these labels, one can compare pseudo-lidar and lidar instance by instance. The 2D instance labels allow to define which instances are correctly segmented. In a next step the chamfer-distance between point clouds of each instance in lidar and pseudo-lidar are calculated. Lidar was used as physically captured ground truth. It has been found that the deviation between lidar and pseudo-lidar is similar for anomalies and non-anomalies. Thus, the dissimilarity between lidar and pseudo-lidar can not be used as an indicator for an anomaly.

In Research Question 2 (RQ2) it was analyzed how good a pseudo-lidar can map anomalies of type novelties in 3D. Therefore, one augmented anomaly dataset, and two real world anomaly datasets were considered. All these datasets are image based. For answering the research questions, both a quantitative and qualitative analysis was carried out. As a quantitative analysis, Monte Carlo Dropout was applied onto these datasets to evaluate the uncertainty of the model. The 3D point clouds estimated with the pseudo-lidar were visualized for the qualitative analysis. The qualitative analysis shows that some anomalies can be mapped well in 3D and others are not mapped at all. Furthermore, it is shown that augmented anomalies can be sometimes mapped ambiguously in 3D. In the quantitative analysis, it is shown for all datasets considered that the pseudo-lidar is more certain for anomaly regions than for non-anomaly regions which is interpreted as a consequence of an overconfidence of the model. Furthermore, the anomaly concept is not always consistent across different modalities.

The Research Question 3 (RQ3) analyzed whether anomalies can be found using flow estimation on pseudo-lidar predicted point clouds. An anomaly would be present in theory if the motion

segmentation model contradicted with a panoptic segmentation model. For this purpose, it was investigated whether the pseudo-lidar estimated point clouds are consistent enough through time to do flow estimation on them. For this purpose, the multi-modal KITTI-360 dataset was used. For each instance in the pseudo-lidar it was determined how much a point cloud of an instance differs from the same instance in the next frame. For the consistency evaluation of static and dynamic instances, the ego motion has to be extracted. The pseudo-lidar prediction is consistent between frames if for static instances the distance is small and for dynamic instances the distance is equal to the motion of the instance. It has been shown that the pseudo-lidar makes inconsistent predictions over time, and therefore one cannot distinguish between static and dynamic instances based on pseudo-lidar point clouds. It follows that a flow-based approach to anomaly detection is not possible for point clouds predicted by current single image based pseudo-lidars.

Contents

1	Introduction	1
2	State-Of-The-Art	3
2.1	Taxonomies of Anomalies in Autonomous Driving	3
2.2	Anomaly Detection Datasets and Benchmarks	4
2.3	Current Anomaly Detection Approaches	6
2.4	Image-Based Anomaly Detection	7
2.5	3D Open World Object Detection for Anomaly Detection	9
2.6	3D Point Reconstruction Based Anomaly Detection	10
2.7	Research Gap and Contribution	11
2.8	Pseudo-Lidar	12
2.9	Panoptic Segmentation	14
3	Method	17
3.1	RQ1: Is There a Dissimilarity Between Point Clouds Generated by Pseudo-Lidar and Those Captured by Lidar for Anomalies, and Are These Dissimilarities an Indicator of an Anomaly?	18
3.1.1	Dataset Requirements	18
3.1.2	Dataset Selection	18
3.1.3	Approach	20
3.2	RQ2: Is a Pseudo-Lidar Capable of Mapping Anomalies of Type Novelities in 3D?	25
3.2.1	Dataset Requirements	25
3.2.2	Dataset Selection	25
3.2.3	Quantitative Analysis	26
3.2.4	Qualitative Analysis	27
3.3	RQ3: Is It Possible With a Flow Estimation Approach, Based on Pseudo-Lidar Data, to Find Anomalies of Dynamic Classes?	27
3.3.1	Dataset Requirements	28
3.3.2	Dataset Selection	28
3.3.3	How to Evaluate Pseudo-Lidar Consistency Through Time	29
3.3.4	Ground Removal	30
4	Evaluation	33

4.1	RQ1: Is There a Dissimilarity Between Point Clouds Generated by Pseudo-Lidar and Those Captured by Lidar for Anomalies, and Are These Dissimilarities an Indicator of an Anomaly?	33
4.1.1	Configuration	33
4.1.2	Answering the Research Question	33
4.2	RQ2: Is a Pseudo-Lidar Capable of Mapping Anomalies of Type Novelities in 3D?	35
4.2.1	Augmented Dataset	37
4.2.2	Non Augmented Anomaly Datasets	44
4.3	RQ3: Is It Possible With a Flow Estimation Approach, Based on Pseudo-Lidar Data, to Find Anomalies of Dynamic Classes?	48
5	Summary and Outlook	53
A	List of Figures	59
B	List of Tables	61
C	Bibliography	63

1 Introduction

The German federal statistical office reports 2,564 traffic fatalities in Germany for the year 2021 [1]. According to the same office, drivers of cars, buses, and trucks account for 68.82% of the causes of personal injuries in traffic. The cause of these incidents is sometimes due to the influence of drugs, inattentiveness or fatigue, which leads to delayed and erroneous reactions and behavior. Autonomous vehicles do not have such influences on the driving behavior. Furthermore, computer driving cars have the chance to enhance, e.g., the convenience of riders and also for, e.g., an optimized use of one car by multiple users. Hence, the hope is that autonomous driving systems can largely prevent fatal traffic accidents and simultaneously enhance the comfort and efficiency of driving

Today's autonomous systems sometimes make fatal errors. One reason is that they misperceive their environment. A Tesla Model S in Autopilot mode failed to recognize a truck trailer in front of it and crashed into it [2]. In this case, the misperception is attributed to glare from the sun. Current systems therefore sometimes recognize objects insufficient, incorrectly, or not at all. Autonomous driving is a safety-critical task, and it is therefore of particular importance that the systems perceive their environment as correctly as possible to effectively reduce dead toll and injuries.

Autonomous driving deals with several components, e.g. localization, prediction, path planning, and perception. It is of great importance to recognize unusual situations, like the one the Tesla Model S was in, so that accidents can be avoided. [2]. The identification of such extraordinary events belongs to the perception component of autonomous driving. Autonomous 'level 4 - high driving automation' [3] and level 5 - full driving automation' [3] systems require the ability to function in very unusual situations. Level 4 systems do not need a driver in their areas of application, level 5 systems are fully autonomous and are able to drive in any situation without a driver [3]. Autonomous vehicles of this level need to be able to perceive the environment with high precision. Neural networks currently define the state-of-the-art for perception systems. They are trained and sometimes do not recognize their surroundings correctly in unfamiliar situations [4]. Therefore, as Andrej Karpathy, the Sr. Director of AI of Tesla, points out, 'It's all about the long tail - 99.9999...%' [4] is about dealing with these rare events in order to be able to implement level 4 and level 5 autonomous systems. Anomaly detection is concerned with identifying such rare events. In perception, the detection of anomalies can be made at different levels. Object level anomaly detection is concerned with finding objects that are anomalies. In the following work, research is carried out with a focus on object level anomaly detection in 3D space for autonomous driving.

Different sensors are used in autonomous vehicles, namely lidar, radar, and cameras. [5]. Lidar and radar sensors record the environment in 3d. Each sensor type has its own set of advantages and disadvantages. For example, lidar sensors do not see black vehicles because they are not

reflective [6]. Radar sensors have problems of inference with other signals in the environment [6]. Cameras can be blinded, or covered with dirt [6].

Most anomaly detection approaches are camera based. There are significantly fewer 3D based and multimodal approaches [7] than camera based approaches. In the field of 3D anomaly detection, there are no flow-based approaches that consider object anomalies yet [7]. The lidar-based approaches look at either single points [8], scene level anomalies [9], a domain shift [10], or at unknown classes [11] [12]. Current radar-based approaches search either for anomalies in terms of entire scenes or point-based anomalies [7].

A pseudo-lidar is a model that, given an image, estimates 3D depth for each pixel in the image. With the addition of the intrinsic camera matrix, a 3D point cloud can be calculated. In this work, first it is analyzed how pseudo-lidar performance is related to object anomalies in images. Therefore, it is analyzed if pseudo-lidar is more accurate for instances that can be correctly instance segmented in 2D than for objects that cannot be correctly segmented in 2D. For this purpose, we compare the point clouds estimated with the pseudo-lidar with the recorded lidar point clouds. Then, it is analyzed if anomaly datasets, that contain real anomalies or augmented anomalies, can be mapped from a pseudo-lidar into a 3D object. These two research approaches are used to determine whether anomaly detection approaches can be built on the basis of pseudo-lidar point clouds. Finally, we will evaluate whether flow-based anomaly detection can be performed based on pseudo-lidar point clouds.

This work is structured as follows. Chapter 2 presents the state-of-the-art for the anomaly definition, anomaly detection approaches, pseudo-lidar approaches, and panoptic segmentation models. In chapter 3, the methodology used to answer the questions is presented. Then, in chapter 4, an evaluation of the results is presented. Chapter 5, summarizes the findings and provides an outlook on areas where further research can be conducted.

2 State-Of-The-Art

In this chapter, first, current state-of-the-art approaches in anomaly detection are discussed¹. Then, the state-of-the-art is evaluated for specific tasks outside of anomaly detection. The tasks are necessary for the implementation of this work.

2.1 Taxonomies of Anomalies in Autonomous Driving

An anomaly, in the domain of autonomous driving, is considered by Heidecker et al. [6] as a sub-category of a so-called corner case. Heidecker et al. [6] follow the definition of Bolte et al. [13] that 'a corner case is given, if there is a non-predictable relevant object/class in relevant location' [13]. Their work considers a corner case as the union of anomalies, outliers and novelties. These three categories can be defined as follows [6]. An *outlier* is an observation that is actually known but by some other mechanism looks quite different in the context [6]. For example, when nothing can be seen on the camera image because the sun is blinding [6]. *Anomaly* is described by Jiang et al. as an event that occurs with a low frequency [14]. By others, an anomaly is described as a pattern that is not consistent with what has been learned [15] [16]. For example, an aircraft that makes an emergency landing on the highway is an anomaly because it is inconsistent with the learned occurrence of an aircraft in the sky. The last part, that belongs to the corner cases according to Heidecker et al. [6], are novelties. *Novelties* are defined as objects that are not known during training. The terms corner-case and anomaly are often used as synonyms [6].

This corner case definition of Bolte et al. [13] was transferred to different sensor systems by Heidecker et al. [6] such as lidar, radar, and camera. For each of these sensor systems, one can view corner cases at different levels. These levels were originally defined by Breitenstein et al. [17] and extended by Heidecker et al. [6] for lidar and radar sensor modalities. The different levels are shown in figure 2.1. The *sensor layer* deals with damaged sensors at the hardware level, such as dead pixels on cameras and other corner cases that are due to, e.g., a sensor that is not operating optimally or has been maladjusted. The *physical level* of the sensor layer considers physical aspects to which a corner case can be attributed. Radar systems, for example, often show an insufficient data quality when it's raining. The *content layer* considers corner cases at three levels. The *domain level* looks at domain shifts. If a network is trained on German data, for example, there is a domain shift if it is used in the USA. For instance, the traffic lights in the USA are mounted in different positions than in Germany. In the *object level*, object corner cases are considered. These can occur when objects of unknown classes occur or when objects of known classes occur in a manner that was not present in the training data. The *scene level* looks at anomalies at a particular point in time and looks at situations or actions rather than individual

¹as of 01 December 2021




	Sensor Layer		Content Layer			Temporal Layer
	Hardware Level	Physical Level	Domain Level	Object Level	Scene Level	Scenario Level
 LIDAR-based corner cases	Laser Error <ul style="list-style-type: none"> • Broken mirror • Misaligned actuator 	Beam-Based Corner Case <ul style="list-style-type: none"> • Black cars disappear • ... 	Domain Shift on Single Point Cloud <ul style="list-style-type: none"> • Shape of Road markings 	Single-Point Anomaly on Single Point Cloud <ul style="list-style-type: none"> • Dust cloud • ... 	Contextual/Collective Anomaly on Single Point Cloud <ul style="list-style-type: none"> • Sweeper cleaning the sidewalk 	Corner Cases on Multiple Point Clouds and Frames <ul style="list-style-type: none"> • Person breaks traffic rule • Overtaking a cyclist • Car accident • ...
 Camera-based corner cases	Pixel Error <ul style="list-style-type: none"> • Dead pixel • Broken lens 	Pixel-Based Corner Case <ul style="list-style-type: none"> • Dirt on lense • Overexposure 	Domain Shift on Single Frame <ul style="list-style-type: none"> • Location (EU-U.S.A.) • ... 	Single-Point Anomaly on Single Frame <ul style="list-style-type: none"> • Animal • ... 	Contextual/Collective Anomaly on Single Frame <ul style="list-style-type: none"> • People on a billboard • ... 	
 RADAR-based corner cases	Impulse Error <ul style="list-style-type: none"> • Low voltage • Low temperature 	Impulse-Based Corner Case <ul style="list-style-type: none"> • Interference • ... 	Domain Shift on Single Point Cloud <ul style="list-style-type: none"> • Weather, e.g., snow, rain, etc. 	Single-Point Anomaly on Single Point Cloud <ul style="list-style-type: none"> • Lost objects • ... 	Contextual/Collective Anomaly on Single Point Cloud <ul style="list-style-type: none"> • Demonstration • Tree on street 	

Figure 2.1: Systematic of corner case levels and sensor modalities adapted from Heidecker et al. [6].

objects. Finally, there is the *temporal layer*. In this layer, there is only the *scenario level*, which does not only look at one point in time, but at the evolution of several points in the data as a function of time. Here, a corner case could be, for example, a motorist overtaking another motorist.

In the following, an anomaly is considered the union of a novelty and an anomaly according to the definition by Heidecker et al. [6]. An outlier does not fall under our definition of an anomaly. Since an outlier is caused by other mechanisms, this part is excluded from the anomaly definition used here. Furthermore, this work deals with the sensor modality of a camera and a lidar. From the categorization, the focus is on the content layer and therein on the object level. This means that no domain shift or scene level is considered here. Furthermore, anomalies and outliers due to corner cases of sensor systems are not considered. In the work we consider in parts two successive point clouds as expected from the temporal layer. But we are looking for object anomalies at the first time point and not for anomalies that exist through time or due to a scenario. Hence, the temporal layer is also not reflected upon.

Object anomalies can be considered as bounding box based or instance based. Bounding boxes for anomalies contain pixels that do not belong to the anomaly, in contrast to instance-based anomalies, which contain only pixels that belong to the anomaly. The points that are in the bounding box but do not belong to the anomaly are mapped in 3D and considered as areas associated with the anomaly. These areas can be very different in 3D from the areas that belong to the anomaly, so a bounding box based anomaly consideration with pseudo-lidar is not advisable. Therefore, anomalies are not considered bounding box based in the following, but pixel-wise instance segmented.

2.2 Anomaly Detection Datasets and Benchmarks

To quantitatively compare anomaly detection approaches, we need dataset suited for the task of anomaly detection and based thereon benchmarks. There are different anomaly detection benchmarks available.

The StreetHazard benchmark [18] consists of a synthetic dataset for anomaly detection. For this reason, it is not considered in this work, as there is a domain gap to the real world. Especially, since the dataset is not rendered photo-realistically.

Another approach is based on an augmented dataset called WD-Pascal [19]. Objects from the

Pascal VOC dataset [20] were inserted into the street scenes of the WildDash dataset [21]. The WildDash dataset [21] contains images from different countries, so the data includes a domain shift. Since domain level anomalies were excluded in the anomaly definition for this work, this benchmark is not included in the evaluation of anomaly detection approaches in the following.

The Fishyscapes anomaly benchmark [22] consists of three parts: FS Web, FS Static and FS Lost and Found. It contains images of car rides from the driver’s point of view in which anomalies are present as well as corresponding semantic segmentation masks to distinguish between anomalies and non-anomalies. The FS Lost and Found dataset extends the Lost and Found dataset of Pingerra et al. [23]. It provides additional annotations and filters out specific images. The original Lost and Found dataset includes annotations for anomalies and coarse annotations for the road. FS Lost and Found adds pixel-level annotations for three classes: objects which represent the anomalies, background for pixels belonging to classes of Cityscapes, and void for the other pixels. Images from Lost and Found in which anomalies can be assigned to Fishyscapes classes were filtered out. FS Lost and Found publishes 100 images as validation set, and 275 images remain as private test set. In the following, we will refer to FS Lost and Found as Lost and Found. FS Static is an augmented anomaly dataset that combines two datasets. Images are taken from the Fishyscapes validation set and Pascal VOC instances are inserted into them. The Pascal VOC instances serve as anomalies. Only instances of classes that cannot be assigned to a cityscapes class are inserted. To reduce the domain shift for a more realistic dataset in the Fishyscapes dataset, classes like mammal were placed with a higher probability on the lower half of the image and classes like birds on the higher half of the image [24]. An adapted shadow is also applied to the inserted instances, and the lighting is adjusted to obtain a more realistic result for the inserted instances. In addition, synthetic fog is applied to the Cityscapes parts of the image. Without this fog, a naïve approach could match the augmented Fishyscapes image with the Cityscapes images and look for the places that differ from each other. It is precisely at these pixels, which differ, that one knows that an anomaly has been inserted. Of FS Static, 30 images are public in the validation set and 1000 images are secret in the test set. FS Web is very similar to FS Static. However, not Pascal VOC instances are inserted but images downloaded from the internet. This involves searching the internet for images using keywords and selecting those that have a transparent background. These instances are then inserted as anomalies. This is more cost-efficient than capturing and labeling completely new scenes, as semantic labeling is expensive [25]. The fine image annotation and quality control for the Cityscapes dataset required 1.5 hours per image [25]. One can generate a semantic label by marking the region as an anomaly, in which one inserts the augmented objects. This gives one a semantic label with minor additional effort by inserting the anomalies. The disadvantages of this approach are that the inserted objects do not fit optimally into the image. This means that reflections and shadows are not transmitted correctly, which is a big disadvantage of this method because the created anomaly dataset is non-realistic. Adjustments were made to the brightness and color distribution of the objects, but no realistic reflections were added. Fishyscapes provides pixel-wise semantic masks that distinguish between anomalies and non anomalies, so the labels are suitable for evaluating anomaly detection approaches that search for anomalies in our sense. Moreover, all three parts consist of street scenes from Germany, so there is no domain shift in

Fishyscapes.

CODA is an anomaly detection benchmark in 2D space from 2022 [26]. CODA selects images containing anomalies from KITTI [27], nuScenes [28] and ONCE [29], and labels them. Since KITTI was recorded in Germany and nuScenes in the USA, there is a domain shift within the benchmark. Therefore, this anomaly benchmark is not considered further because our anomaly term has excluded the domain level.

A newer anomaly detection dataset called SegmentMeIfYouCan [30] consists of real images. It consists of three parts, RoadAnomaly21 with 110 images, which generally refers to anomalies in road traffic. The second part is called RoadObstacle21 and deals specifically with obstacles on the road and contains 357 images. The last part is the Lost and Found dataset.

Compared to CODA and WD-Pascal, there is no domain shift in the Fishyscapes benchmark. The StreetHazard benchmark was not considered because it contains only rendered images and no real world data. This domain shift to the real world does not allow evaluation of anomaly detection approaches in our sense. Since the Fishyscapes benchmark uses aggregated data, it has a larger number of images and is therefore preferred over the SegmentMeIfYouCan benchmark when evaluating the approaches here. As shown, the Fishyscapes benchmark is best suited for the evaluation of camera-based anomaly detection approaches. Since current anomaly detection benchmarks rely on image data as input data, other approaches that aren't based on images data like the one proposed by Masuda et al. [8] cannot participate.

To date, there is no anomaly detection dataset or benchmark in 3D. Therefore, it is difficult to compare anomaly detection methods that are in 3D. However, we can compare pseudo-lidar methods. To do this, one can determine an anomaly score for each point in the pseudo-lidar point cloud by a method and then assign this score to the corresponding pixel in 2D. This allows the method to participate in an image-based anomaly detection benchmark.

2.3 Current Anomaly Detection Approaches

In an autonomous vehicle, there are typically the following sensors: lidar, radar, camera, gps, and imu. For lidar, radar, and camera, there are approaches for anomaly detection [7].

Camera-based object anomaly detection approaches can be divided into four categories: reconstruction-based, confidence-based [7] and open world detection. Chan et al. [31] present a confidence-based approach to anomaly detection. Di Biase et al. present a reconstruction- and confidence-based approach [32]. Lis et al. [33] present a reconstruction-based approach towards anomaly detection. In the following, we try to make a selection of methods based on benchmarks, if possible. In the following, we will consider anomaly detection approaches in our sense, i.e. object-based anomaly detection approaches. As shown, the Fishyscapes benchmark is best suited for the evaluation of camera-based anomaly detection approaches. Therefore, it will be used to select which camera-based approach will be considered in the following.

In the field of 3D based anomaly detection, there are approaches that work on lidar data and approaches that work on radar data [7]. Radar based approaches are not considered herein because they are at the scene level [7], and we are looking for anomalies at the object level. Among the

lidar-based anomaly detection approaches, there are confidence-based and reconstruction-based approaches [7]. There is one reconstruction-based 3D anomaly detection approach from Masuda et al. [8] and one confidence-based anomaly detection approach from Wong et al. [34]. The confidence-based approach considers open world detection. Both approaches are considered in detail below. Another kind of approach is based on detecting abnormal motion through scene flow estimation [35]. The approach estimates the scene flow via point clouds and then considers the closest object to the origin. This object is considered abnormal when it moves towards the car, otherwise not. Therefore, one scenario is searched for anomalies, namely the scenario that the closest object to the car is moving towards it. This is considered as scenario level anomaly. It is not explicitly searched for object anomalies. Since our work is object-based, the approach is not fully applicable. Furthermore, the approach is very limited because it only looks at the nearest object.

2.4 Image-Based Anomaly Detection

First we consider image-based anomaly detection approaches. Di Biase et al. [32] present an approach that is both reconstruction and confidence based. This approach achieves state-of-the-art performance on the Fishyscapes benchmark with a Mean Intersection Over Union (mIoU) of 81.4%. Two approaches on the benchmark leaderboard achieve better results, but the associated publications are not published. The approach also reduced the False Positive Rate at 95% True Positive Rate (FPR95) for Fishyscapes Lost and Found and Fishyscapes Web by 50% in comparison to the second best approach. In other words, they reduced the False Positive Rate (FPR) by half while finding 95% of the anomalies. The confidence based approach by Chan et al.[31] in comparison achieved a mIoU of 70.5%. The two approaches are compared in table 2.1. One can see from the higher Average Precision (AP) that the approach of Di Biase et al. [32] detects more anomalies than the approach of Chan et al. [31] for both FS Lost and Found and FS Static. Furthermore, the approach of Di Biase et al., less often predicts anomalies in non anomaly regions as one can see through the lower FPR95. In the reconstruction-based approaches, there is no approach that has participated in Fishyscapes and therefore these have been excluded from consideration based on our criteria. Overall, therefore, the approach of Di Biase et al. [32] is considered below.

			FS L&F		FS Static
	mIoU Cityscapes	AP	FPR95	AP	FPR95
Di Biase et al.	81.4	43.2	15.8	72.6	18.8
Chan et al.	70.5	34.28	47.43	31.3	84.6

Table 2.1: Fishyscapes benchmark results of two image based approaches. The best values are printed in bold [32] [31].

The approach of Di Biase et al. [32] is both reconstruction and confidence-based. For the confidence-based part, the authors use the uncertainty estimation methods softmax entropy [36] and softmax difference [37]. The problem with this approach is that Convolutional Neural Networks (CNNs) are known to be overconfident [38], and thus it can be difficult to find anoma-

lies through softmax entropy [36] and softmax difference [37]. They use a re-synthesizing approach [33] as a second part of their approach. For re-synthesis one uses two components. First, the image on which anomalies are to be found is segmented semantically. This produces a mask that outputs the corresponding class for each pixel. This mask can now be used to synthesize an image with another neural network. Finally, one compares at which places the re-synthesized image differs from the original image and at these places an anomaly is predicted because the re-synthesis did not work there. These approaches are often limited by the module that performs the comparison between the re-synthesized image and the original image [33]. Di Biase et al. [32] therefore combine two approaches and thereby expect to reduce the deficiencies of each approach alone [32].

The following approach combines two groups of methods, an approach which is based on a re-synthesis loss [33] and uses an uncertainty estimation approach [39] [40]. Both modules are combined in a dissimilarity module to retrieve a pixel-based anomaly detection. It combines softmax entropy [39] and softmax difference [40] for estimating the uncertainty. To retrieve a resynthesis loss, first, a semantic map of the original image is inferred. This semantic map is then resynthesized by a conditional Generative Adversarial Network (GAN). Finally, the perceptual difference between the original and the resynthesized image is retrieved. This approach is similar to Dosovitskiy & Brox [41] which used the perceptual difference instead of the pixel distance. This is superior for comparison than to use the pixel distance as similar objects can have different colors, for example after resynthesis. T-shirts, for example, can have different colors, but a resynthesized t-shirt with a different color than the t-shirt in the original image is no indicator for an anomaly, as many wear a t-shirt. In this case, the pixel-distance would be high and the perceptual distance low. This combined approach therefore assumes that higher perceptual difference corresponds to higher probability of an anomaly. To retrieve the perceptual distance, the following steps are performed. First, they subtract and take the absolute of each feature map pixel at each of the five feature map resolutions from the resynthesized image with the original image. Then they reduce the depth of the feature maps to the next one by applying the arithmetic mean function, resulting in five feature maps, one for each resolution. Then, at each resolution, the resulting map is bicubically extrapolated to the target resolution of the anomaly map. Lastly, each map is combined to one feature map by summing all up and by giving more weight to maps from deeper layers. The second concept that is used relies on an uncertainty estimation approach. It uses softmax difference [40] and softmax entropy [39] to detect anomalies. To retrieve a pixel-wise anomaly probability, a dissimilarity module combines uncertainty estimation maps and perceptual difference maps. This module consists of an encoder, fusion module, and decoder. The encoder uses VGG16 [42] to encode the input image resynthesized image and a CNN to encode the semantic map and the uncertainty maps, namely entropy, softmax difference, and perceptual distance. A linear combination by a 1x1 convolution is applied to the input, the synthesized image, and the semantic map, to obtain the same dimension as the uncertainty map. Afterwards, a point-wise correlation fuses this linear combination with the uncertainty map. The decoder then performs an upsampling of the correlation result using spade normalization, which finally retrieves the anomaly prediction.

The weakness of the approach is that the re-synthesis error is dependent on the ability of the

synthesis module to realistically synthesize things. For example, if a class is poorly synthesized, instances of the class are often recognized as an anomaly because the perceptual difference due to the poor synthesis is higher than in other regions.

However, if the synthesis module works properly, the approach is suitable for detecting anomalies, since anomalies are detected in this case if the segmentation mask is incorrect. This is because the segmentation module would synthesize a completely different image in areas that were not segmented correctly, so the perceptual differences here would be high, and an anomaly would be predicted. And exactly where the segmentation is not correct, there is an anomaly.

2.5 3D Open World Object Detection for Anomaly Detection

One approach for 3D anomaly detection is concerned with open world object detection. Open world detection is a task that involves finding bounding boxes and class labels of objects of known and unknown classes. It additionally enables incremental learning of objects of unknown classes. That means the approach can do active learning. Therefore the objects that are detected and of unknown classes can be manually labeled and incrementally learned by the approach. This enables gradual learning of new objects that appear regularly in the world, and is consequently well suited for autonomous driving. The following approach is considered as it is one of the two anomaly detection approaches in 3D [43].

To achieve this, the authors first train a classifier on samples of known classes [44]. Then the trained model identifies objects of the known classes, and unknown classes which are classified as unknown with a zero label. Someone can then select instances with a zero label of certain classes, label them, and then a new model can be trained. The network is not trained from scratch, but by active learning. What the approach has to deal with in active learning is catastrophic forgetting [45]. Catastrophic forgetting means that if you do continuous learning, like here with active learning, the performance on old classes often becomes very bad when you learn new classes. This would lead to a very bad detection of objects of old known classes while learning new classes. In another publication, it was proven that an optimal continuous learner solves an NP-hard problem which would otherwise require infinite memory [46]. The approach presented here [43] addresses this problem by not learning new classes based on the new examples alone, but additionally retaining a set of old representative examples that contain a minimum number of instances for each old class. Thus, the new training contains the new labeled examples and parts of the old examples. To find objects of unknown classes, they use the region proposal network of Faster R-CNN which gives objectness scores of different regions in the input image. These proposals are class agnostic. Thereby, they can predict those proposals, as unknown object, that do not match to a ground-truth object and have a higher objectness score than unknown objects. The region proposal network is unknown aware and followed by the so called region of interest head. The region of interest head learns the regression of bounding boxes and a classification of these boxes. The classification is achieved using an energy based classifier. The energy function returns low energy values for known classes and high energy values for unknown ones.

The energy based classification head uses contrastive learning. That means that feature vectors

of instances of same classes are forced to be nearby and those of other classes are pushed away. A contrastive loss is therefore defined as follows. To get the distance of an instance to each class, a prototype vector of each class is used that describes the typical vector of features in a class. Then the loss between each class, especially between each prototype vector and the feature vector of the instance, is summed up. The equation (2.1) describes the loss calculation of one prototype vector p_i and an instance feature vector f_c . Where Δ describes the minimum distance between prototype vectors of another class and the feature vector of the instance, D describes any distance metric.

$$l(f_c, p_i) = \begin{cases} D(f_c, p_i) & i = c \\ \max\{0, \Delta - D(f_c, p_i)\} & otherwise \end{cases} \quad [2.1]$$

During training the $|Q|$ newest feature vectors are stored in a feature store and after every I_p iteration the average of the feature store defines P_{new} . Then the new prototype is calculated using the old and the new feature vector using an exponential moving average.

The problem with 3D open world detectors is that they do not primarily try to find anomalies that can be attributed to a pattern that has not yet been trained. The focus of such anomaly detection approaches is on detecting anomalies that are novelties, i.e., belong to an unknown class. This is a limit of open world detection approaches.

2.6 3D Point Reconstruction Based Anomaly Detection

As a third approach, we consider the second 3D anomaly detection approach.

The approach by Masuda et al. [8] is based on a variational autoencoder. The point clouds are encoded in a three channel image, where each channel represents one dimension of the coordinate. The input matrices contain 2048 points. The autoencoder was trained on the ShapeNet dataset [47]. It was only trained on certain classes. For training, 2048 points of each point cloud were randomly sampled to match the input dimension of the network. It therefore was transformed to the input size. Variational autoencoders consist of a latent loss, which ensures that the compressed layer follows certain distributions, and a reconstruction loss that ensures that the reconstruction result is accurate. The encoder is based on FoldingNet [48]. In comparison to the original FoldingNet, the authors used a spherical shape for the grid instead of a plane. It uses the chamfer distance, see equation (2.2), 'because training with the chamfer distance is faster in terms of convergence, and the chamfer distance is less computational expensive than the earth mover distance' [8]. The earth mover distance of two distributions, with the same number of samples, is the minimum effort required to transform one distribution into the other. The chamfer distance is the average distance from one point cloud to another point cloud and back. The distance from one point cloud to another point cloud is defined by the average distance of each point of the first point cloud to the nearest neighbor of the second point cloud. .

$$d_{CD}(S, \hat{S}) = \frac{1}{|S|} \sum_{x \in S} \min_{\hat{x} \in \hat{S}} \|x - \hat{x}\|_2 + \frac{1}{|\hat{S}|} \sum_{\hat{x} \in \hat{S}} \min_{x \in S} \|\hat{x} - x\|_2 \quad [2.2]$$

The variational autoencoder learns a probability distribution over 512 normal distributions. The

network uses skip connections in the encoder, which enables the compressed features to better include global and local features [8].

The anomaly score is finally received for each sample by the chamfer distance between the original point cloud and the reconstructed point cloud. There is no other 3D based anomaly detection approach known to the author. The assumption of the approach is that if a point cloud can be poorly reconstructed, then there is an anomaly. The anomaly determination is therefore based solely on the comparison of the reconstructed point cloud with the original point cloud.

However, this approach does not deal with object based anomalies in point clouds of more than one object. Therefore, it cannot participate in the Fishyscapes benchmark and is not comparable to the other approaches described above. Therefore, a single anomaly score is calculated per 3D point cloud instead. The problem is that in real-world conditions, one receives a point cloud with not one object but many. The network would then only present one anomaly score per point cloud and not per object. Therefore, it would also not output an object wise anomaly score. This means one can do less with the information because one does not know where the anomaly is. In order to use this approach in a real environment to discover objects that have not been found before, one would have to first find bounding box proposals and then second calculate the anomaly scores per proposal. However, since the authors did not do that, the approach is not applicable in the context of this work. The approach has nevertheless a contribution for this work because also in the following work with distances between point clouds is worked. The chamfer distance can be used for the calculation of distances in point clouds as it is done in this approach.

2.7 Research Gap and Contribution

There is currently no anomaly detection approach available which is based on pseudo-lidar. The only approaches in 3D space are open world based [12], reconstruction based approaches [8] [49]. The scene flow based anomaly detection approach, however, evaluates the behavior of objects or people over time as abnormal. Therefore, it considers scenario based anomaly detection and not object-based as considered here [9].

The biggest problem with the unsupervised 3D point cloud anomaly detection [8] approach is that it bases the anomaly score on the chamfer distance equation (2.2) and therefore doesn't provide object level anomaly scores in point clouds of more than one object. This is also the reason why it cannot be benchmarked against other anomaly detection approaches in the Fishyscapes benchmark. To use it under real world conditions one would first have to cut out interesting regions from the 3D point cloud and for each of these regions one could then obtain an anomaly score. This would also require additional execution time. Additionally, the object level anomaly score would have to be transferred onto all pixels of the object, and then it could participate in the Fishyscapes Benchmark. The authors [8] use the Area Under The Curve (AUC) value from the Receiver Operating Characteristic (ROC) as the evaluation measure. This measure is not used for evaluation in the Fishyscapes anomaly detection benchmarks [22] but in the SegmentMeIfYouCan benchmark [30]. In this benchmark, however, it is used for a different reason and not for evaluating the reconstruction error. They do not have an outlier score at the pixel level, therefore they can

only compare their model to 3D point cloud reconstruction models. In this respect, they are state-of-the-art, but the applicability of this metric to anomaly detection approaches is not given. It is unknown today if a high deviation in the reconstruction of the 3D point cloud is an indicator for an anomaly, or if it has other causes.

The Open World Object Detection approach [43] tries to detect anomalies by detecting objects of unknown classes. Instead, our goal is to find anomalies that originate from unknown classes as well as those from known classes that have not been detected for other reasons. Although state-of-the-art closed-set object detectors like YOLOR [50] for example show high AP scores, they sometimes cannot recognize objects of known classes. Due to these weaknesses, this approach is not suitable for achieving our goals and a different approach is required.

The current best state-of-the-art anomaly detection approach [32] uses a combination of different approaches, namely uncertainty estimation and resynthesized error minimization. Both results are fused by late fusion to obtain an anomaly map. But it remains in the 2D space. Although the FPR95 for Fishyscapes Lost and Found and Fishyscapes Web was reduced by 50%, compared to the previous state-of-the-art approach, there is still room for improvement.

Another approach using 3D data could be advantageous as most publications work in the 2D image space, and it should be investigated if one could improve anomaly detection by considering 3D space. Especially the false positive predictions needs to be reduced by future approaches, which means that an anomaly is predicted that is not present [32].

In comparison to the other approaches, the proposed approach of this thesis tries to find out if a flow estimation in 3D can be helpful to find anomalies. A flow based approach is expected to result in a lower false positive rate, as only parts of the image that move together are considered potential anomalies. The use of pseudo-lidar in combination with a 3D flow based approach has four advantages. Firstly, it could be used to participate in the Fishyscapes benchmark, as opposed to the other 3D-based approaches [8] [34] [12]. It presents a bijective mapping from image pixel to 3D points and therefore allows transferring anomalies found in 3D to 2D space. Secondly, if it would be used in cars, no lidar would be needed and only cameras would be sufficient. Thirdly, the problems of lidars in certain weather conditions that appear like absorption, reflection and refraction [51] do not apply to the pseudo-lidar that is based on image data. In addition, the fine-tuning of the architectures could be performed with datasets that do not have corresponding lidar data related to the image data. A disadvantage of the approach is that only dynamic objects are considered, so static objects that are anomalies cannot be detected.

The main goal of this contribution is therefore to evaluate if flow estimation, on pseudo-lidar point clouds, allows detecting anomalies.

2.8 Pseudo-Lidar

One can analyze the performance of a pseudo-lidar using two benchmarks. The first benchmark is the monocular depth estimation benchmark on the NYU-Depth V2 dataset [52] which is based on indoor 3D scenes. The domain gap between indoor scenes and street scenes makes it not preferable for the domain of autonomous driving. Since this work also does not deal with *Domain Level*

anomaly detection, any domain shift is to be avoided. For autonomous driving, the indoor pseudo-lidar performance is much less important than the performance in more realistic driving scenarios. What is not available are benchmarks that analyze how consistently the depth is predicted over time. Video based benchmarks required for this are not available. Therefore, our choice of the pseudo-lidar model depends on the monocular depth estimation benchmark on the KITTI Eigen split [53]. KITTI Eigen split is a subset of the KITTI dataset [54] that is often used for pseudo-lidar comparisons [55] [56] [57]. It was captured from a driving car and is therefore well suited for autonomous driving conditions [54]. There are various criteria on which one could base the decision for choosing the pseudo-lidar model. There are six metrics which are commonly used for the KITTI Eigen split benchmark [53] [55] [57] [56]. Another criterion for choosing the pseudo-lidar model is whether a reference implementation of the approach is publicly available.

No approach leads the benchmark on all metrics of KITTI Eigen split. In terms of the Root Mean Squared Error (RMSE) metric, AdaBins is reported as the best approach [55]. There are also three metrics that are based on other thresholds. They measure what percentage of points have a relative deviation from ground truth that is less than 1.25 , 1.25^2 , 1.25^3 , respectively. AdaBins also has the best performance for all these three metrics [55]. The gaps in this metric are relatively small, namely max. 0.2% . In terms of the absolute relative error, a transformer approach called SwinDepth [58] achieves the best result. It is followed by AdaBins with a 0.1% difference. SwinDepth [58] is an approach for which there is no reference implementation. Hence, this approach is not chosen. Furthermore, the lowest absolute relative difference is achieved by SwinDepth, whereas AdaBins has the second-best result in this metric by only a small margin.

In terms of the metric RMSE log error, SC-Depth (ResNet18) [59] achieves the best results. It punishes underestimation more than overestimation. One advantage of this metric is that it is less sensitive to outliers. However, the approaches that are successful in terms of Root Mean Square Error (RSME) log error are not convincing in the other metrics considered here.

Overall, AdaBins was chosen for this contribution because it ranks first or second in all but one metric and a reference implementation is available. AdaBins mainly consists of two parts. An encoder-decoder architecture and a AdaBins module. The encoder-decoder module consists of the EfficientNet B5 [60] backbone. It achieves a comparatively high top-1 accuracy with other current approaches while having a relatively low memory and computational demand [60]. The AdaBins module works with feature maps and further processes them in a transformer encoder. The transformer is based on a smaller version of an image classification model [55]. The feature map received from the encoder-decoder is embedded through convolutions so that the patch embeddings are received. The stride is exactly as large as the receptive field, so the filter kernel does not overlap but cuts out individual regions. These patch embeddings are then used in a transformer encoder. Transformers learn attention and require a lot of memory and computation, so visual transformers generally work on smaller patches of feature maps to reduce the complexity.

In the AdaBins paper it was shown that the adaptive bins achieve better results than fixed bins and therefore the functionality of these is described below. The transformer encoder output is used twice. First it is followed by a classification multi layer perceptron. It returns the bin widths for depth prediction. Each image gets its own bin widths, so that more relevant regions, meaning

regions with more points, have more intervals than regions with fewer points. The output of the bin widths is a softmax score vector for the i^{th} bin it is b_i . The bin center of the n^{th} bin can be calculated by adding all the softmax scores of the $n-1$ previous bins, i.e. the ones before the value in the vector, and adding half the score of b_i . This can then be multiplied by the considered distance interval, $d_{max} - d_{min}$ which is defined by the maximum distance d_{max} and the minimum distance d_{min} that the model should be able to predict. Finally, the minimum distance d_{min} must be added to it and thus the center point of the bin is obtained. This description is formalized in equation (2.3).

$$c(b_i) = d_{min} + (d_{max} - d_{min})(b_i/2 + \sum_{j=1}^{i-1} b_j) \quad [2.3]$$

The second time, the transformer encoder output is used to retrieve the range attention maps. The output is therefore forwarded through 1×1 kernels. The transformer tends to bring larger relationships into the consideration and analysis, of how the parts of the image relate to each other. Then the original feature maps from the EfficientNet B5 are put through a 3×3 convolution. This result contains more pixel-wise fine granular information of the image. Both sources are then combined with a pixel-wise dot product. One receives the range-attention-maps R . This combination makes it possible to combine high-resolution information from the feature maps with more global information through the transformer.

A dimensionality adaption of the range-attention-maps to n layers is achieved by a convolutional layer. N corresponds to the number of bins. Each layer i now contains the weighting factor p_i per pixel for a cluster center $c(b_i)$. Now one can multiply the cluster centers with the scores for each pixel and add them up and thereby get a smooth depth map as in equation (2.4).

$$d = \sum_k = 1^N c(b_k) p_k \quad [2.4]$$

2.9 Panoptic Segmentation

There are various benchmarks to compare the performance of panoptic segmentation models. One can evaluate the performance on panoptic segmentation on the Cityscapes [25], Mapillary and KITTI benchmark [61]. KITTI does not allow for panoptic segmentation in its published form [62], therefore parts of the dataset were combined to obtain a benchmark for panoptic segmentation. This benchmark was only used in some publications [63] [64] [65] [66]. Hence, most approaches cannot be compared with it. For this reason, methods that were only evaluated on the KITTI benchmark were not included in our analysis. Cityscapes and Mapillary in comparison provide a panoptic segmentation benchmark and were used to evaluate the approaches here. The Cityscapes benchmark includes images from Germany and Switzerland [25]. The Mapillary includes images from 6 continents [67]. Another difference between the two datasets is the number of classes: Mapillary has more classes than Cityscapes. To evaluate the panoptic segmentation, the Panoptic Quality (PQ) metric is used, which can be seen in equation (2.5). The PQ consists of the segmentation quality and the recognition quality. The segmentation quality evaluates how well

the segmentation is. The average Intersection over Union (IoU) is used for this. The recognition quality considers a combination of recall and precision, meaning, how effective the model is at recognition.

$$\text{PQ} = \underbrace{\frac{\sum_{(p,g) \in TP} \text{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}} \quad [2.5]$$

In the two benchmarks Cityscapes test [25] and Mapillary val [67] the three best models hold the top in terms of PQ. The PQ of each model in the two benchmarks is shown in table 2.2. In both benchmarks, the SWideRNet [35] based on the Panoptic DeepLab[35] model has the highest and thus best PQ. For Cityscapes test, EfficientPS [63] is in second place and for Mapillary val Axial-DeepLab-L [68] is in the second place.

	Panoptic Quality (PQ)	
	Mapillary val	Cityscapes test
SWideRNet-(1,1,4.5)	44.8	68.5
EfficientPS	40.6	67.1
Axial-DeepLab-L\XL	41.1	66.6

Table 2.2: Benchmark of State-of-the-Art Panoptic Segmentation Models [35][63] [68].

The SWideRNet model has the best PQ, therefore it is considered the state-of-the-art model for panoptic segmentation.

Panoptic Segmentation can be performed in two ways: top-down or bottom-up. Top-down means that the approach is proposal-based, which is typically achieved by combining Mask R-CNN [69] with an additional stuff segmentation branch [70] [71]. Bottom-up approaches are box-free. They are typically achieved by first starting with a semantic segmentation module, followed by grouping operations to achieve instance segmentation results [72]. SWideRNet is a family of bottom-up approaches that is based on the DeepLab architecture [35]. The family of models is called *SWideRNet* – (w_1, w_2, l) in which w_1 , w_2 and l parameterize the individual models. The parameter w_1 controls the channel size for the first two stages. The first two stages are also referred to as stem models. The two remaining parameters w_2 and l adjust the channel size and number of layers, respectively, for the other layers in the model. The best model of the family on the basis of the PQ metric is found by grid search with the following configuration *SWideRNet* – $(1, 1, 4.5)$.

SWideRNet extends the DeepLab Model by adjusting hyperparameters with w_1 , w_2 and l . The performed grid search determined that the width of the layers seemed to be already large enough, so it is retained. However, the number of layers per block was increased by a factor of 4.5. The authors have therefore determined that the model has sufficient width, but a greater depth brings advantages over the original version [35]. The stem network consists of two convolution layers and thereafter a max-pooling layer. The following three convolution blocks consist of two convolution layers and a channel-wise squeeze and excitation module. This module scales each feature map by one scalar value. The scalar value is determined based on the entire feature map. In the last convo-

lution block, a convolution layer is followed by a switchable atrous convolution layer, then a 1×1 convolution and again a squeeze and excitation module. The other parts remain from the DeepLab model. The switchable atrous convolution gathers information acquired during convolving with different atrous rates [35].

3 Method

This contribution considers if it is possible to do anomaly detection in 3D using a pseudo-lidar and flow estimation. In the following, the three research questions are presented and how they can be answered.

RQ1: Is there a dissimilarity between point clouds generated by pseudo-lidar and those captured by lidar for anomalies, and are these dissimilarities an indicator of an anomaly?

In the following, we will analyze if there is a relation between whether an object is perceived in 2D and whether this object can be mapped well in 3D. The objects that were not detected in 2d are considered as anomalies. Since we only consider objects of known classes, we only consider anomalies that are attributable to an unknown pattern. Then we look for each object if it is well mapped in 3D by the pseudo-lidar. For this, we compare the distance of an object in 3d from pseudo-lidar to lidar. We can compare the pseudo-lidar with the lidar because the lidar was recorded physically and is not a model that makes estimates. Now we have for each object once how well it was mapped in 3d and if it is an anomaly due to an unknown pattern. With this, we can then answer the question if anomalies due to an unknown pattern can be mapped properly in 3D and if dissimilarities between lidar and pseudo-lidar are an indicator for anomalies.

RQ2: Is a pseudo-lidar capable of mapping anomalies of type novelties in 3D? In RQ1, we considered whether anomalies of known classes can be mapped properly in 3D. In RQ2, we want to consider if anomalies of the type of novelties can be mapped properly in 3D. For the analysis, we consider the pseudo-lidar results quantitatively and qualitatively. For the quantitative analysis, we work with Bayesian Neural Network (BNN) and measure the uncertainty of the pseudo-lidar estimate at both anomaly and non-anomaly pixels and compare these uncertainties with each other. In the qualitative analysis, we look at the 3D point clouds predicted by the pseudo-lidar and analyze how well the mapping worked for anomalies. Since we are now analyzing how novelties are mapped in 3D, we consider special anomaly datasets that contain anomalies of type novelties. For two real world anomaly datasets the quantitative analysis is performed and for one of them additionally the qualitative analysis. For one anomaly dataset based on augmented data, both the qualitative and the quantitative analysis are performed.

RQ3: Is it possible with a flow estimation approach, based on pseudo-lidar data, to find anomalies of dynamic classes? The third research question deals with whether the prediction of the pseudo-lidar is suitable to make flow estimation on it. If this were possible, a flow-based anomaly detection approach could be implemented using a pseudo-lidar. Flow based approaches require accurate 3D point clouds from successive points in time. Today's flow based approaches are distance based [73] [49]. Therefore, it must be found out whether the pseudo-lidar applied on successive images provides 3D point clouds on which flow estimation is possible. For this, we consider whether the pseudo-lidar outputs consistent estimates over time.

3.1 RQ1: Is There a Dissimilarity Between Point Clouds Generated by Pseudo-Lidar and Those Captured by Lidar for Anomalies, and Are These Dissimilarities an Indicator of an Anomaly?

The aim of RQ1 is to find out whether anomalies due to unknown pattern can be properly mapped by a pseudo-lidar in 3D or whether the deviations of lidar and pseudo-lidar are an indicator for anomalies.

In order to answer the research question, we need to analyze at least two things. *First*, we need to distinguish between anomalies and non-anomalies on a dataset. In the absence of a multimodal anomaly dataset, a dataset-specific approach to the methodology had to be adopted. *Second*, we need to determine for each instance how well it is mapped in 3D by the pseudo-lidar. For this, we compare an 3D point cloud of the environment captured by a lidar with the prediction of a pseudo-lidar.

3.1.1 Dataset Requirements

The point clouds generated by the pseudo-lidar can be compared with point clouds physically captured by a lidar. For this, one needs a dataset that has 3D point clouds recorded by a lidar synchronously with image data. In addition, we need to know where anomalies are in 2D and 3D. So we can see how well these are mapped in 3D. Furthermore, for non-anomalies we need a segmentation, in 2D and 3D, so that we can make comparisons between anomalies and non-anomalies. If we cannot find an anomaly dataset that provides the multimodal data and labels for anomalies and not anomalies in 2D and 3D, we can proceed as follows. For such cases, we need a dataset that contains 2D instance segmentation masks for the images of the dataset. With this, we can determine if an instance is an anomaly. In order to evaluate the performance of the pseudo-lidar instance by instance, we also need an instance segmentation ground truth in 3D of the lidar point clouds. For these instance masks in 2D and 3D, it must be true that for the same instances the same instance IDs have been assigned. We can only then compare anomalies in 2D with how accurate a pseudo-lidar predicts its surroundings instance by instance. In addition to these hard criteria, it seems to be useful to choose a dataset whose publication is one of the most cited. This is because for those datasets, probably more neural networks are pre-trained and hyperparametrically optimized. Therefore, they can also be used to better select the best neural networks for specific tasks. Besides, there should be enough data with labels for both 2D and 3D instance segmentation to evaluate the research question. If there are only a few labels in a modality, it is difficult to make a valid statement about the research question with this data.

3.1.2 Dataset Selection

There is no multimodal anomaly detection dataset that includes instance labels for both 3D point clouds and images [74]. Therefore, we are looking for a dataset that satisfies the properties we presented in section 3.1.1. There are at least 22 datasets containing lidar and image data according to ad-datasets [74]. The KITTI dataset [62] is the most cited under these 22 datasets autonomous

driving dataset containing lidar and image data [74]. In second place in terms of the number of citations are nuScenes [28] and nuImages [28]. The nuImages dataset contains 2D instance masks and the nuScenes dataset contains 3D bounding boxes. For the nuScenes point clouds, panoptic labels for each point are provided by the nuScenes-lidarseg dataset [75]. However, the problem is nuScenes and nuImages are disjoint [76], i.e. they do not contain corresponding lidar and image data. Since they do not contain synchronous image and point cloud labels, they are not suitable for answering the research question. In fourth place, in terms of most citations, is the Oxford Robot Car dataset [77]. Since it does not contain any labels [74], it does not meet our criteria and was not selected. The next most cited dataset is the Waymo Open Perception dataset [78]. It contains instance labels in 2D and bounding boxes in 3D. For the annotation of objects, bounding boxes are less suitable than point labels because bounding boxes can also contain points that do not belong to the object. Therefore, this dataset was not considered. Now we look at the KITTI dataset and whether it is suitable to answer RQ1. KITTI does not contain any instance labels for the point clouds. But in addition to the KITTI dataset, there is also the SemanticKITTI dataset [79] which provides panoptic labels for all KITTI point clouds. Since panoptic segmentation includes both instance segmentation and semantic segmentation, it fulfills the purpose of instance segmentation. KITTI also includes a monocular depth estimation benchmark that means that pseudo-lidars are trained and optimized on the KITTI dataset. Furthermore, it contains instance segmented images. However, of the 6 hours of trips, only 200 images are training examples and 200 images test examples for instance segmentation [61]. Hence, it is not well suited for the evaluation of the first research question. The KITTI dataset is not considered in the following because it only contains 200 images of ground truth 2D instance segmentations. A similar dataset is KITTI-360 [79]. The KITTI-360 dataset was taken with a very similar sensor setup in the same environment as the KITTI dataset. Therefore, the domain shift is assumed to be not too large between KITTI and KITTI-360. Since KITTI has pre-trained and hyperparametric optimized approaches for pseudo-lidar and other tasks, these models can also be used on KITTI-360. It contains synchronized lidar point clouds and images. It has point clouds panoptically segmented and resulting transformed 2D panoptic labels. In addition, due to the transformed labels from 3D to 2D, the instance IDs are the same. Therefore, KITTI-360 is considered and suitable for the evaluation of this research question. The methodology used to answer RQ1 had to be dataset specific because no multimodal anomaly dataset is available. Therefore, it has to be defined what an anomaly is in a dataset that does not contain anomaly labels. It is also outlined how the pseudo-lidar can be compared to the lidar per instance.

3.1.2.1 Recovery of Panoptic Labels

In order to compare the pseudo-lidar performance instance by instance for each time point in KITTI-360, we need instance labels for the point clouds. The panoptic and motion labels are not available for each point in the point cloud. Instead a fused point cloud contains labels. A total of 9.58 billion raw points were sampled, down to 835 million points. Hence, from the 835 million fused point clouds that have been labeled, we need to recover the labels for the raw points.

Therefore, an already existing recovery script for KITTI-360 was extended [80]. The script was changed similarly to Finn [81] but instead of mapping semantic labels to the raw points, instance labels were mapped to the raw points. The fused points are in the world coordinate system. The raw points taken by the Velodyne at each point in time first have to be transformed from the Velodyne coordinates to the world coordinate system so that both the fused and raw points are in the same coordinate system. For this purpose, transformation matrices of the KITTI-360 dataset were used. However, since these were not published for scans where the speed fell below a threshold, the labels cannot be recovered for these points in time. Out of 81,106 scans, 64,640 could finally be recovered.

For each raw point, the nearest neighbor in the fused points is searched and if this is not greater than 0.5 meters away, the instance label of the nearest fused point is given to the raw point. If there is no fused point within 0.5 meters, the raw point is labeled as "unlabeled". In total, 9% of the points were labeled as "unlabeled". The points that are "unlabeled" and the scans that have no poses were not considered in the following. A total of 6.95 billion points are considered.

3.1.3 Approach

To answer RQ1, two things must be present. For one thing, one has to compare the point clouds of pseudo-lidar and lidar for each instance. And one needs to know for each object whether it is an anomaly or not

3.1.3.1 Instance-Wise Comparison of Lidar and Pseudo-Lidar

The process required to compare pseudo-lidar and lidar point clouds is illustrated in figure 3.1.

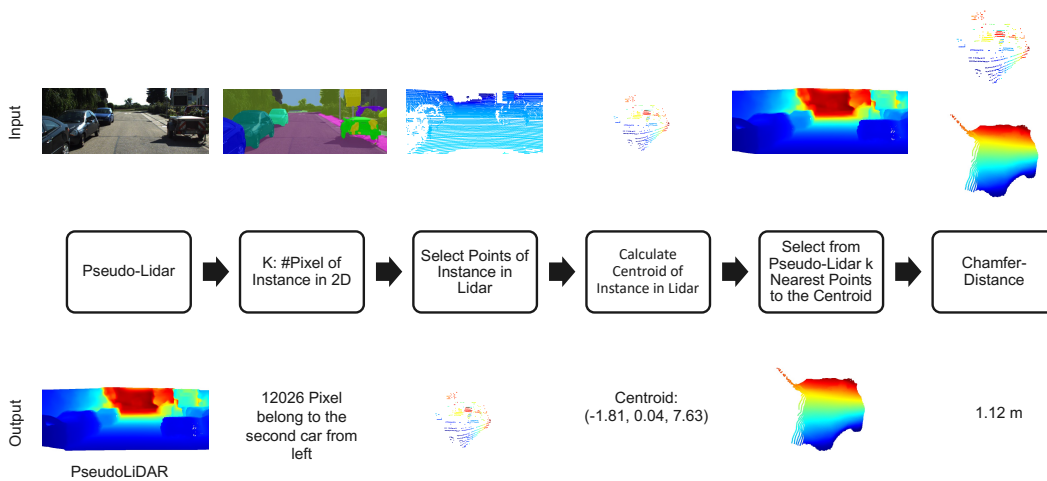


Figure 3.1: Steps required to do instance-wise comparison of lidar and pseudo-lidar.

To compare point clouds instance wise, we need to know which points belong to each instance in the lidar and pseudo-lidar point clouds. The first thing we do is to estimate the 3D environment with AdaBins. To find out how much pseudo-lidar deviates from lidar for an instance, we must find out which points in lidar and pseudo-lidar belong to an instance. For the pseudo-lidar there

is no information for each point to which instance it belongs. One cannot simply select the same number of points in the pseudo-lidar as they were selected in the lidar because the pseudo-lidar has a higher density than the lidar. This approach would result in too few points being selected in the pseudo-lidar. But we know how many points in the pseudo-lidar belong to an instance because we have a 2D instance segmentation. Therefore, we can first take the points that belong to the instance in lidar. With these points, we can determine the centroid. We can use the centroid and select the k -nearest points to the centroid in the pseudo-lidar, with k being the number of pixels in the instance segmentation that belong to the instance. Once the corresponding points in pseudo-lidar and lidar have been found for each instance, the chamfer distance is used as a distance measure to evaluate the performance of pseudo-lidar instance by instance. A large distance means that the pseudo-lidar performs worse compared to the ground truth lidar data, while a small distance means better performance in comparison..

3.1.3.2 Anomaly Classification

As already mentioned, we do not have anomaly labels in KITTI-360, so we have to define anomalies ourselves. An anomaly in our sense 'is given, if there is a non-predictable relevant object/class in relevant location' [13]. Therefore, we consider exactly those instances as anomalies that were not detected by an instance segmentation model. For this task, we used the panoptic segmentation model SWideRNet-(1,1,4.5), which we presented in section 2.9, trained on the Cityscapes dataset.

Determining a Confidence Threshold

KITTI-360 provides transformed instance segmentation from 3D to 2D, and they provide a confidence that the transformation created a correct segmentation ground truth in 2D [82]. The transformation works with Conditional Random Fields. The transformation is performed from the globally labeled point clouds, which is composed of all frames. This is transformed into the images and the labels of the points are transferred to the pixels. lidar does not have as dense data as a camera, so there is not a corresponding point captured for every pixel. Therefore, it can happen that points are transformed to 2D from another point in time that are actually obscured at this point in time. Since the ground truth does not have to be correct, a confidence is added to each pixel indicating whether the ground truth is correct. Low confidence pixels should not be considered because they can provide false ground truth. For such pixels, the model could make a correct prediction, but the evaluation would consider them as a wrong prediction because the ground truth is wrong. Therefore, we do not use the IoU as a metric but introduce a new metric, namely the Confidence-based Intersection Over Union (cIoU). The cIoU is defined in equation (3.1c) for each ground truth instance i . For a pixel at position (x, y) , the confidence is given as $C_{x,y}$, the ground truth as $GT_{x,y}$, and the prediction of the instance segmentation model as $P_{x,y}$. In both the union and the intersection, it considers only pixels that are confident. The segmentation ground of a pixel is confident if its confidence level exceeds a threshold. For these points, the intersection, and union

for each instance is now considered. Finally, the cIoU is defined as the division of both.

$$Intersection_{confident}^i = \sum_{(x,y) \in Pixels} \mathbb{1}_{C_{x,y} > \tau} \cdot \mathbb{1}_{GT_{x,y} \cap P_{x,y} \cap i} \quad [3.1a]$$

$$Union_{confident}^i = \sum_{(x,y) \in Pixels} \mathbb{1}_{C_{x,y} > \tau} \cdot \mathbb{1}_{(GT_{x,y} \cap i) \cup (P_{x,y} \cap i)} \quad [3.1b]$$

$$cIoU^i = \frac{Intersection_{confident}^i}{Union_{confident}^i} \quad [3.1c]$$

The confidence threshold τ must be well chosen. If it is too low, many wrongly segmented GT pixels will be included. This can lead to correctly segmented instances being considered as misclassified. At the other end of the spectrum, a threshold that is too high means that only very few pixels and thus very few parts of the image are taken into account. This could mean that some anomalies are not considered at all in the analysis of the first research question. Additionally, if the threshold is too high, the instance segmentation network may have segmented the instance correctly, but most of the pixels that were segmented correctly were filtered out because of the threshold. Therefore, there is a stronger case for a threshold of 80%.

To determine a suitable confidence threshold, the confidence distribution of the 15,460,208,640 pixels of KITTI-360 were analyzed. Since these confidence values are encoded with 16 bit, they make up a total of over 30 GB Random-Access Memory (RAM). Therefore, a frequency count was first made of all values in order to reduce the RAM requirement. It was then analyzed how different confidence thresholds influence the number of pixels considered. The results can be seen in table 3.1. A threshold of 90% is not preferred, as almost 1/3 of the points are not taken into account.

What seems additionally interesting when choosing a confidence threshold is to look at the Mean Confidence-based Intersection Over Union (mIoU). This is determined by the average cIoU over all instances. If this drops significantly at a certain τ , this could be an indicator that the ground truth transformation from 3D to 2D has led to a false ground truth mask for many pixels that are considered confident. A higher τ would possibly reduce this problem, as fewer pixels of the wrong segmentation mask would be included. The table 3.1 shows the mIoU for different τ as well as the cIoU for different classes. As you can see, the higher the threshold, the better the segmentation result. The rider class has the highest variance for the different confidence thresholds in terms of cIoU. One can see how much the $cIoU^{rider}$ deviates for the different thresholds in table 3.1. Overall, it was shown that the threshold τ has an influence on the cIoU and an influence on the considered pixels. The threshold $\tau = 90\%$ was excluded because only about 2/3 of the image are considered with it. Furthermore, $\tau = 70\%$ was excluded because it is likely to include mislabeled points, since the mIoU as well as the cIoU are strongly reduced for many classes. Therefore, in the following, the threshold of $\tau = 80\%$ will be chosen.

Specification of When an Instance Was Detected

3.1 RQ1: Is There a Dissimilarity Between Point Clouds Generated by Pseudo-Lidar and Those Captured by Lidar for Anomalies, and Are These Dissimilarities an Indicator of an Anomaly?

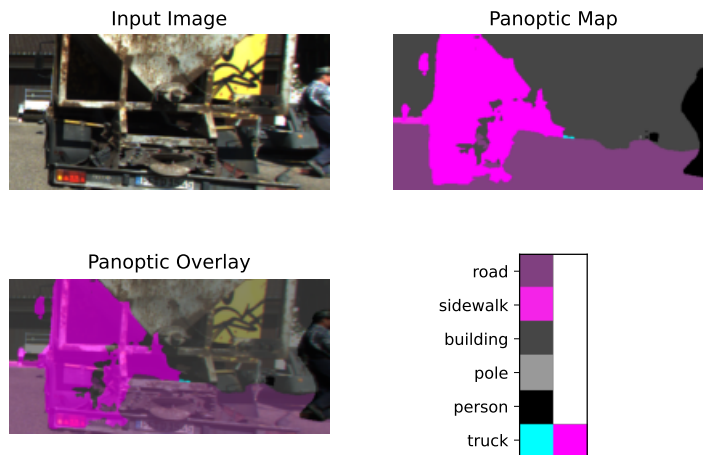
τ	$mcIoU$	$cIoU^{car}$	$cIoU^{rider}$	considered pixels
70	44.01	76.07	53.75	90.85
80	45.87	78.86	58.40	83.52
90	48.09	82.45	63.20	67.18

Table 3.1: IoU analysis for different confidence thresholds τ . Additionally considered pixels for different confidence thresholds.

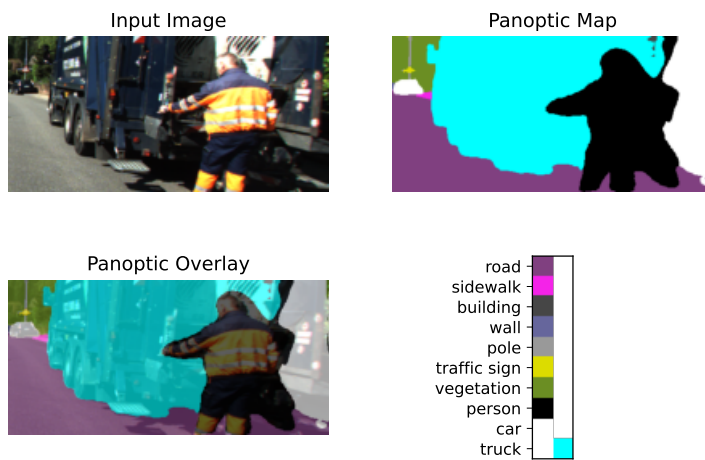
Whether an instance is segmented correctly or not can be defined via the IoU. For this one defines a threshold and if the IoU exceeds this threshold then an instance is considered to be segmented correctly. Since our ground truth has a confidence and is not always correct, we take the cIoU instead of the IoU. The higher you set it, the more instances are considered anomalies. The lower you set it, the fewer instances will be considered anomalies. In figure 3.2 you can see how a segmentation looks like at certain cIoU values. This makes it easier to understand how the cIoU affects what is considered an anomaly and what is considered not an anomaly. The Cityscapes benchmark has an Average Precision With 50% IoU Threshold (AP50) metric that it publishes for each submission [25]. An instance is considered correctly classified if the $IoU > 50\%$. This work is oriented towards this threshold, i.e., the cIoU threshold is set to 50% as in [25].

Different Viewing Angles for Lidar and Camera

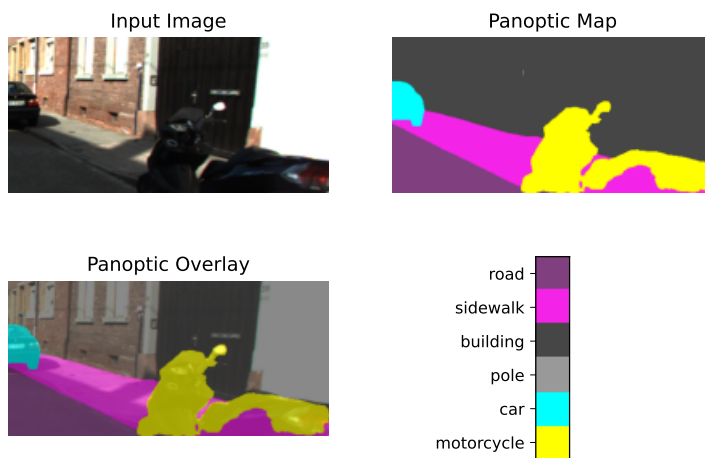
A camera takes pictures from a single point of view. In contrast, the lidar of KITTI-360 captures the 3D point cloud spherically as the lidar rotates 360° . Therefore, instances that were recorded in 3D at a time t were not necessarily recorded by the camera. There are also instances that occur at a time t in the camera instance map but not in the 3D point cloud instance map. This can result from two reasons. First, the camera image has a higher density than a lidar and therefore instances can occur in the camera image that do not occur in the lidar point cloud. Second, labels of points can be incorrectly transformed into 2D and the points can actually be hidden at that point in time. This means that instances can occur in 2D that do not have a corresponding point in 3D at this point in time. In the following sequence 0005 at timepoint 10767 is analyzed to show what an influence the different labels in 3D and 2D have. At this time, four instances were not in 3D, but labeled in 2D. However, these four only cover 23 pixels. Since the instances that are in 2D but not in 3D are small instances, it is assumed that they have no corresponding points in the lidar. Therefore, these instances are not considered in the following analysis. The instances that are in 3D but not in 2D cannot be considered because the 3D point clouds are 360° and the camera has a smaller angle of view. Hence, one can ignore the instances that occur in 3D but not in 2D. Additionally, it was shown above that the labels that appear in 2D but not in 3D are negligible. In summary, it was shown that one should only consider the points and pixels of instances that occur in both 2D and 3D.



(a) The concrete mixer has a cIoU=30.01%. Sequence 0002 at timepoint 11290 of KITTI-360 [82].



(b) Garbage man has a cIoU=50.65%. Sequence 0004 at timepoint 6576 of KITTI-360 [82].



(c) Motorcycle has a cIoU=70.89%. Sequence 0000 at timepoint 6901 of KITTI-360 [82].

Figure 3.2: Illustration of different instances with associated panoptic segmentation maps predicted by SWideRNet-(1,1,4.5) and associated cIoU values.

3.2 RQ2: Is a Pseudo-Lidar Capable of Mapping Anomalies of Type Novelties in 3D?

At the beginning of this work, the question was asked whether the depth can be estimated for anomalies of type novelties. In the following, both a quantitative and a qualitative analysis will be carried out to answer this question. figure 3.3 illustrates which steps are taken to answer RQ2. First, a quantitative analysis is carried out using the uncertainty estimation method of Monte Carlo Dropout. This is followed by a qualitative analysis that evaluates how well the mapping works for anomalies.

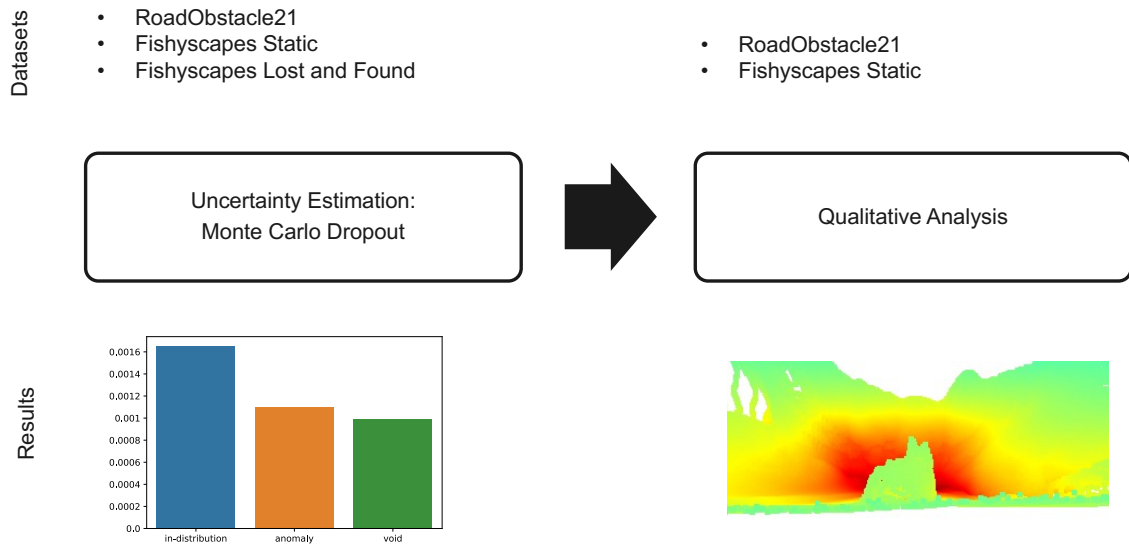


Figure 3.3: Illustration of RQ2 and the steps taken to answer it. For each step it is shown which dataset is considered, and it is illustrated which results will be achieved by the step. First, a quantitative analysis with Monte Carlo Dropout is performed. Then a qualitative analysis is performed using the estimated point clouds to see how well the anomalies were mapped in 3D.

3.2.1 Dataset Requirements

Datasets used to answer RQ2 must contain anomalies that are novelties. Novelties are in the following objects that can not be assigned to classes of cityscapes. Since we want to consider the capability of a pseudo-lidar to map novelties in 3D. In order to quantitatively evaluate the uncertainty of the PseudoLiDAR, we need 2D instance labels to see how uncertain the depth estimate is for anomalies and non-anomalies. The same anomaly masks are needed to visualize the anomalies in the pseudo-lidar point cloud for a qualitative analysis. The optimum would be an anomaly dataset that is multimodal and contains 3D labels for anomalies and non-anomalies.

3.2.2 Dataset Selection

As already explained in section 2.2 there is no multimodal anomaly dataset. Therefore, image-based anomaly datasets are selected in the following. Fishyscapes was chosen as an augmented

anomaly detection dataset [22]. It contains anomalies that are objects with no correspondence to a Cityscapes class and provides pixel wise segmentation of anomalies. Therefore, the dataset is suitable for evaluating the research question. The 30 images of Fishyscapes Static, introduced in section 2.2, that contain augmented anomalies were used in the following. Fishyscapes is based on Cityscapes and inserts cut-out instances into it. Motion blur and depth blur are applied, as well as color matching of the pasted object to the image color schemes [22]. The Fishyscapes datasets divides the images into three regions. The regions are in-distribution for parts that correspond to Cityscapes classes [25]. Void is used as label for regions that do not belong to a Cityscapes class, and anomaly for the anomalies. Void regions in Cityscapes are things like garbage bags that can be somewhere else the next day or your own vehicle [25]. Both the RoadObstacle21 dataset [30] and the Lost and Found dataset [23] contain anomalies that cannot be assigned to Cityscapes classes and offer pixel-by-pixel annotations to the images whether a pixel belongs to an anomaly or not. Therefore, they are suitable for answering our research question. The CODA benchmark [26] is not suitable in our sense because it does not provide pixel wise annotations but bounding box based ones. Other anomaly detection datasets were not considered for the reasons explained in section 2.2 because they consider anomalies that do not fit to our anomaly definition. Therefore, the RoadObstacle21 dataset [30] and the Lost and Found [23] dataset were selected as real world anomaly datasets. The RoadObstacle21 dataset is part of the SegmentMeIfYouCan anomaly benchmark [30]. It consists of images of roads with objects like toys on them. These objects are the anomalies of the dataset. The dataset consists of the images and, corresponding segmentation masks. The segmentation masks divide the images into three parts. The three categories are anomaly, not anomaly and void. Void is a region that does not correspond to a Cityscapes class and is not an anomaly. The 30 training examples were recorded with an iPhone 12 mini. The third dataset considered is the Lost and Found dataset [23]. This resembles the RoadObstacle21 dataset. It also consists of small obstacles lying on the road. It has the same labels as the Fishyscapes dataset.

3.2.3 Quantitative Analysis

The three datasets provide segmentation masks and image data. No ground truth 3D data is provided. Hence, it is difficult to quantitatively assess whether the depth estimate of a pseudo-lidar is correct. However, an uncertainty-based approach is suitable for quantitatively assessing the research question. Therefore, the pseudo-lidar AdaBins is used and Monte Carlo Dropout is applied. Monte Carlo Dropout is used to find out the uncertainty of a model, i.e., the epistemic uncertainty. k forward passes are executed with dropout activated. The average variance of each pixel belonging to a class then gives the uncertainty of the model for that class. In our case, we now calculate the average uncertainty across all images and for each class, i.e., anomaly, not anomaly and void, to find out how uncertain the model is when estimating the depth.

3.2.4 Qualitative Analysis

In addition to the quantitative analysis, a qualitative analysis is also carried out. For the qualitative analysis, one needs the intrinsic camera coordinates to transform the estimated depth image into 3D. In this, the 30 images from RoadObstacle21 that have segmentations are depth-estimated with AdaBins and then the depth estimated 3D point cloud is analyzed for plausibility. The anomaly objects are checked to see if they have been mapped well in 3D. For this the point cloud is color coded in two ways. On the one hand a color coding is used which codes the height so that one can see whether an anomaly stands out from the road for example. On the other hand, a color coding is used that marks in orange which areas are anomalies and the rest is then color coded according to the depth. One is the use of color coding that encodes elevation so that you can see if an anomaly stands out from the road, for example. For the transformation of the estimated depth image into a 3D point cloud, the intrinsic camera matrix of the iPhone12 mini is used. This was determined from the focal length in mm and the pixel width in micrometers. Furthermore, Fishyscapes is analyzed qualitatively as an augmented dataset. Therefore, the depth for Fishyscapes is estimated, and the depth map is transformed into 3D. It is then also checked for plausibility. The qualitative results are then compared to the ones for the Obstacle21 dataset.

3.3 RQ3: Is It Possible With a Flow Estimation Approach, Based on Pseudo-Lidar Data, to Find Anomalies of Dynamic Classes?

The idea behind our theoretical flow-based anomaly detection approach is whether it is possible to use a contradiction between a panoptic segmentation model in 2D and a flow-based model in 3D to find anomalies. In order to find a contradiction between 3D points and 2D pixels, we need a 1:1 mapping from pixels to points. This is made possible by a pseudo-lidar.

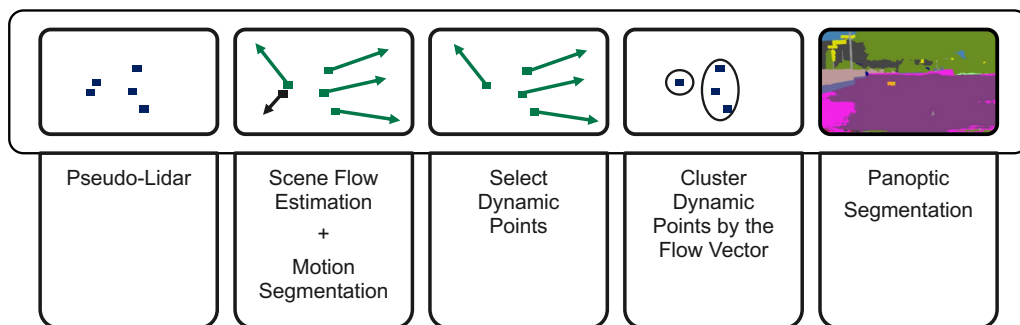


Figure 3.4: Theoretical approach to anomaly detection using a contradiction between motion segmentation in 3D and panoptic segmentation in 2D. For this purpose, an image is mapped in 3D by a pseudo-lidar. The dynamic points are then selected with the help of motion segmentation. These are then clustered by flow into moving objects. Then it is compared whether the pixels belonging to a moving object are segmented by the panoptic segmentation as something that can move. If not there is a contradiction and an anomaly is predicted.

The following approach is illustrated in figure 3.4. First, AdaBins is used to obtain a 3D point cloud for a corresponding image. The input for the flow estimation are two consecutive point clouds. Therefore, in our case, two consecutive images $image_t$ and $image_{t+1}$ of timepoints t and

$t + 1$ are mapped into two point clouds $Pseudo_t$ and $Pseudo_{t+1}$ in 3D by a pseudo-lidar. These two generated point clouds serve as input into a flow estimation and motion segmentation model. The motion segmentation predicts for each point in $Pseudo_t$ if it is static or dynamic. Dynamic means that the point moves, static means that it does not move. The flow estimation indicates how each point in $Pseudo_t$ moves from t to time $t + 1$. One can either use a motion segmentation model to classify the points as static or dynamic, or one can classify a point as dynamic if the closest point in the following point cloud exceeds a certain distance, and otherwise classify the point as static. The dynamic points are then clustered by flow. Finally, the clusters are compared to an instance segmentation prediction.

The contradiction that can be used for anomaly detection looks as follows. If the pixels belonging to a dynamic cluster in 3D are segmented as a non-movable class by the instance segmentation, then the pixels in the image are output as anomalies. In this area, on the one hand the instance segmentation network says there is an immobile class there and on the other hand the motion segmentation model says something is moving there. That is a contradiction.

Current flow estimation models use nearest neighbor distances to predict flow [73] [83]. Therefore, a flow-based approach on pseudo-lidar point clouds is only suitable if they are consistently predicted through time. That is, if the pseudo-lidar predicts successive point clouds inconsistent through time, no flow-based anomaly detection approach is appropriate for pseudo-lidar point clouds.

3.3.1 Dataset Requirements

For flow based anomaly detection on the data of a pseudo-lidar we need a dataset that has both images and instance masks for them. In addition, this data must be available for successive points in time. Without the instance masks, we cannot judge how consistent the depth estimation is through time and therefore whether a flow based approach is suitable. In addition, as already described in section 3.1, instance masks allow us to obtain anomaly ground truth. In order to assess the consistency of two consecutive pseudo-lidar point clouds, we also need a way to remove the ego motion. Transformation matrices are suitable for this purpose. Furthermore, it is of great importance to be able to judge independently how the consistency through time behaves for static and dynamic objects. If the pseudo-lidar gives larger changes through time for dynamic instances than for static ones, it seems suitable to use distance based flow estimation models on the data. Therefore, we need a dataset with motion labels that classify each instance as static or dynamic. In summary, the dataset must include: images with 2D instance labels, corresponding movement labels and data for ego motion removal. In the following, various datasets are examined to see whether they are suitable for this research question.

3.3.2 Dataset Selection

The Fishyscapes [22] anomaly detection benchmark is based on Cityscapes [25] and consequently includes pixel-wise instance labels. However, the instances as well as the inserted anomalies from Fishyscapes are not motion segmented. Therefore, one does not know whether an instance

is static or dynamic. Although ego-motion data is available from the vehicle to remove ego-motion between consecutive pseudo-lidar point clouds in Cityscapes, Fishyscapes does not provide anomalies for subsequent images. Hence, one cannot examine two consecutive point clouds for consistency in the pseudo-lidar and flow estimation.

RoadObstacle21 and Lost and Found are not suitable because neither have motion segmentation labels. The anomalies they show are from static objects. Therefore, the flow-based approach presented here cannot find any anomalies on them. Furthermore, there are no instance labels to analyze the consistency of the pseudo-lidar through time.

The KITTI-Scene Flow Evaluation 2015 dataset contains motion segmentation labels and scene flow labels [84] [85]. The labeling was semi-automatic. The 200 training images from the dataset correspond to the 200 training images from the KITTI Semantic Instance Segmentation Benchmark [86]. Transformation matrices to eliminate the ego-motion are also available. Thus, the combined use of both datasets fulfills our requirements for 2D instance labels, corresponding movement labels and data for ego motion removal.

The second candidate is KITTI-360. It contains recovered panoptic 2D labels, which also contain instance labels. Motion labels are not available for 2D, but for the 3D points. Since the 3D points have the same instance labels as the 2D panoptic labels, we can use the 3D motion labels to have a motion segmentation ground truth for each instance in 2D. KITTI-360 also provides transformation matrices for the removal of the ego-motion. Therefore, KITTI-360 meets all the requirements for this research question.

In the following, RQ3 was analyzed on KITTI-360 because it comprises over 60,000 time points in contrast to Kitti-Scene Flow with 200 time points, which comprises significantly fewer time points. This means that with KITTI-360, a more valid statement can be made about flow-based anomaly detection.

3.3.3 How to Evaluate Pseudo-Lidar Consistency Through Time

To evaluate how consistent the pseudo-lidar is through time, we need to bring each pair of two consecutive pseudo-lidar point clouds into the same coordinate system. Then we can compare the two point clouds for consistency by comparing the pseudo-lidar and lidar point cloud by the chamfer distance for each instance. For static instances the chamfer distance should be small and for dynamic instances the distance should reflect the motion that each instance has moved between t and $t + 1$. The pseudo-lidar is therefore not consistent through time if the chamfer distance is high for instances that are static. Instances that appear in the first point cloud and have disappeared in the second one are not considered in the following. For these, no consistency comparison is possible.

The following explanation of the transformation of the pseudo-lidar point cloud $Pseudo_t$ to the camera coordinates at $t + 1$ is visualized in figure 3.5. $Pseudo_t$ is the estimated point cloud from the image of camera at time t . Therefore, this point cloud is in the camera coordinate system at time t . The goal is now to compare this point cloud to the $Pseudo_{t+1}$ point cloud. $Pseudo_{t+1}$ is in the camera coordinate system at time $t + 1$. So we transform $Pseudo_t$ in the camera coordinate

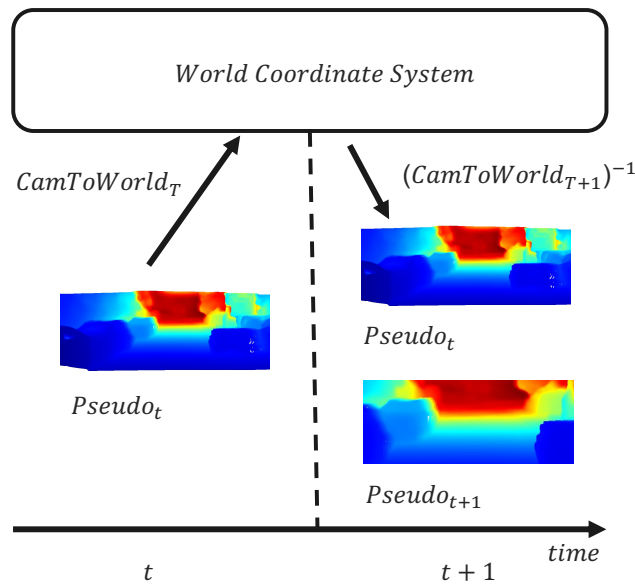


Figure 3.5: Visualization of the transformation of the pseudo-lidar point cloud $Pseudo_t$ from camera coordinate system of t in the camera coordinate system of $t + 1$. Additional visualization of the pseudo-lidar point cloud from timepoint $t + 1$.

system at time $t + 1$. To achieve this, $Pseudo_t$ is transformed to world coordinates and then back to the camera coordinate system of camera at time $t + 1$. The mathematical formulation of this transformation is given in equation (3.2). Since we only have transformations given from the world coordinates to the camera coordinates in KITTI-360, we need to invert the transformations from camera to world in order to obtain a transformation from the world to the camera coordinate system at each point in time.

$$(CamToWorld_{t+1})^{-1} * CamToWorld_T * Pseudo_t \quad [3.2]$$

3.3.4 Ground Removal

We remove the ground first because self-supervised scene flow models like SLIM [73] and Flow-Step3D [83] require a point cloud as input without ground.



Figure 3.6: Visualization of the areas that GndNet segmented as the ground on the pseudo-lidar point cloud. Segmented ground is colored as purple. Considered is KITTI-360 [82] sequence 0005 and time point 752.

For ground removal, you can use GndNet [87], a neural network. GndNet was chosen because

a good reference implementation of this model was available. GndNet, however, is trained on lidar and does not work as well on pseudo-lidar as can be seen in figure 3.6. When viewing the PseudoLiDAR point clouds in 3D, it became clear that the PseudoLiDAR depicts the road as somewhat uneven, even though it is actually flat. We suspect that this is the reason why GndNet has difficulties to remove the ground in PseudoLiDAR point clouds. A different approach towards ground removal would be to cut off everything below a certain height [73]. Since the KITTI-360 cameras are tilted downward, cutting anything below a threshold is not practical because either you cut away too much at the front or the ground is not removed at the back. Instead, one could cut off below a height depending on the distance. Since this is not optimal for the evaluation, we consider another way that is possible on KITTI-360. Ground removal can be carried out using the semantic road mask in 2D of KITTI-360. Since pseudo-lidar has a 1:1 reference between pixel and point, the corresponding points representing a road can be cut out.

4 Evaluation

For RQ1, we will first evaluate if pseudo-lidar performance correlates with instance segmentation performance and whether a deviation between lidar and pseudo-lidar is an indicator for an anomaly. Since the KITTI-360 dataset is not a specific anomaly dataset, we will consider anomalies which are not novelties. According to our anomaly definition, not only unseen patterns count as anomalies, but also new unknown classes. Therefore, we investigate in RQ2, using specific anomaly datasets that contain novelties, whether a pseudo-lidar can map anomalies to 3D. For this purpose, a quantitative and qualitative data analysis is performed. Lastly, RQ3 analyzes whether scene flow based anomaly detection based on a pseudo-lidar is possible. For this, we evaluate if it is possible with today's approaches to do flow estimation on pseudo-lidar data instead of the commonly used lidar data.

4.1 RQ1: Is There a Dissimilarity Between Point Clouds Generated by Pseudo-Lidar and Those Captured by Lidar for Anomalies, and Are These Dissimilarities an Indicator of an Anomaly?

In the following, we will evaluate how the dissimilarity between point clouds generated by pseudo-lidar and those captured by lidar is related to anomalies due to unknown patterns. For this, an instance in 2D was defined as an anomaly if the cIoU does not exceed a threshold. This instance segmentation performance should be put in relation to the pseudo-lidar performance. The pseudo-lidar performance was defined as the instance-wise chamfer distance, equation (2.2), at a timepoint for each instance between pseudo-lidar and lidar. This means that the smaller the distance for an instance the better the pseudo-lidar performance and vice versa.

4.1.1 Configuration

The following experiments were carried out on the KITTI-360 dataset [82]. For the reasons described in section 2.8 we use AdaBins as the pseudo-lidar. As described in section 3.1.3.2 and, section 3.1.3.2 the confidence threshold was set to 80% and the cIoU threshold to 50%.

4.1.2 Answering the Research Question

For the evaluation, the values of the chamfer distance for anomaly instances and non-anomaly instances are considered. If the chamfer distance is usually higher for anomalies than for non-anomalies, then one can see a correlation between anomalies and the pseudo-lidar performance. If the chamfer distances look very similar, no correlation can be found.

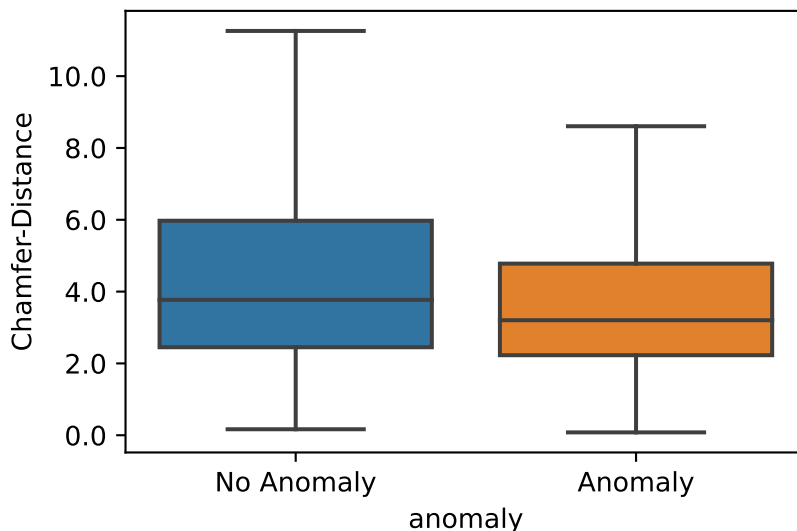


Figure 4.1: Distribution of chamfer distances for anomalies and non anomalies at (cIoU, confidence) = (0.5, 0.8).

The boxplot of figure 4.1 shows in which areas the chamfer distance is situated for instances that are no anomalies and for instances that are anomalies. We see that the chamfer distance is larger for non-anomalies. The median chamfer distance for anomalies is 3.21 and for non-anomalies 3.77. The upper quantile is 5.97 for non-anomalies and 4.78 for anomalies, which is a significantly larger difference than both the median and the lower quantile. Therefore, there is a correlation that the chamfer distance is slightly lower for anomalies than for non-anomalies in KITTI-360. We expected the pseudo-lidar performance to be poor when the instance segmentation performance is poor. This did not prove to be true. It is shown that a dissimilarity between lidar and pseudo-lidar is not an indicator for an anomaly.

It should be taken into account that a pseudo-lidar has greater error at greater distances to the camera [55]. Therefore, we now additionally consider the distances of anomaly and non-anomaly instances. Figure 4.2 shows that on average, anomaly instances are further away from the camera than non-anomaly instances. The median of non-anomalies towards the distance to the camera is below the lower quantile of anomalies. This shows how substantially different the distances of anomaly and non-anomaly instances are in relation to the distance to the camera. It becomes apparent that the panoptic segmentation model has greater difficulties in segmenting more distant objects. However, it also shows that although pseudo-lidar has more difficulty estimating depth at greater distances [55], in our case anomaly instances are better mapped in 3D than non-anomaly instances.

The distance to the camera can not explain why anomalies are better mapped by pseudo-lidar than non-anomalies. The difference could be that different classes can be mapped differently well in 3D. But classes that can be mapped well in 3D are not the same classes that can be segmented well by the panoptic segmentation model. In order to analyze this, one can show the chamfer

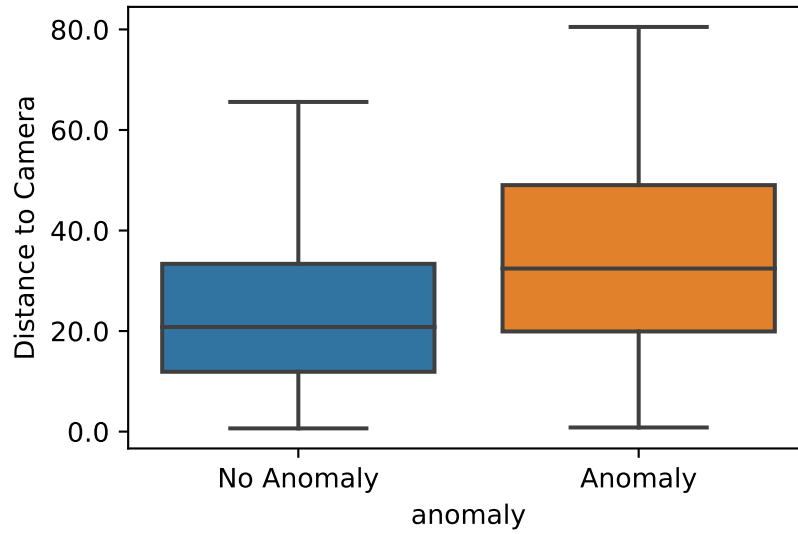


Figure 4.2: Distribution of object distances to the camera for anomalies and non anomalies at (cIoU, confidence) = (0.5, 0.8).

distance per class and the frequency of occurrence of that classes, both for anomalies and non-anomalies. The more frequently a class appears, the greater its influence on the outcome of the evaluation of this question is. Figure 4.3 illustrates this very point: the frequency of occurrence of instances and the chamfer distance for anomaly and non-anomaly instances. It can be seen that the class distribution is not equally distributed. There are many cars and many buildings and garages. This was to be expected, since KITTI-360 is captured on streets where there are many parked cars and houses with garages. On the one hand, one can see that the panoptic segmentation model has great difficulties with buildings and garages. On the other hand, cars are detected with a high cIoU by the panoptic segmentation model. Therefore, there are many cars that are non anomalies and many anomalies under buildings and garages. However, the chamfer distance of buildings and garages is lower from anomalies than the chamfer distance of cars from non-anomalies. This strongly influences the overall result that indicates anomalies are similar to slightly better mapped in 3D than non-anomalies. For anomalies and non-anomalies, the distances between pseudo-lidar and lidar are similar. This means that the dissimilarity between lidar and pseudo-lidar point clouds is not an indicator of anomalies.

4.2 RQ2: Is a Pseudo-Lidar Capable of Mapping Anomalies of Type Novelties in 3D?

RQ1 assessed the relationship between panoptic segmentation performance and pseudo-lidar performance. Since this was evaluated on the KITTI-360 dataset, it considered anomalies according to Heidecker [6] but not novelties, which are included in our anomaly term. This is because KITTI-360 is not a special anomaly dataset. Therefore, we now consider the pseudo-lidar performance on anomaly datasets that deal with novelties of our anomaly taxonomy. Since our panoptic seg-

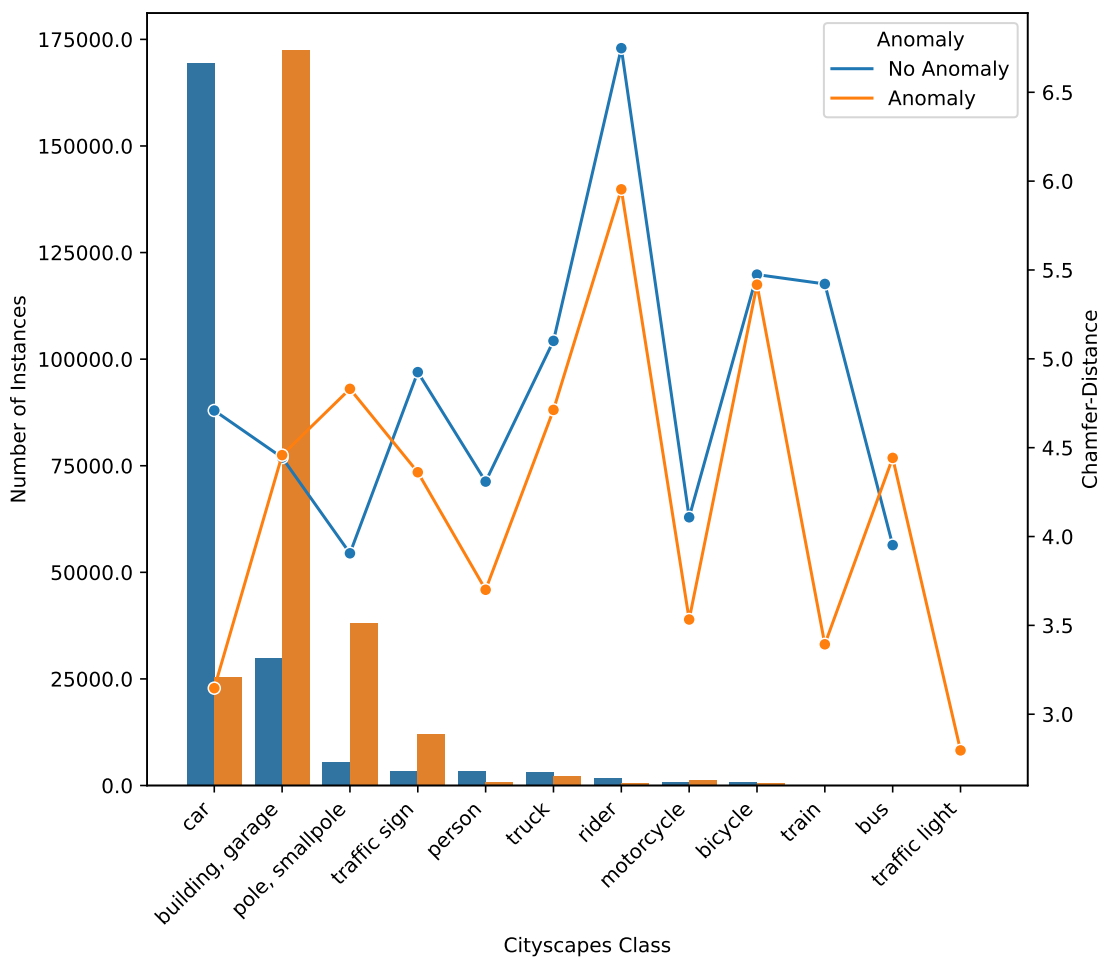


Figure 4.3: Number of instances for each class and corresponding average chamfer distance for both anomalies and no anomalies. The box plot shows the frequency of occurrence and the line plot shows the average chamfer distance. The classes are sorted according to the frequency of occurrence.

mentation model is trained on Cityscapes classes, all instances that belong to classes that are not part of Cityscapes are novelties. The presented anomaly datasets contain toys, animals like dogs, as well as boxes that could have fallen from a truck onto a road. All these elements are not part of Cityscapes classes.

First, we consider whether a pseudo-lidar can map augmented instances in 3D. For this purpose, we perform a quantitative and qualitative analysis on the augmented anomaly dataset Fishyscapes [22].

In order to consider the performance on novelties and the performance on augmented datasets separately, we analyze a novelty-based dataset based on real data. When examining Fishyscapes on its own, we can't be sure if there is uncertainty in mapping anomalies in 3D because of the augmentation of the data or because the instances are novelties. For this purpose, the two datasets Lost and Found and RoadObstacle21 are considered.

4.2.1 Augmented Dataset

The Fishyscapes Static dataset is considered first. It contains segmentation masks for anomalies for 20 images.

4.2.1.1 Quantitative Analysis

Fishyscapes is based on Cityscapes and for the datasets anomalies were augmented into Cityscapes scenes, there is no ground truth of depth data available for both datasets. Therefore, it is challenging to perform a quantitative analysis on it. As described in the methodology, it was decided to use Monte Carlo Dropout to consider the uncertainty for both the class anomaly and non anomaly. The measure we use for uncertainty is the average uncertainty over all uncertainties of a class. This means that we consider the uncertainty for each pixel of a class. For Monte Carlo Dropout, this is the variance of the different outputs of the forward passes at each pixel of the class. From all pixels of a class, we now take the average of these uncertainties and thus obtain an indicator for the uncertainty.

Figure 4.4 shows that the uncertainty is the lowest for anomalies. This is counterintuitive. We actually expected that the uncertainty is higher for anomalies because these were often not present in the training data of the pseudo-lidar. However, the average uncertainty does not show the distribution of uncertainties. It is possible that the median uncertainty is the highest for anomalies even though the arithmetic mean is lower for anomalies. If outliers in in-distribution and out-distribution classes make the uncertainty higher, then the mean could be higher for these classes than for anomalies. Therefore, we consider the distribution of the uncertainties for each class.

In figure 4.5 one sees that the distribution of uncertainties is very different between the different classes. For anomaly the lower-, upper quantile as well as the median is the lowest. This indicates that the anomaly class has the lowest uncertainties. The assumption that the average uncertainty of anomalies is higher than that of non-anomalies due to outliers has therefore not proved to be true. It confirms what the average uncertainty has shown, that the pseudo-lidar is most certain for anomalies compared to in-distribution and void. It is suspected that the model is overconfident for the anomaly regions and therefore makes an incorrect uncertainty estimate.

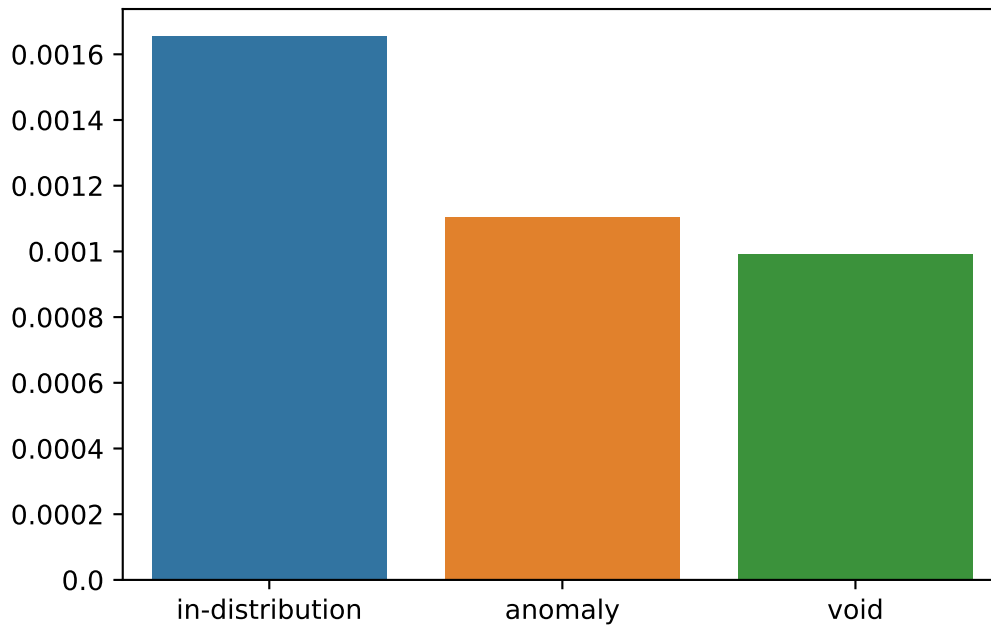


Figure 4.4: Average uncertainty of different classes on Fishyscapes Static.

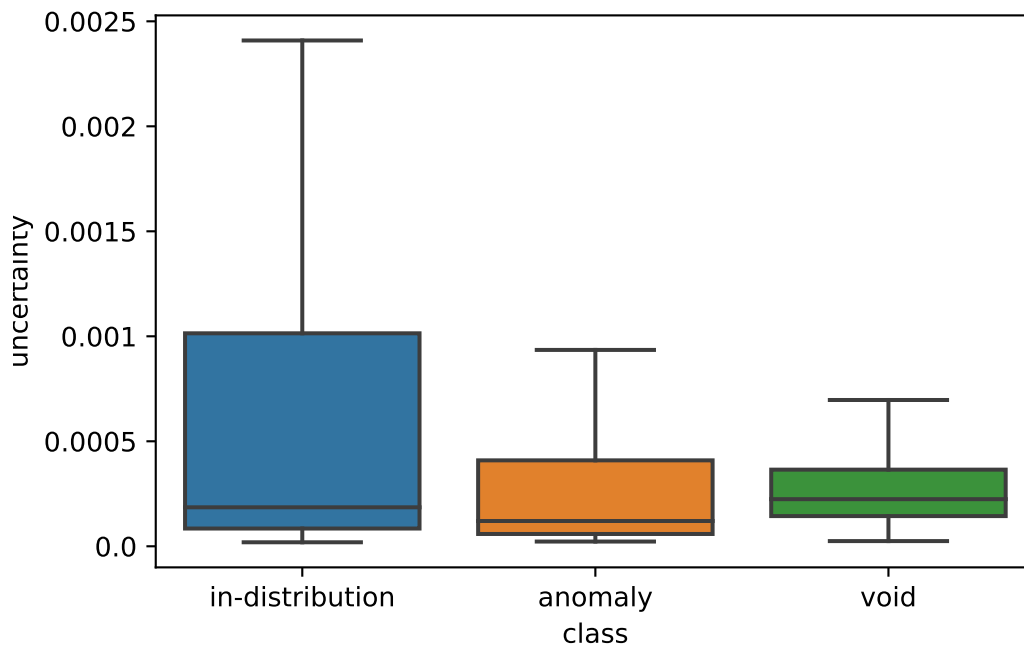


Figure 4.5: Distribution of uncertainty from pseudo-lidar on Fishyscapes Static for in-distribution, anomaly and void regions.

4.2.1.2 Qualitative Analysis

For the qualitative analysis we have to consider each of the three spatial dimensions. At first we will evaluate how the pseudo-lidar mapped the anomalies in 3D from birds eye view. Figure 4.6 shows that the pseudo-lidar has difficulties in mapping the anomalies. It shows results for all publicly available Fishyscapes Static images. The instances are distributed very widely in the z-axis, although they should not have such a great depth. The first evaluation image, for example, has two inserted horses that should not extend so far along the z-axis. The third image contains three birds and the mapped anomalies cover several meters in depth. It is clear that the pseudo-lidar cannot map these augmented anomalies well in depth. In the following we will evaluate particular images of Fishyscapes and their corresponding point clouds.

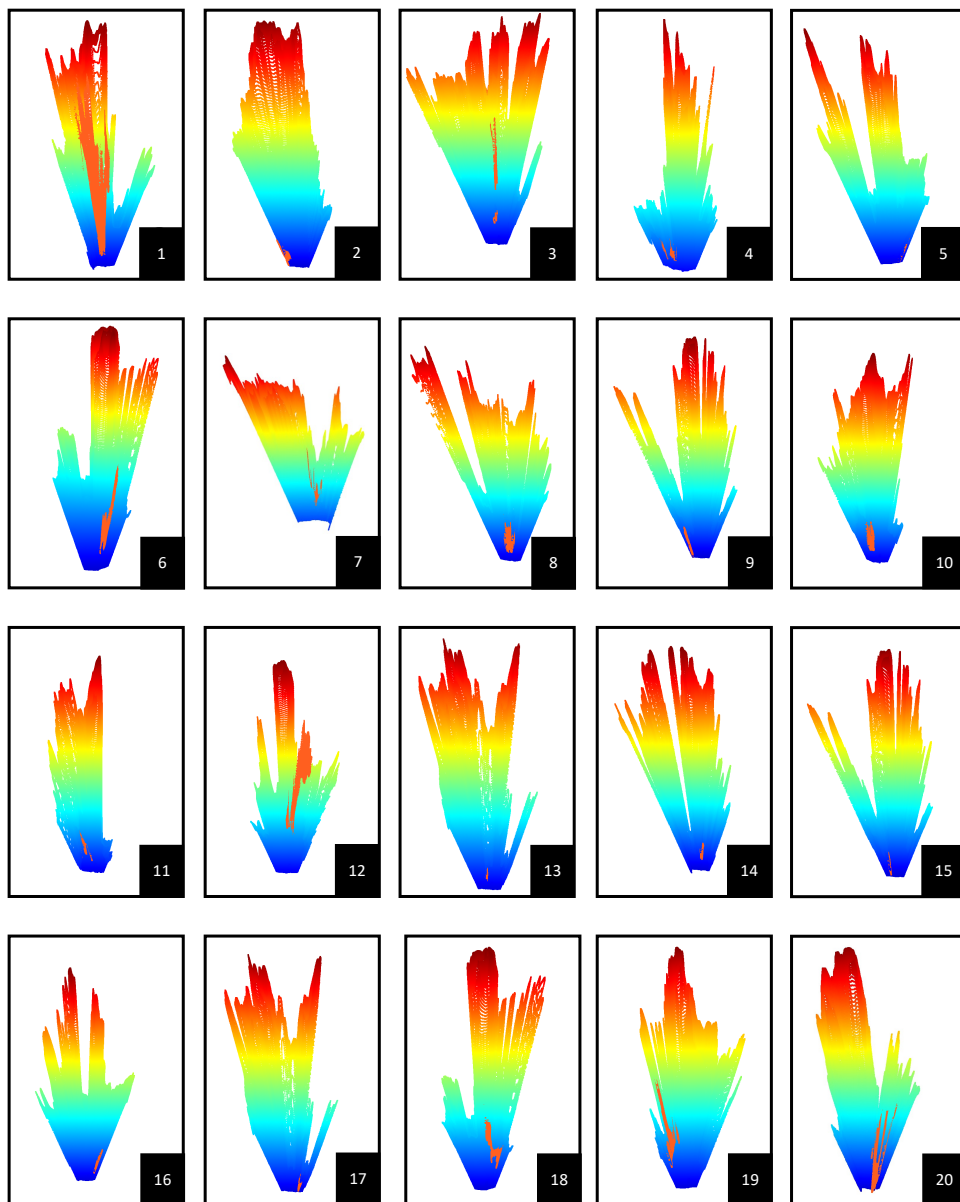
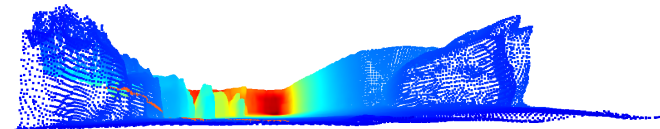


Figure 4.6: All Pseudo-Lidar point clouds of Fishyscapes evaluation images from birds eye view. Anomalies in orange. Color gradient for z-axis for non-anomaly points.

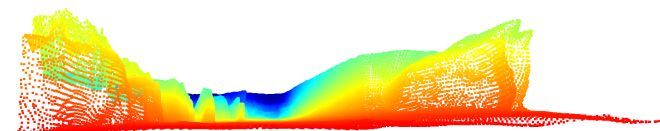
The first finding is that in certain cases the pseudo-lidar is not able to predict the depth at all. This means that instead of mapping an object in 3D, what is presumably behind the object is mapped in 3D, here a street. We categorize anomalies mapped like this in 3D with background-mapped. This means that one cannot see the plasticity of the anomaly objects at all. For this insight, we will look at two examples from Fishyscapes Static, specifically images 11 and 18. As one can see in figure 4.7 the bird that is augmented into the dataset is not mapped in 3D at all. The reason for that could be that the bird that has to be mapped was augmented with color shifts before inserting and thereby the colors look quite similar to the color scheme of the ground. In general, it is quite an unrealistic insertion and barely visible at all. Another reason could be that the bird was not in the training samples of KITTI, on which AdaBins was trained. The network could not generalize enough to predict birds for two reasons. Either the dataset is not large enough, or the network does not have enough capacity to generalize sufficiently.



(a) Fishyscapes Static image 11 [25] [22].



(b) Pseudo-Lidar Fishyscapes Static image 11. Orange for anomaly.



(c) Pseudo-Lidar Fishyscapes Static image 11. Color gradient for visualizing height.

Figure 4.7: Qualitative analysis of Fishyscapes Static image 11.

Fishyscapes Static image 18 shows another bird in another setting. It is also not mapped in 3D, as one can see in figure 4.8. Since the bird from the 18th image of Fishyscapes Static is more distinct in color from the background than the bird from the 11th image and yet it was not mapped in 3D, it is suspected, like stated before, that birds were not or not enough in the training data of AdaBins.

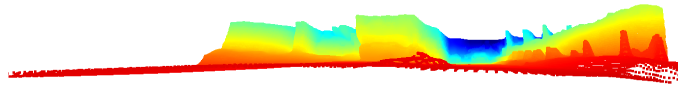
In the second step, we now look at another aspect that the qualitative analysis has brought to



(a) Fishyscapes Static image 18 [25] [22].



(b) Pseudo-Lidar Fishyscapes Static image 18. Orange for anomaly.



(c) Pseudo-Lidar Fishyscapes Static image 18. Color gradient for visualizing height.

Figure 4.8: Qualitative analysis of Fishyscapes Static image 18.

light. The anomalies of image 12 and 13 of Fishyscapes Static consist of dogs. Dogs are not part of a class of Cityscapes [25]. Thereby they are novelties in our sense and thus anomalies. One can see in figure 4.9c that the dog is way better mapped in 3D than the birds. The dog is plastic three-dimensional. In contrast to the birds where one does not see birds in any sense figure 4.8c.

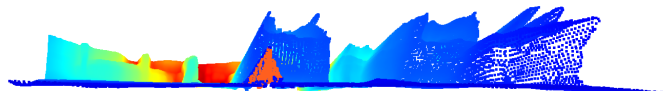
To confirm this, we consider the 13th image of Fishyscapes Static. The dog inserted in this image is platically well visible in figure 4.10c. By human standards, he is a smaller dog than the dog in image 12. The dog is also mapped smaller in 3D by AdaBins. AdaBins was trained on KITTI. Dogs are an anomaly for Cityscapes because they are not labeled as a class. However, in the KITTI dataset used for training AdaBins they are found, for example, in sequence 0 at time 814. Therefore, they are not a novelty and thus not an anomaly for AdaBins but an anomaly for the panoptic segmentation model. Therefore it is assumed that novelties like birds can be mapped worse by AdaBins in 3D than dogs. We classify anomalies mapped like the 12th and 13th images as correctly mapped.

It must be said that for a pseudo-lidar the depth images on which it is trained do not have to be labeled manually. However, the instance masks and panoptic masks have to be laboriously labeled. Therefore, it can still be advantageous to use an AdaBins with pseudo-lidar to detect objects compared to a Panoptic Segmentation Model like SWideRNet, since the pseudo-lidar does not require manual labeling.

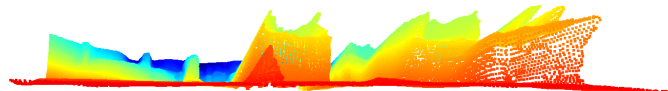
Lastly, we look at ambiguously mapped images where AdaBins has neither mapped nothing of the instance in 3D nor mapped an instance properly in 3D. For this, we look at image 1 from Fishyscapes Static. Figure 4.11a shows two augmented horses inserted into a Cityscapes image.



(a) Fishyscapes Static image 12 [25] [22].



(b) Pseudo-Lidar Fishyscapes Static image 12. Orange for anomaly.

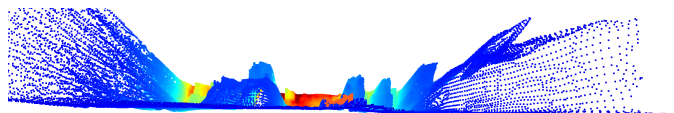


(c) Pseudo-Lidar Fishyscapes Static image 12. Color gradient for visualizing height.

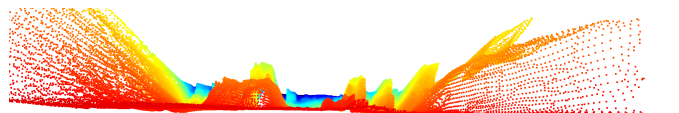
Figure 4.9: Qualitative analysis of Fishyscapes Static image 12.



(a) fishyscapes static image 13 [25] [22].



(b) Pseudo-Lidar Fishyscapes Static image 13. Orange for anomaly.

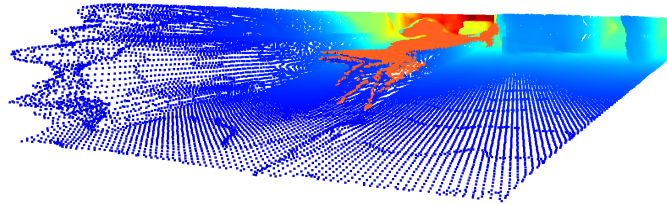


(c) Pseudo-Lidar Fishyscapes Static image 13. Color gradient for visualizing height.

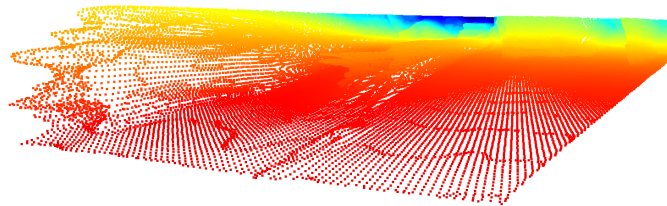
Figure 4.10: Qualitative analysis of Fishyscapes Static image 13.



(a) fishyscapes static image 1 [25] [22].



(b) Pseudo-Lidar Fishyscapes Static image 1. Orange for anomaly.



(c) Pseudo-Lidar Fishyscapes Static image 1. Color gradient for visualizing height.

Figure 4.11: Qualitative analysis of Fishyscapes Static image 1.

One can see quite well in the image that the horses are blended into the image with alpha blending. The background is still visible through the horses. If you now look at the pseudo-lidar point cloud, you see that the legs of the horses are mapped in a place that is roughly correct, but towards the head the mapping goes far back. We think the horses are badly mapped in 3D because of the strong blending. The bus for example shines through the horse. Furthermore, horses seem to be unknown to the network. We refer to anomalies mapped as image 1 as ambiguously-mapped. It has been shown that pseudo-lidar estimates of augmented anomalies can be divided into three categories. The first category is well-mapped for plausible mappings in 3D. The second category is background-mapped for objects that are not visible in 3D, but the model estimates what is behind the object in 3D. The third category is called ambiguously-mapped and means that the anomaly is partially plausible and partially poorly mapped in 3D which occurs for anomalies that are highly augmented. In order to estimate the influence of augmentation on depth estimation, two true anomaly datasets are considered in the following.

4.2.2 Non Augmented Anomaly Datasets

In the following, we will evaluate two real anomaly datasets quantitatively, namely RoadObstacle21 and Lost and Found. For RoadObstacle21 we further evaluate the depth predictions qualitatively. With this, we aim to find out why AdaBins is not able to predict the depth of several anomaly objects. Thereby, we want to find if augmentation caused the bad mapping or if it is because the instances are anomalies that are mapped.

4.2.2.1 Quantitative Analysis

One can see in figure 4.12 that the uncertainty of the depth for Lost and Found [23] is the highest for in-distribution pixels. Anomaly has the second highest uncertainty and void the lowest uncertainty. This means that even on the non-augmented Lost and Found dataset the uncertainty for anomalies is not the highest of the classes.

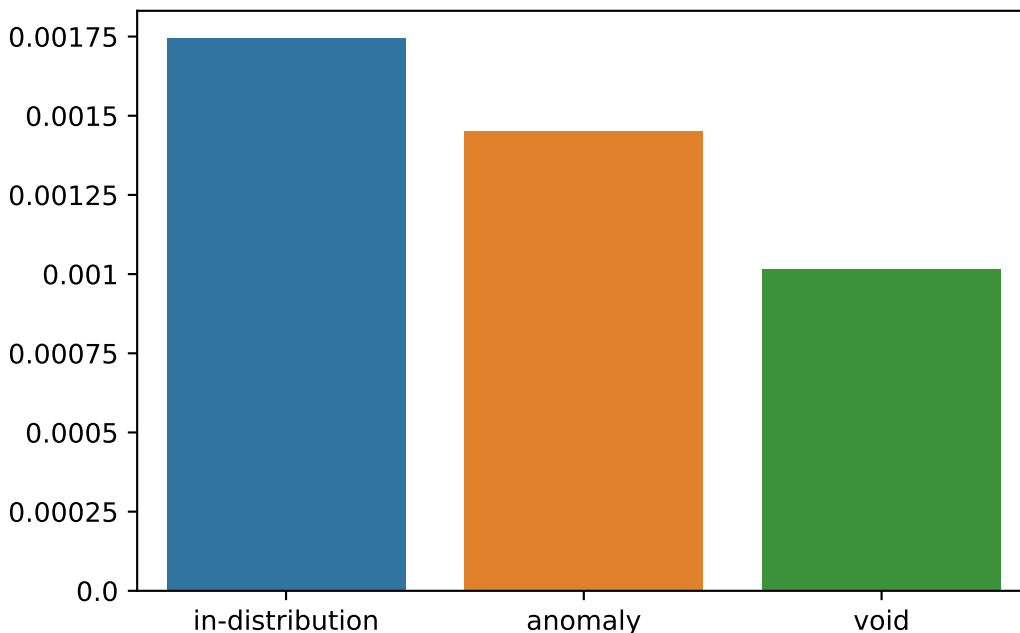


Figure 4.12: The average uncertainty of AdaBins on Lost and Found [23].

In RoadObstacle21 the uncertainty for street is higher than for anomaly as can be seen in figure 4.13. Thus, it is now shown on all three datasets that the AdaBins depth estimation network is not most uncertain for anomalies. For RoadObstacle21, interestingly, the road is the most unsafe region in the deep estimate. This is unexpected since roads have appeared a lot in the training of AdaBins. Perhaps it is due to a domain shift, since the topology of RoadObstacle21 was taken in Switzerland, which is very different from the environment in Karlsruhe. It can also be due to the different road surfaces or the different quality of the road in RoadObstacle21. There are also gravel roads in contrast to KITTI, where there are tarred roads.

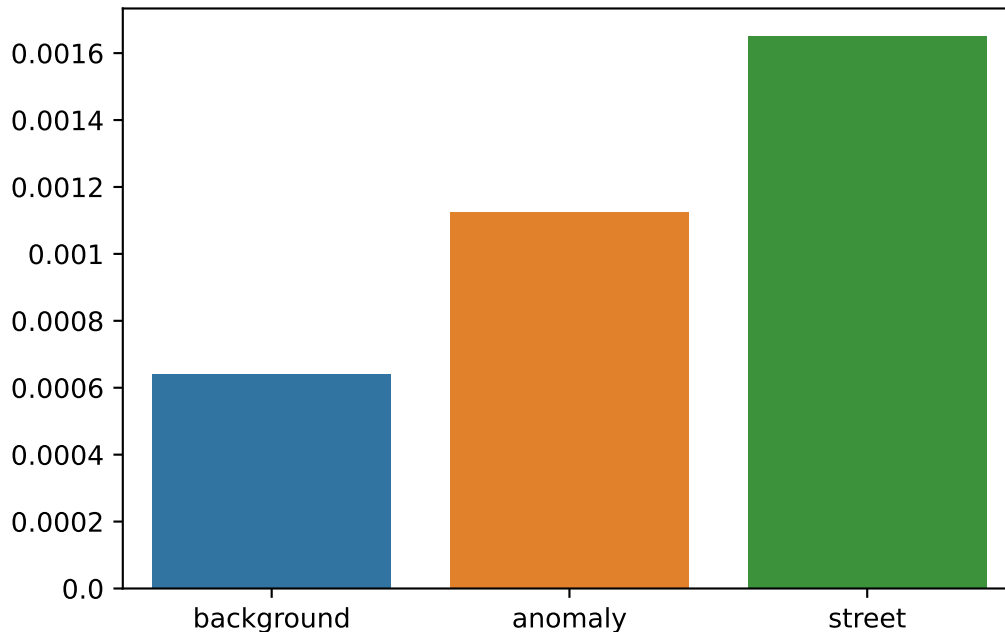


Figure 4.13: The average uncertainty of AdaBins on RoadObstacle21 [23].

4.2.2.2 Qualitative Analysis

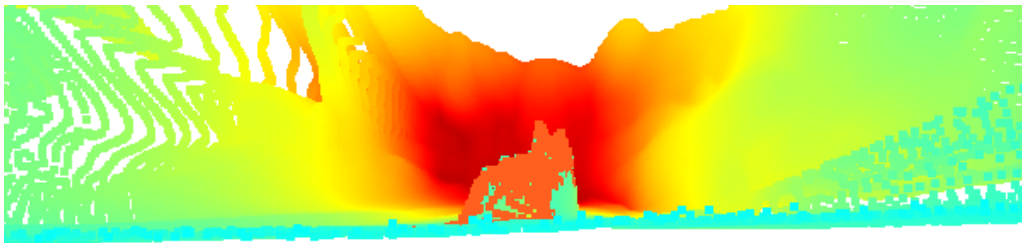
In qualitative analysis, we look at different things. We analyze if the three different aspects found in Fishyscapes are present in the depth estimation of RoadObstacle21. These are background-mapped, well-mapped and ambiguously-mapped.

First we determine if there are well-mapped anomalies in RoadObstacle21. As in Fishyscapes, there are also dogs as anomalies in RoadObstacle21. In figure 4.14a there is a dog sitting on the street. You can see in the pseudo-lidar generated point cloud in figure 4.11b that the dog is well mapped in 3D. Interestingly, there are ears sticking out of its head, although the dog in the image does not have ears sticking out. The presumption suggests that the pseudo-lidar has seen dogs with ears sticking out in KITTI and therefore is trained to map a dog into a dog with sticking out ears. Umbrellas, which also occur as anomalies, can sometimes be mapped well in 3D.

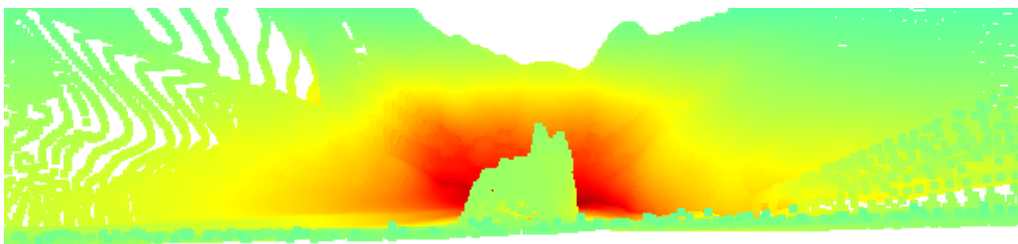
Now the background-mapped category of mapped anomalies is analyzed on RoadObstacle21. In figure 4.15a you can see a soccer ball which represents the anomaly of the image. In the point cloud, one can see that the soccer ball does not appear at all. The neural network simply continues the mapping of what is near the ball in the image, in this case the street. This means that the pseudo-lidar predicts in 3D what is probably behind the anomaly on the image. In all the pictures where the soccer ball appeared in RoadObstacle21 not the soccer ball but what was behind the ball was mapped in 3D. The soccer ball serves as an example for background-mapped anomalies in RoadObstacle21. Toys are mostly background-mapped in 3D. In rare cases something sticks out a little where the toy should be. The same is true for watering cans and trash cans.



(a) Image of RoadObstacle21 with dog as anomaly [30].



(b) Pseudo-Lidar of RoadObstacle21 with dog as anomaly. Orange for anomaly.

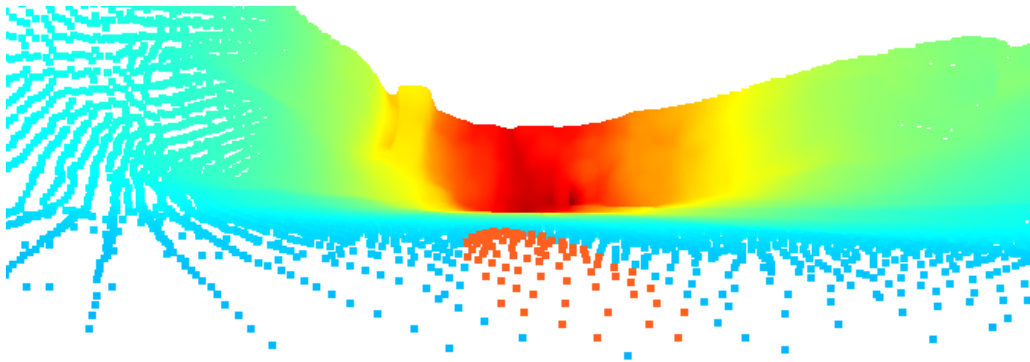


(c) Pseudo-Lidar of RoadObstacle21 with dog as anomaly. Color gradient for visualizing height.

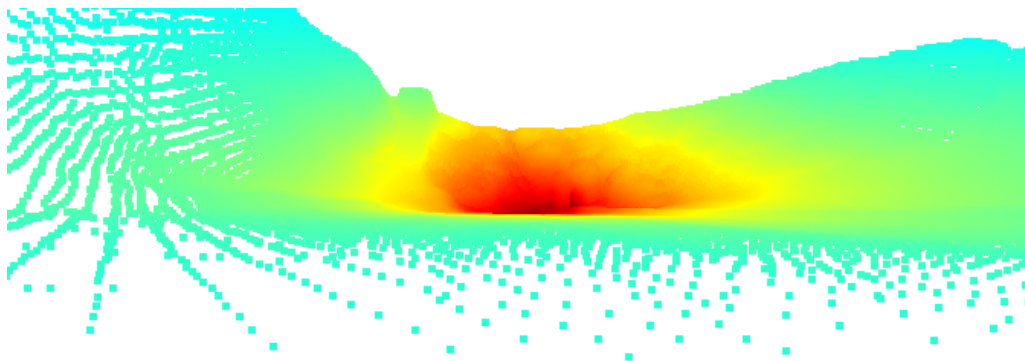
Figure 4.14: Qualitative analysis of RoadObstacle21 image with dog as anomaly.



(a) Image of RoadObstacle21 with soccer ball as anomaly [30].



(b) Pseudo-Lidar of RoadObstacle21 with a soccer ball as anomaly. Orange for anomaly.



(c) Pseudo-Lidar of RoadObstacle21 with a soccer ball as anomaly. Color gradient for visualizing height.

Figure 4.15: Qualitative analysis of RoadObstacle21 image with a soccer ball as anomaly.

The last category in which we categorized the mapping quality of anomalies are the ambiguously-mapped instances. In all 30 images of RoadObstacle21 that are available for evaluation, not a single anomaly was mapped ambiguously. An anomaly was either well mapped or not mapped at all. Whereas with background-mapped anomalies, the network simply mapped in 3D what is probably behind the anomaly. So if there is ground behind an anomaly from the camera's perspective, the network has predicted the ground at the location of the anomaly.

In summary, both RoadObstacle21 and Fishyscapes cannot map unknown classes to 3D in the vast majority of cases. This means that an anomaly detection approach that works on point clouds generated from pseudo-lidar tend to fail for objects that are unknown to the pseudo-lidar. In Fishyscapes, the horses were predicted ambiguously and they are strongly augmented with alpha blending. It is assumed that the blending was responsible for the ambiguous prediction, since no instances were ambiguously mapped in RoadObstacle21. The quantitative analysis has shown that the network is very confident for areas where anomalies occur. The assumption is that the network is very overconfident. As shown in the qualitative analysis of fishyscapes, many instances are not well-mapped in 3D. Therefore, we assume that the uncertainty estimation for reasons of overconfidence did not show any significant deviations from anomalies to other classes.

4.3 RQ3: Is It Possible With a Flow Estimation Approach, Based on Pseudo-Lidar Data, to Find Anomalies of Dynamic Classes?

To evaluate if flow based anomaly detection is possible on pseudo-lidar point clouds, it has to be evaluated if flow estimation works on pseudo-lidar point clouds.

Modern approaches towards flow estimation are self-supervised. This allows you to benefit from large unlabeled datasets and saves expensive labeling. These state-of-the-art unsupervised approaches [88] [89] [90] [73] incorporate the k-Nearest Neighbor (kNN) loss as one of their main losses. The kNN loss work as follows. A model estimates the scene flow between two points in time. As input the model gets two consecutive point clouds between which it is to estimate the scene flow. The estimated flow is applied to the first point cloud, and then for each of these points, the nearest neighbor in the second point cloud is searched. The nearest neighbor distances of all these points now make up the loss. Thus, today's flow estimation models can only be re-trained with pseudo-lidar data if the pseudo-lidar outputs consistent point clouds through time. This means that if one adjusts the own motion between two points in time then a model should give the same distances for the same points of an object. If the noise of the pseudo-lidar is too large, it is not possible to distinguish between static and dynamic instances using distance-based methods.

To compare the point clouds taken from camera coordinates at different times, the transformation from section 3.3.3 is applied. We remove the ground first as in section 3.3.4. First, we consider the deviations of the total point clouds after removal of the ground. Figure 4.16 shows that most of the deviations of the point clouds are in the range of less than one meter. More than 99.7% of time points have a deviation of less than 2 meters.

To facilitate the interpretation, one should additionally consider the time. The time between two recordings moves predominantly between 0.1043 and 0.1047 seconds. Therefore, in the following,

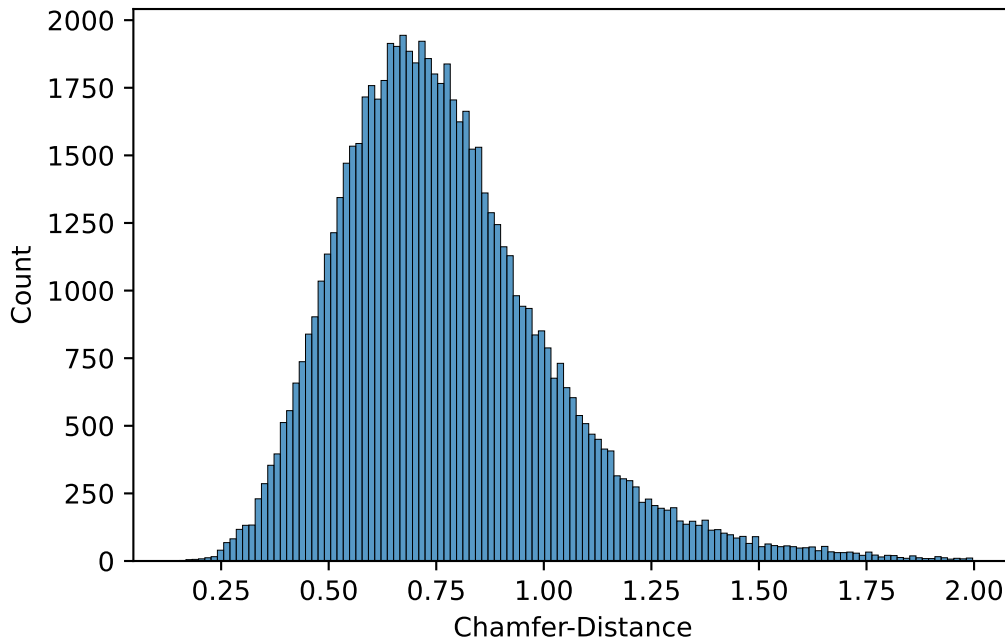


Figure 4.16: Distribution of the average consistency through time, evaluated by the chamfer distance.

the time interval is simplified with 0.1. With this time interval, one can now interpret this distance as the distance of the instance and thereby convert it into km/h. For this you must first halve the distance because it is the average distance from each point in point cloud A to the closest point in point cloud B and back. Then we convert this distance into km/h.

The figure 4.17 shows the distribution of average speeds of all timepoints in km/h. This means that for each point in time, the average speed of all points is determined. This is only an approximation because it is based on the kNN towards the next pseudo-lidar point cloud but with egomotion removed. It can be seen that the point cloud is very inconsistent through time, especially when considering that a large portion of the pixels in the image belong to static classes/instances.

Up to this point, we have considered only one distance per pair of consecutive point clouds to evaluate consistency over time. The problem is that outliers can make this value very large. Therefore, the next step is to determine the distances and velocities per instance. To do this, we cut out the corresponding parts of an instance from each point cloud. Only the chamfer distance between the instance point clouds are calculated. Additionally we show two plots: one for the distance of dynamic instances and one for the distance of static instances. In figure 4.18, you can see that the pseudo-lidar is also not consistent per instance through time. Interestingly, the dynamic instances have fewer deviations than the static ones. Although actually the static instances should be consistent and the dynamic instances should reflect their motion.

In figure 4.19, one sees that the static instances have a very high velocity according to the pseudo-lidar. Static and dynamic instances on pseudo-lidar point clouds cannot be distinguished from each other based on distance as one can see. Almost every static instance can be considered

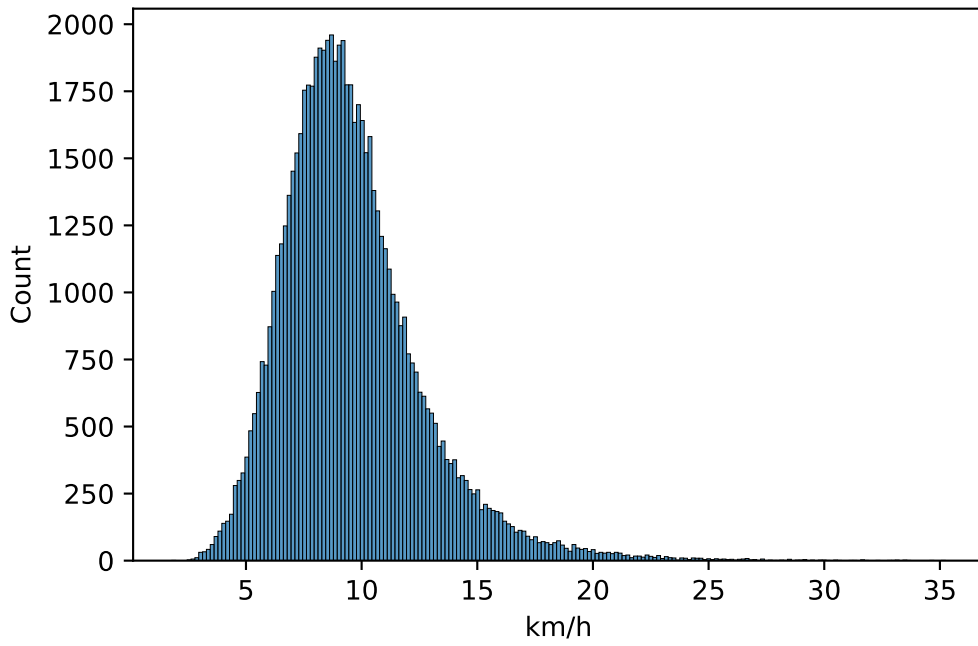


Figure 4.17: Distribution of average speeds of instances km/h.

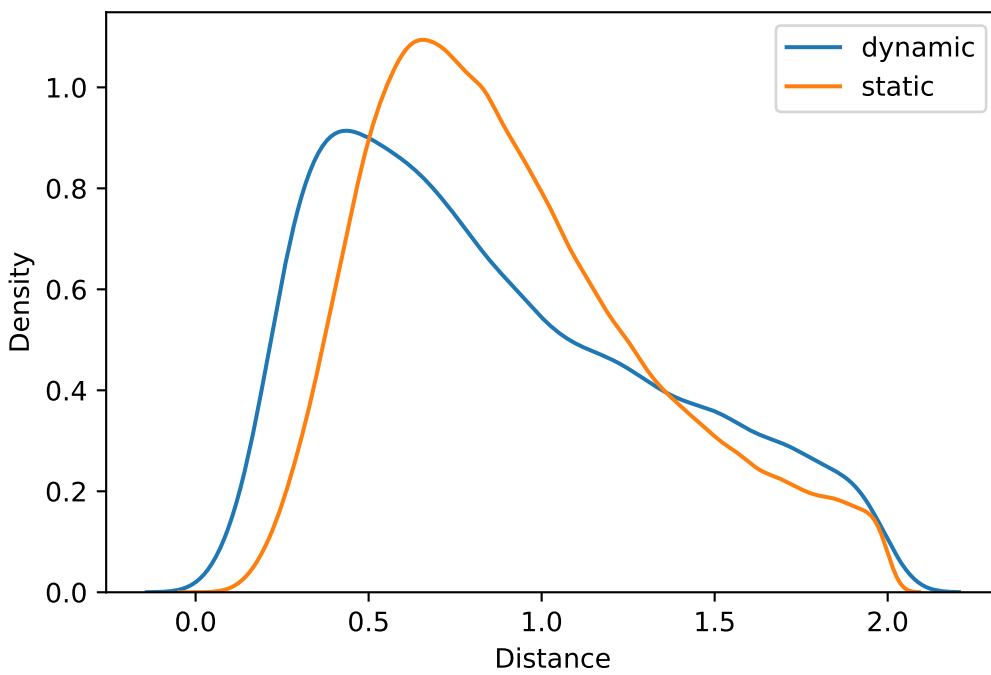


Figure 4.18: Evaluation of the movement of KITTI-360 instances through time with the chamfer distance.

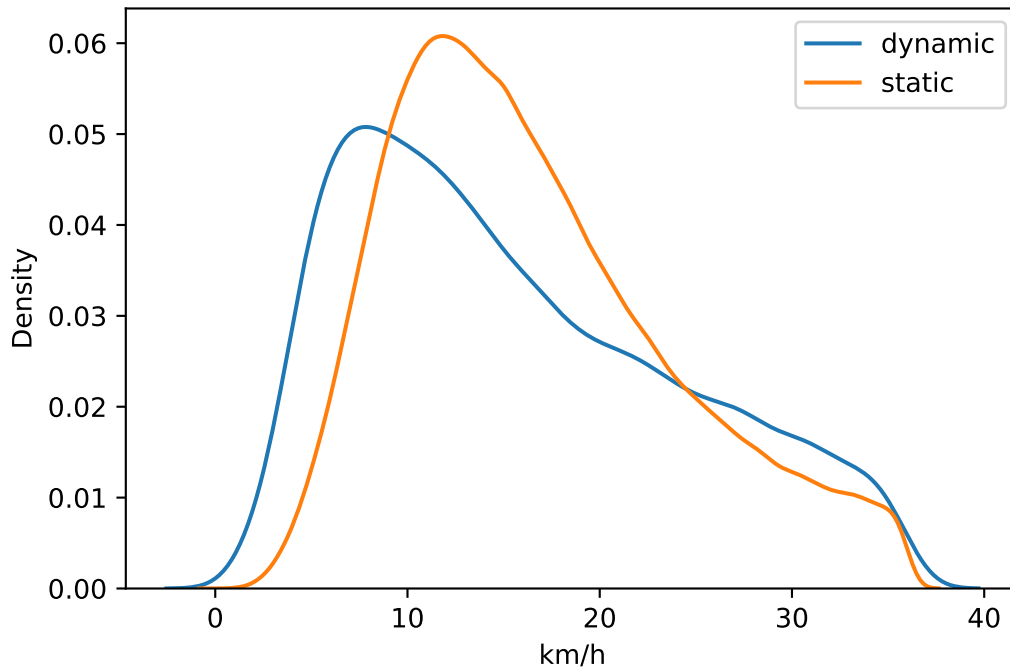


Figure 4.19: Evaluation of the movement of KITTI-360 instances through time in km/h.

dynamic with it. Whether the measured speed for static instances is too high is not decisive for whether flow estimation can be performed. Important is whether the measured velocity of static instances caused by the inconsistency is lower and well separable from that of dynamic instances. Since this is not the case, flow estimation based on the pseudo-lidar AdaBins cannot be made. The goal of RQ3 was to find out whether it is possible to do anomaly detection with a flow based approach on pseudo-lidar estimated point clouds. For this purpose we wanted to find out if pseudo-lidar estimated point clouds are suitable for flow estimation. Because the pseudo-lidar does not provide consistent enough point clouds through time, we cannot distinguish static and dynamic instances, so it is not possible to perform flow estimation on it.

5 Summary and Outlook

The focus of this contribution is to test if anomaly detection can be performed using pseudo-lidar point clouds and subsequent flow estimation. An anomaly can result primarily from the fact that an object is a novelty or that the pattern of the object has not been learned yet. Both cases were considered.

First, in RQ1, we analyzed if anomalies had higher deviations from lidar to pseudo-lidar than non-anomalies. Since there is no multimodal anomaly dataset, we had to work with another type of dataset. KITTI-360 was used, which has corresponding 3D point clouds and 2D images as well as corresponding instance labels. In KITTI-360, we defined anomalies as those instances that were not properly segmented in 2D. Then, for all instances in KITTI-360, the distance between pseudo-lidar and lidar was determined using the chamfer distance. Subsequently, it was analyzed whether this dissimilarity in the form of the chamfer distance is greater for anomalies than non-anomalies. In RQ1, it could be shown that the dissimilarity between the pseudo-lidar and the lidar data is quantitatively comparable for anomalies and non anomalies. As a result, we determined that the pseudo-lidar is capable to map anomalies that originate from an unknown pattern in 3D with similar performance as non-anomalies. Thus, the dissimilarity between lidar and pseudo-lidar is not suitable for detecting anomalies. It is important to note that for RQ1 the anomaly was defined by the instance segmentation performance. Only anomalies based on unseen patterns were found, and not novelties of unknown class. The challenge here is that an anomaly was defined over the cIoU threshold and thus, depending on the threshold, more or fewer objects are considered an anomaly. One important caveat is that the pseudo-lidar was trained on KITTI while the instance segmentation model, in our case a panoptic segmentation model, was trained on Cityscapes. Therefore certain patterns may be unseen for the pseudo-lidar but seen for the panoptic segmentation model and vice versa. This may have influenced the result of RQ1.

For RQ2 it was analyzed whether anomalies of type novelty can be mapped in 3D. Since no multimodal anomaly dataset is available, this question could not be evaluated with ground truth. Therefore, Monte Carlo Dropout was used as uncertainty estimation method and the pseudo-lidar estimated point clouds were qualitatively analyzed. The result was that for Fishyscapes, RoadObstacle21 and Lost and Found the network is not most uncertain on anomaly regions. For Fishyscapes and Lost and Found, the network is more uncertain for areas that correspond to Cityscapes classes than for areas that are anomalies. On RoadObstacle21, the used network was more uncertain for road than for anomaly. Therefore, one can conclude that the quantitative analysis does not give any reliable information about anomalies. In the qualitative analysis, it was possible to divide the depth prediction results into three categories. Anomalies that were available in KITTI as depth maps were mostly well-mapped. Due to blending, some anomalies from Fishyscapes were mapped ambiguously in 3D. It is therefore likely that participation in the Fishyscapes benchmark will not

be successful with an approach based on pseudo-lidar predicted point clouds. Most interesting were the predictions of the net for classes that the net has never seen. Those anomalies were mostly not mapped in 3D at all, and the neural network simply predicted the background of the anomaly in those places. This was true for the augmented dataset Fishyscapes as well as for the real dataset RoadObstacle21. Therefore, it is also suspected that participation with a pseudo-lidar approach to anomaly detection will not be successful in the SegmentMeIfYouCan benchmark, which includes the RoadObstacle21 dataset. One might test the hypotheses that finding anomalies via pseudo-lidar is only possible if either the network has enough data to generalize or the anomaly classes are present in the training data from pseudo-lidar. Therefore, it can be concluded that the network is overconfident and therefore anomalies cannot be detected using the uncertainty estimation approach with Bayesian Neural Networks. Finally, the qualitative analysis showed that the results of the quantitative analysis were dominated by overconfidence. The network mostly did not get the anomaly mapped in 3D and still the uncertainty for anomaly was not higher than for the other regions that were mapped correctly in 3D.

In RQ3, it was analyzed whether the depth estimation is consistent through time so that dynamic and static objects can be distinguished from each other via flow estimation. It has been shown that AdaBins delivers too noisy results and the noise will have such a big influence on the flow estimation that it will not enhance the capability in pseudo-lidar to improve the quality of anomaly detection.

In summary, we conclude that current pseudo-lidar models trained on public depth estimation datasets are not capable of helping to improve anomaly detection. However, it could be shown that anomaly detection using pseudo-lidar may work if sufficient data were available.

Therefore, future work could provide larger datasets so that the networks reach higher levels of generalization. Thereby, it is hoped that also unseen anomalies of type novelty can be mapped in 3D. The KITTI depth estimation dataset with its nearly 80,000 images is apparently not sufficiently suitable for this purpose. Furthermore, one could also evaluate whether neural networks that work on video data [91] are suspected to be capable of predicting depth more consistent and thereby enable their use for anomaly detection through flow. An alternative approach would be to remain in 2D and predict scene flow on images. This way, one could also look for contradictions with the panoptic segmentation model. Thereby, one could predict an anomaly if the flow estimation says a pixel is moving while the panoptic segmentation model says the pixel is of an unmoving class like street. The advantage of the pseudo-lidar over scene flow approaches is that the pseudo-lidar does not require hand-labeled datasets. On the other hand, the intermediate step via the pseudo-lidar could cause information loss, as has been shown in this contribution with the current approaches. Thus, the pseudo-lidar, when trained on very large datasets, could possibly provide better results with a self-supervised flow estimation model than a 2D based approach.

Acronyms

AP Average Precision. 7, 12

AP50 Average Precision With 50% IoU Threshold. 23

AUC Area Under The Curve. 11

BNN Bayesian Neural Network. 17

cloU Confidence-based Intersection Over Union. 21–24, 33, 35, 53, 59

CNN Convolutional Neural Network. 7

FPR False Positive Rate. 7

FPR95 False Positive Rate at 95% True Positive Rate. 7

GAN Generative Adversarial Network. 8

IoU Intersection over Union. 15, 21, 23, 61

kNN k-Nearest Neighbor. 48, 49

mIoU Mean Confidence-based Intersection Over Union. 22, 23

mIoU Mean Intersection Over Union. 7

PQ Panoptic Quality. 14, 15

RAM Random-Access Memory. 22

RMSE Root Mean Squared Error. 13

ROC Receiver Operating Characteristic. 11

RQ1 Research Question 1. v, vii, 17–21, 23, 53

RQ2 Research Question 2. v, vii, 17, 25, 59

RQ3 Research Question 3. v, vii, 17, 27, 29, 31, 54

RSME Root Mean Square Error. 13

A List of Figures

2.1	Systematic of corner case levels and sensor modalities adapted from Heidecker et al. [6].	4
3.1	Steps required to do instance-wise comparison of lidar and pseudo-lidar.	20
3.2	Illustration of different instances with associated panoptic segmentation maps predicted by SWideRNet-(1,1,4.5) and associated cIoU values.	24
3.3	Illustration of RQ2 and the steps taken to answer it. For each step it is shown which dataset is considered, and it is illustrated which results will be achieved by the step. First, a quantitative analysis with Monte Carlo Dropout is performed. Then a qualitative analysis is performed using the estimated point clouds to see how well the anomalies were mapped in 3D.	25
3.4	Theoretical approach to anomaly detection using a contradiction between motion segmentation in 3D and panoptic segmentation in 2D. For this purpose, an image is mapped in 3D by a pseudo-lidar. The dynamic points are then selected with the help of motion segmentation. These are then clustered by flow into moving objects. Then it is compared whether the pixels belonging to a moving object are segmented by the panoptic segmentation as something that can move. If not there is a contradiction and an anomaly is predicted.	27
3.5	Visualization of the transformation of the pseudo-lidar point cloud $Pseudo_t$ from camera coordinate system of t in the camera coordinate system of $t + 1$. Additional visualization of the pseudo-lidar point cloud from timepoint $t + 1$	30
3.6	Visualization of the areas that GndNet segmented as the ground on the pseudo-lidar point cloud. Segmented ground is colored as purple. Considered is KITTI-360 [82] sequence 0005 and time point 752.	30
4.1	Distribution of chamfer distances for anomalies and non anomalies at (cIoU, confidence) = (0.5, 0.8).	34
4.2	Distribution of object distances to the camera for anomalies and non anomalies at (cIoU, confidence) = (0.5, 0.8).	35
4.3	Number of instances for each class and corresponding average chamfer distance for both anomalies and no anomalies. The box plot shows the frequency of occurrence and the line plot shows the average chamfer distance. The classes are sorted according to the frequency of occurrence.	36
4.4	Average uncertainty of different classes on Fishyscapes Static.	38

4.5	Distribution of uncertainty from pseudo-lidar on Fishyscapes Static for in-distribution, anomaly and void regions.	38
4.6	All Pseudo-Lidar point clouds of Fishyscapes evaluation images from birds eye view. Anomalies in orange. Color gradient for z-axis for non-anomaly points. . .	39
4.7	Qualitative analysis of Fishyscapes Static image 11.	40
4.8	Qualitative analysis of Fishyscapes Static image 18.	41
4.9	Qualitative analysis of Fishyscapes Static image 12.	42
4.10	Qualitative analysis of Fishyscapes Static image 13.	42
4.11	Qualitative analysis of Fishyscapes Static image 1.	43
4.12	The average uncertainty of AdaBins on Lost and Found [23].	44
4.13	The average uncertainty of AdaBins on RoadObstacle21 [23].	45
4.14	Qualitative analysis of RoadObstacle21 image with dog as anomaly.	46
4.15	Qualitative analysis of RoadObstacle21 image with a soccer ball as anomaly. . .	47
4.16	Distribution of the average consistency through time, evaluated by the chamfer distance.	49
4.17	Distribution of average speeds of instances km/h.	50
4.18	Evaluation of the movement of KITTI-360 instances through time with the chamfer distance.	50
4.19	Evaluation of the movement of KITTI-360 instances through time in km/h.	51

B List of Tables

2.1	Fishyscapes benchmark results of two image based approaches. The best values are printed in bold [32] [31].	7
2.2	Benchmark of State-of-the-Art Panoptic Segmentation Models [35][63] [68]. . .	15
3.1	IoU analysis for different confidence thresholds τ . Additionally considered pixels for different confidence thresholds.	23

C Bibliography

- [1] Statistisches Bundesamt. Hauptverursacher von Unfällen mit Personenschaden. <https://web.archive.org/web/20220614095236/https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/hauptverursacher-fahrzeugart.html>, 2022. Accessed: 2022-05-19.
- [2] Tesla. A Tragic Loss. <https://web.archive.org/web/20220610053904/https://www.tesla.com/blog/tragic-loss>, 2016. Accessed: 2022-05-25.
- [3] On-Road Automated Driving (ORAD) committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2018.
- [4] Tesla. Tesla Autonomy Day. <https://www.youtube.com/watch?v=Ucp0TTmvq0E>, 2019. Accessed: 2022-05-09.
- [5] Mobileye Vision Technology Ltd. Mobileye Self-Driving Mobility Services. <https://web.archive.org/web/20220215060755/https://www.mobileye.com/mobility-as-a-service/>, 2022. Accessed: 2022-06-13.
- [6] Florian Heidecker, Jasmin Breitenstein, Kevin Rösch, Jonas Löhdefink, Maarten Bieshaar, Christoph Stiller, Tim Fingscheidt, and Bernhard Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021.
- [7] Daniel Bogdoll, Maximilian Nitsche, and J. Marius Zöllner. Anomaly detection in autonomous driving: A survey. *arXiv:2204.07974*, 2022.
- [8] Mana Masuda, Ryo Hachiuma, Ryo Fujii, Hideo Saito, and Yusuke Sekikawa. Toward Unsupervised 3d Point Cloud Anomaly Detection Using Variational Autoencoder. In *2021 IEEE International Conference on Image Processing (ICIP)*, 2021.
- [9] Hafsa Iqbal, Abdulla Al-Kaff, Pablo Marin, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. Detection of Abnormal Motion by Estimating Scene Flows of Point Clouds for Autonomous Driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.
- [10] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2018.

- [11] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.
- [12] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-set 3D Object Detection. *2021 International Conference on 3D Vision (3DV)*, 2021.
- [13] Jan-Aike Bolte, Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. Towards corner case detection for autonomous driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.
- [14] Fan Jiang, Junsong Yuan, Sotirios A. Tsafaris, and Aggelos K. Katsaggelos. Video anomaly detection in spatiotemporal context. In *2010 IEEE International Conference on Image Processing*, 2010.
- [15] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.
- [16] Oluwatoyin P. Popoola and Kejun Wang. Video-Based Abnormal Human Behavior Recognition—A Review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2012.
- [17] Jasmin Breitenstein, Jan-Aike Termöhlen, Daniel Lipinski, and Tim Fingscheidt. Systematization of corner cases for visual perception in automated driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [18] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling Out-of-Distribution Detection for Real-World Settings. *ICML 2022*, 2020.
- [19] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift. *GCPR 2019*, 2019.
- [20] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010.
- [21] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domínguez. WildDash - Creating Hazard-Aware Benchmarks. In *Computer Vision – ECCV 2018*, 2018.
- [22] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *International Journal of Computer Vision*, 129, 2021.
- [23] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

- [24] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *International Journal of Computer Vision*, 2021.
- [25] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [26] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, Xiaodan Liang, Zhenguo Li, and Hang Xu. CODA: A Real-World Road Corner Case Dataset for Object Detection in Autonomous Driving. *arXiv preprint: 2203.07724*, 2022.
- [27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [28] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [29] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Chunjing Xu, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021.
- [30] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMelfYouCan: A Benchmark for Anomaly Segmentation. *NeurIPS 2021*, 2021.
- [31] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. *IEEE/CVF International Conference on Computer Vision*, 2020.
- [32] Giancarlo Di Biase, Hermann Blum, Roland Y. Siegwart, and César Cadena. Pixel-wise anomaly detection in complex driving scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [33] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [34] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying Unknown Instances for Autonomous Driving. *3rd Conference on Robot Learning (CoRL 2019)*, 2019.

- [35] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling Wide Residual Networks for Panoptic Segmentation. *arXiv:2011.11675*, 2021.
- [36] Matthias Rottmann, Pascal Colling, Thomas Hack, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.
- [37] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.
- [38] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *Advances in Neural Information Processing Systems*, 2021.
- [39] Matthias Rottmann, Pascal Colling, Thomas Hack, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities. In *Conference: 2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.
- [40] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.
- [41] Alexey Dosovitskiy and Thomas Brox. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.
- [42] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015.
- [43] K. J. Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N. Balasubramanian. Towards Open World Object Detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [44] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.
- [45] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 1999.
- [46] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is NP-hard. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

- [47] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv:1512.03012*, 2015.
- [48] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [49] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [50] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *arXiv:2105.04206*, 2021.
- [51] Chen Zhang, Zefan Huang, Marcelo H. Ang, and Daniela Rus. Lidar degradation quantification for autonomous driving in rain. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [52] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.
- [53] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014.
- [54] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.
- [55] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [56] Minsoo Song, Seokjae Lim, and Wonjun Kim. Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [57] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [58] Zeyu Cheng, Yi Zhang, and Chengkai Tang. Swin-Depth: Using Transformers and Multi-Scale Fusion for Monocular-Based Depth Estimation. *IEEE Sensors Journal*, 2021.
- [59] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised Scale-consistent Depth Learning from Video. *International Journal of Computer Vision*, 2021.

- [60] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- [61] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.
- [62] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [63] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision (IJCV)*, 2021.
- [64] Lorenzo Porzi, Samuel Rota Buló, Aleksander Colovic, and Peter Kotschieder. Seamless scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [65] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [66] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [67] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Buló, and Peter Kotschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [68] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *Computer Vision – ECCV 2020*, 2020.
- [69] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [70] Qizhu Li, Xiaojuan Qi, and Philip H.S. Torr. Unifying training and inference for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [71] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

- [72] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [73] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Bjorn Ommer, and Andreas Geiger. SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [74] Daniel Bogdoll, Felix Schreyer, and J. Marius Zöllner. ad-datasets: a meta-collection of data sets for autonomous driving. In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2022.
- [75] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscnets: A large-scale benchmark for lidar panoptic segmentation and tracking. In *ICRA*, 2022.
- [76] How to use nusc and nuim at the same time.
<https://web.archive.org/web/20220614123650/https://forum.nuscenes.org/t/how-to-use-nusc-and-nuim-at-the-same-time/493>, 2020. Accessed: 2022-06-14.
- [77] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 2017.
- [78] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [79] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.
- [80] Jules Sanchez. recoverkitti360label.
<https://web.archive.org/web/20220525101943/https://github.com/JulesSanchez/recoverKITTI360label>, 2022. Accessed: 2022-05-24.
- [81] Finn Sartoris. Anomaly detection in lidar data by combining supervised and self-supervised methods. Bachelor’s thesis, Karlsruhe Institute of Technology (KIT), 2022.

- [82] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv preprint:2109.13410*, 2021.
- [83] Yair Kittenplon, Yonina C. Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [84] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.
- [85] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.
- [86] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.
- [87] Yunze Man, Xinshuo Weng, Xi Li, and Kris Kitani. Groundnet: Monocular ground plane normal estimation with geometric consistency. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [88] Ivan Tishchenko, Sandro Lombardi, Martin R. Oswald, and Marc Pollefeys. Self-Supervised Learning of Non-Rigid Residual Flow and Ego-Motion. In *2020 International Conference on 3D Vision (3DV)*, 2020.
- [89] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. PointPWC-Net: Cost Volume on Point Clouds for (Self-)Supervised Scene Flow Estimation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, 2020.
- [90] Himangi Mittal, Brian Okorn, and David Held. Just Go With the Flow: Self-Supervised Scene Flow Estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [91] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Trans. Graph.*, 2020.