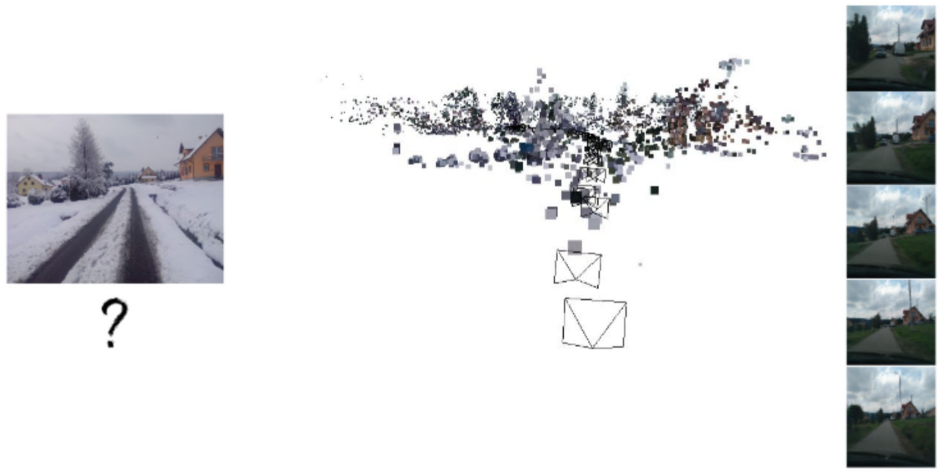
CHALMERS
UNIVERSITY OF TECHNOLOGY



Localization in changing environments
A semantic visual localization approach

Master’s thesis in Electrical Engineering

Ara Jafarzadeh

Department of Electrical Engineering
Chalmers University of Technology

Master's thesis 2020:EENX30

Localization in changing environments A semantic visual localization approach

Ara Jafarzadeh

Department of Electrical Engineering
Image Analysis and Computer Vision
Chalmers University of Technology
Gothenburg, Sweden 2020

Localization in changing environments
A semantic visual localization approach
Ara Jafarzadeh

Supervisor: Torsten Sattler, Department of Electrical Engineering
Examiner: Torsten Sattler, Department of Electrical Engineering

Master's Thesis 2020:EENX030
Department of Electrical Engineering
Image Analysis and Computer Vision
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of querying an image affected by seasonal changes against a scene, images from Mapillary.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Localization in Changing Environments : A Semantic Visual Localization Approach
Ara Jafarzadeh
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Visual localization is the problem of estimating the position and orientation of an image with regards to a reference scene. It is a fundamental capability for many applications in robotics and computer vision. Currently, most pose estimation methods are based on finding visual correspondences and they work well as long as the appearance of objects is the same in the different images. When there are drastic changes in illumination or view-point between image-pairs, it is rare to find algorithms being capable of reliable performance.

In this thesis, we have introduced a new benchmark dataset designed for investigating the current performance of visual localization algorithms under changing conditions and focusing on the failure cases of the existing algorithms. Furthermore, we have moved the problem of localization from 2D images to 3D scenes; we have proposed a method for 3D registration of two scenes instead of using single query images and have benefitted from semantic information of the point clouds for improving the performance. The results show promising results on wide-baseline localization.

Keywords: Localization, registration, benchmarking

Acknowledgements

First of all, I want to thank Torsten Sattler. His excellent supervision and great support not only helped me during this project but also made me fall in love with Computer Vision, and made me more confident that I want to continue doing research.
I want to thank the great team at Mapillary, Pau Gargallo, and Manuel Lopez for the great lead and support, and Yubin Kuang for his guidance.
This thesis is dedicated to my parents.

Ara Jafarzadeh, Gothenburg, August 2020

Contents

List of Figures

List of Tables

1

Introduction

Visual localization is the problem of estimating the position and orientation of an image with regards to a reference scene and is a fundamental capability for many applications in robotics and computer vision. Most methods [2–6] for doing 3D reconstruction from images or localizing an image with respect to a scene are based on finding visual correspondences. There are currently many methods [7–10] for finding visual correspondences between similar images. These methods work well as long as the appearance of objects is the same in the different images. When there are drastic changes in illumination or viewpoint between image-pairs, it is rare to find algorithms being capable of a good performance [11, 12].

There is a widely known trade-off [13] between discriminative power and invariance of the local descriptors. On the extreme sides of the discriminative power - invariance spectrum, a local descriptor that is constant is not discriminative but invariant, and taking an image patch as a descriptor is highly discriminative but not invariant. This makes the local descriptors only capable of handling small changes between the query and reference scene. At the same time, what is observed in outdoor localization is that due to the dynamic nature of the world around us, outdoor scene description should be robust to changes in illumination, seasons, etc.

There are only a few works [11, 12] in the literature showing the impact of changing conditions on 6DoF localization. One reason is that finding reliable poses that would work as ground truth for localization algorithms has also been challenging in the changing environments. Traditionally, [14–16] have used Structure-from-Motion pipelines to generate the 3D models of the scene and have used the generated poses as the accurate poses for localization. However, as the whole SfM pipeline is based on local feature description, the generated models are only able to include slight changes and with more dramatic changes, the pipeline fails to generate reliable poses . The datasets that provide reliable poses and show changing environments such as Aachen [11, 17], CMU Seasons [11, 18, 19], and Robotcar [11, 20] only cover a few challenges or have been geographically limited to one location in the world. In these datasets, for generating reliable poses, the authors have relied heavily on human work.

In this thesis, we have revisited multiple common scenarios, such as night, seasonal and viewpoint changes, that decrease the performance of localization. In this regard, in order to provide challenging scenarios for localization, we have mined our crowd-sourced data source for sequences that SfM fails. To improve the poses, we have relied on human annotations and have created 43 sets of query-reference sets with broader geographical coverage and more challenging scenarios compared to current benchmark datasets [11, 17–20]. We have applied state-of-the-art localization algorithms to measure their performance under these conditions and have analyzed how different challenges affect their performance. Additionally, we have proposed a method that uses the 3D point clouds of the scenes and its semantic information to generate more reliable poses in cases with strong viewpoint changes.

1.1 Overview

This thesis is organized as follows :

After a brief introduction in chapter 1, in chapter 2, we underline the most important properties of the current benchmark datasets and highlight the characteristics that make our proposed dataset different. Based on a thorough literature review, we have discovered the current limitations of existing benchmark datasets. Today, a geographically diverse dataset with reliable poses and showing real-world driving scenarios is lacking in the literature. Constructing such a dataset would require hundreds of people from all around the world to gather. We have addressed that by choosing Mapillary as our crowd-sourced database. By careful mining and selection of images from our database and improving the poses with manual annotations, we have been able to generate such dataset. The process of generation of this dataset is covered in chapter 3.

In chapter 4, we propose a new approach for automatically generating more reliable poses with moving from 2D to 3D. We have tried both monocular and multi-view depth estimation approaches for generating the point clouds. Also, we have benefited from the current 3D point cloud description networks [21] to match query and database scenes and later have improved the poses using the semantic information of the point clouds.

In chapter 5, we have analyzed how current algorithms perform under changing conditions and how we can address some of the challenges using our approach. Also by applying the current state-of-the-art localization methods [22–24] on our dataset, we have shown that the generated dataset is very challenging and can push the current research forward.

In the appendix, sample images from the curated dataset could be found.

2

Theory and Related Work

Visual localization is the problem of estimating the camera pose of an image relative to a scene representation and has been studied extensively. In this section, after explaining different approaches and previous usage of semantic information for improving the localization performance, we go through benchmarking visual localization. Later, as a part of this thesis benefitted from 3D point cloud description, we breifly visit 3D point cloud classification.

2.1 Visual Localization

Different methods have been derived by looking at localization from different perspectives. They can be categorized into the following :

1. Image-retrieval-based: Methods that use image-level descriptors for finding the closest image in the database and estimating the pose using that image, [25–29].
2. Local-feature-based: These methods use depth information and estimate the pose using 3D-3D correspondences, or 2D-3D matches when there is no information on depth for the query side [30–35]. These approaches build a 3D model of the scene and then estimate the camera pose based on 2D-3D matches between features in a test image and 3D model points [36–38]. Currently, these approaches surpass the performance of other methods for localization in changing environments. [22, 24, 39].
3. Learning-based: These methods represent the scene by a learned model and then predict the matches and do pose estimation [40–43]. The performance of these methods usually does not go beyond image retrieval methods [44]. In contrast, learning-based approaches that do not learn the full localization pipeline but only the 2D-3D matching part [45–50] can outperform feature-based approaches in terms of pose accuracy in small scenes. However, they currently do not scale well to larger scenes [50] and do not handle conditions not seen during training. Therefore, they are currently not used for localization in changing environments.
4. Sequence-based: Methods that utilize multiple images instead of single image for localization as image retrieval [51] or model the image sequence as a generalized camera [52].
5. Hybrid methods: Methods that combine image-retrieval, geometric and learning methods [39].

Table 2.1: A comparison between different localization datasets. CrowdDriven is the most diverse in term of scene types and changes in viewing conditions.

Dataset		Nordland [60]	Pittsburgh [61]	Tokyo 24/7 [62]	NCLT [63]	RobotCar Seasons [11, 20]	Aachen Day-Night [11, 17]	San Francisco [11, 64]	CMU Seasons [11, 65]	**CrowdDriven**
Scene Type	urban	✓	✓	✓	✓	✓	✓	✓	✓	✓
	suburban	✓				✓				✓
	natural	✓								✓
	road									✓
	indoors				✓					
condition changes	weather					✓			✓	✓
	seasonal	✓		✓	✓	✓			✓	✓
	strong viewpoint									✓
	day/night			✓			✓	✓		✓
	intrinsics								✓	✓
	snow								✓	✓
	rain					✓			✓	✓
sequential										✓
6DoF query poses							✓	✓	✓	✓
#images	reference	14k	254k	174k	3.8M	20k	3k	610k	7k	**1.3k**
	query	16k	24k	1k		12k	369	0.4k	75k	**1.7k**

2.1.1 Semantic Visual Localization

Multiple localization methods have benefitted from the semantic information of the scene for generating more reliable poses. One of the most recent works is the semantic visual localization method proposed in [53]. In the semantic visual localization (SVL) paper, after computing the semantic segmentations of every image associated with depth map, both segmented query and database images are voxelized and turned into semantic voxel maps, $M_{Database}$ and M_{Query} [54]. The goal of the localization is to estimate the transformation between the $M_{Database}$ and M_{Query} through 3D-3D matches. For this purpose, SVL learns an embedding function that outputs similar descriptors for similar subvolumes that is invariant to large viewpoint and illumination changes through semantic scene completion [55] as an auxiliary task. Later, the descriptors are used to create a bag of semantic word representation (BoSW) [56] of the database and query maps. Note that only descriptors of occupied subvolumes are considered in the BoSW representation. The matching of subvolumes is done via nearest neighbor search in descriptor space using the Euclidean metric, however, for a more efficient matching, an offline semantic vocabulary trained on the training dataset (the same one that is used for descriptor learning) is used. In this regard, only K = 5 nearest database words for each query word are considered candidates for matching. Given these matches, the best match in terms of geometrical and semantic alignment that aligns the query to the database map is chosen. Later, for improving the poses, ICP [57] is used.

Another approach for leveraging semantic information of the scene and improving localization performance is outlier removal. In this respect, Cohen et. al [58] use the reprojection on non-matching labels as a metric for quantifying the wrong alignment of the images. They project the 3D points of reference scene into images of the query scene and use the number of points projected to sky as an outlier metric and later choose the alignment with the least semantic violations (number of outliers). Toft et al. [59] take a similar approach and use semantic consistency for outlier filtering. In this respect, after generating a set of plausible poses for each match, they project the 3D points into 2D query images and count the number of matched labels between 2D and 3D points as a semantic consistency score and use it as a measurement for how likely it is that the match is correct. On 2D descriptor matching, [56] uses the semantic neighborhood to secure an exact match; this approach is explained further in the methods section.

2.2 Benchmarking outdoor localization under changing conditions

Tab. 2.1 provides an overview of visual localization datasets commonly used to measure visual localization performance under changing conditions. There are multiple published datasets that are used by the community,*e.g.*, the indoor 7 Scenes [45], 12 Scenes [66], and InLoc [67] indoor datasets or outdoor datasets such as Dubrovnik [30], Rome [68], Vienna [69], San Francisco [35,70], or Cambridge Landmarks [71] datasets, however, neither are tailored for measuring the impact of changing conditions.

There are also multiple place recognition datasets such as Nordland [60], Pittsburgh [61], and Tokyo 24/7 [62], however, they cannot be used for localization and also coarse-scale location information can be obtained relatively easily at scale via GPS measurements. On the other hand, getting accurate 6DoF poses under changing conditions is a difficult task and needs manual intervention as the current algorithms cannot handle strong changes in the scenes [11]. While current long-term localization datasets such as Aachen Day-Night [11, 72], CMU Seasons [11, 73], and RobotCar Seasons [11, 20] offer accurate ground truth poses, they only cover a few locations, in Europe and the United States, while our dataset has a wider range of coverage as can be seen in Fig. 3.1.

In addition to current localization datasets released by the Computer Vision research community, self-driving companies such as Lyft [74], Waymo [75], Aptiv (nuScenes) [76], Baidu (Apolloscape) [77] have also released outdoor datasets. However, the focus for most of these datasets have been 2D-3D object detection, semantic segmentation, depth estimation rather than localization.

Our goal for benchmarking has been to generate a dataset for long-term visual localization that (1) shows different challenging scenarios such as day-night, seasonal, daylight illumination, structural and viewpoint changes (2) with reliable poses refined using human annotations (3) using crowd-sourced data to imitate real driving scenarios, (4) with reliable calibration and geographical information. The procedure for generating this dataset is described in Section 3. Later, in the methods section, we present a method for finding more reliable poses than the 3D model that was built based on structure from motion pipeline run on those images.

2.3 3D Point cloud classification

Unlike 2D images, point clouds do not have any order and could be represented as sets of points. Following this property, the description of point clouds should also be order-invariant. Researchers used to transform point clouds to 3D voxels or collections of images (i.e. views); however, recent deep architecture models such as PointNet [78] model them as sets to make them permutation-invariant.

PointNet has three key modules: the max-pooling layer as a symmetric function to aggregate information from all the points, a local and global information combination structure, and two joint alignment networks that align both input points and point features. The architecture of PointNet is simple. The classification network takes n points as input and uses multi-layer perceptrons (MLP) for mapping from

3 dimensions to a 64 dimensions feature vector and later to 1024 dimensions. After taking the 3-dimensional network, a transformation network (T-Net) is used for estimating a transformation that aligns the points to a canonical space and makes it invariant to certain geometrical transformations (e.g., rigid transformation). Note that each of n points goes through the network. Later max-pooling is used to get a global feature vector of size 1024. A simple classification MLP then transforms the global feature vector to classification scores.

Following the success of PointNet, the authors also have proposed PointNet++ [79]. PointNet++ improves pre-steps to the classification network. In this regard, given n points, it samples and groups (clusters) the point cloud into K clusters. A PointNet is then applied to make a d-dimensional feature vector of each cluster member. This process is applied twice, and later another PointNet is used to get the classification scores. The 3D descriptor networks that are used in this thesis [21] use PointNet++ for point clustering.

3

Benchmarking Visual Localization : CrowdDriven

As explained earlier, current datasets are either not suitable for measuring the effect of changing environments on localization or only cover a few geographical locations. Additionally, if we take autonomous driving as our goal and look at the related datasets, the images in datasets such as the RobotCar Seasons [11, 20], and the (extended) CMU Seasons [11, 73] are usually taken using only a few cameras.
In this section, we provide an overview of CrowdDriven, our new benchmark dataset designed to have wide geographical coverage and sourced with crowd-sourced data. The dataset not only covers extreme changes, such as snow, but we have also visited moderate changes that still could lead to failure of the SfM pipeline and therefore could decrease the localization accuracy. Fig. 3.4 shows that the CrowdDriven contains images from each month of the year with a slight bias to the recent years, as we chose to include images with higher quality and more reliable metadata.

3.1 Data Source

Mapillary is a collaborative, street-level imagery platform that hosts images collected by community members while driving or walking on public spaces and roads. Images in Mapillary cover more than 190 countries with varying camera models at different times of day and weather conditions. They are well-suited to evaluate problems in conditions similar to those faced in self-driving scenarios since most images are

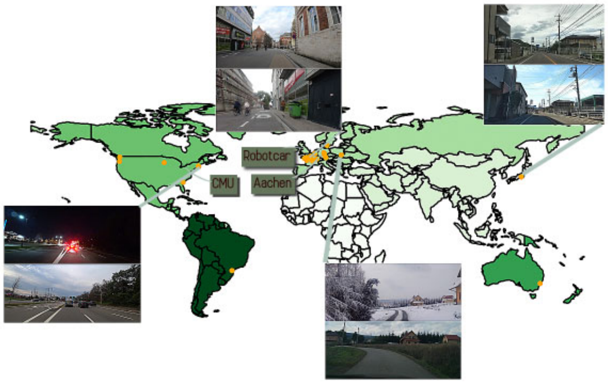


Figure 3.1: Geographical coverage of CrowdDriven.

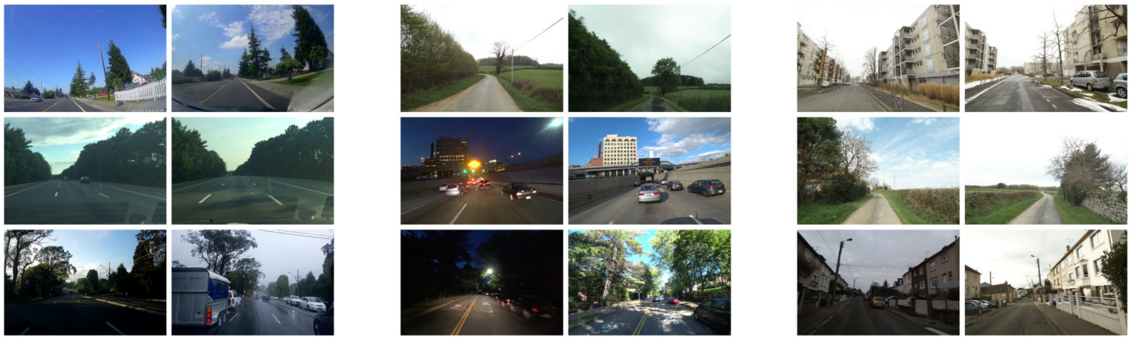


Figure 3.2: Left to right: Images from easy, medium, and hard datasets. For each category, we show test (left) and reference images (right).

captured with consumer-grade devices such as smartphones and dashcams mounted on vehicles traversing the scene with a forward motion. Thus, we have chosen Mapillary as our data source.

3.2 Sequence Selection

To generate a dataset that is challenging for current state-of-the-art algorithms, we have carefully selected the neighboring sequences that are taken at different times and show different conditions, such as illumination, seasonal, and viewpoint changes. We start by randomly selecting an image sequence as the seed sequence that is at least 0.2 images/meter dense, and has a length between 20 and 100 images. Note that we have consciously chosen to select smaller sequence-sets, as with today's consumer-grade GPS systems, it is reasonable to assume that we have acceptable pose priors, and the problem is mostly localization on a finer scale. Then, we select another neighboring sequence that is less than 3 meters apart from the seed sequence that satisfies the same density criteria. Assuming the viewing direction of the sequence as the average of raw compass angles of the images, we put the images based on the angle difference into three classes based on the failures we expect to see in the Structure-from-Motion pipeline:

(1) If SfM can accurately reconstruct the sequences in a common coordinate frame, we consider the pair as *easy*.

(2) If SfM fails and the sequences show similar viewing direction, we use manual annotations to correct the errors and classify the pair to be of *medium* difficulty.

(3) If SfM fails and the sequences show opposite viewing direction, we use manual annotations to obtain a joint 3D model and consider the pair as *hard*.

The sequence-sets that have an angle difference of less than 45 degrees would either show good localization or fail because of dramatic appearance changes, while the angle difference of more than 135 would likely result in failures caused by both appearance and viewpoint changes.

In the following, we describe how we reconstruct reference poses for our datasets. Examples for easy, medium, and hard datasets are shown in Fig. 3.2.

Table 3.1: Summary of the CrowdDriven dataset: scene type, number of test images, number of reference images, number of 3D points, average number of observations per image, reference model conditions, test conditions, changes and whether the scene is vegetated or not. Easy category: light gray, medium category: gray, difficult category: dark gray

scene type	identifier	#test	#ref. imgs.	#3D pts	#obs.	ref. conditions	test conditions	considerable changes	foliage
road	Sydney	10	22	2973	567	day, partly cloudy	day, rain,	illumination	
	Massachusetts1	10	29	1606	274	day, partly cloudy	day, overcast,	illumination	✓
	Poing	21	41	2902	272	day, clear sky	day, overcast	illumination	✓
	Washington	19	20	3164	678	day, clear sky	day, cloudy	illumination	
	Melbourne	14	28	1332	183	day, cloudy	day, overcast	illumination	
	Burgundy2	12	36	3096	616	day, sunny	day, rain	illumination, rain	
	Eden Prairie	26	26	4840	937	day, sunny	day, snow	small viewpoint, seasonal, illumination	
	Burgundy3	9	26	2046	386	day, sunny	day, rain	seasonal, rain	✓
	Thuringia	25	49	1717	201	day, sunny	day, clear sky	illumination	✓
	Massachusetts2	20	56	2848	209	day, overcast	night	day-night	
	Burgundy1	15	21	5175	1112	day, rain	day, overcast	rain	✓
	Besançon2	18	56	6207	494	day, overcast	day, cloudy	illumination, strong viewpoint	✓
	Besançon4	17	32	1396	168	day, cloudy	day, overcast	strong viewpoint, illumination	✓
	Besançon3	32	49	1860	205	day, overcast	day, vegetated, strong viewpoint	illumination	✓
	Brittany	50	50	4291	450	day, sunny	day, partly cloudy	strong viewpoint	✓
suburban	Portland	18	30	1292	153	day, clear sky	day, overcast	illumination	
	Curitiba	36	50	3724	373	day, cloudy	day, overcast	illumination	
	Tsuru	8	26	59	5	day, cloudy	day, overcast	illumination	
	Clermont-Ferrand	24	35	5774	986	day, sunny	day, overcast	illumination	
	Savannah	44	56	12900	963	day, clear sky	day, cloudy	illumination	
	Subcarpathia	48	52	11112	1358	day, cloudy	day, overcast	snow, seasonal	
	Massachusetts3	49	51	9691	1743	day, clear sky	night	day-night	✓
	Besançon1	31	40	6989	923	day, sunny	z day, sunny	seasonal, small viewpoint, illumination	
	Skåne	11	17	2537	604	day, cloudy	day	small viewpoint, illumination	
	Angers2	39	45	13369	1810	day, cloudy	day, clear sky	strong viewpoint, illumination	
	Ile-de-France	34	41	3895	439	day, sunny	day, cloudy	strong viewpoint, illumination	
	Orleans2	34	47	13470	2208	day, clear sky	day, sunny	strong viewpoint, illumination	
	Pays de la Loire	21	24	3798	723	day, cloudy	day, overcast	strong viewpoint	✓
	Brourges	33	34	7863	1088	day, partly cloudy	day, clear sky, illumination	strong viewpoint	✓
	Nouvelle-Aquitaine2	50	50	17118	1497	day, sunny	day, sunny	strong viewpoint	
urban	Muehlhausen	31	31	4752	755	day, cloudy	day, overcast	slight illumination	
	Bayern	47	49	17614	1720	day, cloudy	day, overcast	illumination	
	Boston5	20	21	1456	176	day, sunny	night	day-night	
	Boston1	50	50	6498	551	day, sunny	night	day-night	
	Boston3	50	50	11417	937	day, clear sky	day,clear sky	small viewpoint	
	Massachusetts4	50	50	5895	538	day, clear sky	night	day-night	
	Boston2	42	58	21161	1701	day, clear sky	night	day-night	
	Boston4	49	51	10887	952	night	day, sunny	day-night	
	Le-Mans	50	50	4710	472	day, overcast	day, overcast	illumination, strong viewpoint	
	Nouvelle-Aquitaine1	31	53	15911	1483	day, overcast	day, overcast	strong viewpoint, seasonal, snow	
	Angers1	22	23	3310	618	day, cloudy	day, clear sky	strong viewpoint	
	Orleans1	46	47	10835	1244	day, sunny	day, clear sky	strong viewpoint, illumination	
	Leuven	45	46	10258	877	day, cloudy	day, overcast	strong viewpoint	

3.3 Reference Pose Generation

For generating the reference poses, we have taken several steps: first, we have run SfM using OpenSfM [80] Open-source Library on the sequences to generate approximate poses, then using GPS priors, we have scaled the 3D reconstruction to a metric one. We have also divided the reconstructions into test and training sequences, based on the number of images in each sequence; the images from the smaller sequence are considered as the test (query), while the images from the bigger one, are the training (reference/ database) images. It is also noteworthy that as the chosen images are from **neighboring** sequences and the sequences are at least 8 meters long, the images see the same scene and have visual overlap.[1] Furthermore, all the sequences were annotated by individuals; therefore, enough correspondences could have been spotted by humans, meaning that the query scene was recognizable for humans.

In the following we go through each dataset category and the procedure for reference pose generation.

[1] In practice the length of sequences are considerably longer than 8 meters.

3.3.1 Easy Datasets

When the SfM pipeline can reconstruct both sequences in the same coordinate frame, we mark them as easy. Note that the sequences in this category still show challenging changes, such as snow; however, the structures in the scenes were distinctive enough that enough matches were found. Therefore, both sequences could have been registered in the same frame. After reconstruction, the resulting 3D scenes and the alignment of sequences were inspected[2], and only those with acceptable alignment and no noticeable misalignments by humans have been accepted. After the reconstruction, we use the GPS priors for (approximately) upgrading to a metric reconstruction. We expected and have observed later, that the current state-of-the-art algorithms are able to achieve satisfactory results in this category, see Tab. 5.1. Note that the Structure-from-Motion library used for performing reconstructions is OpenSfM [80]; however, we have later reconstructed the sequences with the COLMAP [81] library and have achieved similar results.

Tab. 3.1 provides statistics over the 13 datasets in the easy category (marked in light gray). As can be seen, challenges such as weather (snow or rain) and illumination changes have been visited under different environments such as roads, suburban and urban areas. Fig. 3.2(left) shows example images.

3.3.2 Medium and Hard Datasets

The changes between the query and database sequences of these categories are significant to an extent that SfM fails to register both sequences in the same frame. This means that they present a more significant challenge for current visualization algorithms and would be more interesting to the community.

Same as easy datasets, we run SfM pipeline on these sequences and use the GPS priors for metric upgrading. However, as SfM cannot align the sequences, we introduce manual annotations. In this regard, we have manually annotated visual correspondences between the sequences.

The annotated matching points are then triangulated to generate 3D points. Then, we estimate a rigid transform between these points and apply it to the query sequence to globally improve the alignment of the sequences. Later, we do bundle adjustment [82] to improve the poses. An example of sequences before and after pose refinement is shown in Fig. 3.3.

3.3.3 Manual Annotation

For recording the annotated matches between the query and database images, an annotation tool has been developed. In addition to recording the matches, it also hints the annotator with probable line that the match could be on using the epipolar geometry of the scene and is put in the OpenSfM Open Source Library [80] and could now be used by everyone. It is also noteworthy that manual annotation of these sequences was a time-consuming task, taking over one hour for each sequence; besides, we have discarded the sequences that are not well aligned after the annotation

[2] The visual inspection has been done by people with knowledge of 3D-vision.

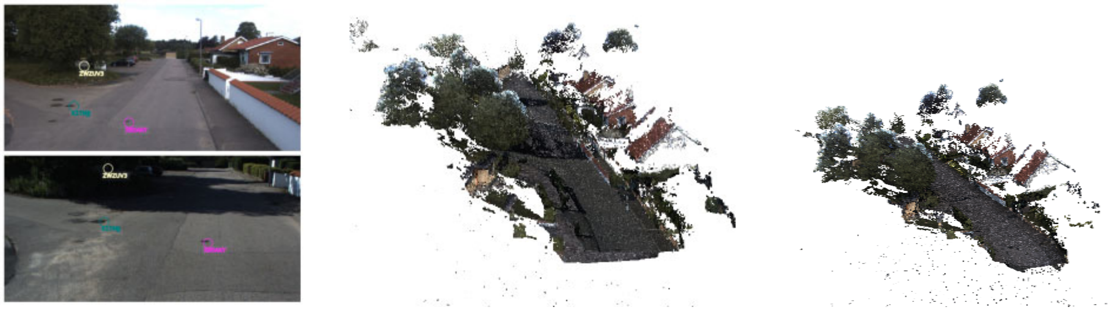


Figure 3.3: Left: To register the sequences in the medium and hard datasets, we manually annotated points in the images to create correspondences (shown together with automatically generated labels); Middle: Initial dense reconstruction of the scene; Right: Dense reconstruction with refinement after registration.

to guarantee high quality of reference poses.

Tab. 3.1 provides statistics over the 15 medium (gray) and 15 hard (dark gray) datasets in our benchmark set. As can be seen, nearly all the conditions that can challenge the current localization algorithms have been included in these datasets. In the medium category, conditions such as day-night changes, rain, snow and other seasonal variations that can impact the geometry of the scene, have been met under different scene types, while the main focus of the difficult category is to include large viewpoint changes in addition to less extreme variations in illumination, weather, and seasonal conditions. Fig. 3.2(middle) and (right) show examples for medium and hard datasets, respectively. More examples could be found in the appendix.

3.4 Reference Pose Verification

For the easy categories, that are those where SfM is able to register both sequences in the same frame, if there are no humanly noticeable artifacts and the sequences seem to be visually well aligned with respect to each other, we accept the poses estimated by SfM as the reference poses. However, in cases of medium and hard datasets, we needed manual annotations for generating the reference poses, and ways to measure the accuracy of poses after refinement [3]. For the sequences in these categories, in addition to the visual inspection step, we have developed some metrics to estimate the improvement of the reconstruction after the refinement. If a model fails these checks, we either annotate more correspondences or discard that model. We have used uncertainty estimation as well as reprojection errors as a metric for improvement of the reconstructions that will be described in the following paragraphs.

[3] It is noteworthy that manually-annotated correspondences are less accurate than SIFT features and only accurate up to 7 pixels.

3.4.1 Uncertainty estimation using reprojection errors

We have done a procedure similar to a 5-fold Leave-One-Out where we do bundle adjustment only using one-fifth of the points (training points) and we use the rest as the validation points, the metric is the reprojection errors. In this respect, we use training points for bundle adjustment as described in Sec. 3.3. We measure the reprojection error of the 3D points generated from the training points and compare it against the reprojection error of the 3D points generated from the validation points. Ideally, both errors should be relatively similar and within a reasonable range. However larger errors in reprojection error mean that the pose refinement only describes one part of the image that has had the training points on; and a high ratio indicates a higher uncertainty.
Tab. 3.3[4] [5] shows statistics of the reprojection errors for the training and validation ground control points on the medium and hard datasets. Considering that the images in our dataset have high resolution, we believe that the generated poses are accurate enough, however, in cases with higher validation/train ratio, such as Besancon2, a higher uncertainty should be taken into account.

3.4.2 Uncertainty estimation using uncertainty of bundle adjustment problem

To get an estimate on the uncertainty of the recovered camera positions, we compute the covariance of the solution given by the bundle adjustment problem [82] that includes SIFT and GCP correspondences.
To fix the gauge ambiguity [83, 84] and get metric estimates of the uncertainty, we fix the camera poses of the first sequence and optimize only the poses of the second sequence and the intrinsics of both. From the full covariance matrix, we look only at the 3x3 sub-matrices corresponding to the positions of the cameras of the second sequence. We then compute the size of the principal axes of the corresponding ellipsoid as a measure for uncertainty. Table 3.2 shows the median and maximum uncertainty of the shots in each dataset. Notice that the images with the largest uncertainty typically correspond to the last images in a sequence, *i.e.*, those images with the fewest constraints. As can be seen from the table, we measure smaller positional uncertainties for the hard datasets (dark gray rows). This is likely due to the fact that these datasets were created later in the benchmark creation process and that more time was spent on these datasets to ensure sufficiently many annotations.

3.5 Baselines

In order to show that our dataset introduces new challenges, we have tested the state-of-the-art localization algorithms on our dataset. We have chosen the algorithms

[4] Note that a reprojection error of tens of pixels is acceptable as it corresponds to about 1% of the image diagonal and due to the fact that our manual annotations are not pixel-accurate.

[5] The datasets where we were able acceptable alignment was achieved by only a small number of points were discarded in the table, as in these cases the test set was small and could have not been representative of the whole points.

Table 3.2: Max and median position uncertainties for the test images in each dataset (in meters). medium category: gray, difficult category: dark gray

names	max(m)	median(m)
Burgundy1	2.70	0.49
Burgundy2	1.19	0.29
Besançon1	3.25	0.26
Burgundy3	1.39	0.79
Eden Prairie	4.68	0.17
Massachusetts2	1.32	0.32
Massachusetts3	1.07	0.24
Boston1	1.71	0.88
Boston2	2.76	1.56
Boston3	1.83	1.04
Thuringia	0.43	0.25
Massachusetts4	0.96	0.40
Boston4	0.88	0.31
Boston5	0.17	0.17
Skåne	0.40	0.06
Orleans1	0.17	0.17
Nouvelle-Aquitaine1	0.17	0.17
Orleans2	0.20	0.06
Angers1	0.33	0.20
Leuven	0.10	0.04
Besançon2	0.70	0.10
Ile-de-France	0.28	0.08
Besançon3	0.21	0.08
Pays de la Loire	0.33	0.18
Le-Mans	0.26	0.10
Besançon4	0.17	0.17
Brittany	0.54	0.18
Brourges	0.24	0.14
Angers2	0.52	0.32
Nouvelle-Aquitaine2	0.47	0.11

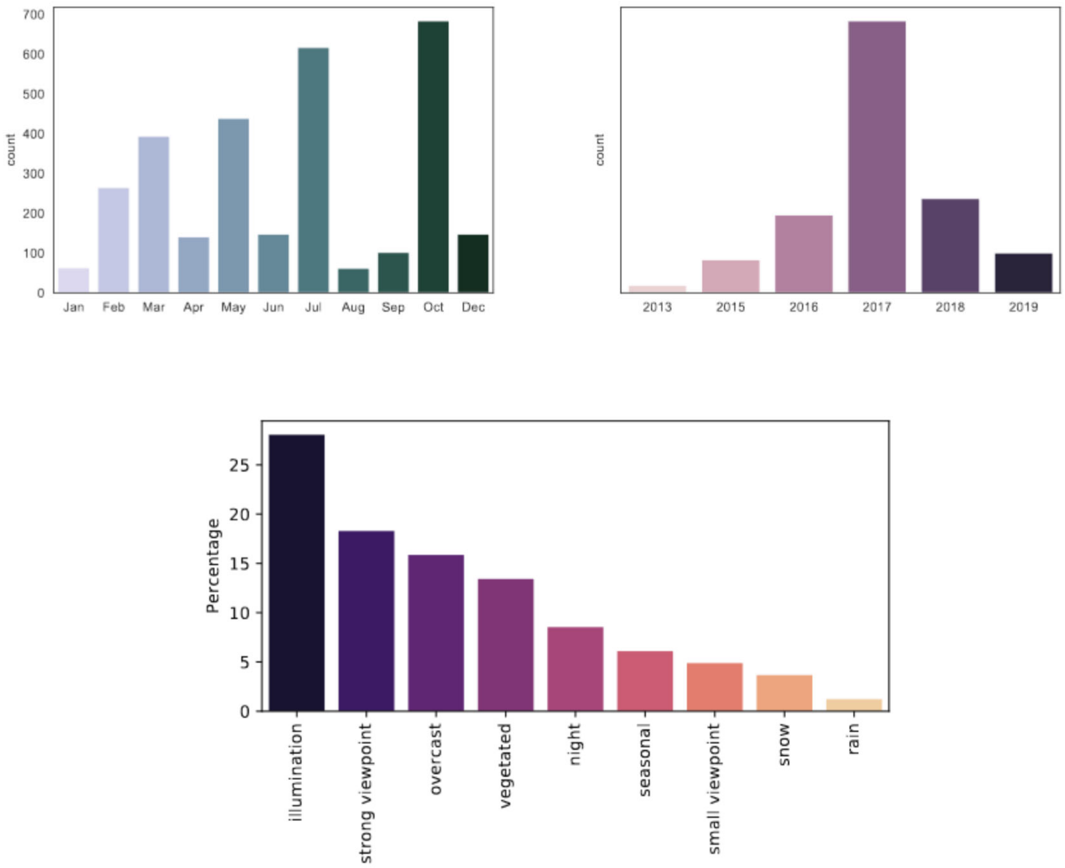


Figure 3.4: (a) The number of images taken each month in our dataset, (b) The number of images taken each year in our dataset, (c) Percentage of images that show each change in our dataset

Table 3.3: Mean reprojection error in training and validation set, the ratio between validation error train error, and the sizes of the images.

hard datasets						medium datasets					
reconstruction name	mean reproj. err. tr.	mean reproj. err. val.	ratio	size - ref.	size - test	name	mean reproj. err. tr.	mean reproj. err. val.	ratio	size - ref.	size - test
Angers1	10.88	14.43	1.33	(2048, 2448)	(2048, 2448)	Besançon1	3.89	13.16	3.39	(2050, 2448)	(2050, 2448)
Angers2	7.69	32.38	4.21	(2048, 2448)	(2048, 2448)	Boston5	13.35	17.51	1.31	(2160, 3840)	(2160, 3840)
Besançon2	0.33	8.84	26.43	(2048, 2448)	(2048, 2448)	Burgundy1	13.56	44.78	3.30	(2048, 2448)	(2048, 2448)
Besançon3	3.70	7.07	1.91	(2048, 2448)	(2048, 2448)	Burgundy2	3.89	19.59	5.04	(2048, 2448)	(2048, 2448)
Besançon4	1.56	4.62	2.96	(2048, 2448)	(2048, 2448)	Massachusetts2	7.11	13.10	1.84	(2160, 3840)	(2160, 3840)
Brourges	6.41	7.41	1.16	(2048, 2448)	(2048, 2448)	Skåne	1.13	1.96	1.73	(1024, 1360)	(960, 1280)
Ile-de-France	1.59	4.20	2.64	(2048, 2448)	(2048, 2448)	Thuringia	9.82	13.29	1.35	(3024, 4032)	(3024, 4032)
Le-Mans	3.87	5.70	1.47	(2048, 2448)	(2048, 2448)						
Leuven	3.37	4.31	1.28	(3000, 4000)	(3000, 4000)						
Nouvelle-Aquitaine1	9.29	9.76	1.05	(2048, 2448)	(2048, 2448)						
Nouvelle-Aquitaine2	4.79	7.66	1.60	(2048, 2448)	(2048, 2448)						
Orleans1	4.43	11.92	2.69	(2048, 2448)	(2048, 2448)						
Pays de la Loire	7.23	8.97	1.24	(2048, 2448)	(2048, 2448)						
mean	5.01	9.79	3.84	-	-	mean	7.54	17.63	2.57	-	-

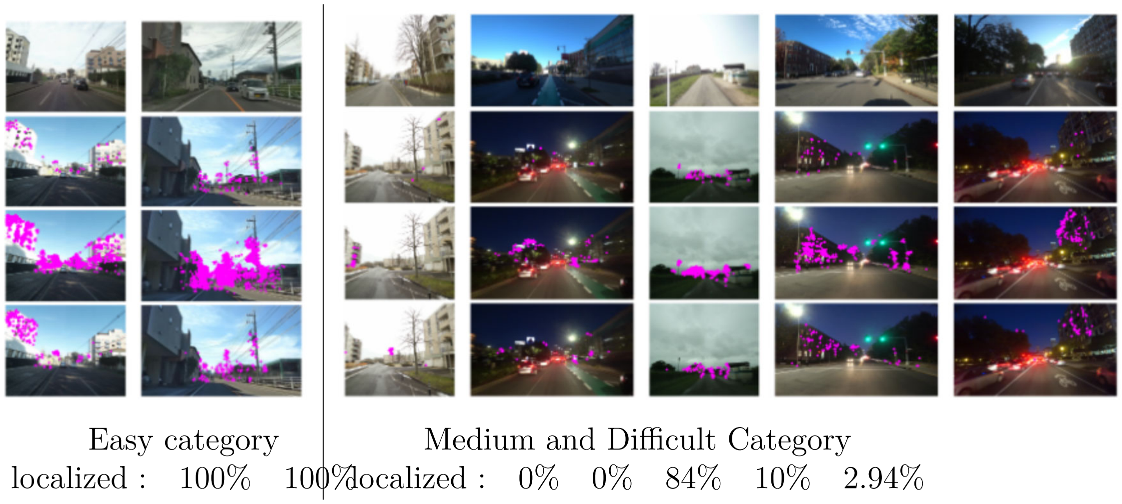

Easy category
localized : 100% 100%
Medium and Difficult Category
localized : 0% 0% 84% 10% 2.94%

Figure 3.5: Inlier plots of the evaluated methods (Top: closest reference image; Top-middle: HF-Net Results, Bottom-middle: D2-Net, Bottom: S2DHM), and the percentage of localized under the medium-precision regime for the best method.

based on the results on the current outdoor long-term visual localization under changing conditions benchmark [1] as well as availability of code. We have chosen to focus on the feature-based methods for localization. The reason behind is that recently it has been shown [44] that current pose regression [71, 85–87] approaches do not yet outperform the feature-based methods on complex scenes such as Aachen dataset [29].

3.5.1 SIFT

As the first baseline, we have implemented a simple localization pipeline using COLMAP [81], where we do pose estimation based on the 2D-3D SIFT-based [7] matches found exhaustively between the query image and the reference images. The query images are then localized using COLMAP image registerator and the estimated pose is used for evaluation.

3.5.2 HF-Net

HF-Net [39] is a state-of-the-art localization method approach that performs localization with a hierarchial approach. It uses a monolithic CNN for simultaneously predicting local features and global descriptors for accurate 6-DoF localization. The method first does a prior global retrieval,using MobileNetVLAD [88], for estimating the candidate locations, called the prior frames. The $K = 10$ nearest neighbors of the query image represent the candidate locations on the scene. Later the prior frames are clustered based on the 3D structure that they co-observe, and connected components are selected, called places. For each place, by 2D-3D matching between the query image keypoints, that are based on SuperPoint [8], and the 3D model points, a more accurate 6DoF pose is estimated using PnP [89].
In addition to achieving state-of-the-art results on the previous benchmark datasets, see Tab. 5.2, HF-Net can run faster than the other baselines, due to restrictions in the search space as well as using MobileNetVLAD [88] for faster retrieval.

3.5.3 D2-Net

The challenges confronted in our dataset, *e.g.* illumination, considerably change the number of detected keypoints, this is due to the fact that strong changes have significant impact on low-level information used by detectors. Traditionally, methods have followed a detect-then-describe approach [7, 8, 90–92]. However, D2-Net [24] uses a representation that is both a detector and descriptor for addressing the problematic keypoint detection in changing environments. For the detection stage, it applies a CNN on the input image and outputs a 3D tensor with the $h \times w$ resolution and n channels $\mathbb{R}^h \times w \times n$, the resolution of the output is one fourth of the input. During training, these descriptors are trained to produce similar descriptors for the same points in the presence of strong appearance changes. Each of n channels could also be thought of as a response map analogous to detectors such as Difference of Gaussians [93].
Traditionally, in order to sparsify the detections and avoid an interesting point being detected multiple times in neighboring regions, a non-maximum suppression is applied on top of the feature detector response. In D2-Net, this non-maximal suppression is done in two steps; first across the channels and then across the response map. In order to make the non-maximal suppression usable in backpropagation, the detection is *softened* by defining a local softmax function defined in the neighboring region for each point in the output.
This approach has made D2-Net more robust to strong changes between query and database scenes; in previous benchmark datasets [11,67], in combination with image retrieval, D2-Net have achieved state-of-the-art results, see Tab. 5.2. We have skipped the retrieval stage for testing on our dataset. The same pipeline as the SIFT baseline has been used, and SIFT was replaced with D2-Net features.

3.5.4 Sparse-to-Dense-Hypercolumn-Matching (S2DHM)

Sparse-to-Dense-Hypercolumn-Matching (S2DHM) [22] boosts the feature matching step in localization with extracting and exhaustively matching dense features on

the query side with the sparse features in the reference image. With doing dense feature extraction on the query side, it can bypass feature detection on the query images. This helps the method when the feature detection could be erroneous, such as in night images. In this method, the 2D coordinates on the reference and their corresponding 3D points are computed in an offline reconstruction stage. Also, same as HF-Net, it uses image retrieval as the first step of localization, where it retrieves the 30 first closest neighbors to each reference image.
Using this strategy for matching has been shown to achieve better performance under strong changes, especially day-night changes, see Tab. 5.2.

3.5.5 Sequence-based localization

In addition to single-image localization, we also utilize the sequential information of the images. Using known relative poses, we can model a sequence of images as a generalized camera [52], *i.e.*, a camera with multiple centers of projections. This enables the matching part to benefit from other detected matches in the other images and enables localizing images even if not enough matches are found in each. We use a solver [36] (inside a RANSAC [94] loop) for estimating both pose and intrinsic scale of the generalized camera. For each dataset, for modeling the generalized camera, $k+1$ images i to $i+k$ in the sequence are selected, for varying values of k. Therefore, each image is present in multiple generalized cameras; however, we select the pose using the generalized pose with the highest number of inliers.

4

Our Method

While images capture a 2-dimensional projection of the scenes, point clouds leverage the 3D spatial and geometrical information. When there are strong viewpoint changes in the query and database images, there might be only a small overlap between the projections in the query and database images. However, the 3D scene made by query and database images separately, preserve the captured similarities between the two scenes. Also, point clouds exhibit less variation than images and are less affected by illumination and seasonal changes [21].

Previously, Schönberger et al. [53] have proposed a method using semantic and geometric information of the scenes, which achieved promising results on several challenging large-scale localization datasets. Their method suggests learning the descriptors while using semantic scene completion as an auxiliary task, explained more in the Related Work section. Furthermore, the idea of using registration for localization has been previously investigated by Gilbaz et. al. [95], they proposed finding interesting points, called superpoints, by selecting overlapping spheres and then filtering non-salient or low-quality superpoints. The superpoints later are described by deep auto-encoders, and the descriptors are used for coarse matching between the point clouds. Later, ICP is applied to the point clouds for improving the poses.

We have used multi-view depth estimation for generating point clouds of both query and reference scenes. Later, using both deep-learned [21] as well as hand-crafted point descriptors [96], we have matched the points. Using the matches, we have fitted a rigid transformation that transforms the query point cloud to the reference point cloud. Following, using the semantic information of the point clouds, we have improved matching with ignoring matches with the wrong label.

4.1 Estimating depth from 2D images

The first works for generating denser point clouds in this thesis was about using the monocular depth estimation networks for assigning the depth of every point on the image. However, our earlier experiments have shown us that, right now, even the state-of-the-art monocular depth estimation neural networks cannot infer depths accurately enough to be comparable to multi-view depth estimation in our case. This probably might rise from the fact that our dataset, as opposed to landmark datasets, comprise of images of objects that are far away, where the accuracy of monocular depth estimation networks is less.

A brief description of both methods, *i.e.* monocular and multi-view depth estimation, is provided in this section.

4.1.1 Multi-view depth estimation

After doing simple Structure-from-Motion using the OpenSfM, we have used the semantic segmentations[1] to ignore the regions that are not of interest, such as sky, cars, etc. At this point, we have a sparse reconstruction of the scene with scale ambiguity. Having the images' GPS data, we can resolve the scale ambiguity and yield metric reconstruction (accurate up to GPS precision). Using poses from SfM, we run a simple PatchMatch [99,100] based multi-view stereo algorithm to get denser reconstructions and, thus denser depths. Later, we clean the depths by only keeping the depth values that are consistent with at least 3 neighbors. Figs. 4.1 and 4.3 show the depths estimated by multi-view depth estimation and corresponding 3D maps, only points that are seen by the image is shown.

4.1.2 Monocular depth estimation

For monocular depth estimation, we have used a network trained on the Mapillary Planet-Scale Depth Dataset [101] in an off-the-shelf manner. The network has a U-net [102] structure with a dilated ResNet-50 [103] that is trained on ImageNet [104] as the encoder. In the contraction stage of the architecture (encoding part), the image is reduced to a feature-map 16 times smaller than the input image. After the encoder, a DeepLabV3 [105] head is added to incorporate contextual information. The extracted feature map is then upsampled to the original size, and at every stage, features from the corresponding contraction stage are concatenated with it. The used loss function can be seen in eq. 4.1, with predicted depth map z, ground truth depth z^*, focal length f, and $d_i = \log(z) - \log(z^*)$. The loss is only evaluated on those pixels with known depth and n is the number of valid depth points in the image and λ is a scale-invariant hyper-parameter, set to $\lambda = 0.5$ here.

$$L(z, z^*, f) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2 \tag{4.1}$$

An example of the 3D map generated by the estimated depths by this method is shown in Fig. 4.2.

4.2 3D point cloud description

3D point cloud descriptors can be categorized into two classes : (1) Handcrafted point cloud descriptors [96, 106–108] and (2) Learned point cloud descriptors [21, 109–111]. Inspired by the success of 2D image descriptors such as SIFT, there have been multiple early works in designing handcrafted 3D point cloud descriptors, which were mostly proposed before the popularity of deep learning and are largely affected by domain knowledge. One of the most popular handcrafted 3D point cloud

[1] Semantic segmentations have been generated in an off-the-shelf manner, using a network with a similar architecture to [97] that has been trained on Mapillary Vistas [98].

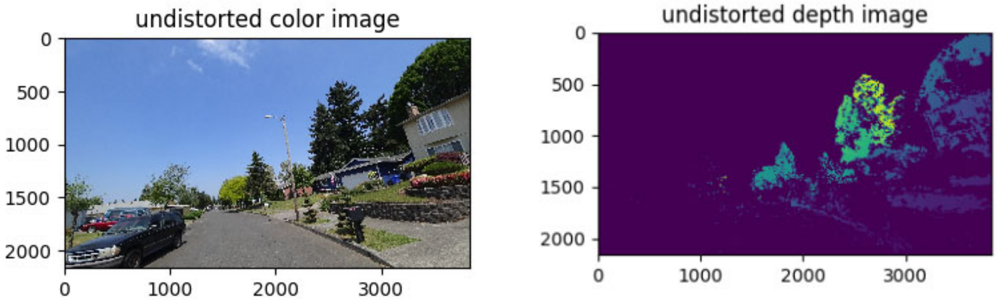


Figure 4.1: Undistorted image and its depth map, computed by the multi-view depth estimation method.

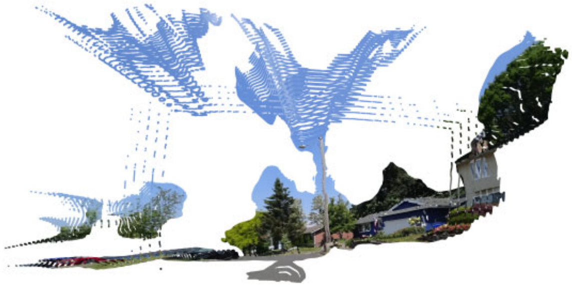


Figure 4.2: 3D map of an image, where the depths are estimated by monocular depth estimation network.

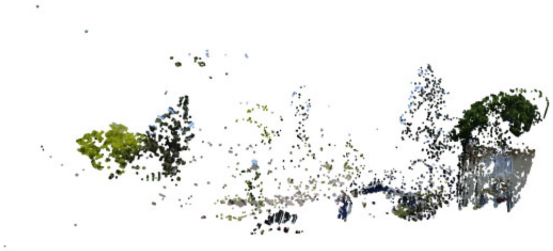


Figure 4.3: 3D map of an image, where the depths are estimated by the multi-view depth estimation method.

descriptors is FPFH [96]. FPFH uses mean curvature around a point's neighborhood as a distinctive property and puts it in a histogram.

However, compared to the success of SIFT, handcrafted descriptors mostly do not work well on real world and noisy point clouds [21]. Meanwhile, after advances in point cloud description, e.g. in PointNet [78] and PointNet++ [79], deep-learned descriptors have achieved better results [95, 110, 111]. In this regard, 3DMatch [109] uses 3D CNNs for learning and uses RGB-D indoor scenes with voxel input format. PPFNet [110] uses a PointNet architecture to get the descriptors. It takes local patches and computes point-pair features [112] between them and gives the neighboring points, normals, and point-pair features and gives as an input to the network instead of only coordinates. Also, it uses the global context for improving the descriptor matching. LORAX [95] uses deep learning for reducing the dimensions of its handcrafted descriptor. Among all these, neither use a detector; on the other hand, 3D-FeatNet [21] uses two networks for detection and description. It utilizes a week-supervised framework for generating feature correspondences where it benefits from GPS/INS tagged 3D point clouds, therefore we decided to use 3DFeat-Net as the deep point cloud feature detector and descriptor in our pipeline.

4.2.1 Deep point cloud description

Our first ideas were on trying to use the state-of-the-art point cloud classification networks, such as PointNet++, to use for point cloud description, training from scratch. However, in an ablation study, [21] suggests that solely using PointNet++, for point cloud description decreases the performance of the registration success[2] from 98.2 % (3DFeat-Net) to 48.6 % (PointNet++), and therefore current deep-learned descriptors would have dramatically surpassed that approach.

As for point cloud descriptor, we have chosen 3DFeat-Net, because of using both detection and description and the similarity of the dataset it has been trained on, i.e. Oxford Robotcar [20] and the loss function it uses for training. Using two networks for detection and description in a Siamese architecture setting, 3DFeat-Net uses feature alignment for mining for harder negative, and has given good results on commonly used benchmarks such as KITTI [113].

As advised in the paper, we have chosen the descriptor dimension to be 32, as based on the metric error at 95% recall [21], that dimension gives the best result with small number of dimensions.

4.2.2 3D point cloud matching

Having the point cloud descriptors, we use Euclidean distance as the metric and match each detected point from the reference point cloud against its nearest neighbor from the query point cloud in the descriptor space. Using a RANSAC loop and Euclidean distance between matched points as metric, we estimate a rigid transform that can register the query on the database point cloud with the most inlier matches.

[2] Success is defined as localizing within 2 meters and 5°

4.3 Using semantics for a more robust matching

We have used semantic labels to improve the matches. For labeling the point clouds with the correct semantic labels, we use the labels from 2D images. For getting the semantic segmentations in the 2D-level, we use a network similar to [97] trained on Cityscapes [114] and Mapillary Vistas [98] dataset in an off-the-shelf manner. The network uses a ResNet50 backbone with a Feature Pyramid Network [115] on the top. In this regard, we have taken two approaches: (1) We only accept the matches with similar labels. In this regard, after matching with the nearest neighbor, we reject the match in case of the labels are not same. (2) We use a a semantic twist approach [56]. In this respect, followed by [56], we take a 3D patch around our interest points. The semantic word of that patch is defined as the one-hot coded vectors of the semantic labels present in that region. In this regard, by assuming a sphere with a certain radius around our interest point, we put all the labels present within that sphere in a one-hot coded vector, where one means presence of that label. After testing with different sphere radii, the best trade-off between the accuracy of localization and speed was achieved with $50cm$ sphere radius, that is also close to radius training set patches of 3DFeat-Net that is $30cm$. From the segmentation network, we have 97 different classes, and the one-hot encoded vector theoretically could have $2^{97}-1$ different values; however, the actual present classes all considerably less. Later for matching the 3D points, we only accept those matches with equal semantic words. In the original paper, matching patches are done based on a bag of visual word approach; however, we only have tried simple nearest-neighbor matching as done in [21].

Although in 2D descriptor matching, the results of [56] is better than simple rejection of matches with dissimilar labels, we could not reproduce that in 3D and on our dataset and the result of the second approach is same the first one. The reason behind is that because of only having a few present semantic labels in our dataset, in most of the cases, taking the neighboring labels into account does not help with further rejection of wrong matches. It is likely that in case of a wrong match, when the label is same, the label of the neighboring point is same too.

5

Experimental Setup and Results

5.1 Experimental Evaluation

In this section, we evaluate the performance of the state-of-the-art localization algorithms on our dataset; in order to show that the new benchmark dataset introduces new challenges that are currently unsolved. We also evaluate the performance of our method on the difficult category, as the approach is likely to perform better in case of viewpoint changes.
After introducing the evaluation measures, we focus on the different types of challenges and how solved they are by the current state-of-the-art, as well as if moving to 3D can partly solve the problem. We identified three main conditions in our dataset; slight illumination changes, day-night changes, and strong viewpoint variations and have gone through each and how it can affect the localization performance.

5.1.1 Evaluation Measures

We present the accuracy by computing the distance between the estimated and reference poses. We follow the evaluation protocol of [11]; the position error is defined as the Euclidean distance between the camera centers, while we follow common practice [116] and compute the rotation angle α from $2cos(|\alpha|) = trace(R_{ref}^T R_{est}) - 1$, the minimum rotation angle needed for alignment of the rotations, where R_{Ref} is the reference and R_{est} is the estimated rotation.
Similar to [11], we introduce 4 levels of localization accuracy in order to reflect

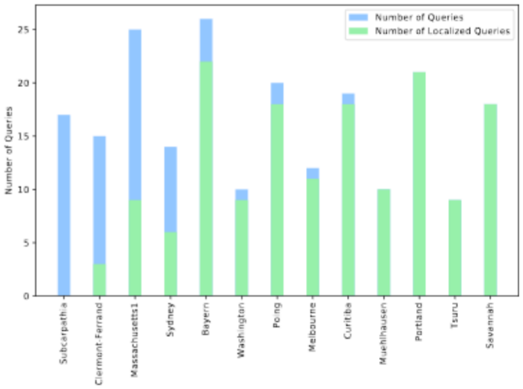


Figure 5.1: The number of images localized with COLMAP Image Registrator and total number of queries per dataset.

Table 5.1: Localization performance of the baseline methods on our CrowdDriven benchmark. We report the median position (in meters) and orientation (in degrees) errors, as well as the percentage of test images localized within certain error bounds on the position and orientation errors. Easy, medium, and hard datasets are color-coded in light, standard, and dark gray, respectively. The right side of the table provides information about the type of change between the training and test sequences: il.: illumination, fo.: foliage, sn.: snow, se.: seasonal, ng.: day-night, sm. v.: small viewpoint, rn.: rain, st. v.: strong viewpoint.

	S2DHM			HF-Net			D2-Net			Changes							
name	pos. err.	rot. err.	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	pos. err.	rot. err.	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	pos. err.	rot. err.	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	il.	fo.	sn.	se.	ng.	sm.v.	rn.	st.v.
Muehlhausen	0.29	0.33	80/100/100/100	0.09	0.18	100/100/100/100	**0.07**	**0.08**	**100/100/100/100**								
Tsuru	0.16	0.34	88.89/100/100/100	0.24	0.38	66.67/88.89/100/100	**0.07**	**0.06**	**100/100/100/100**	✓							
Poing	0.42	0.60	55/85/100/100	0.39	0.72	65/90/100/100	**0.11**	**0.06**	**100/100/100/100**	✓	✓						
Bayern	**0.16**	0.30	88.46/92.31/100/100	0.19	0.46	80.77/80.77/92.31/96.15	**0.16**	**0.14**	**96.15/100/100/100**	✓							
Savannah	0.31	0.33	72.22/88.89/100/100	**0.16**	0.17	**100/100/100/100**	0.24	**0.15**	83.33/100/100/100	✓							
Curitiba	0.39	0.38	78.95/94.74/100/100	**0.16**	0.29	84.21/100/100/100	0.26	**0.08**	**100/100/100/100**	✓							
Melbourne	**0.22**	0.27	**100/100/100/100**	0.35	0.53	66.67/91.67/100/100	0.38	**0.11**	**100/100/100/100**	✓							
Sydney	1.11	0.91	21.43/42.86/100/100	1.61	1.98	0/21.43/64.29/78.57	**0.41**	**0.23**	**78.57/100/100/100**	✓							
Clermont-Ferrand	**0.23**	0.50	**100/100/100/100**	0.37	0.91	86.67/93.33/93.33/93.33	0.43	**0.15**	73.33/100/100/100	✓							
Besançon1	90.39	77.41	0/0/0/0	89.76	131.84	0/0/0/0	**0.53**	**1.74**	**38.89/61.11/77.78/77.78**	✓			✓		✓		
Massachusetts1	1.83	0.68	0/32/68/92	3.90	1.22	4/24/52/60	**0.58**	**0.26**	**44/64/100/100**	✓	✓						
Boston4	34.83	26.70	11.76/17.65/41.18/41.18	35.36	96.44	14.71/17.65/23.53/26.47	**0.81**	**0.82**	**14.71/64.71/100/100**					✓			
Portland	0.68	**0.52**	28.57/80.95/100/100	**0.54**	0.54	**47.62/76.19/100/100**	0.83	0.34	19.05/66.67/100/100	✓							
Burgundy2	0.60	0.74	42/84/100/100	**0.49**	**0.71**	**50/84/98/100**	0.89	10.09	0/0/14/100	✓						✓	
Washington	19.47	1.59	0/0/0/0	5.41	3.94	**10/20/50/70**	**1.06**	**0.17**	0/20/90/100	✓							
Skåne	12.91	8.29	0/0/14.29/38.10	75.71	96.35	23.81/28.57/42.86/47.62	**1.74**	**3.55**	**0/4.76/85.71/95.24**	✓					✓		
Burgundy1	3.53	**8.53**	0/0/71.88/96.88	4.40	8.86	0/0/50/81.25	**1.83**	11.25	**0/0/0/100**		✓						
Burgundy3	4.47	3.08	0/13.89/63.89/86.11	12.42	9.27	0/0/25/38.89	**2.03**	7.50	**0/0/100/100**		✓		✓				
Subcarpathia	5.96	4.70	0/5.88/47.06/52.94	5.24	4.09	0/17.65/47.06/70.59	**2.06**	**1.50**	**17.65/41.18/94.12/94.12**			✓	✓				
Massachusetts4	**1.33**	11.38	0/0/0/97.44	2.33	11.78	0/0/0/64.10	2.22	**3.89**	**0/10.26/94.87/100**					✓			
Massachusetts2	246.10	94.89	0/0/0/0	98.83	133.94	0/0/0/0	**5.73**	**4.49**	**4.17/4.17/20.83/41.67**					✓			
Massachusetts3	683.86	121.47	**0/0/6.82/6.82**	504.57	123.50	0/0/2.27/4.55	**6.53**	**29.55**	0/0/0/0		✓			✓			
Boston2	5873.53	113.46	0/0/2.04/2.04	32.61	43.39	0/0/4.08/14.29	**7.67**	**5.08**	**0/0/4.08/85.71**					✓			
Brittany	201.63	**143.33**	0/0/0/0	22.43	162.19	0/0/0/0	**14.79**	146.82	0/0/0/0		✓						✓
Boston5	**30.57**	5.35	**0/2.94/35.29/35.29**	81.97	120.67	0/0/0/0	**15.04**	9.78	0/0/0/26.47					✓			
Leuven	58.52	146.80	0/0/0/0	52.11	**154.32**	0/0/0/0	**15.27**	154.70	0/0/0/0								✓
Boston3	302.16	137.32	0/3.23/6.45/16.13	19.68	138.77	0/0/3.23/16.13	**16.42**	**6.66**	**0/0/16.13/32.26**	✓					✓		
Boston1	146.87	117	0/0/18.75/18.75	35.17	108.03	0/0/6.25/8.33	**16.50**	**4.59**	**0/0/2.08/25**					✓			
Brourges	60.37	162.98	0/0/0/0	37.31	**149.82**	0/0/0/0	**17.22**	176.58	0/0/0/0		✓						✓
Orleans1	26.45	176.72	0/0/0/0	48.49	**149.82**	0/0/0/0	**17.64**	178.25	0/0/0/0	✓							✓
Thuringia	**0.82**	**0.72**	**18.18/63.64/100/100**	1.99	1.58	18.18/27.27/72.73/90.91	18.16	132.50	0/0/36.36/45.45	✓							
Nouvelle-Aquitaine2	21	170.39	0/0/0/0	83.22	**130.89**	0/0/0/0	**20.16**	155.51	**0/0/4.44/6.67**								✓
Angers1	165.33	144.33	0/0/0/0	42.75	**130.50**	**0/2.13/4.26/6.38**	**23.02**	178.14	0/0/0/0								✓
Pays de la Loire	36.93	172.02	0/0/0/0	24.08	**149.13**	**0/2.38/2.38/2.38**	**24.06**	177.26	0/0/0/0		✓						✓
Angers2	564.12	**143.75**	**0/0/2.17/2.17**	89.19	152.95	0/0/0/0	**24.13**	166.39	0/0/0/0	✓							✓
Orleans2	224.69	164.96	0/0/0/0	34.34	**137.63**	0/0/0/0	**30.12**	172.52	**0/0/3.23/3.23**								✓
Ile-de-France	78.55	156.45	0/0/0/0	88.67	**117.09**	0/2/2/2	**32.07**	158.74	**0/6/10/10**	✓							✓
Besançon3	102.97	**145.03**	0/0/0/0	81.76	148.22	0/0/0/0	**32.44**	167.99	0/0/0/0	✓	✓						✓
Nouvelle-Aquitaine1	288.45	157.33	0/0/0/0	60.70	**144.84**	0/0/0/0	**34.13**	177.58	0/0/0/0			✓	✓				✓
Le-Mans	46.16	162.29	0/0/0/0	45.90	**149.95**	**0/0/0/2.04**	**34.46**	169.44	0/0/0/0	✓							✓
Besançon2	73.58	158.38	0/0/0/0	69.93	134.47	0/0/0/0	**43.47**	**172.62**	0/0/0/0		✓						✓
Besançon4	3137.38	138.70	0/0/0/0	303.66	124.22	0/0/0/0	**53.02**	**163.24**	0/0/0/0	✓	✓						✓
Eden Prairie	-	-	-	110.88	150.73	0/0/0/0	**56.53**	**101.90**	0/0/0/0	✓		✓	✓		✓		

the accuracy required for autonomous driving : (1) high (2) medium (3) coarse (4) very-coarse with upper bounds : 0.5, 1, 5, 10 meteres for position error and 2, 5, 10, 20 degrees for rotation error, respectively.

In order to represent how the changes have affected the localization performance, we also present the changes seen in the scenes. As could be seen in Tab. 5.1, the main challenges that are observed in our benchmark datasets are slight illumination, day-night, and strong viewpoint changes. Also, we have sorted the table by D2-Net errors, which is mostly the method with better results, for observing how changes affect the errors. We also have analyzed how any of these changes have affected the pose errors.

5.1.2 Slight Illumination Changes

When the changes between reference and query images are only slight illumination changes, such as in Muehlhausen, Tsuru, Poing, Bayern, Savannah, Curitiba, Melbourne, Sydney, and Clermont-Ferrand, we can see that almost all the methods can perform high-precision localization (see rows 1 to 9 of Tab. 5.1 and figure 5.2). Also, the descriptor vector normalization that is mostly done in handcrafted descriptors

Table 5.2: Baseline results on the previous benchmark datasets [1].

Method	Condition	D2-Net + (Net/Dense)VLAD	S2DHM	HfNet
Aachen day-night	Day	84.8 / 92.6 / 97.5		80.5 / 87.4 / 94.2
	Night	43.9 / 66.3 / 85.7		42.9 / 62.2 / 76.5
CMU Seasons	Urban			91.7 / 94.6 / 97.7
	Suburban			74.5 / 81.5 / 91.3
	Park			54.3 / 62.5 / 79.0
Extended CMU-seasons	Urban	94.0 / 97.7 / 99.1	65.7 / 82.7 / 91.0	89.5 / 94.2 / 97.9
	Suburban	93.0 / 95.7 / 98.3	66.5 / 82.6 / 92.9	76.5 / 82.7 / 92.7
	Park	89.2 / 93.2 / 95.0	54.3 / 71.6 / 84.1	57.4 / 64.4 / 80.4
Robotcar	Day	54.5 / 80.0 / 95.3	46.4 / 77.6 / 95.1	53.1 / 79.1 / 95.5
	Night	20.4 / 40.1 / 55.0	30.0 / 68.8 / 94.6	7.2 / 17.4 / 34.4

such as SIFT enables the descriptor to be more robust to illumination changes as can be seen in the mentioned datasets. We can conclude that having slight illumination changes is not a challenge for current algorithms.

5.1.3 Day-night Changes

Although the small illumination changes seem to be a solved problem, when the illumination changes are more significant, as seen between day and night, most methods perform poorly when localizing night images, as seen in Figure 5.2. The best results for datasets that are affected with day-night illumination changes are mostly achieved by D2-Net, in some of the datasets such as Massachusetts4 and Boston4. This is expected as the descriptor is designed to be more robust to illumination changes. However, a big problem in night images is the presence of artificial lights that would either mask or considerably change the appearance of the features found, making D2-Net fail on Massachusetts2, Massachusetts3, Boston2, and Boston5.

The errors of methods such as HFNet and S2DHM are considerably higher, as seen in Tab. 5.1. While matching between query and reference images in day-night cases, the inlier ratio is usually considerably low. Taking approaches such as dense matching does not work as long as the extracted features are not describing the same feature in the same way in illumination changes.

5.1.4 Strong Viewpoint Changes.

The most challenging visited condition for both single and multi-image localization has been strong viewpoint changes. For both single and multi-image queries, the orientation errors are always above 160°, meaning a complete failure to localize the images.

When the reference and query images look at the scene from opposite directions, the appearance of objects could totally change; subsequently, a significant decrease in the number of matches is expected. Also, as the field of view of the images is less than 90 degrees, the overlap of the scenes could also be small. The overlaps are also on smaller scales.

According to Tab. 5.1 and Tab. 5.4, as long as the only change visited in a scene is strong viewpoint changes, and both sequences have been captured at the same time of the day and the scene is not very vegetated, D2-Net performs better than the

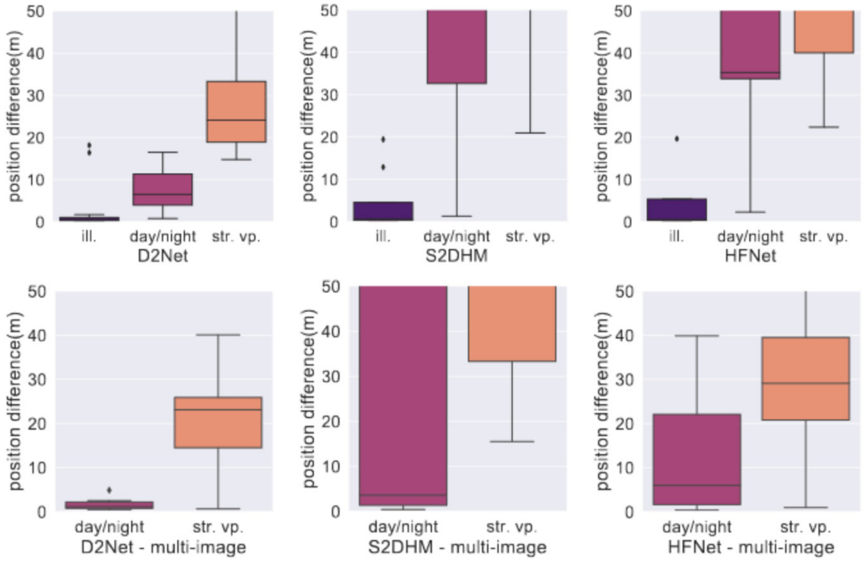


Figure 5.2: Position errors under different conditions, Top plots : single-image localization, Bottom plots : multi-image localization. Left : D2-Net, Middle : S2DHM, Right : HF-Net (the plots has been cut at 40 meters as error more than 40 meters is not of interest.)

other algorithms, in cases such as Brittany and Leuven, although the pose error is still high. However, when the strong viewpoint change co-occurs with illumination changes or the presence of foliage, the errors rise considerably, resulting from the decrease in a previously small number of matches and leading to a complete failure in localization. Even with taking the multi-image localization approach, the position errors still do not reach the very-coarse precision. This has motivated us to move from 2D to 3D.

5.1.5 Our method

Using multi-view depth estimation and moving from 2D to 3D and then doing 3D registration, we can resolve localization for a third of sequences in the hard category. Given the nature of our reference scenes and the fact that there is only small rotations between the images of each scene, the 3D points that are constructed from either scene might only co-observe *e.g.* the facades of building, and not contain enough overlap in the point cloud. This could result in failure in most of the cases. Approaches such as using scene completion [55] taken by [53], is expected to improve this problem. One other big problem, also observed in 2D, is repetitive textures, as can be seen in Fig. 5.3.

Using semantic data for ignoring the wrong matches has improved our result, see Fig. 5.6. However, although semantic twist [56] has shown promising results in 2D, we observed that this approach does not surpass simple semantic matches result on our dataset, because of the nature of scenes and the fact that the labels are not as diverse as the test cases presented in [56]. Therefore, only results of simple semantic matching is shown in Tab. 5.3.

Comparing our approach to the state-of-the-art localization methods, interestingly

Table 5.3: Localization performance of our pipeline, using 3D-FeatNet and FPFH descriptors, on difficult part of CrowdDriven benchmark. We report the median position (in meters) and orientation (in degrees) errors, as well as the percentage of test images localized within certain error bounds on the position and orientation errors. The right side of the table provides information about the type of change between the training and test sequences: il.: illumination, fo.: foliage, sn.: snow, se.: seasonal, ng.: day-night, sm. v.: small viewpoint, rn.: rain, st. v.: strong viewpoint.

	3D-FeatNet			FPFH			conditions								
name	pose	rot_err	percentage	pose	rot_err	percentage	il.	ov.	fo.	sn.	se.	ng.	sm.v.	rn.	str. v
Le-Mans	0.23	0.35	87.76/95.92/100/100	3.96	0.66	0/0/100/100		✓							✓
Ile-de-France	0.30	5.28	0/0/100/100	133.10	89.43	0/0/0/0									✓
Besançon2	0.73	0.93	14.00/100/100/100	5.99	1.76	0/0/0/100			✓						✓
Besançon3	1.00	0.40	0/50/100/100	1.40	2.38	0/0/100/100			✓						✓
Besançon4	1.11	0.77	0/24.00/100/100	3.86	0.93	0/0/97.96/100		✓	✓						✓
Pays de la Loire	2.43	0.61	0/0/100/100	76.07	151.98	0/0/0/0			✓						✓
Leuven	15.79	10.33	0/0/0/0	46.45	175.14	0/0/0/0		✓							✓
Brourges	20.34	11.62	0/0/0/0	48.73	179.77	0/0/0/0			✓						✓
Brittany	23.87	176.88	0/0/0/0	35.72	4.82	0/0/0/0			✓						✓
Nouvelle-Aquitaine2	26.36	1.39	0/0/0/0	1.17	0.64	0/16.00/100/100									✓
Angers2	32.65	8.61	0/0/0/0	24.06	136.89	0/0/0/0									✓
Orleans1	33.98	179.81	0/0/0/0	0.64	0.35	0/100/100/100									✓
Angers1	46.01	177.95	0/0/0/0	101.90	46.67	0/0/0/0									✓
Orleans2	46.01	90.26	0/0/0/0	34.43	157.70	0/0/0/0									✓
Nouvelle-Aquitaine1	51.11	168.72	0/0/0/0	34.95	178.98	0/0/0/0				✓	✓				✓

the cases that are *solvable* by 3D point cloud matching, such as Besançon2, Besançon3, Besançon4, are the cases with vegetation, however, with a rather static geometry in the query and reference scene. We believe that the improvement in the result has risen from the denser point cloud of these scenes because of its more textured nature of vegetation. As this is not the general case for vegetation, we can not conlcude that this method will always work well for vegetated scenes.

In 3DFeat-Net [21], the authors have tested their method on the LIDAR data of Oxford Robotcar [20] that contains challenging conditions including day-night changes. Although no specific experiment was done on the method's robustness to these changes, the figures (see figure 1 in the original paper) show that the method has some robustness to these changes on the LIDAR data. In the early attempts, we have also tried to test the method on the datasets with strong illumination changes, such as day-night changes seen in the Massachusetts2, Massachusetts3; however, as the point-clouds are computed from the images, the night point clouds are still sparse, so we could not reproduce the results using the point clouds from multi-view point clouds.

Table 5.4: Performance of the baseline methods on our CrowdDriven benchmark when using multi-image localization. We report the median position (in meters) and orientation (in degrees) errors, as well as the percentage of test images localized within certain error bounds on the position and orientation errors. We report results for different sequence lengths. Medium and hard datasets are color-coded in standard and dark gray, respectively. The right side of the table provides information about the type of change between the training and test sequences: il.: illumination,ov. : overcast, fo.: foliage, sn.: snow, se.: seasonal, ng.: day-night, sm. v.: small viewpoint, rn.: rain, st. v.: strong viewpoint.

	S2DHM			HF-Net			D2-Net			conditions								
name	pos. err.	rot. error	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	pos. err.	rot. error	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	pos. err.	rot. error	% of localized 0.5/1.0/5.0/10.0 (m) 2/5/10/20 (°)	il.	ov.	fo.	sn.	se.	ng.	sm.v.	rn.	st.v.
Skåne	1.66	1.69	0/23.81/71.43/76.19	0.61	0.54	47.62/71.43/80.95/80.95	0.15	0.37	95.24/95.24/95.24/95.24	✓						✓		
Massachusetts2	0.51	0.46	45.45/100/100/100	0.92	0.25	18.18/54.55/90.91/100	0.62	0.49	36.36/63.64/81.82/90.91	✓								
Thuringia	0.32	0.63	76.00/96.00/100/100	0.27	0.60	74.00/88.00/98.00/100	0.30	0.52	72.00/96.00/100/100	✓	✓						✓	
Burgundy2	1.60	0.96	26.47/44.12/55.88/55.88	34.02	18.57	26.47/32.35/32.35/32.35	0.64	0.56	44.12/67.65/100/100						✓			
Boston4	3438.04	128.29	0/0/0/0	18.26	7.95	0/0/6.12/10.20	1.35	1.22	10.20/36.73/100/100						✓			
Boston5	297.34	173.65	0/0/0/0	31.19	117.01	0/0/0/0	3.34	1.78	37.50/50/50/62.50						✓			
Boston2	72.12	95.37	0/0/0/0	97.81	145.06	0/0/0/0	0.65	1.81	5.56/77.78/77.78/77.78	✓				✓		✓		
Besançon1	29.82	2.57	0/26.47/38.24/38.24	59.75	125.11	0/0/0/0	28.90	1.85	0/0/35.29/35.29						✓			
Burgundy3	2.44	3.14	0/25.00/91.67/94.44	4.41	3.05	0/2.78/58.33/72.22	2.82	2.91	0/13.89/83.33/100			✓		✓				
Boston3	386.77	17.57	0/0/29.03/29.03	10.36	21.43	0/0/6.45/35.48	3.20	3.33	0/6.45/54.84/100	✓						✓		
Boston1	149.96	132.81	0/0/18.75/18.75	13.60	25.97	0/0/22.92/33.33	2.63	4.26	0/0/58.33/72.92						✓			
Nouvelle-Aquitaine2	2.16	8.56	0/0/90.62/100	3.11	8.87	0/0/75.00/90.62	2.47	8.63	0/0/100/100		✓	✓						
Burgundy1	1.17	11.29	0/0/0/100	1.55	11.26	0/0/0/76.92	1.15	11.21	0/0/0/100						✓			
Massachusetts4	1393.84	77.81	0/0/4.55/4.55	16.81	97.11	0/0/0/0	4.94	28.48	0/0/0/0			✓			✓			
Massachusetts3	57.31	142.47	0/0/0/0	57.31	142.47	0/0/0/0	65.22	40.99	0/0/0/0	✓			✓	✓		✓		
Brittany	134.22	149.22	0/0/0/0	46.80	169.11	0/0/0/0	12.99	146.07	0/0/0/0			✓						✓
Eden Prairie	20.90	170.85	0/0/0/0	55.55	68.27	0/0/0/0	23.57	153.63	0/0/6.67/8.89									✓
Orleans2	95.19	129.57	0/0/0/0	40.08	126.87	0/0/10/10	26.50	163.24	8.00/16.00/16.00/16.00									✓
Ile-de-France	748.20	151.03	0/0/0/0	92.56	147.16	0/0/0/0	23.11	168.35	0/0/0/0									✓
Angers2	59.57	131.43	0/0/0/0	19.66	158.46	0/0/0/0	8.15	169.56	0/0/0/0		✓							✓
Le-Mans	46.24	161.56	0/0/0/0	43.87	152.15	0/0/0/4.08	32.72	171.41	0/0/0/0		✓							✓
Leuven	48.73	134.82	0/0/0/0	77.69	90.79	0/0/2.00/2.00	24.36	171.92	0/0/0/0			✓						✓
Besançon3	458.73	157.18	0/0/0/0	32.49	170.67	0/0/0/0	43.48	174.15	0/0/0/0									✓
Besançon2	5670.75	121.32	0/0/0/0	40.12	148.52	0/0/0/0	45.07	174.47	0/0/0/0		✓	✓						✓
Brourges	88.22	153.89	0/0/0/0	43.69	137.36	0/0/0/0	26.06	174.94	0/0/0/0			✓						✓
Nouvelle-Aquitaine1	131.24	175.56	0/0/0/0	39.86	166.11	0/0/0/0	17.93	176.79	0/0/0/0			✓						✓
Angers1	315.34	140.58	0/0/0/0	20.34	102.01	0/8.51/27.66/36.17	20.91	178.15	0/0/0/0									✓
Pays de la Loire	510.92	153.42	0/0/0/0	47.80	156.94	0/0/0/0	35.87	178.19	0/0/0/0				✓	✓				✓
Besançon4	20.77	176.90	0/0/0/0	38.87	146.69	0/0/0/0	14.85	178.94	0/0/0/0									✓
Orleans1	42.48	161.58	0/0/0/0	22.56	160.46	0/0/0/0	26.43	178.95	0/0/0/0			✓						✓

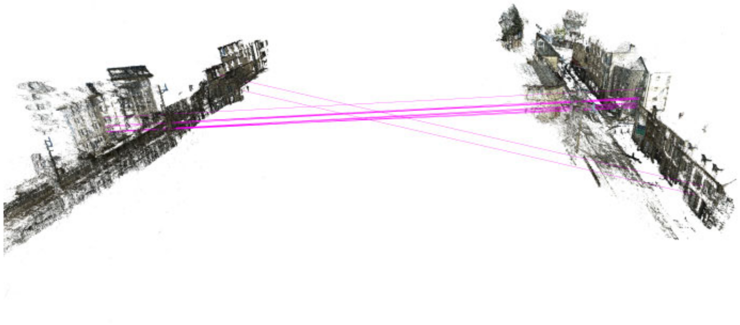


Figure 5.3: Database and query scene and their 3D matches, failure case of our method - reason : repetitive structure.

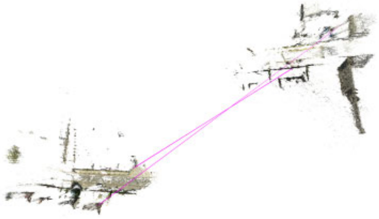


Figure 5.4: Database and query scene and their 3D matches, failure case of our method - reason : lack of enough overlap.

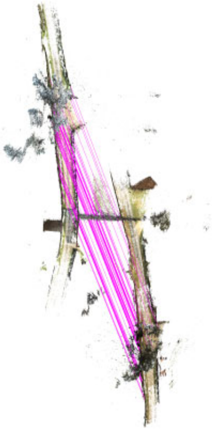


Figure 5.5: Database and query scene and their 3D matches, Success case of our method.

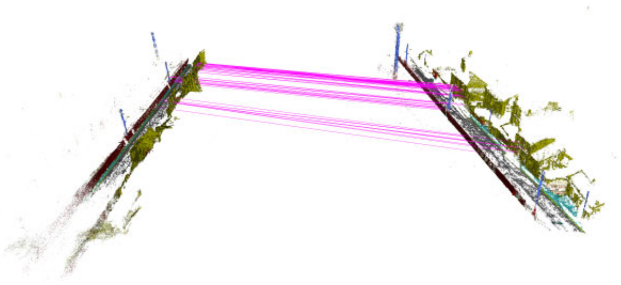


Figure 5.6: Database and query scene and their 3D matches, matching using semantic labels and descriptors.

6
Conclusion

6.1 Conclusion

In this thesis, we have introduced a new benchmark dataset for outdoor visual localization focusing on the failure cases of the SfM pipeline. Applying the current state-of-the-art and analyzing the results, we can conclude that (1) slight illumination changes is already a solved problem; and even early methods such as SIFT are able to handle those conditions, (2) localizing night-time queries is still a challenging problem for current algorithms; however, some methods (such as D2-Net) were able to get better results by revising the detection stage. (3) by far, the most challenging problem for current algorithms is strong viewpoint changes, and most methods fail in correctly estimating the orientation.

We have also moved from 2D to 3D, with the idea that the 3D structure is less affected by the changes. For some of the scenes affected by the strong viewpoint changes, we have seen improvements; however, the dataset remains challenging. In the future, statistical approaches that model the scene using statistical models and are able to handle outliers such as [117] could be investigated. Also, approaches that replace RANSAC with inlier confidence networks such as [118] could be investigated. Also, for the problem of lack of overlaps, scene completion using generative neural networks might be a good idea.

6.2 3D registration in other domains

3D representations are frequently used in other fields, such as the medical field. For example, one of the most common modules in the diagnosis, CT scan, is a series of 2D axial slices accumulated to represent the anatomy in a 3D structure [119]. CT scans are frequently used in colon health inspections, angiographies, pulmonary conditions diagnosis, and it is also used in 3D tumor simulations, surgical planning, and classification fusion [120].

When applying for 3D-3D registration, by estimating a rigid transform between the query and database, the assumption is that no scaling or distortion between the query and database scenes has been made. While this assumption holds in most street-level scenes, it would not be accurate for a general case, for instance, the medical applications of 3D-registration as the internal anatomy of the subject might be distorted between these two scenes. There are many cases where we can have this assumption in medical imagery, too, while in others, anatomical deformations between two scenes should be taken into account.

The most common scenario for non-rigid 3D registration in medical images is having the 3D representation of an object (e.g., an organ) in two different times (t and $t^{'}$) and registering one to another. In these scenarios, usually, the transformation estimation is broken in two terms (eq. 6.1) [121], global and local, where the global part $T_{\text{global}}(x, y, z)$ models the affine transformation of the object (rotation, orientation, and scale) and an additional $T_{\text{local}}(x, y, z)$ term describes the free-form deformations of the object. To model free-form deformations (FFD) of the anatomy, there are powerful modeling tools for deformable objects, such as [122]. These methods follow the idea of deforming an object by manipulating the underlying mesh of control points and are based on B-splines.

$$T(x, y, z) = T_{\text{global}}(x, y, z) + T_{\text{local}}(x, y, z) \tag{6.1}$$

To use our method for medical 3D object registration, the same considerations should be taken. In this regard, after point cloud description, instead of the rigid transformation between the scenes, first an affine transformation between two scenes (at t and $t^{'}$) should be estimated. Further deformations between two scenes should be estimated using by modeling deformations of the underlying control points.
On the other hand, as medical imagery is taken under more considerations, the scale of the scene is better recovered as the exact calibration information of the scanning devices is available [120].

Bibliography

[1] *The Visual Localization Benchmark.*

[2] F. Camposeco, T. Sattler, A. Cohen, A. Geiger, and M. Pollefeys, "Toroidal constraints for two-point localization under high outlier ratios," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4545–4553, 2017.

[3] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *European conference on computer vision*, pp. 15–29, Springer, 2012.

[4] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[5] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1455–1461, 2016.

[6] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2704–2712, 2015.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[9] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*, pp. 467–483, Springer, 2016.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[11] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6DOF Urban Visual Localization in Changing Conditions," in *CVPR*, 2017.

[12] F. Radenovic, J. L. Schonberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas, "From dusk till dawn: Modeling in the dark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5488–5496, 2016.

[13] M. Varma and D. Ray, “Learning the discriminative power-invariance tradeoff,” in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[14] J. L. Schönberger and J.-M. Frahm, “Structure-From-Motion Revisited,” in *CVPR*, June 2016.

[15] N. Snavely, S. Seitz, and R. Szeliski, “Modeling the World from Internet Photo Collections,” *IJCV*, vol. 80, no. 2, pp. 189–210, 2008.

[16] J. Heinly, J. L. Schönberger, E. Dunn, and J. M. Frahm, “Reconstructing the world* in six days,” in *CVPR*, 2015.

[17] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, “Image retrieval for image-based localization revisited.,”

[18] H. Badino, D. Huber, and T. Kanade, “Visual topometric localization,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 794–799, IEEE, 2011.

[19] H. Badino, D. Huber, and T. Kanade, “Real-time topometric localization,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 1635–1642, IEEE, 2012.

[20] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000km: The Oxford RobotCar Dataset,” *IJRR*, vol. 36, no. 1, pp. 3–15, 2017.

[21] Z. J. Yew and G. H. Lee, “3dfeat-net: Weakly supervised local 3d features for point cloud registration,” in *ECCV*, 2018.

[22] H. Germain, G. Bourmaud, and V. Lepetit, “Sparse-to-dense hypercolumn matching for long-term visual localization,” in *2019 International Conference on 3D Vision (3DV)*, pp. 513–523, IEEE, 2019.

[23] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” *CoRR*, vol. abs/1812.03506, 2018.

[24] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint description and detection of local features,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 8084–8093, 2019.

[25] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *CVPR*, 2016.

[26] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford, “Deep Learning Features at Scale for Visual Place Recognition,” *ICRA*, 2017.

[27] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, “24/7 Place Recognition by View Synthesis,” in *CVPR*, 2015.

[28] A. Torii, J. Sivic, and T. Pajdla, “Visual localization by linear combination of image descriptors,” in *Proceedings of the 2nd IEEE Workshop on Mobile Vision, with ICCV*, 2011.

[29] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, “Large-Scale Location Recognition and the Geometric Burstiness Problem,” in *CVPR*, 2016.

[30] Y. Li, N. Snavely, and D. P. Huttenlocher, “Location recognition using prioritized feature matching,” in *European conference on computer vision*, pp. 791–804, Springer, 2010.

[31] T. Sattler, B. Leibe, and L. Kobbelt, "Fast Image-Based Localization using Direct 2D-to-3D Matching," in *ICCV*, 2011.

[32] T. Sattler, B. Leibe, and L. Kobbelt, "Improving Image-Based Localization by Active Correspondence Search," in *ECCV*, 2012.

[33] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization," *PAMI*, vol. 39, no. 9, pp. 1744–1756, 2017.

[34] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *ICCV*, 2015.

[35] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide Pose Estimation Using 3D Point Clouds," in *ECCV*, 2012.

[36] Z. Kukelova, J. Heller, and A. Fitzgibbon, "Efficient Intersection of Three Quadrics and Applications in Computer Vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *IJCV*, vol. 13, no. 3, pp. 331–356, 1994.

[38] M. Fischler and R. Bolles, "Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *CACM*, vol. 24, pp. 381–395, 1981.

[39] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[40] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4400–4408, 2015.

[41] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013.

[42] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692, 2017.

[43] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European conference on computer vision*, pp. 536–551, Springer, 2014.

[44] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images," in *CVPR*, 2013.

[46] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *CVPR*, 2016.

[47] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC - Differentiable RANSAC for Camera Localization," in *CVPR*, 2017.

[48] E. Brachmann and C. Rother, "Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC," 2020.

[49] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. Di Stefano, and P. H. S. Torr, "On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation," in *CVPR*, 2017.

[50] E. Brachmann and C. Rother, "Expert Sample Consensus Applied to Camera Re-Localization," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[51] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *ICRA*, 2012.

[52] R. Pless, "Using Many Cameras as One," in *CVPR*, 2003.

[53] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6896–6906, 2018.

[54] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3d scene reconstruction and class segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 97–104, 2013.

[55] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1746–1754, 2017.

[56] R. Arandjelović and A. Zisserman, "Visual Vocabulary with a Semantic Twist," in *ACCV*, 2014.

[57] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.

[58] A. Cohen, T. Sattler, and M. Pollefeys, "Merging the Unmatchable: Stitching Visually Disconnected SfM Models," in *ICCV*, 2015.

[59] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic match consistency for long-term visual localization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 383–399, 2018.

[60] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons,"

[61] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 883–890, 2013.

[62] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1808–1817, 2015.

[63] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[64] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, *et al.*, "City-scale landmark identification on mobile devices," in *CVPR 2011*, pp. 737–744, IEEE, 2011.

[65] H. Badino, D. Huber, and T. Kanade, "The CMU Visual Localization Data Set." `http://3dvis.ri.cmu.edu/data-sets/localization`, 2011.

[66] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, and C. Keskin, "Learning to Navigate the Energy Landscape," in *3DV*.

[67] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor visual localization with dense matching and view synthesis," in *CVPR*, 2018.

[68] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location Recognition using Prioritized Feature Matching," in *ECCV*, 2010.

[69] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From Structure-from-Motion Point Clouds to Fast Location Recognition," in *CVPR*, 2009.

[70] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-Scale Landmark Identification on Mobile Devices," in *CVPR*, 2011.

[71] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *ICCV*, 2015.

[72] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image Retrieval for Image-Based Localization Revisited," in *BMVC*, 2012.

[73] H. Badino, D. Huber, and T. Kanade, "Visual topometric localization," in *Intelligent Vehicles Symposium (IV)*, 2011.

[74] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, "Lyft level 5 av dataset." urlhttps://level5.lyft.com/dataset/, 2019.

[75] "Waymo open dataset: An autonomous driving dataset." https://waymo.com/open/, 2019.

[76] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *CoRR*, vol. abs/1903.11027, 2019.

[77] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," *arXiv: 1803.06184*, 2018.

[78] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2016.

[79] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, pp. 5099–5108, 2017.

[80] "OpenSfM." `https://github.com/mapillary/OpenSfM`.

[81] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[82] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[83] K.-i. Kanatani and D. D. Morris, "Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 2017–2028, 2001.

[84] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[85] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *CVPR*, 2017.

[86] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[87] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-Based Localization Using LSTMs for Structured Feature Correlation," in *ICCV*, 2017.

[88] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," 2018.

[89] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *2011 International Conference on Computer Vision*, pp. 383–390, IEEE, 2011.

[90] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks.,"

[91] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[92] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 43–57, 2010.

[93] A. Bundy and L. Wallen, "Difference of gaussians," in *Catalogue of Artificial Intelligence Tools*, pp. 30–30, Springer, 1984.

[94] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[95] G. Elbaz, T. Avraham, and A. Fischer, "3d point cloud registration for localization using a deep neural network auto-encoder," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[96] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3212–3217, 2009.

[97] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8277–8286, 2019.

[98] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5000–5009, 2017.

[99] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, (New York, NY, USA), Association for Computing Machinery, 2009.

[100] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *BMVC*, 2011.

[101] M. Lopez-Antequera, P. Gargallo, M. Hofinger, S. Rota Bulò, Y. Kuang, and P. Kontschieder, "Mapillary planet-scale depth dataset," in *Preprint*, 2020.

[102] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, p. 234–241, 2015.

[103] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[104] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[105] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[106] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[107] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3d data description," in *Proceedings of the ACM workshop on 3D object retrieval*, pp. 57–62, 2010.

[108] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[109] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.

[110] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 195–205, 2018.

[111] M. Khoury, Q. Zhou, and V. Koltun, "Learning compact geometric features," *CoRR*, vol. abs/1709.05056, 2017.

[112] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 998–1005, Ieee, 2010.

[113] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[114] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban

scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[115] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[116] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International journal of computer vision*, vol. 103, no. 3, pp. 267–305, 2013.

[117] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1397–1410, 2017.

[118] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[119] R. Acharya, R. Wasserman, J. Stevens, and C. Hinojosa, "Biomedical imaging modalities: a tutorial," *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, pp. 3–25, 1995.

[120] F. E.-Z. A. El-Gamal, M. Elmogy, and A. Atwan, "Current trends in medical image registration and fusion," *Egyptian Informatics Journal*, vol. 17, no. 1, pp. 99–124, 2016.

[121] M. M. Letteboer, P. W. Willems, M. A. Viergever, and W. J. Niessen, "Non-rigid registration of 3d ultrasound images of brain tumours acquired during neurosurgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 408–415, Springer, 2003.

[122] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151–160, 1986.

A

Appendix

A.1 Dataset Visualization

This section provides visualizations for the datasets from our CrowdDriven benchmark. Figures A.1 to A.43 show example images from all of our datasets.

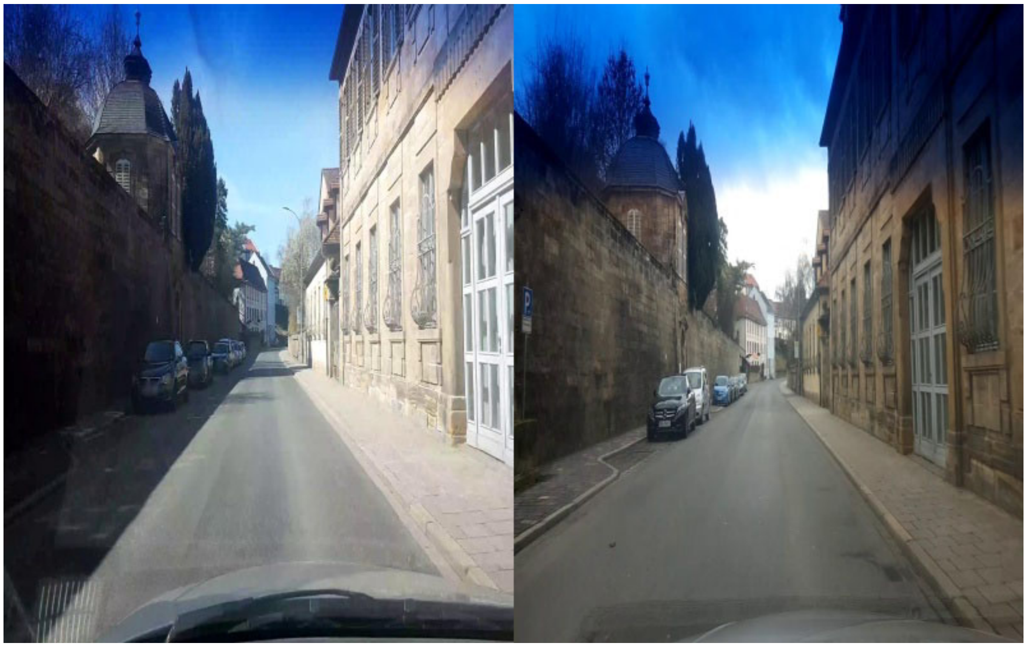


Figure A.1: Bayern, Category : Easy

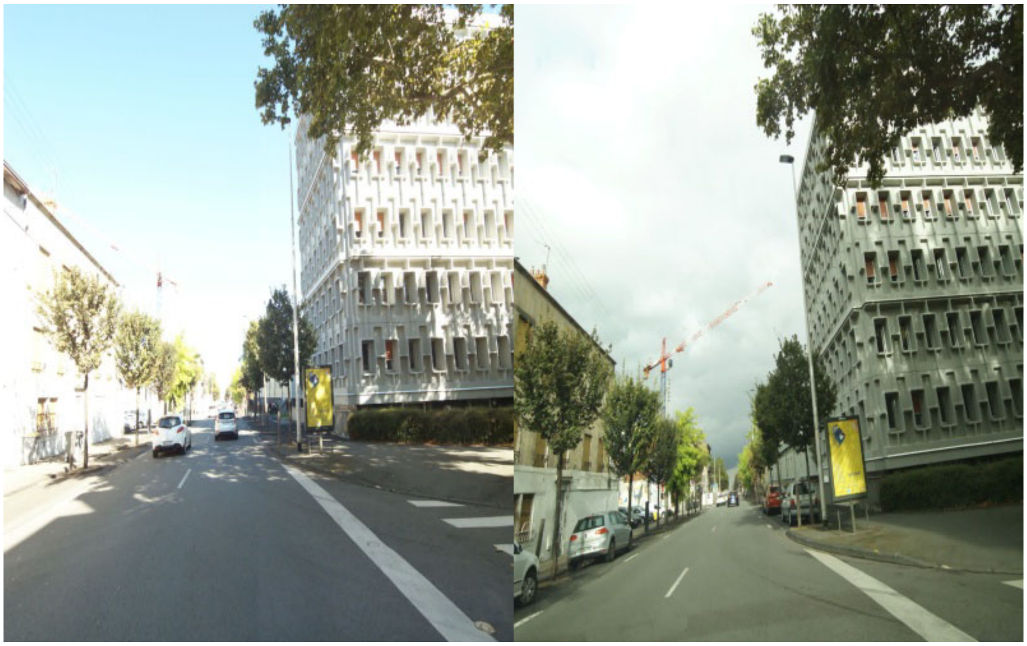


Figure A.2: Clermont-Ferrand, Category : Easy

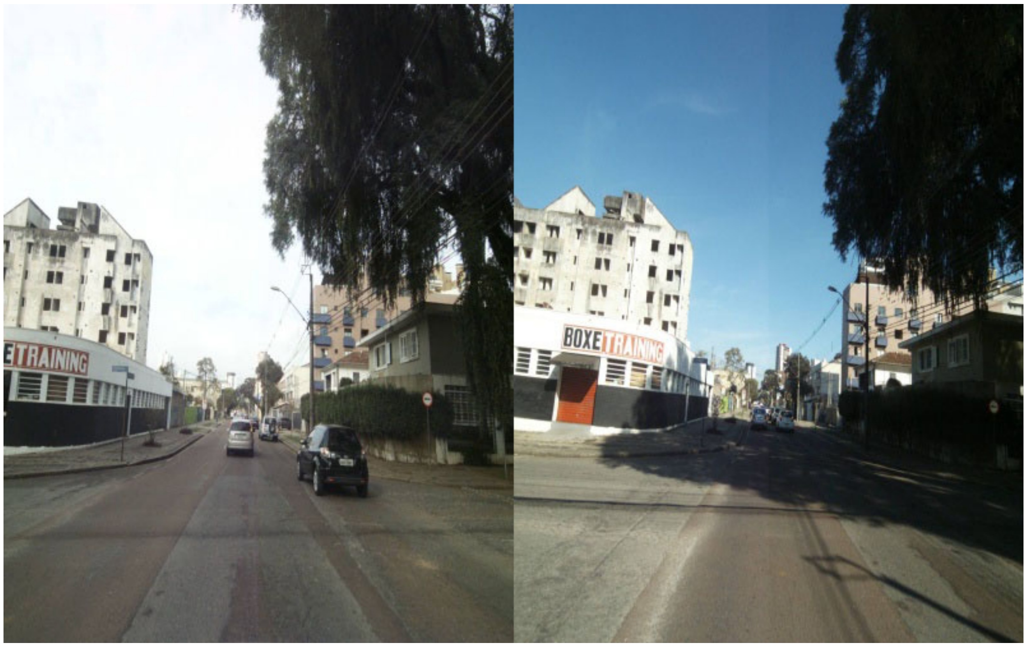


Figure A.3: Curitiba, Category : Easy

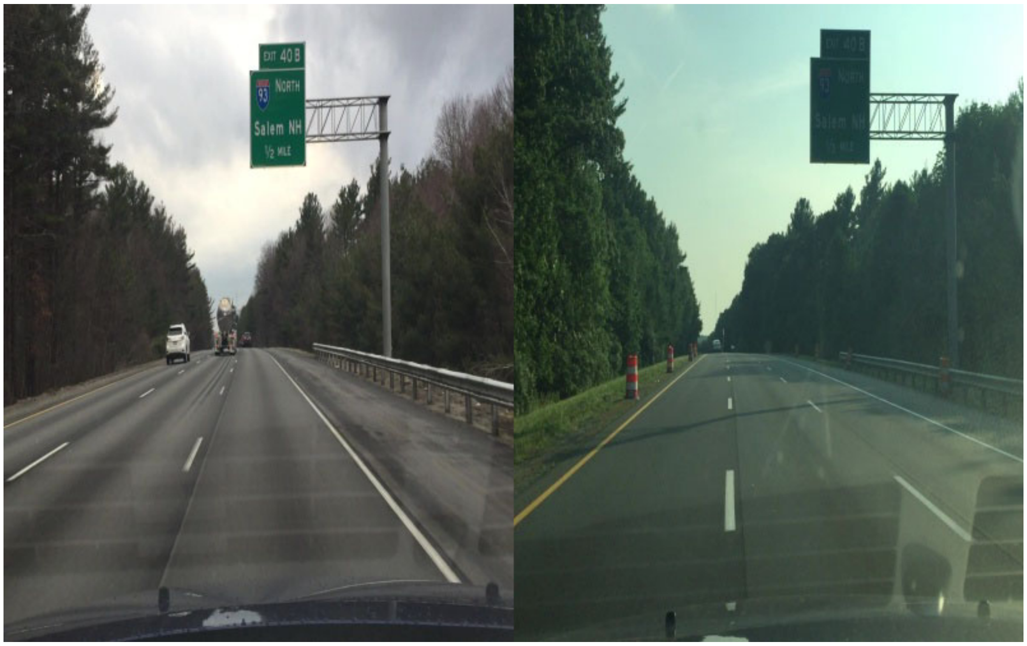


Figure A.4: Massachusetts1, Category : Easy

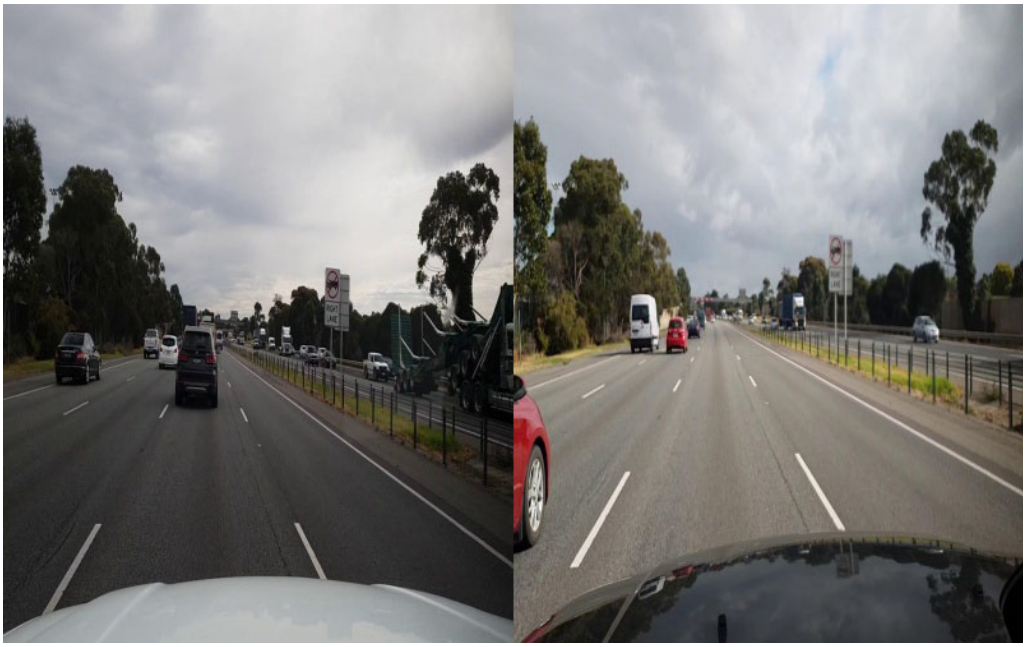


Figure A.5: Melbourne, Category : Easy

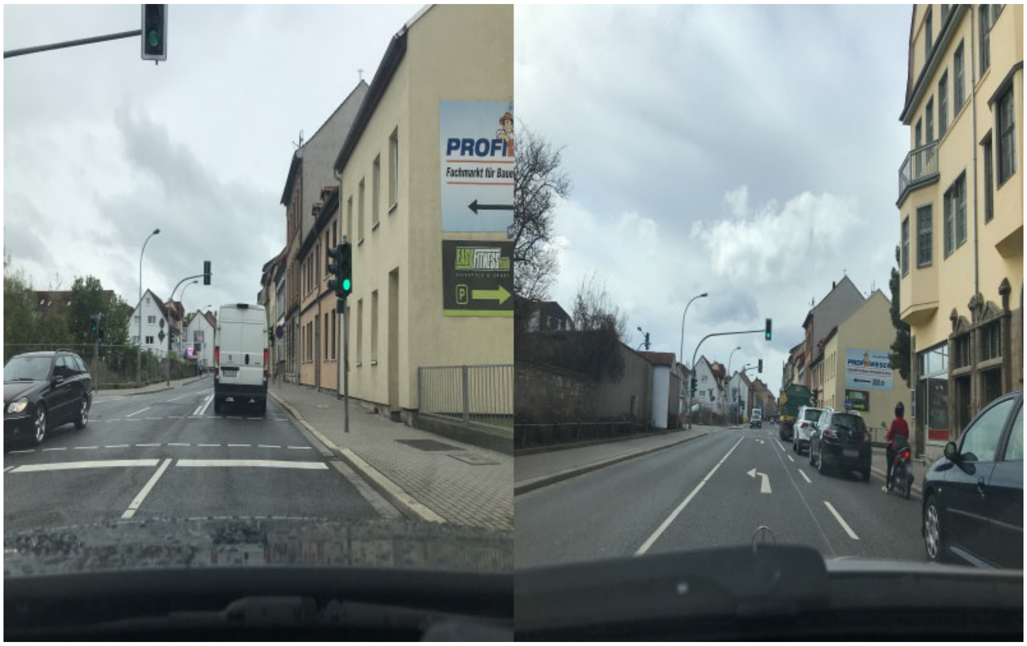


Figure A.6: Muehlhausen, Category : Easy

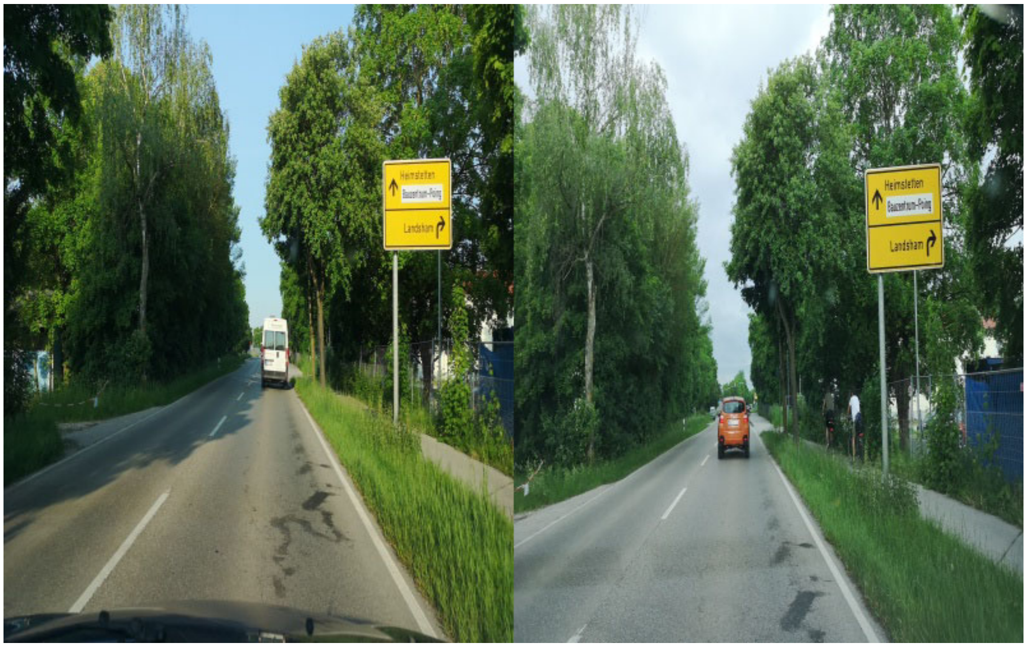


Figure A.7: Poing, Category : Easy

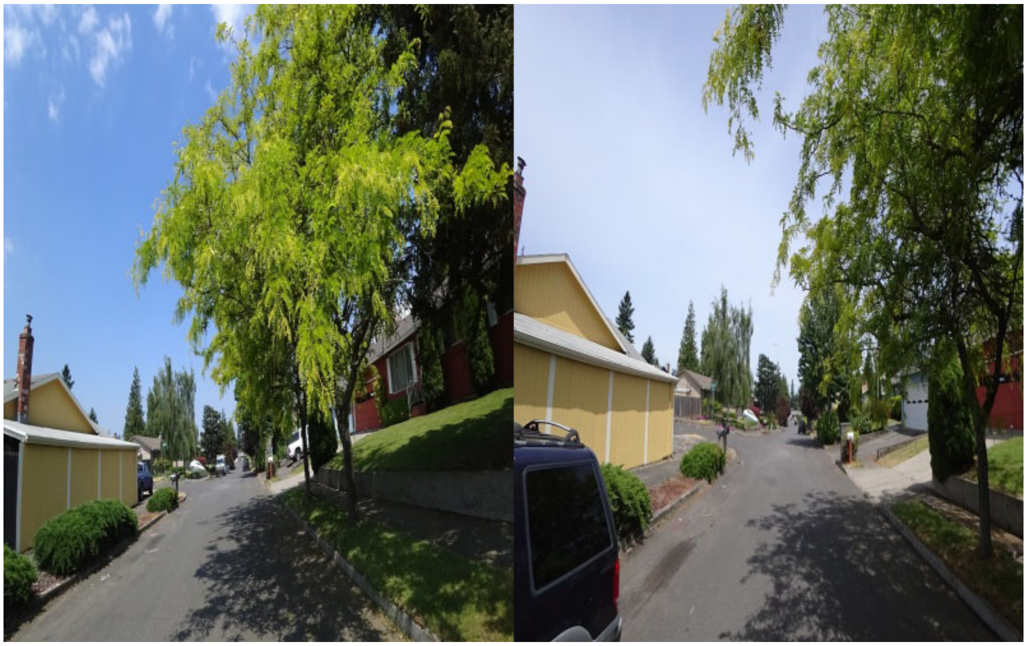


Figure A.8: Portland, Category : Easy

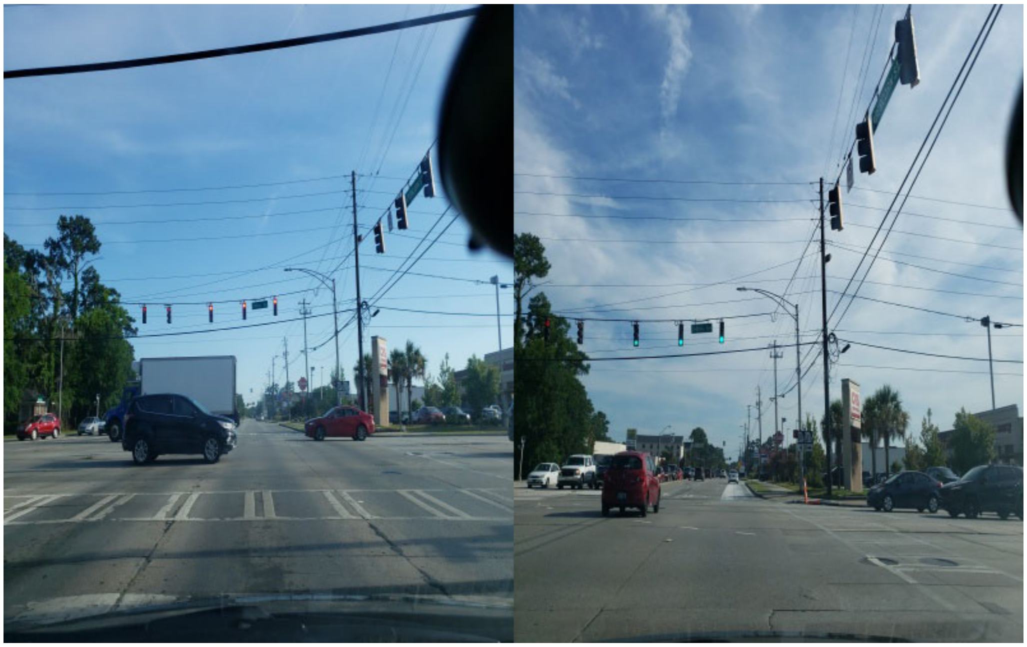


Figure A.9: Savannah, Category : Easy

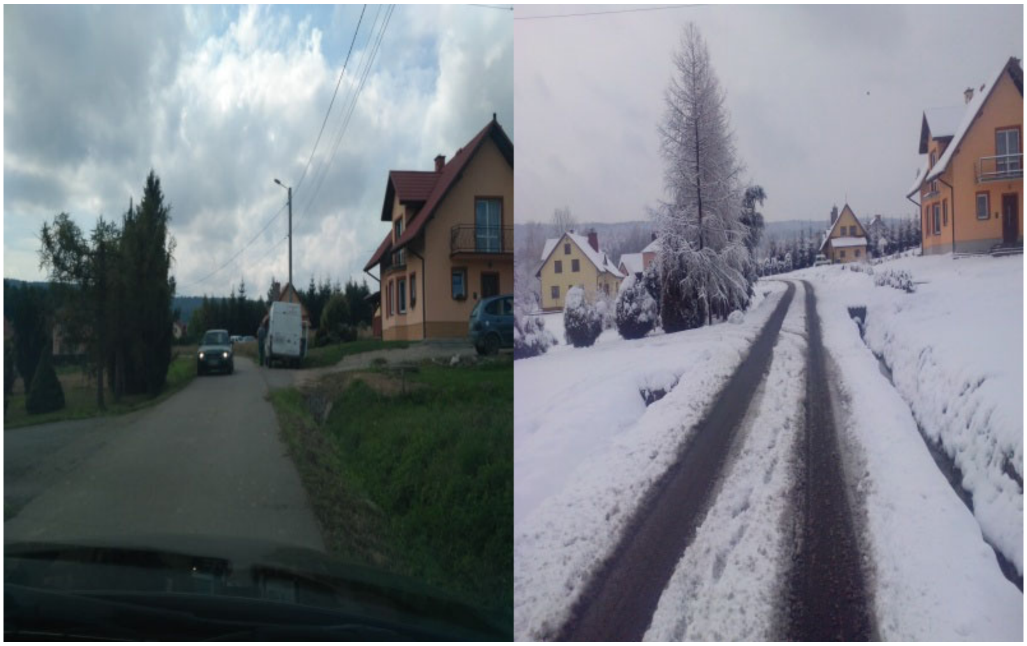


Figure A.10: Subcarpathia, Category : Easy

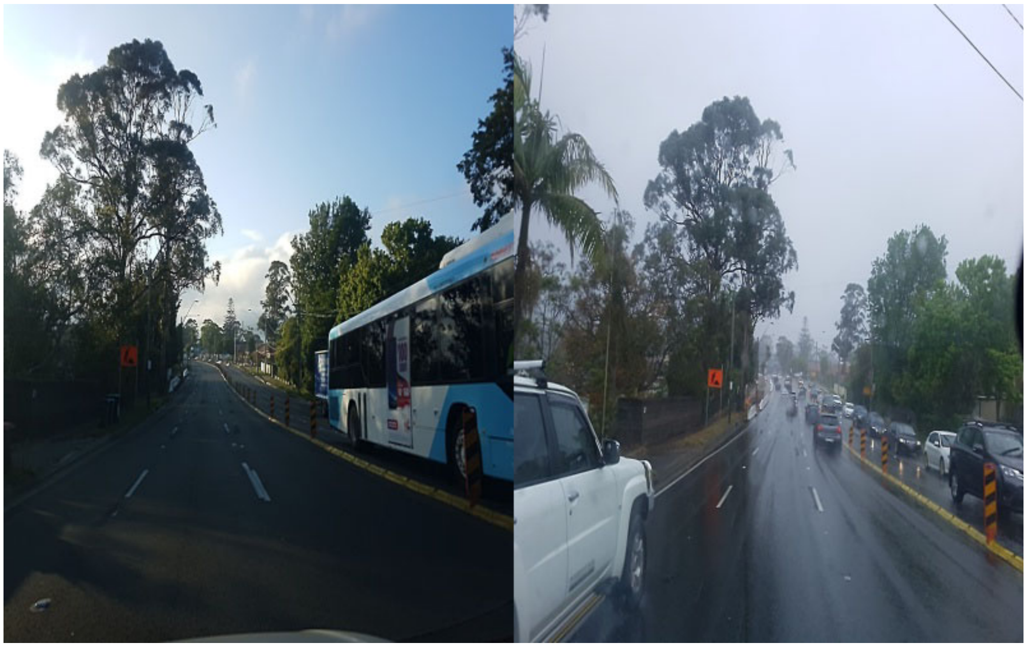


Figure A.11: Sydney, Category : Easy

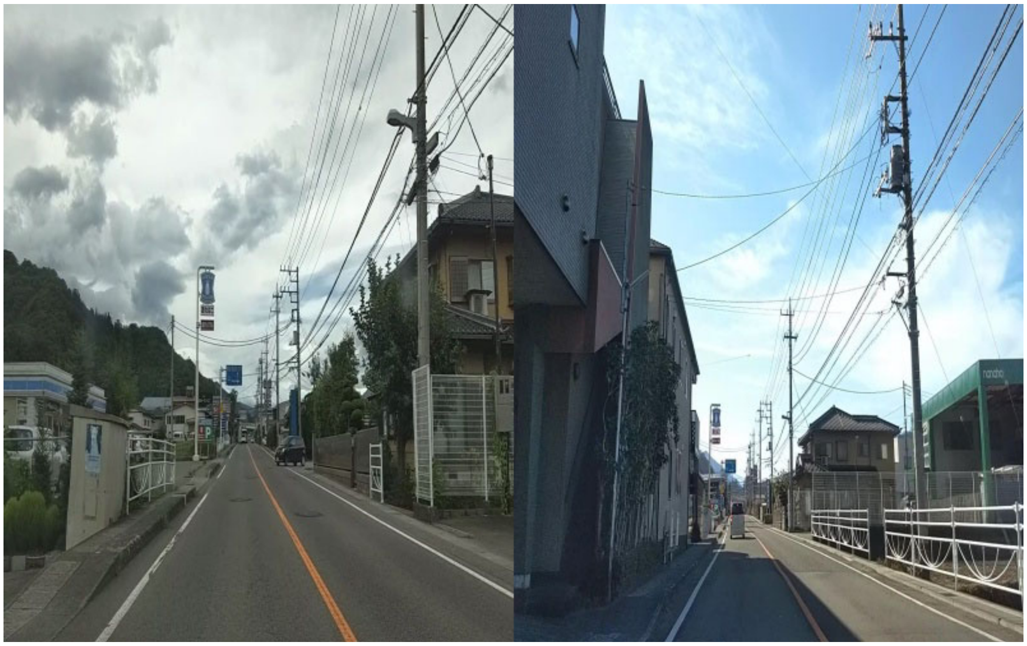


Figure A.12: Tsuru, Category : Easy

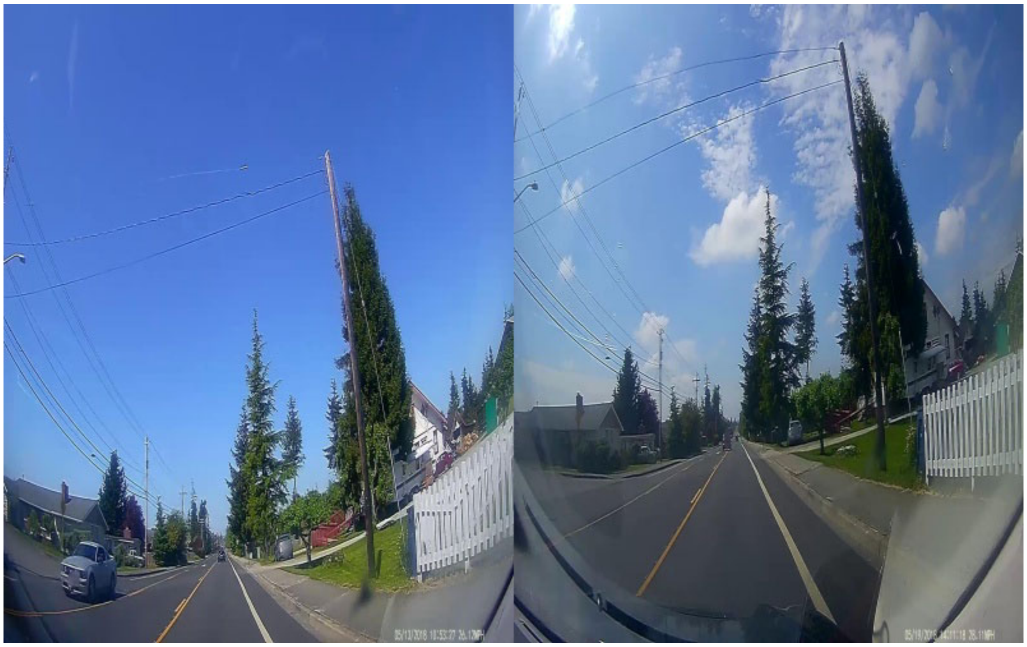


Figure A.13: Washington, Category : Easy

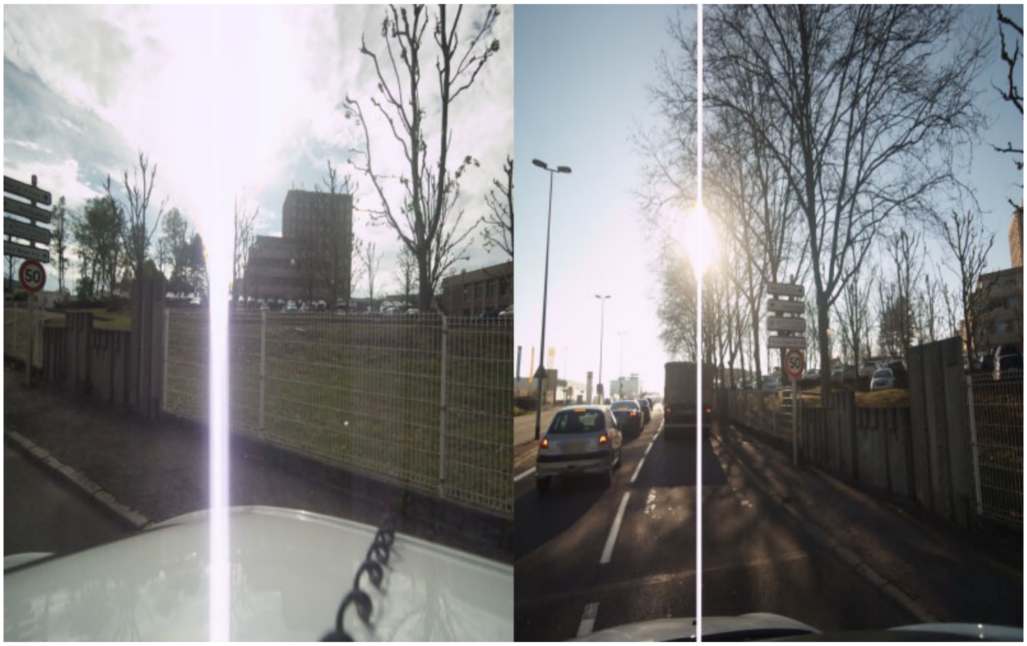


Figure A.14: Besançon1, Category : Medium

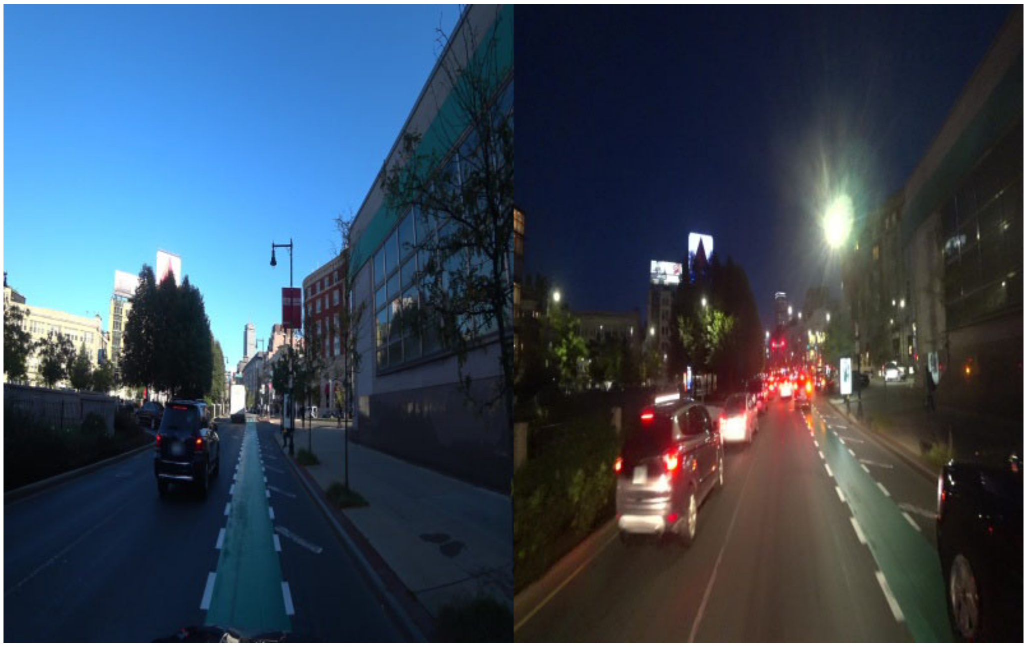


Figure A.15: Boston1, Category : Medium

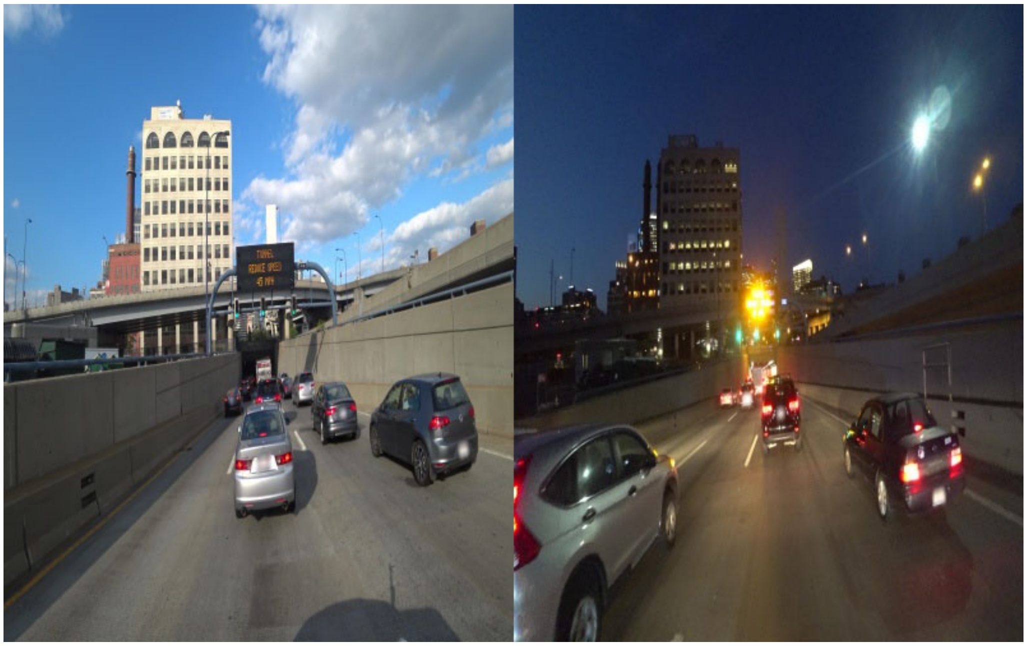


Figure A.16: Boston2, Category : Medium

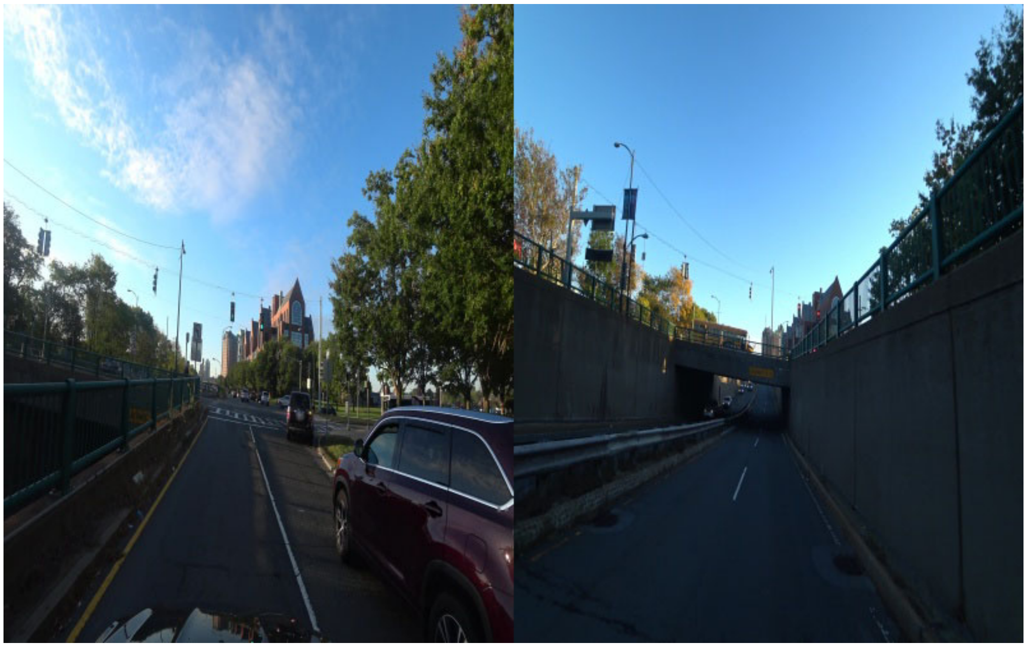


Figure A.17: Boston3, Category : Medium

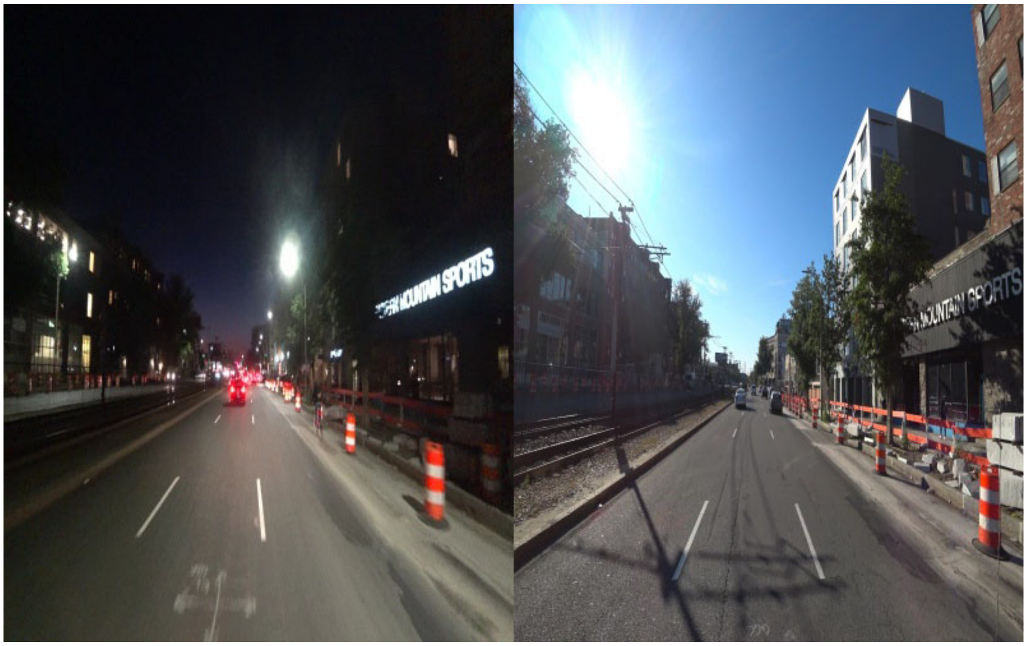


Figure A.18: Boston4, Category : Medium

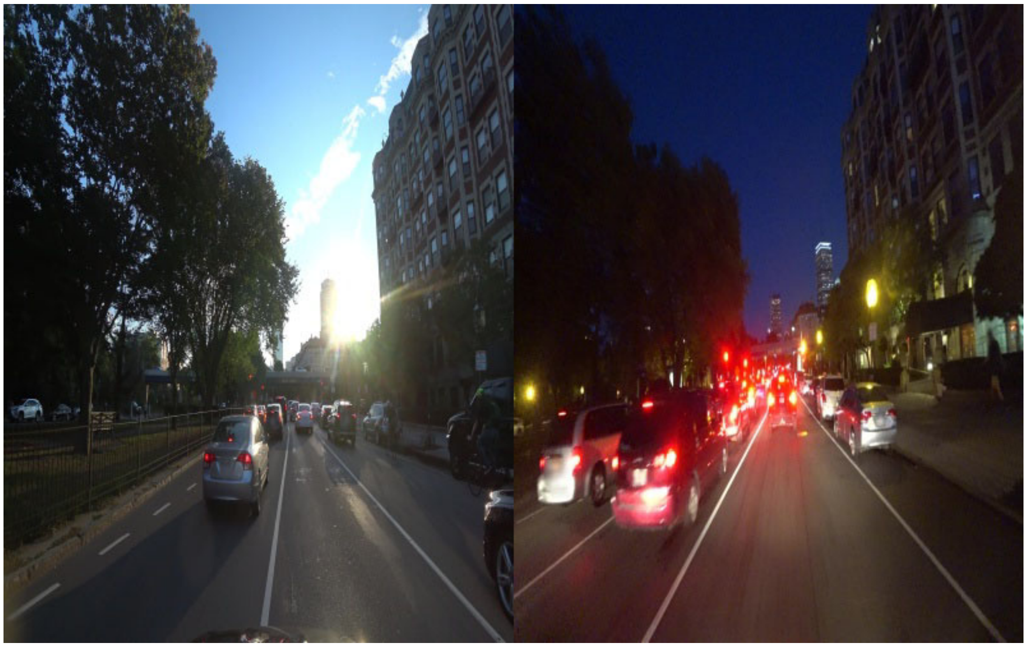


Figure A.19: Boston5, Category : Medium

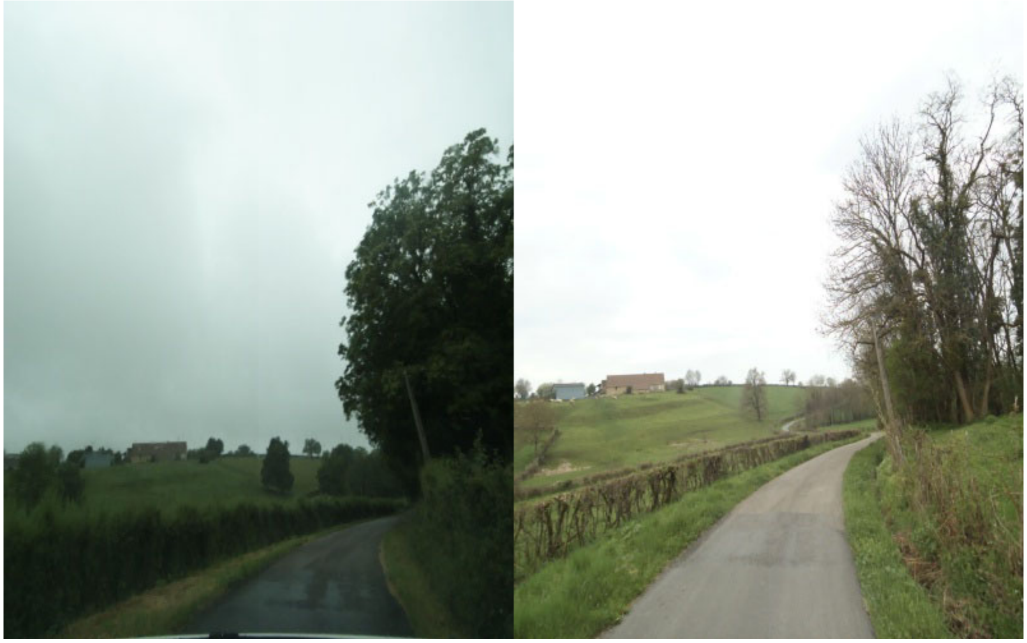


Figure A.20: Burgundy1, Category : Medium

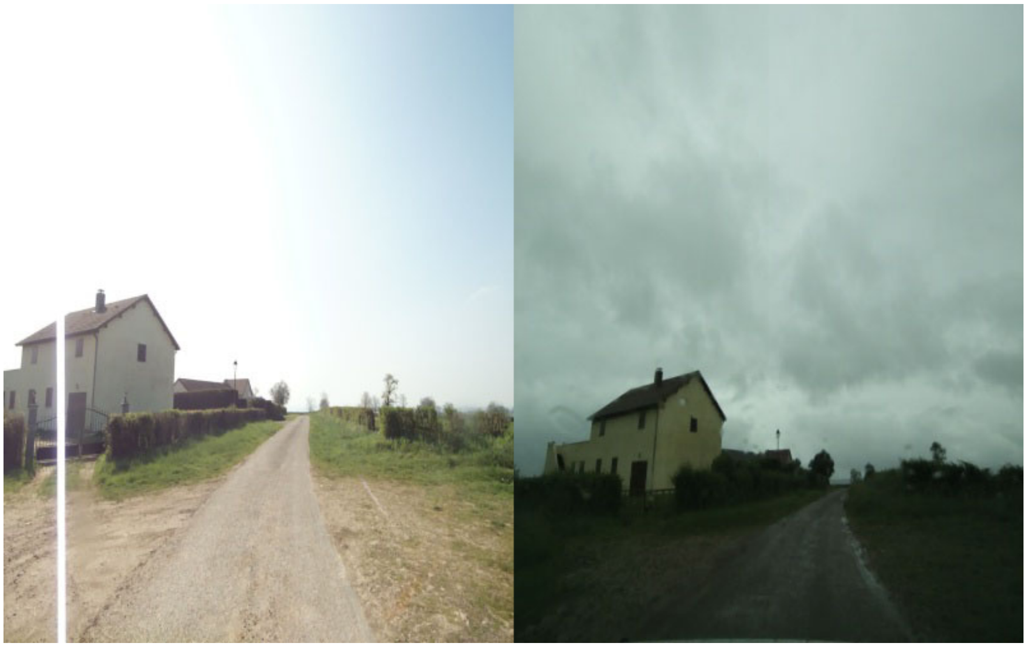


Figure A.21: Burgundy2, Category : Medium

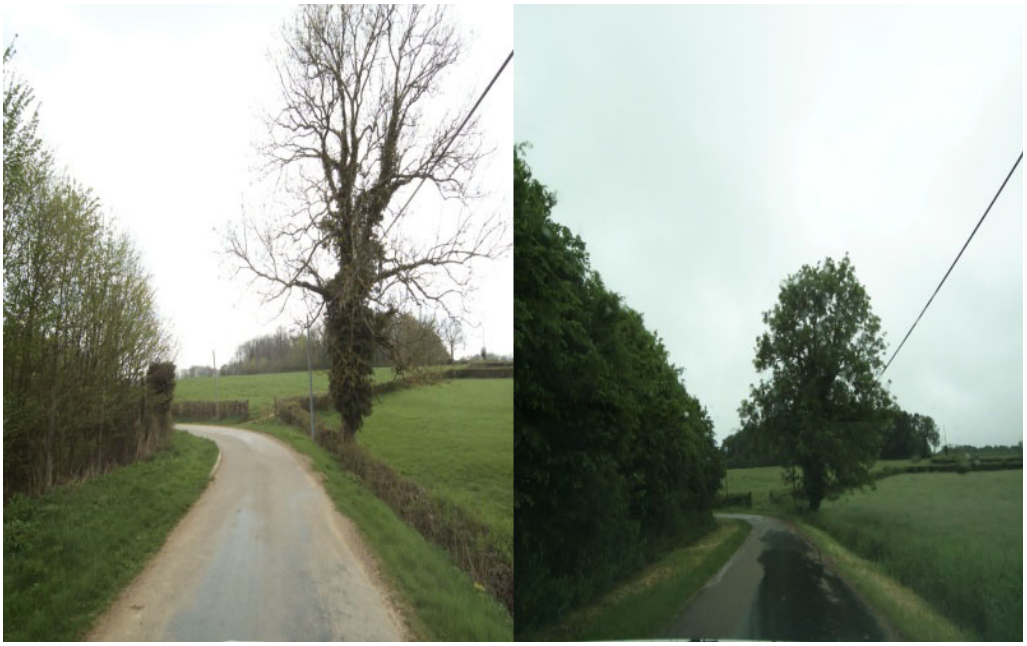


Figure A.22: Burgundy3, Category : Medium

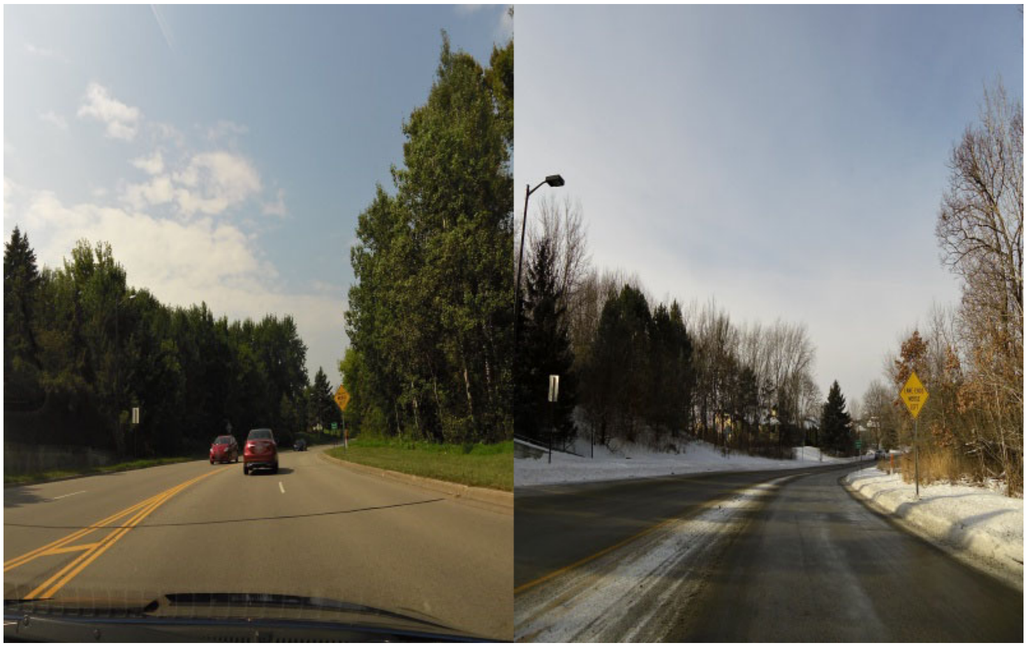


Figure A.23: Eden Prairie, Category : Medium

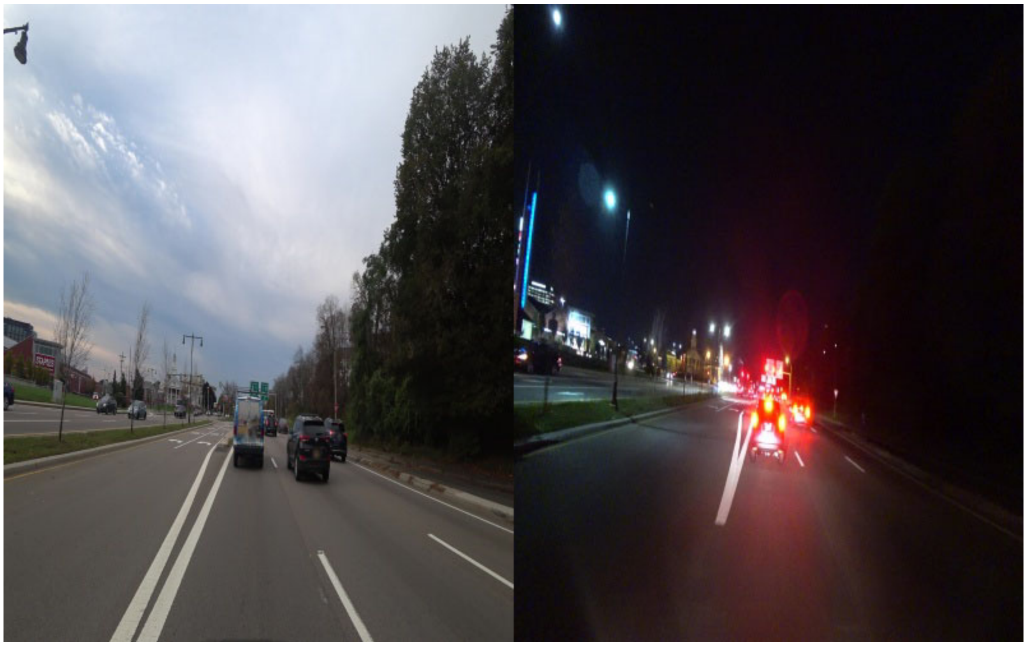


Figure A.24: Massachusetts2, Category : Medium

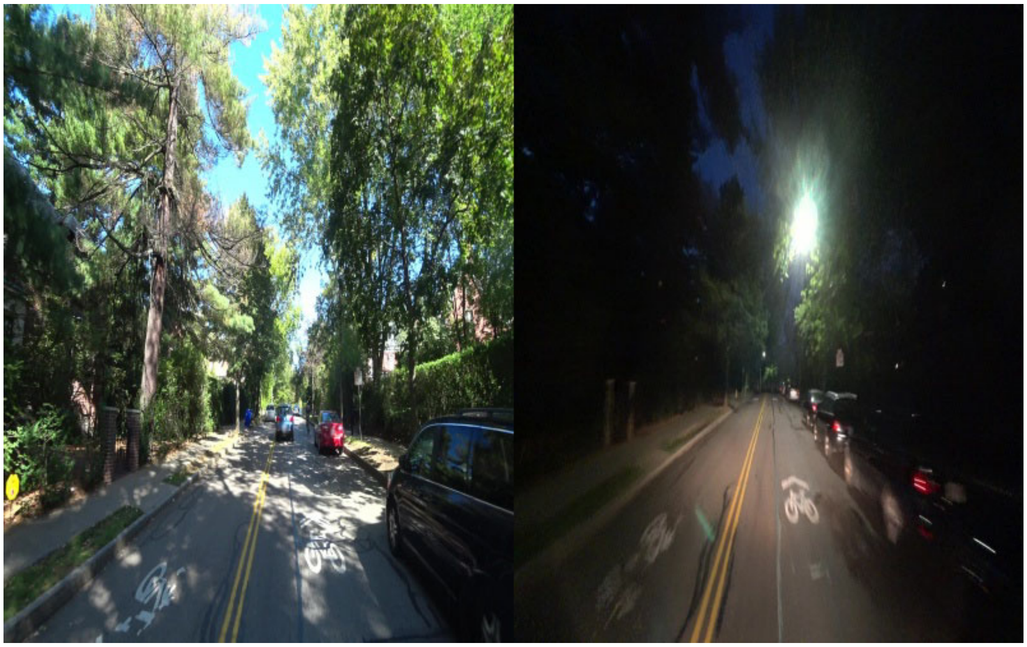


Figure A.25: Massachusetts3, Category : Medium

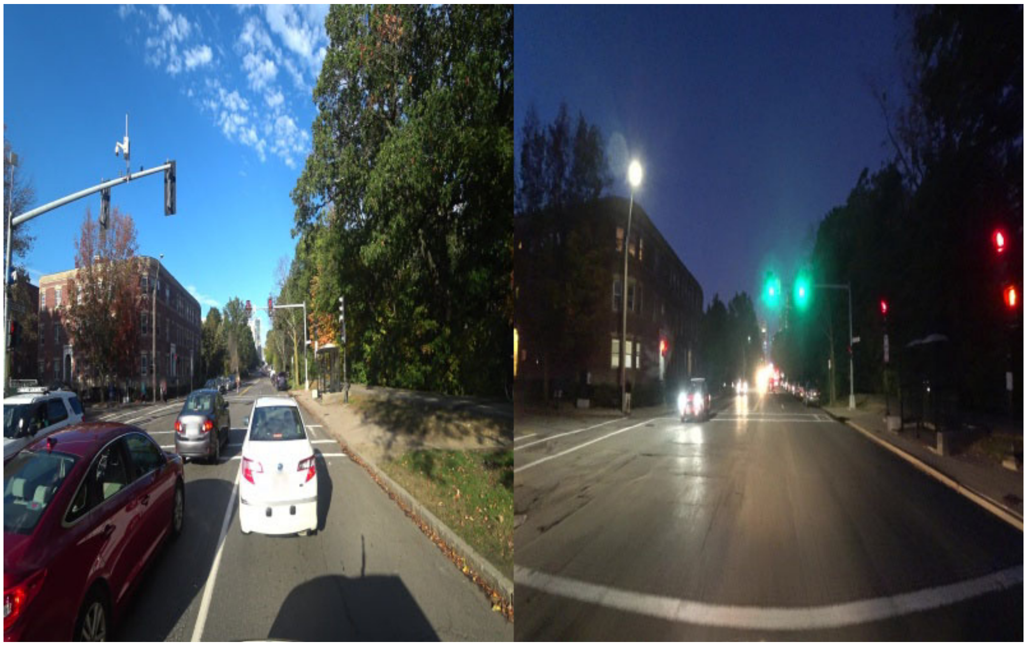


Figure A.26: Massachusetts4, Category : Medium

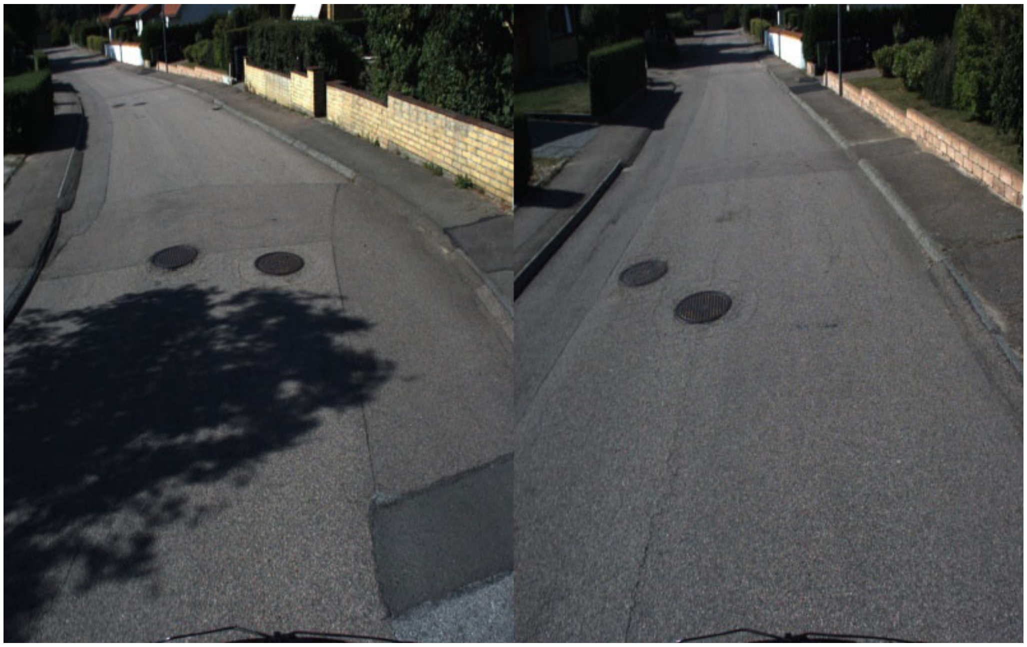


Figure A.27: Skåne, Category : Medium

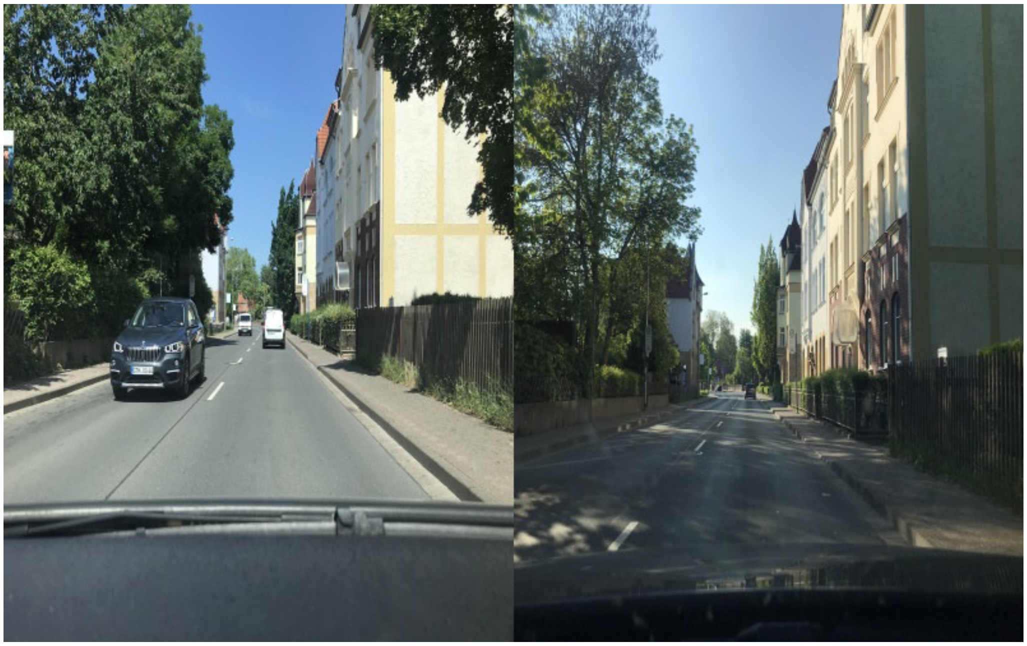


Figure A.28: Thuringia, Category : Medium

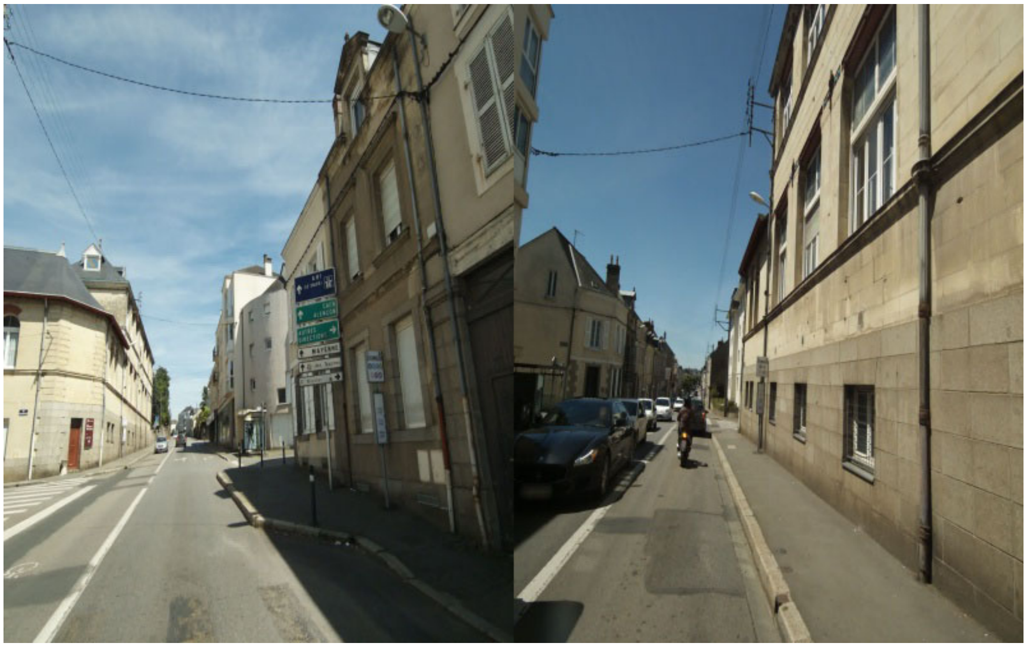


Figure A.29: Angers1, Category : Difficult

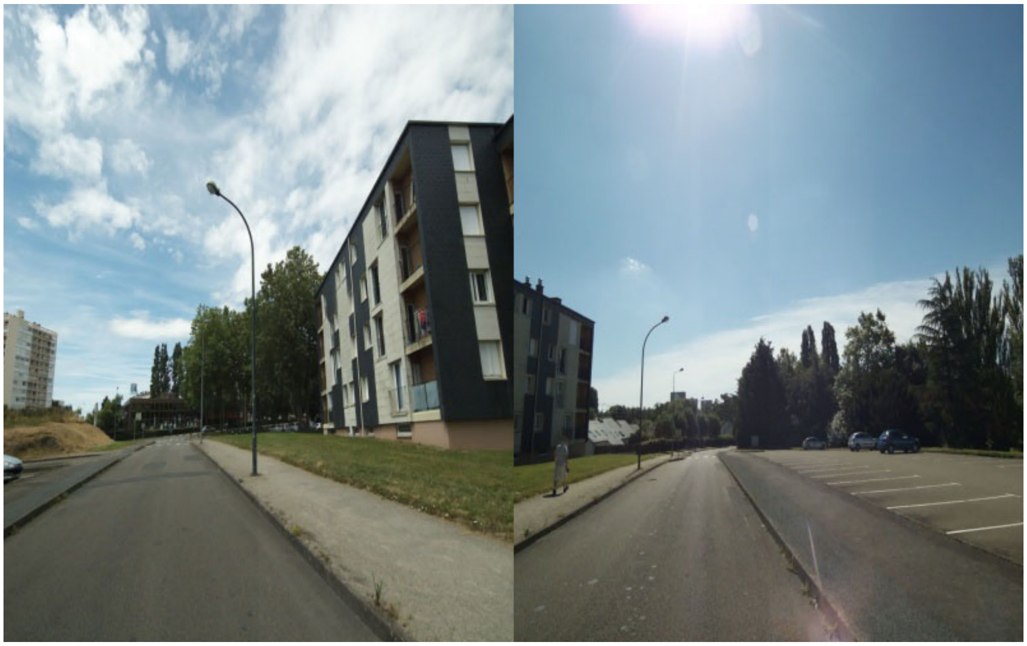


Figure A.30: Angers2, Category : Difficult

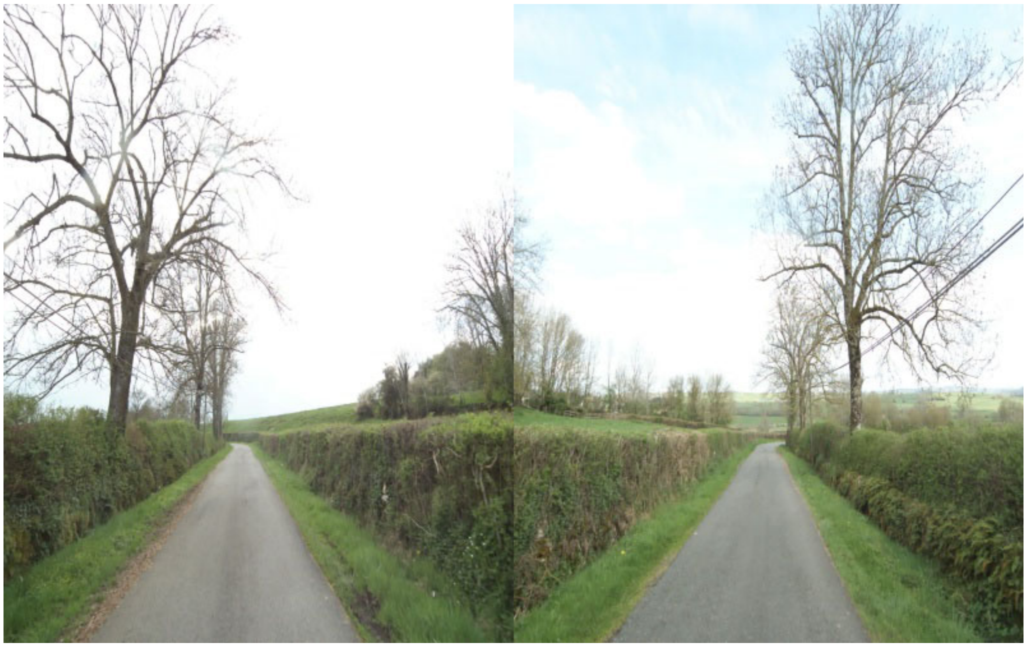


Figure A.31: Besançon2, Category : Difficult

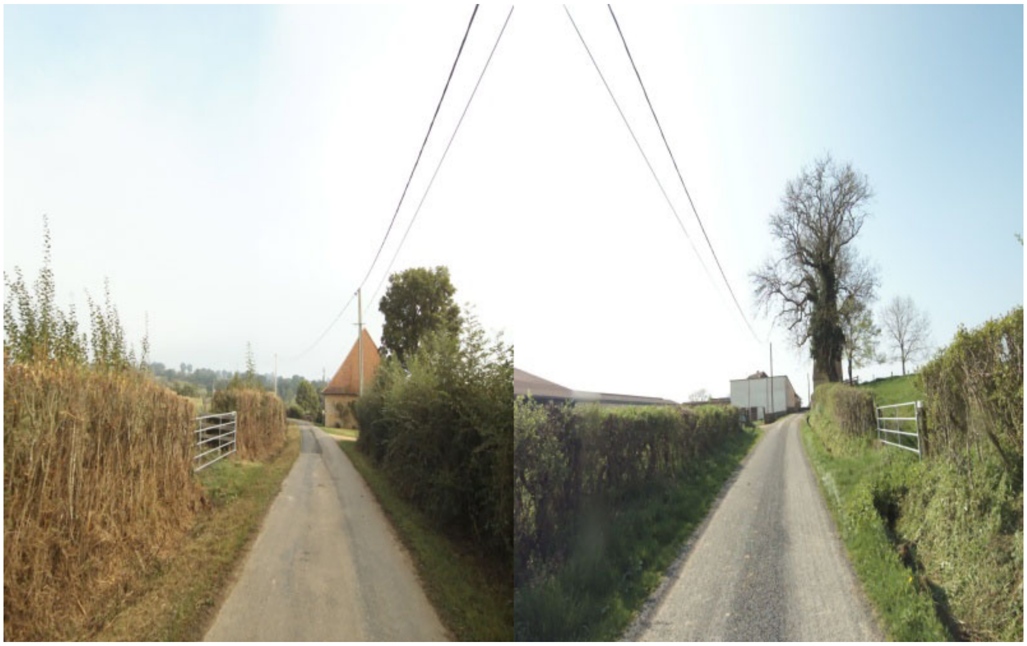


Figure A.32: Besançon3, Category : Difficult

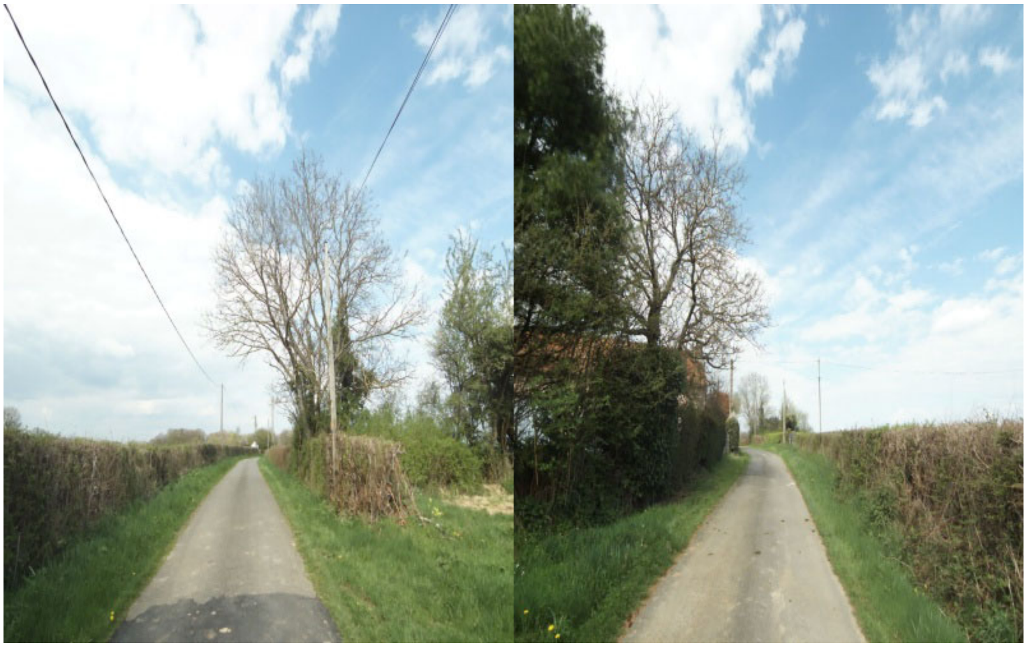


Figure A.33: Besançon4, Category : Difficult

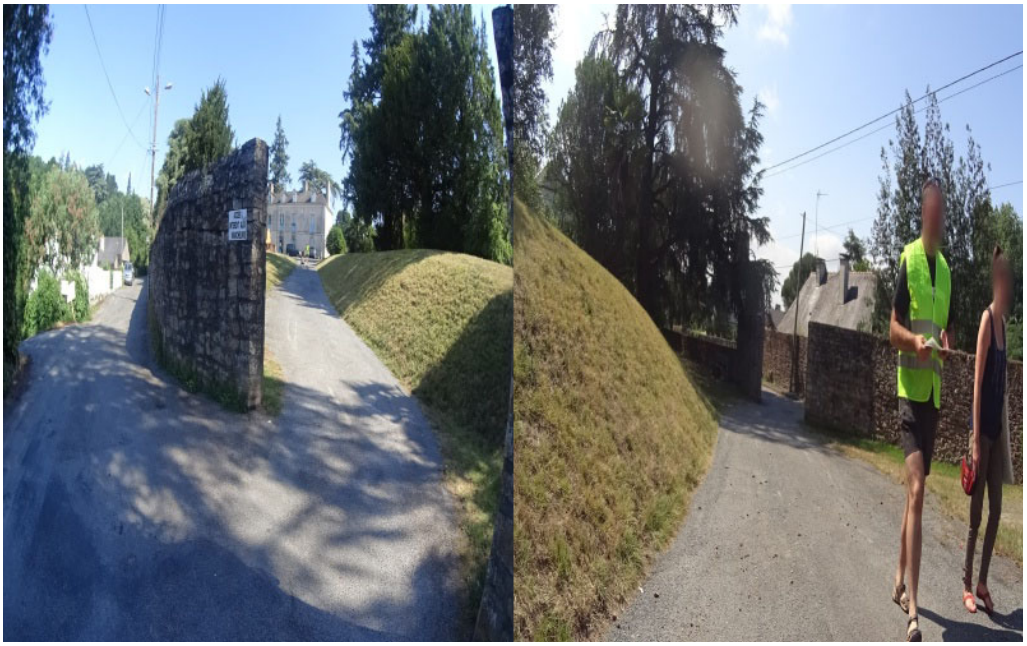


Figure A.34: Brittany, Category : Difficult

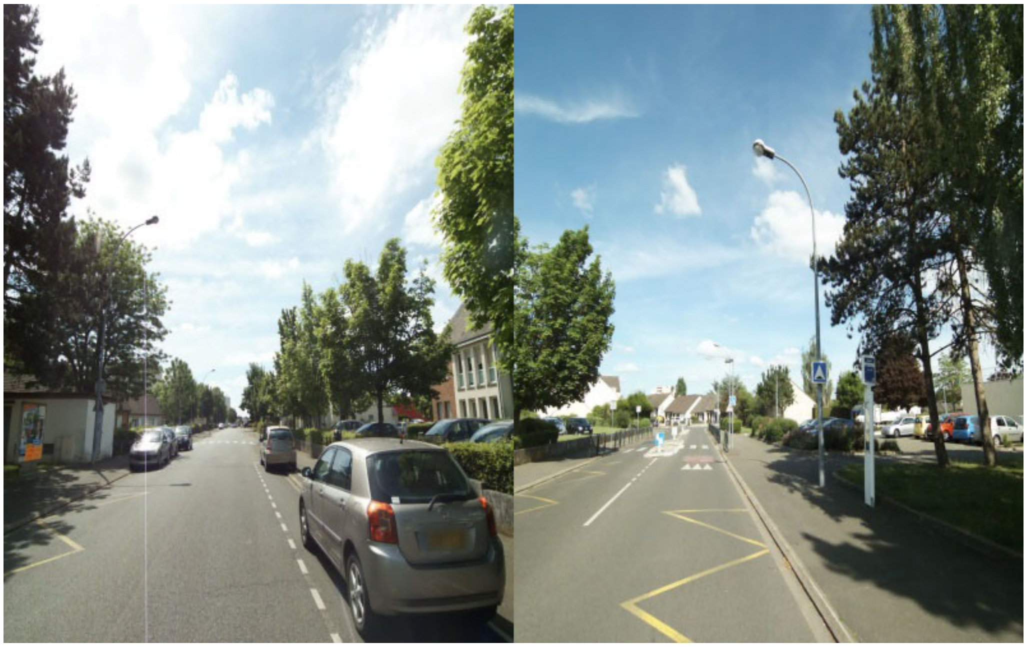


Figure A.35: Brourges, Category : Difficult

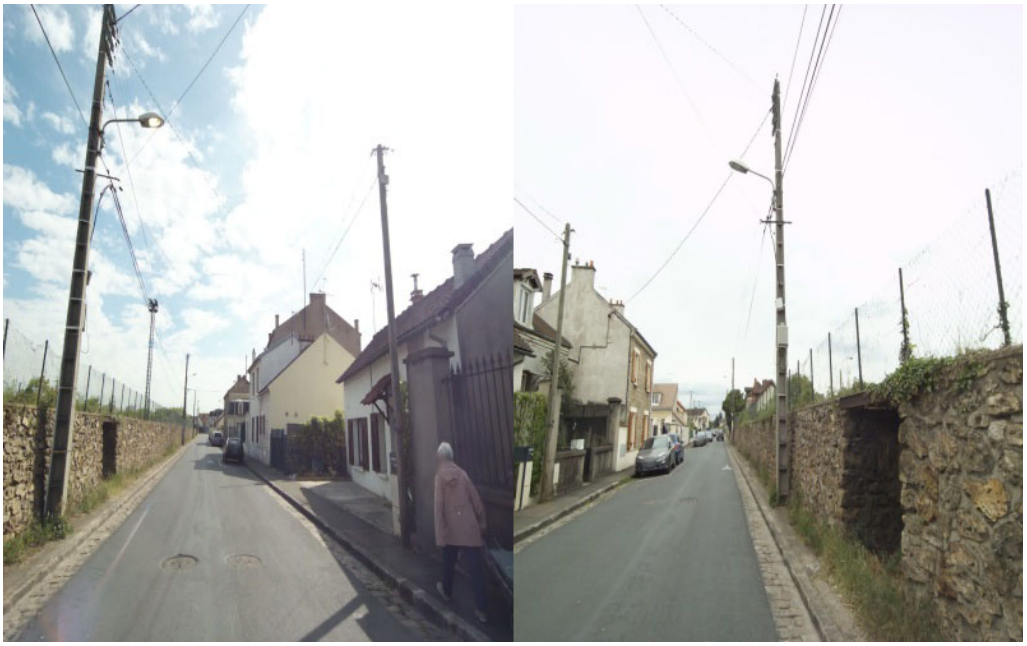


Figure A.36: Ile-de-France, Category : Difficult

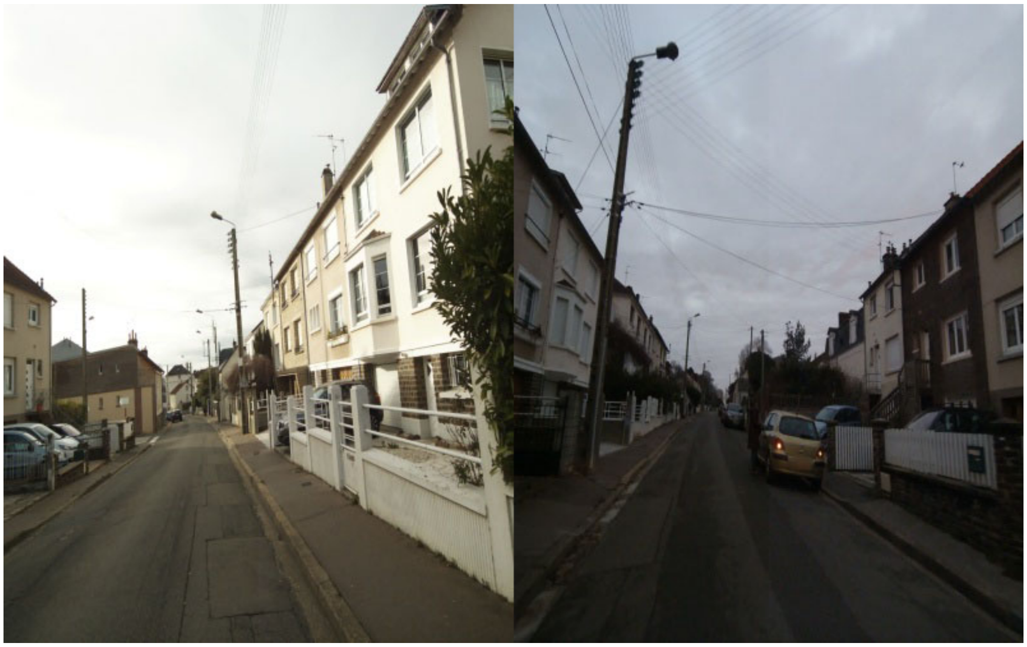


Figure A.37: Le-Mans, Category : Difficult

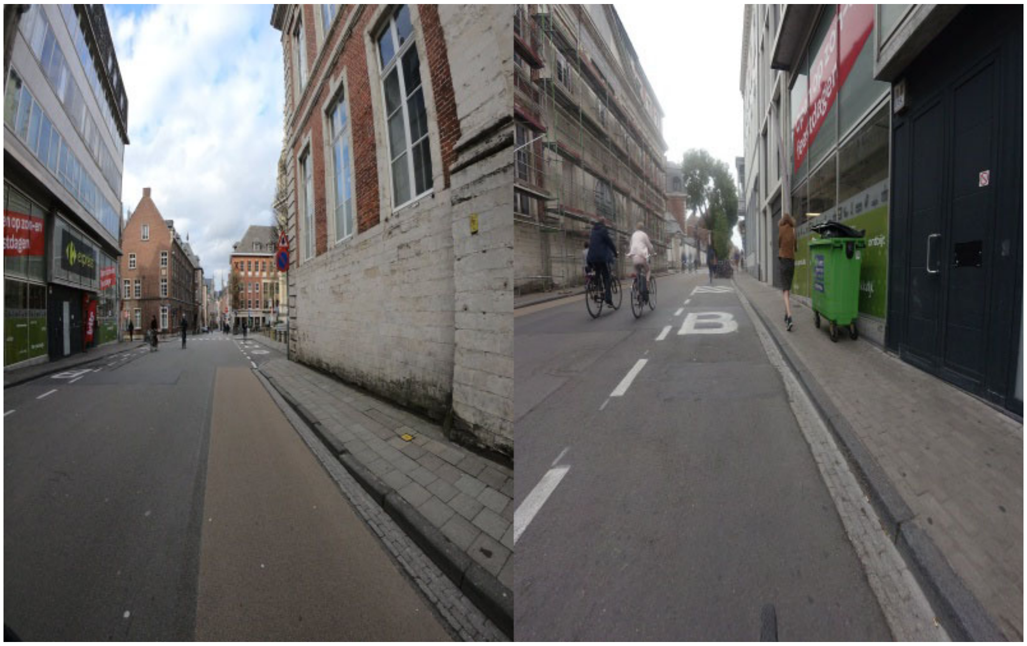


Figure A.38: Leuven, Category : Difficult

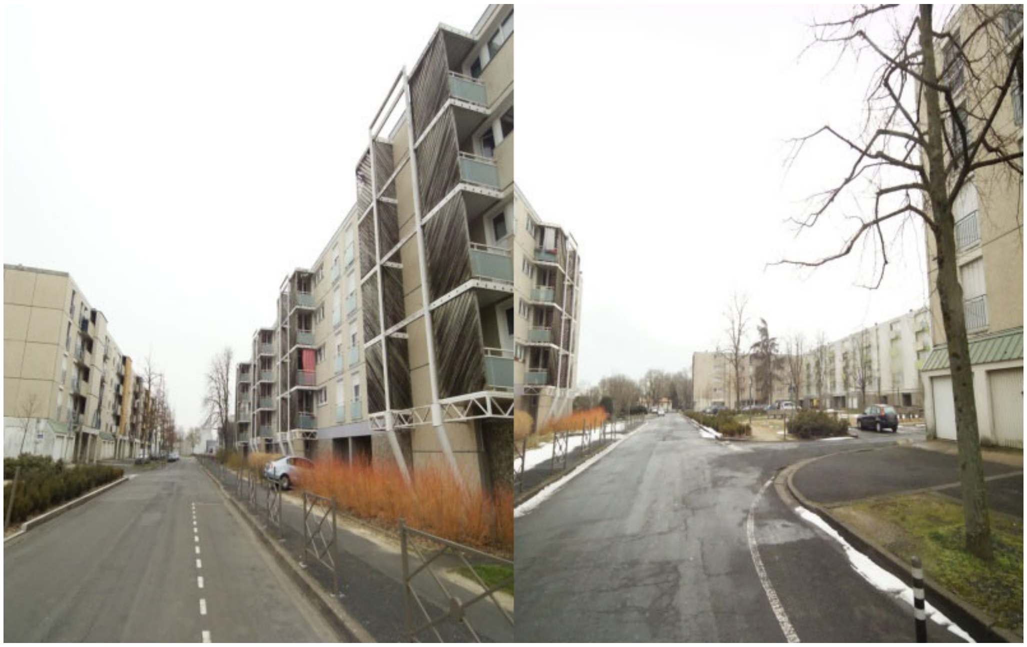


Figure A.39: Nouvelle-Aquitaine1, Category : Difficult

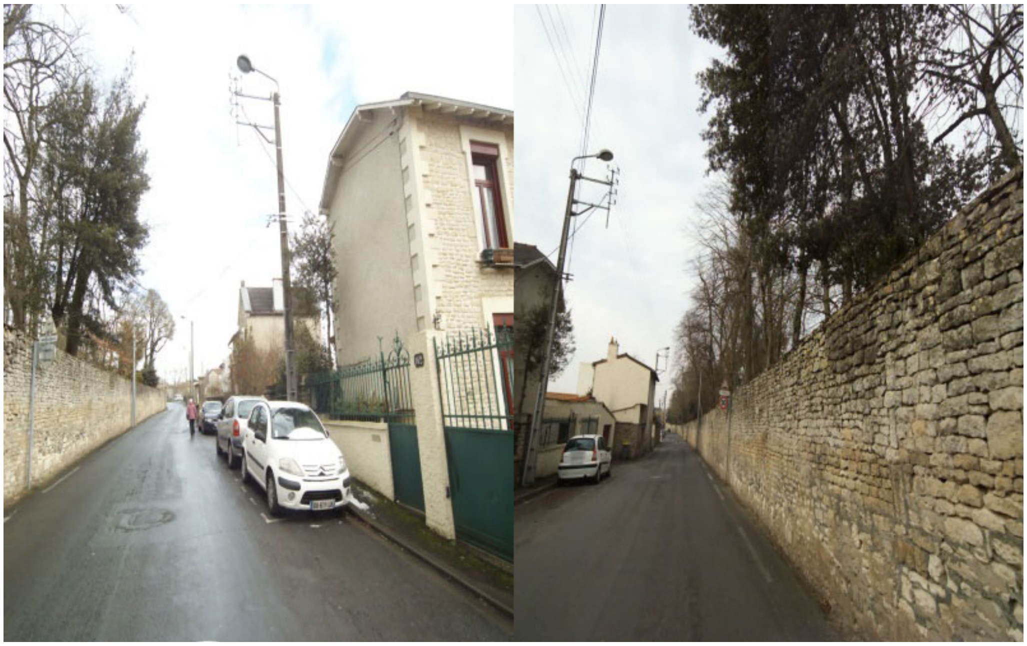


Figure A.40: Nouvelle-Aquitaine2, Category : Difficult

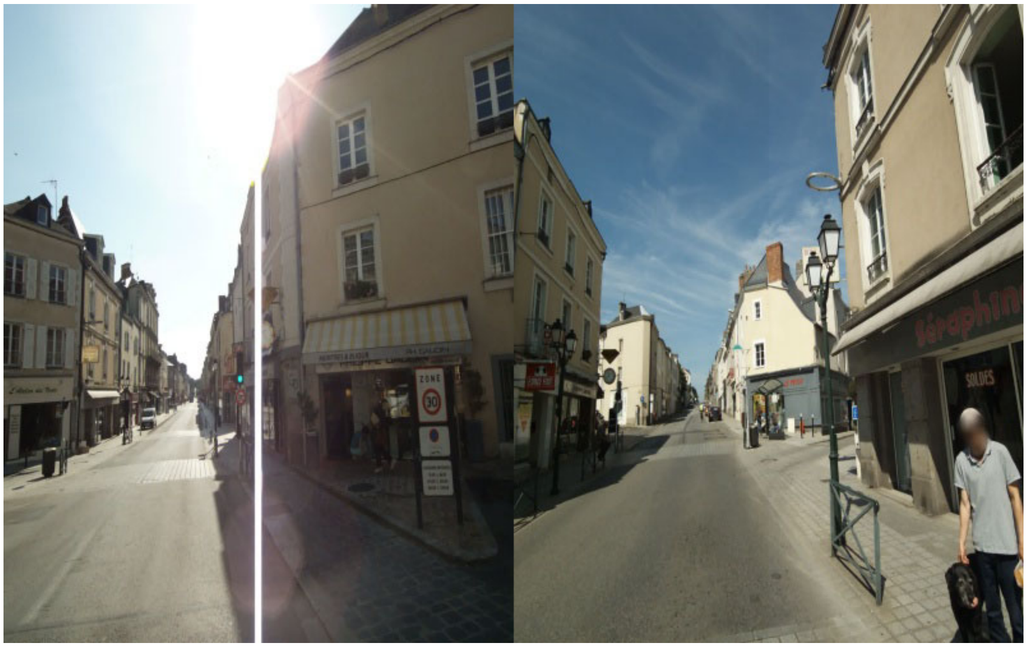


Figure A.41: Orleans1, Category : Difficult

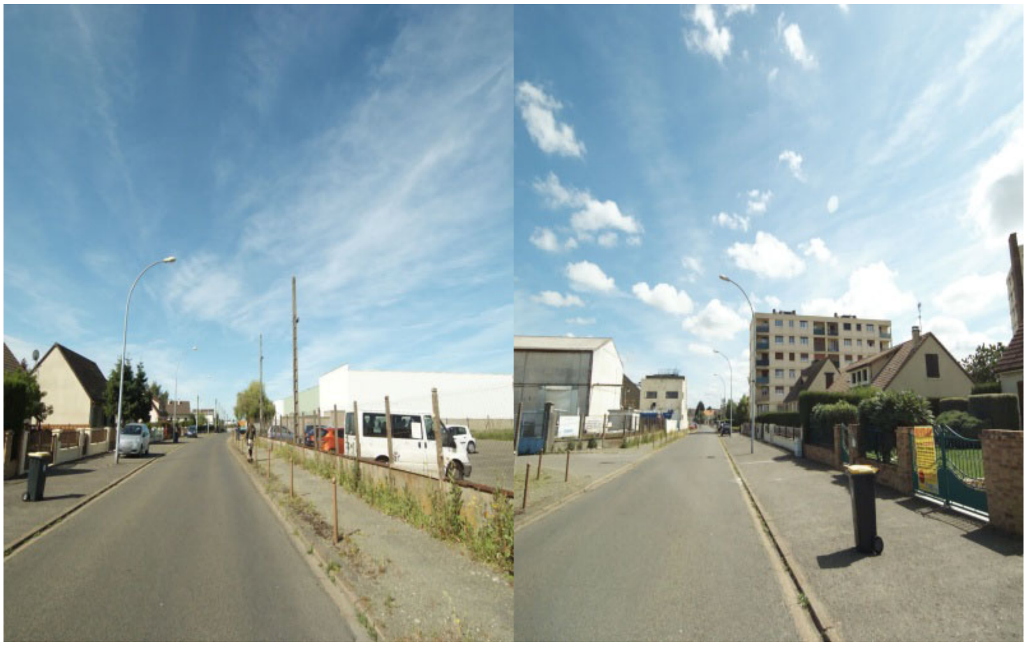


Figure A.42: Orleans2, Category : Difficult

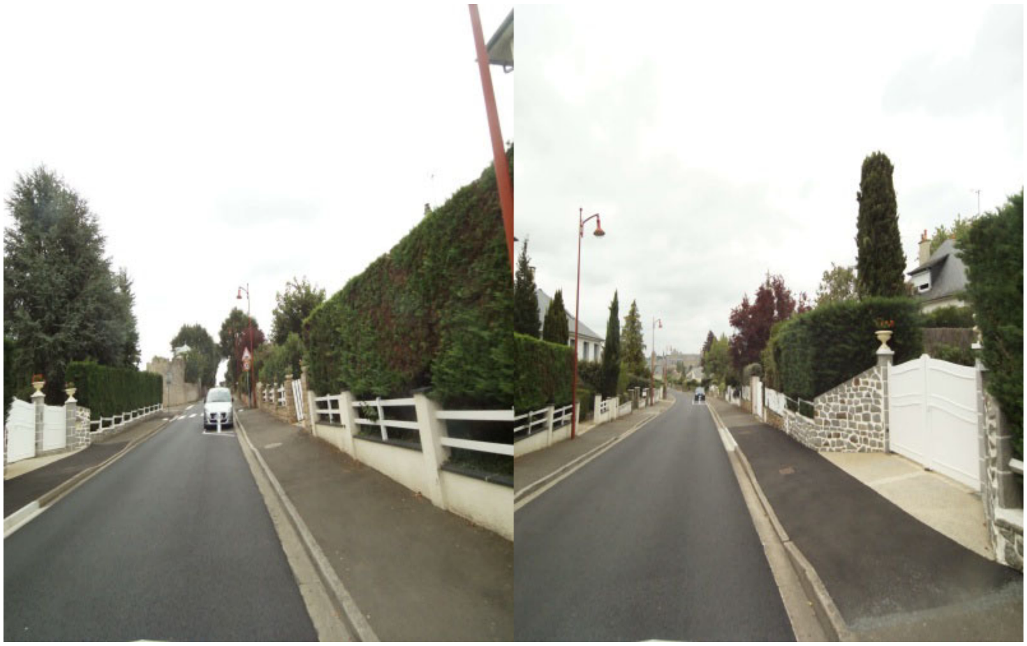


Figure A.43: Pays de la Loire, Category : Difficult