THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Towards Robust Visual Localization in Challenging Conditions

CARL TOFT

CHALMERS

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2020

Towards Robust Visual Localization in Challenging Conditions
CARL TOFT
ISBN 978-91-7905-423-6

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4890
ISSN 0346-718X

Department of Electrical Engineering
Computer Vision and Medical Image Analysis Group
CHALMERS UNIVERSITY OF TECHNOLOGY
SE–412 96 Göteborg, Sweden

Typeset by the author using LaTeX.

Chalmers Reproservice
Göteborg, Sweden 2020

Towards Robust Visual Localization in Challenging Conditions
CARL TOFT
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Visual localization is a fundamental problem in computer vision, with a multitude of applications in robotics, augmented reality and structure-from-motion. The basic problem is to, based on one or more images, figure out the position and orientation of the camera which captured these images relative to some model of the environment. Current visual localization approaches typically work well when the images to be localized are captured under similar conditions compared to those captured during mapping. However, when the environment exhibits large changes in visual appearance, due to e.g. variations in weather, seasons, day-night or viewpoint, the traditional pipelines break down. The reason is that the local image features used are based on low-level pixel-intensity information, which is not invariant to these transformations: when the environment changes, this will cause a different set of keypoints to be detected, and their descriptors will be different, making the long-term visual localization problem a challenging one.

In this thesis, five papers are included, which present work towards solving the problem of long-term visual localization. Two of the articles present ideas for how semantic information may be included to aid in the localization process: one approach relies only on the semantic information for visual localization, and the other shows how the semantics can be used to detect outlier feature correspondences. The third paper considers how the output from a monocular depth-estimation network can be utilized to extract features that are less sensitive to viewpoint changes. The fourth article is a benchmark paper, where we present three new benchmark datasets aimed at evaluating localization algorithms in the context of long-term visual localization. Lastly, the fifth article considers how to perform convolutions on spherical imagery, which in the future might be applied to learning local image features for the localization problem.

Keywords: Visual localization, camera pose estimation, long-term localization, self-driving cars, autonomous vehicles, benchmark

Included publications

Paper I C. Toft, C. Olsson and F. Kahl. ”Long-term 3D Localization and Pose from Semantic Labellings”. *Presented at the 3D Reconstruction Meets Semantics (3DRMS) Workshop at the International Conference on Computer Vision (ICCV) 2017*

Paper II C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler and F. Kahl. ”Semantic Match Consistency for Long-Term Visual Localization”. *Presented at the European Conference on Computer Vision (ECCV) 2018.*

Paper III C. Toft, D. Turmukhambetov, T. Sattler, F. Kahl and G. Brostow. ”Single-Image Depth Prediction Makes Feature Matching Easier”. *Presented at the European Conference on Computer Vision (ECCV) 2020.*

Paper IV C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl, and T. Sattler ”Long-Term Visual Localization Revisited”. *Accepted for the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 2020.*

Paper V C. Toft, G. Bökman, and F. Kahl. ”Azimuthal Rotational Equivariance in Spherical CNNs”. *Submitted for review at the the International Conference on Learning Representations (ICLR) 2021.*

Subsidiary publications

(a) V. Larsson, J. Fredriksson, C. Toft and F. Kahl. ”Outlier Rejection for Absolute Pose Estimation with Known Orientation”. *British Machine Vision Conference (BMVC) 2016.*

(b) E. Stenborg, C. Toft, and L. Hammarstrand. ”Long-term visual localization using semantically segmented images”. *Presented at the International Conference on Robotics and Automation (ICRA) 2018.*

(c) 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 Outdoor Visual Localization in Changing Conditions”. *Conference on Computer Vision and Pattern Recognition (CVPR) 2018.*

(d) M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler and F. Kahl. ”Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization”. *Presented at the International Conference on Computer Vision (ICCV) 2019.*

(e) E. Stenborg, C. Toft, and L. Hammarstrand. ”Semantic Maps and Long-Term Self-Localization for Self-Driving Cars using Image Segmentation”. *To be submitted to the IEEE Transactions on Robotics.*

(e) A. Jafarzadeh, M. Antequera, P. Piracés, Y. Kuang, C. Toft, F. Kahl and T. Sattler. ”CrowdDriven: A New Challenging Dataset for Outdoor Visual Localization”. *Submitted to the Conference on Computer Vision and Pattern Recognition (CVPR) 2021.*

(e) P. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl and T. Sattler. ”Back to the Feature: Learning Robust Camera Localization from Pixels to Pose”. *Submitted to the Conference on Computer Vision and Pattern Recognition (CVPR) 2021.*

Acknowledgements

First of all, I would like to thank my supervisor Fredrik Kahl for introducing me to computer vision, and always being patient and willing to spend time to provide guidance throughout my PhD. I have always felt supported, and for that I feel very grateful.

I would also like to extend a thank you to Torsten Sattler, who has acted as some form of inofficial co-supervisor during the last few years. You have taught me a lot about how to perform high-quality research and experimental work. I have appreciated all the time, energy and attention you have given to our projects.

Throughout my PhD, I have been blessed to have fantastic and supportive coworkers. Thank you to Måns Larsson, Jennifer Alvén and Erik Stenborg, who started before me and to whom I have always looked for guidance throughout the program.

Also, thanks to all other members of the computer vision group, and our signal processing sibling group. In particular, thank you Lucas Brynte, Mikaela Åhlen, Huu Le, Georg Bökman, Kunal Chelani, José Pedro Lopes Inglesias, Rasmus Kjær Høier, Carl Olsson, Olof Enqvist, Anders Karlsson and Christopher Zach.

Last of all, a large thank you to all my friends and family. In particular, I would like to express my gratitude to my parents, Jan and Charlotte, and my brothers, Johan and Fredrik, for being great family members and always being there to support and encourage me. And a huge thank you to my fiancée, Joanne, for her constant love, patience and support throughout this period.

Carl Toft
Gothenburg, December 2020

Contents

Part I

Introductory Chapters

Chapter 1

Introduction

At the most fundamental level, the field of visual localization aims to answer the question ”Where am I?” based on one or more images. This is a problem which humans seem to solve almost effortlessly as we go about our daily tasks: we track our position in the world with little effort, sometimes in previously unseen environments, and successfully use this information to navigate and plan the path to our destination. Getting lost is the exception rather than the norm.

However, as often seems to be the case, tasks which humans find easy and intuitive turn out to be very challenging to find a general algorithmic solution to. The problem of visual localization is no exception.

In order to provide a satisfactory answer this problem, the system needs some internal representation of the world, relative to which the answer can be provided, and it is possible to imagine many different forms in which an answer may be given. For some applications, an answer such as ”in the living room” may be sufficient, whereas other applications, such as navigation of autonomous vehicles, may require considerably more precision in the answer. For these applications, the absolute position in terms of x-, y- and $z-$coordinates, as well as orientation, relative to some coordinate system may be desired.

Providing such a six degree-of-freedom position would require a 3D model of the environment to be constructed beforehand, and the localization would occur with respect to this map. Map construction and representation are thus closely linked to the localization problem. In fact, camera-pose estimation forms a core building block of many 3D reconstruction (often called Structure-from-Motion, or SfM for short) pipelines, where the 3D model is incrementally extended by triangulating the position of cameras, one at a time [65, 67].

Fig. 1.1 illustrates the basic goal of the visual localization problem.

While the single-image localization problem is important in its own right, a common scenario in robotics is that of sequential localization where we wish to compute the most likely position of a robot, based on all measurements which have been collected up until the current time. These measurements may contain,

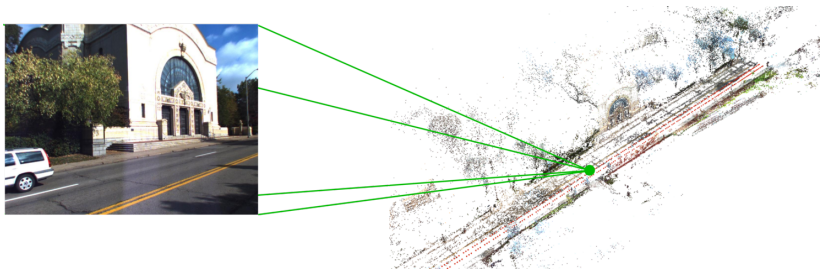


Figure 1.1: The camera pose estimation problem. Given one or more input images, we wish to compute the position of the camera which captured the image, relative to some map representation of the external world. A very common type of map is the three-dimensional point-cloud.

for example, a video feed, or a sequence of images from one or more cameras, as well as measurements from inertial measurement units (IMUs) and wheel-speed sensors. The goal is then to combine these measurements to obtain a reliable estimate of the current pose of the vehicle.

The problem of visual localization has received a substantial amount of attention in both the research community and industry the past few years, much due to the surge in interest in autonomous vehicles such as drones and self-driving cars. The camera is often seen as an affordable but information-rich sensor, which might be used instead of, or as a complement to, more expensive sensor setups with Lidars and radars. Automotive grade GPS receivers may be used to aid in positioning, but is often not accurate enough: the positioning error in these kinds of GPS receivers is often on the order of meters [36], much too high for the application. The camera may then be a viable alternative to (or complement to), these GPS receivers. Of course, there exist more accurate (but also considerably more expensive) survey-grade GPS receivers, but these may still be subject to signal acquisition failure in tunnels or indoors. They may also be unreliable in densely populated cities, where there may be no line-of-sight to the GPS satellites in the ”urban canyons”, deteriorating the localization performance as the signal reflects off of the tall buildings on the way down to street-level. The camera, if its challenges can be overcome, is thus seen as an attractive potential sensor for vehicle localization.

Today, one of the most common pipelines for camera pose estimation utilizes a 3D model in the form of a point cloud, where each point, in addition to its Cartesian coordinates, also has an associated descriptor vector (often 128 bytes, such as for SIFT-descriptors [48]), which encodes the local appearance of the point as it was seen in the cameras during map building.

In order to localize an image taken somewhere in the map, local image features are extracted from the query image, and 2D-3D correspondences are established between the features in the query image and the 3D map points, as

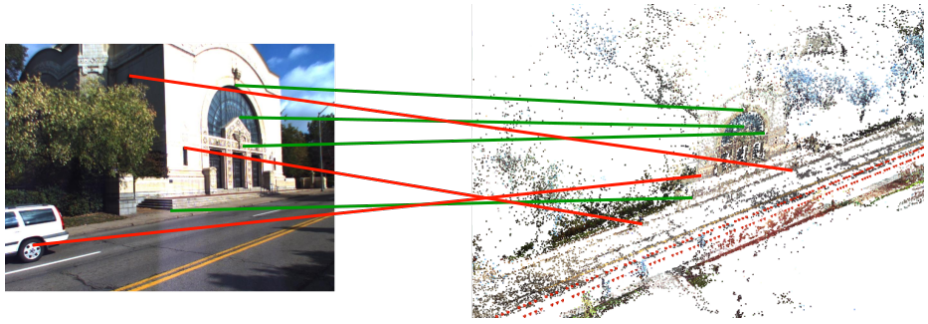


Figure 1.2: Before pose estimation, matches between the query image and the 3D model are established. Note that some of the matches will typically be incorrect (shown in red). These 2D-3D correspondences are then fed to a camera pose estimation method which computes the pose of the camera for which as many of the 3D matches in the map as possible are projected down onto the corresponding 2D points in the image.

illustrated in Fig. 1.2. This is typically done by, for each keypoint in the image, finding its approximate nearest neighbour in the map in terms of descriptor distance using a kd-tree search [54]. Using these matches, the pose which agrees with the largest number of these matches is then estimated. This may be done using a perspective-three-point solver [37, 39], in combination with a robust estimation technique such as RANSAC [26].

This purely geometrical approach based on local image appearance typically works well when the query image is taken under similar conditions as the map. However, if the appearance variation is too large, for instance due to large changes in viewpoint, lighting, or even seasonal changes, then the local image appearance may not be discriminative enough to correctly match the 2D features to the 3D map. This may then yield too few correct matches to accurately estimate the camera pose.

Fig. 1.3 illustrates how the same scene can change in appearance across seasons. If an autonomous vehicle is to employ camera-based navigation over a long period of time, then it should ideally be able to handle these kinds of visual changes. Creation of accurate maps is a time-consuming and expensive process, and creating maps for every single conceivable visual condition is unfeasible. Rather, we desire our localization systems to be robust to these kinds of changes.

While the last few years have seen great progress in this area, the visual localization systems of today are not able to fully handle these sorts of visual changes, and developing robust localization methods is a field of active research. It is also the topic of this thesis.

Figure 1.3: Two images from the CMU Visual Localization dataset[8], taken from approximately the same position, but during different seasons. While clearly the same spot, matching features based only on local appearance information would yield very few correct matches.

1 Thesis aim and scope

In this thesis we address the problem of long-term visual localization, i.e., localizing images which are taken under conditions very dissimilar to the condition under which the mapping images were captured. Specifically, we address the problem of single-image localization, as well as local feature matching between image pairs. We try to make the localization pipeline more robust by incorporating higher-level information in the form of semantic information and pixelwise depth information. By employing a semantically labelled 3D model, we show how this additional information can be used directly for pose estimation as a replacement for e.g. SIFT, and how it can be used as a complement to a traditional feature-based pipeline by using semantics to identify mismatched correspondences (the red correspondences in Fig. 1.2). We show how pixelwise depth information, obtained from a monocular depth estimation network, can increase the robustness of local features to viewpoint-changes, and also introduce three new datasets aimed at evaluating long-term visual localization algorithms. To conclude the thesis, we also include a paper on deep learning for omnidirectional images, since we believe these methods (when they are mature enough) will be useful for localization and mapping applications.

2 Thesis outline

This thesis consists of two parts. In the first part (of which this is the first chapter), background material is presented in Chapter 2. This is intended as a ”warm up” containing the necessary background material to comfortably tackle the ap-

pended papers in the second part of the thesis. Readers already familiar with the camera-pose-estimation problem can likely skip this chapter without any loss of comprehension when reading the remainder of the thesis. Chapter 3 summarizes the contents of the five papers, and clarifies the author's contribution to each of them. Lastly, Chapter 4 concludes the first part of the thesis and presents a final outlook on possible future directions for research.

The second part consists of five appended papers, and represents the main content and novel research work of the thesis.

Chapter 2

Background

The main part of this thesis consists of the papers appended in Part II. However, research articles are often quite terse and do not elaborate significantly on the background material since the reader is assumed to already be more or less familiar with it. This can make reading papers in a new area very challenging, since the overall setting in which the paper takes place may not be fully explained.

The purpose of this chapter is to serve as such a warm-up and summarize the relevant background material necessary to understand the contents of the papers included in Part II, and elaborate some more on the general context that may be lacking in the individual papers themselves. The structure of the chapter is as follows: Sec. 1 gives an coarse taxonomy of the visual localization problem, and some common approaches that may be used to solve it. Sec. 2 introduces what local image features are, how they can be used to establish matches between images, and how they often break down in the long-term localization scenario. Sec. 3 discusses the geometry of camera projection, and how correspondences between 2D image points and 3D map points may be used to calculate camera pose relative to a map. Sec. 4 explains how image-retrieval based visual localization works. Lastly, Sec. 5 brings up the topic of semantic segmentations, and suggests how they may potentially aid in visual localization tasks.

1 Visual localization

As mentioned in the introduction, the problem of visual localization is to determine the position of the camera which captured one or more images with respect to a map. There exist many different variations of this problem, with accompanying solutions, depending on what input data is available to aid in the localization (such as the number of images, any prior information on location, data from additional sensors such as from an inertial measurement unit, Lidar, etc) as well as what representation is used for the map.

One may roughly categorize the most common localization methods into two categories [10]: topological and metric. This is not a strict classification, and any given method may fall somewhere in between.

In topological localization, the map is represented as a discrete set of places, often encoded as nodes in a graph. The nodes may then represent places, and adjoining edges correspond to possible paths between these places. The visual localization problem is the to select, from this finite set of nodes, the one which corresponds to the position where the image was captured. Depending on what the graph represents, this may correspond to an answer such as ”in the kitchen”, or a specific street intersection. This thus corresponds to a discrete classification problem. Finer-grained localization could be obtained by placing nodes densely, and attaching metric coordinates to each node, as done in e.g. [9, 10, 76]. Assigning the query image to a node thus also yields approximate metric coordinates for the position of the camera.

This discrete form of visual localization is sometimes referred to as visual place recognition, and is often solved using image retrieval methods, which will be discussed briefly in Sec. 4.

The other class of localization methods are the metric methods. Here, the goal is to output the camera position in metric coordinates, such as latitude, longitude and altitude, or more generally, the coordinates with respect to some pre-defined coordinate system. In full six degree-of-freedom camera pose estimation, the goal is to also estimate the three rotational degrees of freedom, in addition to the three translational degrees of freedom, for a total of six real numbers.

Metric localization methods often employ a pre-constructed 3D map of the environment. By associating landmarks detected by the vehicle’s sensors to landmarks with known position in the map, it is possible to reason about the position of the vehicle in the map. However, it should be noted that not all metric localization methods need to rely on an explicit 3D map of the environment. Other approaches, such using a neural network to directly regress the six degree-of-freedom pose directly from the image, have been explored [38].

For cameras, the landmarks typically consist of point-features detected in the image by an image feature detector. These image features form the backbone of modern 3D reconstruction and visual localization pipelines, and understanding them is central to understanding the shortcomings of current visual localization systems in the long-term localization scenario. We will thus describe these local features more in depth in the following section.

2 Local image features

Image feature (or keypoint) detection and description is one of the most fundamental problems in computer vision, and forms the foundation on which a large body of other methods rest [71]. The purpose of the feature detector is to extract from an image a set of interest points we believe we will be able to redetect in a different image of the same scene, and the purpose of the descriptor is to encode the appearance of the keypoints into a descriptor vector, such that keypoints in the first image can be associated to keypoints in the second image (or map).

If the same set of points can be detected in a different image, we can establish point-correspondences between images, or between an image and a 3D model. Image-to-image correspondences can be used for calculating relative camera pose, which enables subsequent 3D reconstruction of the scene [32]. Image-to-model correspondences allow us to compute the camera pose relative to the model.

In this section we will first briefly discuss keypoint detectors and descriptors, and then have a look at them in the context of long-term visual localization.

2.1 Feature detectors

Feature detectors have been studied since the early days of computer vision, and as such there exist a large number of feature detectors (see e.g. [75] for a survey). Common among most of them is that they try to find corner- or blob-like features in the images; flat areas with uniform brightness are not distinctive enough to match unambiguously across images, and the same is true for edges, see Fig. 2.1.

Corner detectors work by computing some statistics of the extracted image patch. For example, they might examine the Hessian, the Laplacian-of-Gaussian [45], or the Difference-of-Gaussian (DoG) [49] of the image. Other detectors examine the auto-correlation function of the image [30, 74]: imagine extracting a rectangular patch centered around the point. If the contents of the patch changes considerably as we slide the window in any direction (with differences measured in, perhaps, sum-of-squared difference in pixel intensity between the original patch and the translated one), then the point corresponds to a corner. On the other hand, if the patch contents only change in one direction, but are more or less constant in the other direction, the point likely lies on an edge, and it seems unlikely we would be able to accurately re-identify it in a different image of the same scene.

Figure 2.1: Three example image patches from an image. A feature detector should trigger on corner-like points we believe we could re-identify in a different image. Do you think you could find and correctly match the three extracted patches in a different image of the same scene?

2.2 Feature descriptors

After having identified the keypoints in an image by running a feature detector on it, we need some way to encode the appearance of the keypoint, so that we can match it across images, or match it to a corresponding 3D point in a point cloud. While it is certainly possible in some cases to use a very simple similarity metric between image patches such as correlation or sum-of-squared-differences (SSD), the perhaps most widely used way of encoding image patch appearance is the gradient histogram [19]. The very popular SIFT and SURF features [11, 50] are examples of features which utilize a gradient histogram for describing the keypoints.

To compute the gradient histogram of a patch, the gradient (horizontal and vertical derivatives of the pixel intensity) is first computed for each pixel. The pixelwise gradients are then binned into eight different bins, depending on their direction. I.e., the sum of the lengths of all gradients pointing in a direction between 0° and 45° are put in the first bin, the sum of lengths of all gradients pointing in a direction between 45° and 90° are put in the next bin, etc. This yields a total of eight numbers. An image patch can thus be compressed into eight numbers, representing, in some sense, in which directions any edges are oriented, and how strong these are.

However, when condensing a patch centered on a keypoint, the patch is first subdivided into 4×4 sub-patches, and each of these sub-patches is compressed into a vector of eight numbers using the above method. All these 16 vectors are

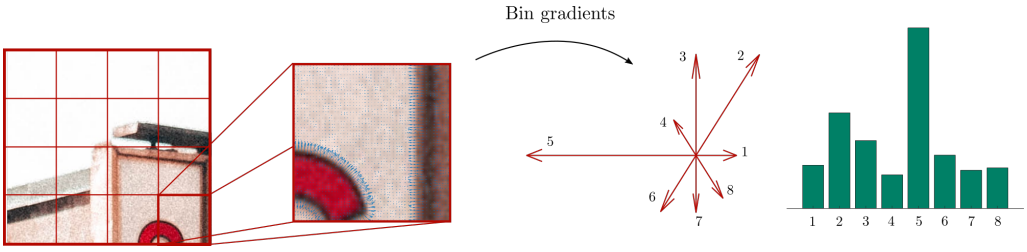


Figure 2.2: Illustration of gradient-histogram computation. The image is divided into 4×4 sub-patches. For each sub-patch, gradients are computed at each pixel and their magnitudes are added into one of eight bins depending on their orientation. The total values in these bins yield a descriptor-vector with eight entries.

then stacked into a single vector of 128 numbers. This vector is then the SIFT-descriptor of the patch. This process is illustrated in Fig. 2.2 for one of the image patches from Fig. 2.1.

Now, in order to find point correspondences between two images, or an image and a map, these local features (consisting of both the set of detections returned by the feature detector, as well as the corresponding feature descriptors) can be matched by, for each feature in the image, finding the nearest neighbour in the descriptor space in the other image (or map).

The SIFT descriptor, while simple, is remarkably effective at matching features across images, even under moderate changes of illumination and perspective distortions. However, while it is illustrative to know how these traditional feature detectors and descriptors work, it should be noted that recently, learned features extracted using deep neural networks have started to consistently outperform SIFT features on matching tasks. Most notable of these at the time of writing are perhaps the SuperPoint [20] and D2-Net [21] features. That said, the SIFT descriptor still remains one of the most popular baselines for comparison when developing new local image features.

2.3 Feature matching across large appearance changes

While very powerful, the described local image features are not without limitations. They struggle to find correct matches between images taken from very different viewpoints, such as between two images showing the same street intersection but taken from perpendicular streets, or between two images of the same scene taken during dissimilar environmental conditions. For example, reliably establishing feature matches between daytime and nighttime images, or between images taken during different seasons (see Fig. 2.3), remains very challenging and is still an open problem, and this is also why long-term visual localization is

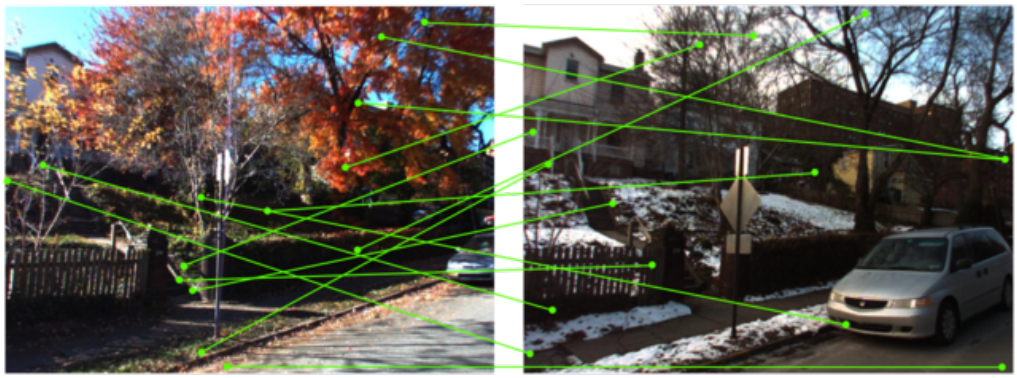


Figure 2.3: An example of SIFT-feature matching between two images taken from approximately the same viewpoint, but during different seasons. A Lowe ratio of 0.9 was used in the matching process. Not all feature matches are shown, but only those consistent with the estimated epipolar geometry.

difficult.

Fig. 2.3 shows an example of SIFT-feature matching between two images taken from approximately the same viewpoint, but far apart in time (during different seasons). Keypoints are extracted using the DoG detector, and then each feature in the left image is matched using approximate nearest neighbour matching (in descriptor space) to the features in the image to the right. A relative camera geometry which is consistent with as many matches as possible is calculated, and the figure shows the surviving, geometrically verified matches. None of them are correct.

The reason for this failure is two-fold: first of all, the detector does not trigger on the same set of points in the two images (i.e., the detector is not *repeatable* over these appearance variations), and secondly, the descriptor for a given point changes too much between the images to be reliably matched. The low-level pixel intensity in a local neighbourhood around any given patch looks completely different in the two images, even though they correspond to the same point. In the article [1], several extensive experiments are performed showing (and quantifying) the non-repeatability of the most popular feature detectors for varying degrees of viewpoint and lighting changes.

If a map is constructed from images captured during the same condition as in the left image, and we then revisit the same area at a later time, when the environment looks as in the figure to the right, how do we perform robust visual localization if traditional local-feature matching yields mostly incorrect matches? This is the core problem which is discussed in the appended papers in this thesis.

At this stage, we can make one key observation regarding this simple experiment. If I asked you to manually provide a set of, say, ten point correspondences

between the two images, would you be able to? Ideally, if they are to be used for camera pose estimation, the error when ”clicking out” the correspondences should not be more than a few pixels.

Most likely you would. Though the low-level content of the image may have changed drastically in any given image patch due to changes in lighting, added snow, removed leaves and so on, we can (perhaps with some slight effort) match points based on a higher-level, semantic reasoning. We could match the corners of the building roof, the corners of the street sign, the tips of the poles in the picket fence, and so on. None of this would be possible by simply comparing the low-level image content between pairs of patches; we must reason using considerably more high-level information about the image contents to find the correct matches.

It is likely by learning these kinds of higher-level features that local features based on machine learning have started outperforming traditional features in the recent literature. Additionally, this thought experiment seems to suggest that extracting higher-level scene semantics may be useful for the purpose of visual localization. This idea is explored in the first two appended papers in this thesis.

3 Structure-based localization

Now that we have understood how to identify local features in images, and how these can be matched between images, we will now have a look at how these features can be used for the task of visual localization. We will first have a look at structure-based methods, i.e., methods which employ a 3D representation of the environment (such as a point cloud) to aid in the localization process. In order to understand how this is used, we must first have a quick look at the image formation process for cameras, i.e., the mechanism by which the 3D world is projected down into a two-dimensional image.

3.1 The camera model

The standard pinhole camera model is a subject which has been written extensively about before, and every course on optics, computer vision and computer graphics covers this, so it seems somewhat redundant to write yet another introduction. We will here only write down the very basics of camera projection. For more information, see e.g. Chapter 6 in [31].

The simplest and perhaps most popular camera model is the pinhole camera model. A pinhole camera, or a *camera obscura*, is simply a box with a tiny hole (a pinhole) in one of its sides. When light is emitted from, or scatters off of, an object in the scene, some of that light might pass through the hole, which we will call the center of projection O, and will fall onto a point on the opposite side of

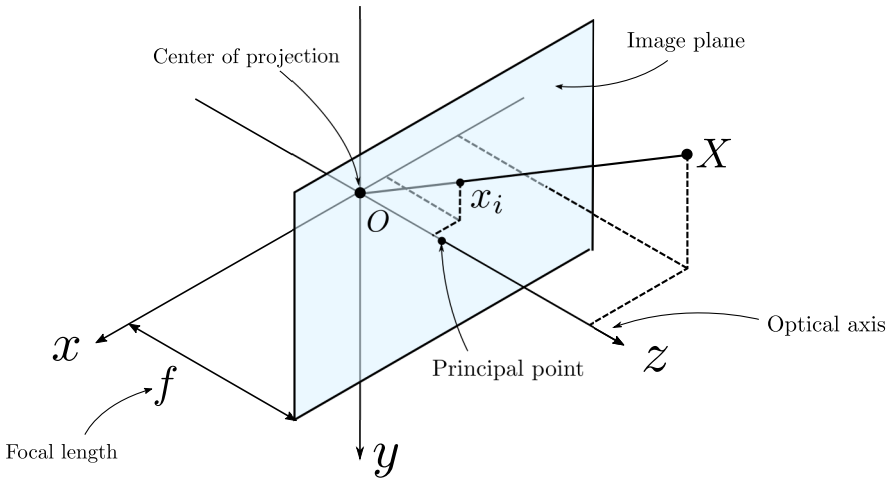


Figure 2.4: Projection of a 3D point X onto the image plane of a pinhole camera.

the box. An image of the scene will thus form on the inside of the box. If we were to place a photographic plate on the side opposite from O, an image would form on the plate. We have thus created a simple camera.

Fig. 2.4 shows the projection process for a point X, with one slight modification: the image plane where the image forms is now imagined to be in front of the center of projection. This is of course not what happens in practice; the image is formed on a plane a distance f behind the center of projection. However, the image which forms on the physical imaging plane will be a mirror image of what an observer would see when "looking out" from O. When displaying the captured image, it will have to be mirrored in order to display what was actually seen by the camera. To simplify the calculations, it is more convenient to imagine the image being formed on the *image plane* placed a distance f in front of the center of projection.

The relationship between the world point X and the imaged point x_i can now be deduced from similar triangles. Let $X = (X_w, Y_w, Z_w)$ and $x_i = (x, y, f)$, then clearly $x = X_w \cdot f/Z_w$, $y = Y_w \cdot f/Z_w$.

The equations for camera projections are most conveniently expressed in the framework of projective geometry. In this framework, it is customary to express points in 2D and 3D in terms of their homogeneous coordinates: the point (x, y) in $\mathbb{R}^2$ is now represented as $(x, y, 1)$. Furthermore, we identify all points along the same ray through the origin, i.e., the point (x, y) in Euclidean coordinates is identified with all points $\vec{x} = \lambda(x, y, 1)$ in homogeneous coordinates, with $\lambda \neq 0$. Similarly, the point $X = (X_w, Y_w, Z_w)$ in $\mathbb{R}^3$ is identified with all points with homogeneous coordinates $\vec{X} = \lambda(X_w, Y_w, Z_w, 1)$.

Using this formalism, the camera projection equations derived above can be

written as

$$\lambda\vec{x} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3\times3} | & \vec{0} \end{bmatrix} \vec{X}, \tag{2.1}$$

where $\vec{x}$ and $\vec{X}$ are the homogeneous representations of the points x and X, f is the focal length, $I_{3\times3}$ is the 3×3 identity matrix, and $\vec{0}$ is the vector of all zeroes in $\mathbb{R}^3$.

To obtain the final projection equations for the pinhole camera, two additional things need to be modified. First, the vector x is measured in meters (or whichever unit of length is used to express the focal length f). These coordinates are often called the normalized image coordinates. However, when working with images, one initially obtains the image features in terms of their pixel coordinates in the image, counting rows from up to down, and columns left to right, with the pixel $(1, 1)$ being in the top left corner. To transition from normalized coordinates to pixel coordinates, one multiplies the normalized coordinates by the camera calibration matrix K given by

$$K = \begin{bmatrix} \alpha_x & s & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

where α_x and α_y is the focal length in terms of pixels (these are identical if the pixels are square), s is called the skew and is zero for rectangular pixels, and the point (c_x, c_y) is the principal point in pixel coordinates, i.e., the point where the optical axis intersects the image plane.

Lastly, the camera may not actually be located at the origin in the coordinate system of the 3D world, and it need not be oriented as shown in Fig. 2.4. Suppose instead that the center of projection is located at some arbitrary position C, and that the camera has some orientation described by the rotation matrix R, which brings the camera coordinate axes to the world coordinate axes, then the final projection equation may be written

$$\lambda\vec{x} = P\vec{X}, \tag{2.3}$$

where

$$P = KR[I_{3\times3} | -C] \tag{2.4}$$

is called the camera matrix, which fully determines the projection of world points into image coordinates.

Note that the camera matrix contains five intrinsic parameters, related to the camera's internal workings (sensor size, focal length, skew, etc.), and six extrinsic parameters (related to the position of the camera in the external world), for a total of eleven parameters.

In the visual localization problems discussed in the papers in this thesis, the internal parameters have all been determined beforehand in a calibration procedure. This is typically done by taking several pictures of a calibration object whose physical dimensions are known very precisely [79].

One final thing worth bringing up before moving on to structure-based camera pose estimation, is that the pinhole camera model is an idealized, mathematical model for the image formation process. In practice, since the pinhole camera only lets in a very small amount of light, a larger aperture is needed (the exposure time for the world's first pinhole camera was around eight hours [33]). However, increasing the aperture size leads to blurry images in a pinhole camera, so a lens is placed in the aperture to focus the light. Real optical systems can be quite sophisticated, and the lenses introduce several different kinds of aberrations in the optical system, one of which is that of non-linear distortion.

An ideal pinhole camera maps lines in 3D to lines in the image, but non-linear distortion causes these lines to map onto curved arcs in the image, often with increasing distortion as one moves away from the principal point. However, this type of distortion has been studied for a long time, with its roots in the photogrammetric community [13, 25], and simple and accurate models for the non-linear distortion have been created. These distortion parameters are typically estimated during the camera calibration procedure [79], and distorted images can then be *rectified* into corresponding pinhole camera images with sub-pixel accuracy.

3.2 Correspondence-based camera pose estimation

Now that we understand the camera model, we are in a position to look at how structure-based pose estimation works. The problem of estimating the camera pose from a set of point or line correspondences is perhaps one of the oldest problems of computer vision, and there exists a vast literature on the subject. In this section we will summarize the most important points needed to understand the papers in Part II.

Minimal solvers

Assume that we have identified a set of points $\{x_i\}_{i=1}^n$ in the image, and have matched these to a set of 3D points $\{X_i\}_{i=1}^n$ in a 3D point cloud. We can then write down the projection equation (2.3) for each 2D-3D correspondence. Note that the projection equations yield three equations per correspondence, but each correspondence also introduces an extra unknown λ. Each correspondence thus fixes a net two degrees of freedom in the camera matrix. If the camera intrinsics are known, e.g., if we have calibrated the camera beforehand, there are a total

of six unknown degrees-of-freedom in the camera matrix, corresponding to the position of the camera center, and the camera orientation.

It thus seems a total of three correspondences should in general suffice to compute the camera pose. This is indeed the case. This problem is known as the perspective-three-point problem (P3P), and many different methods for solving this problem have been presented over the years [27, 37, 39], with the first known solution dating back to 1841 [28].

A classical way of solving the problem is to, for each pair of points, use the law of cosines to formulate a quadratic equation in the unknown depths of the 3D points [26, 46, 53]. The three possible pairs thus yield three quadratic equations in the three unknown depths. One may solve this system in a variety of ways. One way is to use the Sylvester resultant to successively reduce the system into a single fourth degree equation, which will yield up to four possible solutions to the problem [27]. In general, a fourth correspondence is needed to disambiguate the solution. For four or more point correspondences, there exist linear methods for calibrated camera pose estimation [2, 58].

In computer vision, methods which solve a problem using only the minimum number of correspondences required theoretically are often called *minimal solvers*. They play a central role in robust estimation, which we will see below.

The P3P solvers mentioned above is one kind of minimal solver, but there exist other kinds of minimal solvers for camera pose depending on what kind of information is available. For example, if the gravity direction is known in the camera reference frame (perhaps supplied by an IMU, or estimated from vanishing lines in the image [12]), the camera pose only has four extrinsic degrees of freedom left, enabling pose estimation from only two point correspondences [41].

Robust estimation

Once correspondences between the image and a map have been established, feeding all correspondences to an n-point pose solver to estimate the camera pose is a very bad idea. The reason is the presence of *outliers*, in the data, i.e., mismatched correspondences which are completely wrong. Including these in a pose estimation routine which minimizes an algebraic error or the geometric reprojection error of the correspondences will introduce gross errors in the estimated pose. A way to remove outliers, or to downweigh their contribution to the error function must be devised.

The undoubtedly most popular technique for outlier detection and removal in computer vision is the *random sample consensus maximization* (RANSAC) procedure [26]. This method is not limited to only camera pose estimation, but works in a wide variety of robust estimation problems, such as line and plane fitting, 2D and 3D homography estimation, and so on.

The main idea is to randomly sample a minimal subset of the correspondences, i.e., extract a subset which is just large enough to apply a minimal solver on it. For calibrated camera pose estimation, a subset of three correspondences would be extracted. Once the camera pose (or homography, or whichever type of model it is we are estimating) has been computed from the minimal subset, we check how many of the remaining correspondences agree with this model, within some pre-specified error tolerance. For camera pose estimation, we may project down the 3D points of the remaining correspondences, and measure the reprojection error. We count the number of inliers (the size of the *consensus set*) for the estimated model. We repeat this procedure a pre-specified number of times, and simply return the model with the largest consensus set.

Typically, the final pose is then refined using the whole consensus set, using an iterative local optimization procedure to minimize the reprojection error of the correspondences in the consensus set.

RANSAC is not the only possible approach to handling outliers, and there exists a large body of work concerned with developing other kinds of robust estimation techniques, such as globally optimal model fitting under the L_2 norm [7], and methods based on branch-and-bound [22, 23]. However, these methods tend to suffer from an exponential worst-case runtime.

Due to its simplicity and generality, RANSAC remains one of the most popular robust estimation method. However, it has several shortcomings worth mentioning. First, RANSAC is clearly not guaranteed to return the optimal solution. Secondly, the algorithm may be sensitive to the selected error threshold: the model calculated from a set of correct correspondences may not actually have a large consensus set due to noisy observations. Lastly, the runtime of RANSAC increases very rapidly as the ratio of outliers approaches one. This is especially a problem in long-term visual localization, where a large fraction of the correspondences are expected to be incorrect due to the large appearance variations of the scene, as well as in large-scale localization, where the number of correct matches typically decreases with model size due to visual ambiguity resulting from repetitive structures in the environment [78].

Due to the rapid increase of the runtime with the outlier ratio, it is sometimes desired to prune outliers if possible before feeding the correspondences into the RANSAC procedure. This leads us to a class of methods often called *outlier filters*, or *outlier rejection schemes*.

Outlier rejection

If a large amount of outliers are expected, it may be beneficial to utilize an outlier rejection scheme. Typically, these use prior information about the camera pose to reason about which correspondences may be outliers. For example, [44] presents an outlier rejection scheme where the camera rotation is fully known; in this

Figure 2.5: Visual place recognition formulates the localization problem as an image retrieval problem. Given a query image and a set of database images, find the database image most similar to the query image (according to some metric).

scenario it is possible to, for each correspondence, calculate an upper bound on the number of inliers possible for a camera pose which also has that particular correspondence is an inlier. If this score is low, it can then be discarded. The article [70] presents an outlier rejection scheme which utilizes prior information about camera height and vertical direction. One of the included papers in this thesis presents a soft outlier rejection scheme based on exploiting the semantics of the observed image.

4 Image-retrieval based localization

In the beginning of the chapter, we discussed that visual localization methods can be roughly categorized into metric and topological. The metric methods are typically 3D structure based, i.e., they employ a 3D model of the environment and use this to calculate the pose of the camera which captured the query image, as discussed in the previous section.

The topological localization methods, also called *visual place recognition* methods, work in a different manner. Here, the localization problem is instead formulated as an image search problem: given a set of database images and a query image, find the database image which most closely resembles the query image, see Fig. 2.5.

If metric information is included in the map, for example if the images are geotagged and have associated GPS metadata, then the position of the query

image could be approximated by the position of the database image [63].

One of the advantages of image retrieval based approaches is that image retrieval is a well-studied problem in computer vision, and efficient search strategies using vocabulary trees [56] and inverted file indices have been developed for this problem.

A common implementation of image retrieval is to compute a whole-image descriptor for all database image. The whole-image descriptor could be based on a bag-of-visual-words (BoW) approach [15, 34, 35, 73], or use a vector-of-aggregated-descriptors (VLAD) [3, 72]. The basic idea behind these methods is to extract local features from the image, such as SIFT descriptors. If the descriptor space has been quantized beforehand, for example by clustering all descriptors from a different set of training images using a k-nearest-neighbour approach, each descriptor extracted from the query image can be assigned to the nearest cluster. The corresponding whole-image descriptor would then be the histogram over the number of features assigned to the different clusters (visual words). Learning based approaches such as NetVLAD [4], which compute a whole-image descriptor using a neural network, also exist and perform very well.

To localize an image, this whole-image descriptor is computed for all database images, as well as for the query image. The nearest-neighbour (or k-nearest neighbours) of the query image are then extracted from the database. The pose of the query image can then be approximated using the pose of the top-ranked database image.

Computing a global descriptor from extracted local descriptors yields a more compact representation of the image, but spatial information is lost. To compensate for this, spatial verification and re-ranking [57] can improve the results. This means that the top-ranked images after image retrieval are then re-ranked based on how well it is possible to fit some geometric transformation that maps correspondences from the query image to the database images in question. In other word, this approach tries to take the spatial layout of the features into account to reconsider the ordering of the suggested best matches. For a survey of the visual place recognition problem, see [51].

Image retrieval methods have a significant advantage over structure-based methods: a database of geotagged images is significantly easier to construct, maintain and extend than a metric 3D reconstruction. It was believed that structure-based approaches yielded more accurate pose estimates than image retrieval based methods, but there is recent evidence that estimating the camera pose using a local 3D model created ”on-the-fly” from the top-ranked database images can yield just as accurate, if not more accurate, poses than pure structure-based methods [61]. Additionally, using image retrieval as a pre-processing step in a structure-based system almost always increases the localization performance, as

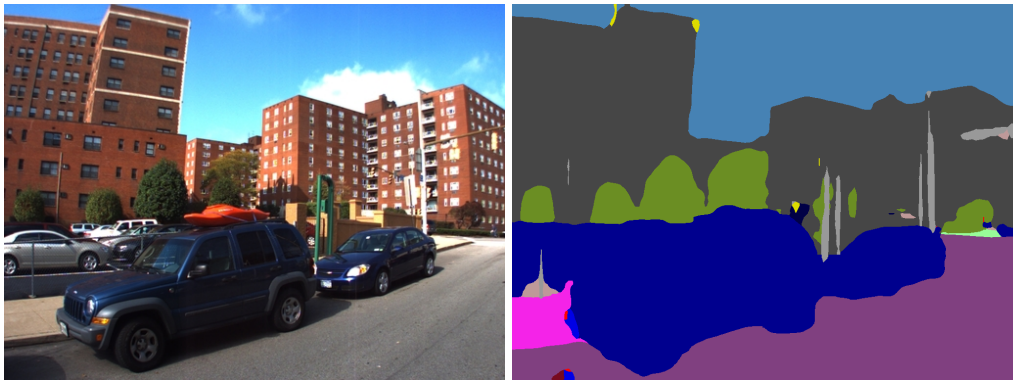


Figure 2.6: An image from the CMU Visual Localization dataset and its corresponding semantic segmentation.

we will see in Paper IV.

5 Semantic segmentation

When discussing local image features in Sec. 2, we discussed the problem of non-invariance of local features. That is, under viewpoint and illumination changes, the feature detector may trigger on a different set of points, and the feature descriptor may change to such an extent that feature matching based on descriptor distances fails completely.

The problem is that traditional descriptors only contain low-level intensity information in a patch around the feature points. They contain no higher-level, holistic understanding of the scene. Even in a fairly challenging scenario, a human would likely be able to provide matches between two images by utilizing this high-level information.

In the computer vision community, much progress has been made in recent years on the problem of semantic segmentation. This problem consists of assigning, to each pixel in an image, a label from a pre-defined set of classes. Fig. 2.6 shows an example of an image together with its semantic segmentation. The image comes from the CMU Visual Localization dataset [10] and the segmentation is performed using the network [77] trained on the Cityscapes classes [17, 18]. Cityscapes is a dataset and public benchmark for semantic segmentation of street-view images into semantic classes such as road, sidewalk, car, pedestrian, pole, building, sky and so on. These aim towards a high-level understanding of images taken in street scenes, which may be relevant for the task of visual navigation for autonomous cars.

Today, semantic segmentation is most commonly performed using some variants of convolutional neural networks (CNNs) [14, 47, 60, 80]. Sometimes, a

conditional random field (CRF) model is added on top of the network output to encourage a structured segmentation [6, 14, 43].

Even though the local image features may not be invariant to illumination, seasons, day-night changes and so on, a good semantic segmentation algorithm should ideally be robust to these kinds of variations. If semantic information can be reliably extracted under these conditions, it may be possible to utilize this during the localization process. For example, it may be used for identifying incorrect matches (a feature detected on a street sign should be matched to a 3D point with the corresponding label), as done in e.g. [5] in the context of object retrieval.

As observed in [5], the idea of integrating semantic information into the classical computer vision pipelines is something of an emerging theme in the computer vision literature, since this has been shown to improve the performance across several different tasks. For example, in [42], the authors show that for the problem of dense stereo reconstruction in road scenes, jointly reasoning about depth and semantic classes improves the performance of the reconstruction dramatically. The article [64] reaches similar conclusions, where they instead jointly infer semantic labels and depths for ”stixels”. Similarly, in [29] a voxelized multi-view-stereo reconstruction is performed by joint geometric and semantic reasoning. It is found that the geometry helps enforce the semantics across images, and the semantics help the depth reconstruction e.g. in areas (such as the ground) for which depth values are sampled more sparsely.

5.1 Semantics for visual localization

Lastly, to wrap up this chapter, we arrive at semantics for visual localization. As mentioned above, integrating semantics into the classical, geometrical pipelines has been shown to improve performance. The question is then whether this is also the case for visual localization. There exists evidence this is indeed the case (see for example [55, 66, 68] for examples where higher-level features such as lane-markings and pole-structures are used for localization), and it is also the topic of the first two articles appended in Part II.

Chapter 3

Thesis Contributions

The topic of this thesis is that of long-term visual localization. Current visual localization approaches typically work well when the mapping images are similar in appearance and view-point to the images to be localized, but under change of viewpoint, illumination or seasons, the localization performance typically degrades rapidly. The non-repeatability of the feature detector, and the non-invariance of the feature descriptor, are the two main culprits.

We have mentioned in the last chapter that the reason behind the failure of the traditional local feature approach in this scenario is that they rely only on low-level pixel-intensity information, and that incorporating a higher-level scene understanding via the semantic segmentation may potentially alleviate some of these problems. This is the topic of two of the included papers (Papers I and II).

While working on the first two articles, we noticed the lack of suitable long-term visual localization datasets to evaluate our methods on. There were a variety of localization datasets available, but they either did not have sufficient variation in weather, seasons or illumination to evaluate the performance of the methods in the long-term localization scenario, or, if they did include this variation, they did not come with accurate six degree-of-freedom camera poses.

Motivated by this, we joined forces with a quite large group of researchers and put together three datasets (where we at Chalmers were responsible mainly for one) specifically aimed at evaluating six degree-of-freedom long-term visual localization. This work resulted in Paper IV, which is the journal version of a conference article published at CVPR 2018. The benchmark we present in this article has turned out to be fairly popular, and we have hosted several workshops at CVPR and ICCV on this work, where we accept submissions to a long-term visual localization challenge.

During the summer of 2019, I had the opportunity to do an internship at Niantic's research lab in London. The work I did there resulted in Paper III, in which we tackle the problem of increasing the robustness of local features to viewpoint changes by utilizing pixelwise depth information gained by running

the images through a monocular depth-estimation network.

The last paper, Paper V, differs in theme from the other articles. This paper considers the problem of how to perform convolutions on spherical data, such as images obtained from an omnidirectional camera. The original idea for the article came up when considering an aspect of the visual localization problem. Specifically, a known challenge is to recognize a previously visited area when it is revisited, but from a different direction. This could be, for instance, a car driving down the same road twice, but in opposite directions.

When using a regular perspective camera with limited field of view in this scenario, the scene will look very different in the two different traversals. Each traversal will observe different parts of the scene, and the parts that are covisible in both traversals will tend to suffer from quite severe perspective distortion.

It seemed to me that this problem would be almost completely bypassed if an omnidirectional camera was employed instead. In this case the two traversals would observe essentially the same scene, and it should be possible to approximately align the images with only a rotation. Learned local features have recently started outperforming traditional local features for long-term localization, and I wondered if these learned features could be applied to omnidirectional imagery.

However, it turned out that the question of how to perform convolutions on spherical data has not yet been fully settled, and it felt necessary to perform some more foundational work in this area to better understand this problem before turning our attention to applications. The result of this work is presented in Paper V.

In the remainder of this chapter we present a high-level summary of each of the included papers in turn. The full papers are appended in Part II.

Paper I

C. Toft, C. Olsson, F. Kahl. ”Long-term 3D Localization and Pose from Semantic Labellings”. *Presented at the 3D Reconstruction Meets Semantics Workshop at the International Conference on Computer Vision 2017.*

This paper was something of a pilot-study where we examined how well it is possible to perform single-image visual localization, using only the semantic segmentation of the query image. In other words, is the semantic information alone sufficient for visual localization? The method was evaluated on two small subsets of the Oxford RobotCar dataset, each the size of a few city blocks. The method uses an image-retrieval method (based only on the semantic segmentation), and then refines the 6 degree-of-freedom (DoF) camera pose by minimizing a cost function based on how well the reprojection of a semantically labelled 3D model, consisting of semantically labelled points and curves of the envi-

Figure 3.1: Illustration of the method in Paper I. Any given pose may be evaluated by how well the semantically labelled structure projects down into the given pose. Only the semantic 3D curves are shown (poles, road edge, vegetation-sky contour). 3D points (not shown) were used as well.

ronment, matches the observed segmentation in the image. See Fig. 3.1 for an example. The results suggest that semantics is sufficient for localization in small environments when the query image is sufficiently ”semantically interesting”, and that the proposed 6 DoF semantic pose refinement does indeed improve the pose.

Author contribution. I did most of the implementation work of the method and the evaluation on the datasets. The third author created the datasets used and contributed the original ideas. The second author contributed to the discussions and assisted greatly in writing the article.

Paper II

C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, F. Kahl. ”Semantic Match Consistency for Long-Term Visual Localization”. *Presented at the European Conference on Computer Vision 2018.*

This paper presents another semantics-based single-image localization method. Unlike Paper I, which was based on semantics alone, this article aims to increase localization performance by incorporating semantic information in a classical geometrical pose-estimation pipeline based on local image features.

Specifically, we address the problem of the high-outlier ratios which are often encountered when employing feature-based methods in the long-term localization scenario. Inspired by the geometric outlier-filtering methods, we devise a

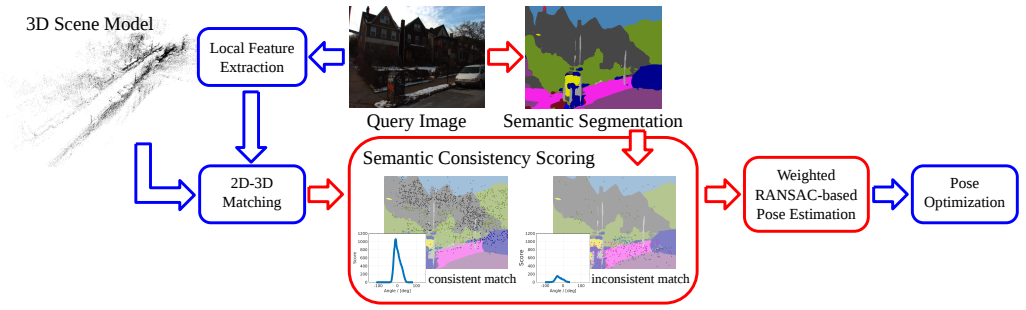


Figure 3.2: The general pipeline of the method presented in paper II. The method is based on a traditional feature-based pipeline, but each feature is scored depending on how well it agrees with the overall semantics of the scene and image. This score is then used to weight the sampling probabilities in the RANSAC loop.

semantic outlier filter, which, like the geometric outlier filters, uses prior knowledge about the camera height above the ground plane as well as the vertical direction, to reason about the likelihood of each correspondence being an inlier. The method is evaluated on two self-driving car datasets, and we show that, under the conditions mentioned, the method can significantly increase the performance of a classical P3P RANSAC based pipeline.

One of the main advantages of the proposed method is that it is fully parallel in the number of correspondences: each correspondence can be scored independently of the others, and the method of scoring correspondences depends only on projections and angle calculations. As such, it would be very suitable for an efficient GPU implementation, though in the paper only a sequential MATLAB implementation was performed.

Author contribution. The last author supplied the original idea, and I implemented the method and all evaluation scripts. The other authors helped with discussions, writing, figures and provided semantic segmentations for the used images.

Paper III

C. Toft, D. Turmukhambetov, T. Sattler, F. Kahl, and G. Brostow. "Single-Image Depth Prediction Makes Feature Matching Easier". *Presented at the European Conference on Computer Vision 2020.*

When matching local features between two images, the performance degrades with increasing viewpoint difference between the images. In this paper,

Figure 3.3: When matching features between images taken from very dissimilar viewpoints (such as the left and middle image here), we can use pixelwise monocular depth predictions (show next to the images) to render a top-down view of the images and match features in these top-down views (right).

we propose a method that utilizes the output from a monocular depth estimation network to extract features that are less sensitive to viewpoint changes.

The main idea is to use the depthmap to identify parts of the image that are planar. This is done by computing a surface normal for each pixel, and then clustering these normals in order to identify connected parts of the image sharing the same normal. The parts of the scene that are deemed to be planar can then be rectified using a homography (i.e., rendered from a top-down view), computed from the estimated surface normal (see Fig. 3.3). The clustering also tells us which parts of the image is rectified by which homography, allowing us to bypass some of the problems encountered when using vanishing point based rectification methods.

The proposed method is evaluated on a new dataset that consists of eight scenes, each with around 1500 image pairs, ranked by difficulty according to the relative viewpoint difference between the images. Each scene has been captured under different environmental conditions for a large range of viewing angles.

Author contribution. This work was performed during an internship at Niantic. The original idea came from a discussion between me, Daniyar and Gabriel, and I implemented the pipeline and ran the experiments.

Paper IV

C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl and T. Sattler ”Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions”. *Accepted for the IEEE Transactions on Pattern Analysis and Machine Intelligence.*

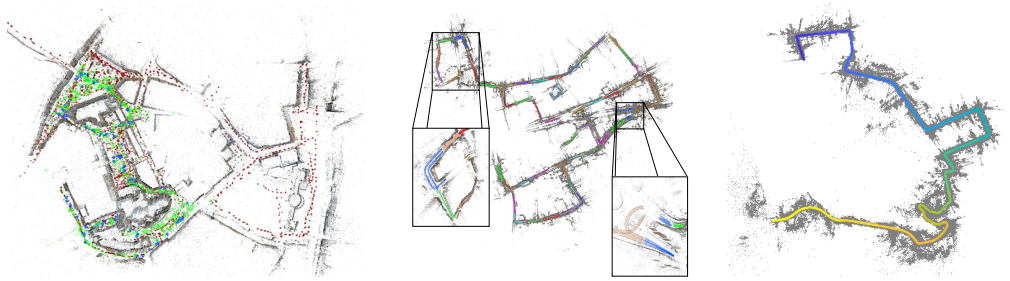


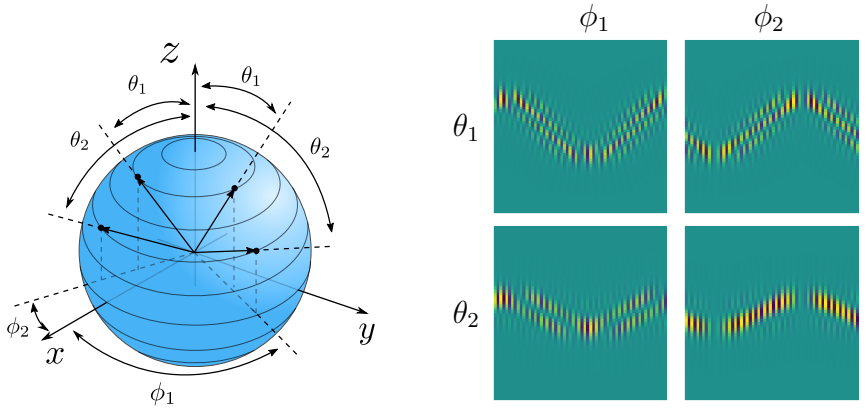
Figure 3.4: Maps of the three new datasets we present in Paper IV: the *Aachen Day-Night dataset* (left), the *RobotCar Seasons dataset* (middle) and the *Extended CMU Seasons dataset* (right).

This is a dataset and review paper, where we present three challenging new datasets for visual localization, and document the performance of current state-of-the-art methods on these datasets. This is the journal version of a conference article presented at CVPR in 2019. In this article, we address our observation that there were no public datasets suitable for evaluation of six degree-of-freedom visual localization in the long-term scenario. There were several datasets containing images taken under a variety of different conditions (such as several traversals of the same road in winter, summer, spring, day, night etc.), however these did not come with known reference poses, making them unsuitable for evaluation of long-term localization algorithms.

In this article, we augmented three of these publicly available datasets with reference poses by reconstructing each condition individually, and then registering the different models into the same coordinate system. The dataset we augmented were the CMU Visual localization dataset [8], the Aachen dataset [62] and the Oxford RobotCar dataset [52]. Fig. 3.4 shows top-down views of the reconstructions of our datasets.

The article also presents a comprehensive evaluation of the current state-of-the-art methods on these datasets, based on submissions to our evaluation server.

Author contribution. This work was a large collaboration between many researchers and research groups. At Chalmers, we were mainly responsible for the CMU Seasons dataset. Fredrik did most of the bundle adjustment and establishing ground truth poses. I triangulated the reference models, ran some of the baseline methods, and formatted and organized the raw dataset into its current, published form, and was responsible for putting together the journal article and setting up the benchmarking server and website. Torsten did most of the writing for the original CVPR article.



(a) (b)

Figure 3.5: We show that a general linear SO(2) equivariant transform from $L^2(S^2) \rightarrow L^2(S^2)$ can be interpreted as computing the correlation between the input signal f and a θ-dependent filter h_θ, and how these take the form of block-diagonal matrices in the spherical Fourier domain.

Paper V

C. Toft, G. Bökman, and F. Kahl. "Azimuthal Rotational Equivariance in Spherical CNNs". *Submitted for review to the International Conference on Learning Representations 2021.*

In this paper we consider the problem of designing neural networks for spherical data. Convolutional neural networks have achieved outstanding success in the analysis of planar data, but it is not immediately clear how to generalize these networks to operate on spherical data.

Two of the defining characteristics of a (planar) convolution are its linearity and equivariance to translations. Using these as guiding principles, one can define convolutions on the sphere equivariant to arbitrary SO(3) rotations, as done by e.g. Cohen et al., Esteves et al. and Kondor et al. [16, 24, 40]. However, several recently proposed high-performing networks on the sphere apply convolutional operators that are not fully SO(3) equivariant (for example, they may include horizontal and vertical derivatives). We note, however, that these convolutions are linear and exhibit SO(2) equivariance.

Thus, in this paper we ask how an operator from $L^2(S^2) \rightarrow L^2(S^2)$ must look in order for it to be linear and SO(2) equivariant. We show that these operators must admit a block-diagonal representation in the spherical Fourier domain if they operate on band-limited input data. Additionally, we show how these may be intepreted as a correlation with an azimuthal-dependent filter (see Fig. 3.5).

Lastly, we show how an existing state-of-the-art SO(2) equivariant pipeline

for spherical data can be recreated in our framework, showing increased invariance to azimuthal rotations of the test data.

Author contribution. I came up with the idea for the paper, and proved the result in proposition 1 regarding the spectral characterization of SO(2) equivariant linear operators from $L^2(S^2) \to L^2(S^2)$. Additionally, I ran the experiments on the MNIST dataset, while the second author ran the experiments on the ModelNet40 dataset.

Chapter 4

Conclusion and Future Outlook

In this thesis, we have presented work on the problem of robust visual localization, mainly by adressing the following points:

- We have examined how to incorporate semantic information to aid in the visual localization process.
- We have examined a possible way to utilize depth estimates to extract features less sensitive to viewpoint changes.
- We have developed new datasets, and a public benchmark, to enable accurate and fair evaluation of the performance of visual localization methods in the long-term localization scenario.
- We have worked on untangling some of the theory of spherical convolutions, which may eventually lead to learned local features for omnidirectional cameras.

The field of visual localization has seen considerable progress since I embarked on this journey over four years ago. Looking back to my earliest work on incorporating semantic information in localization, it is already somewhat outdated and has been superseded by new ideas, but I believe that work served as an important stepping stone that provided valuable insights.

For instance, I believe one of the main takeaways from Papers I and II is that it is possible to perform visual localization without relying only on the matched 3D points; the unmatched points also provide important information regarding the likelihood of a given hypothesized pose being correct. Additionally, in the first paper we saw some indications that it might even be possible to perform localization without relying on local features at all. This line of thinking was pursued in later work, with some promising results in the sequential localization scenario, where we saw that good localization results can be obtained in the long-term scenario without relying on traditional feature matching at all. Instead, only

the consistency of the projected semantic model was used. See e.g. subsidiary publications (d) and (e).

When speaking of localization performance, a question which has recurred continually throughout the work has been that of: *What localization error is acceptable?* How do we know when the localization problem has been solved? What framerate do we need to be able to handle? At the moment, I am not aware of any work which provides general answers to these questions. The localization performance of the state-of-the-art algorithms on our benchmark suggest that the localization methods are rapidly improving, but there is still room for improvement for the very challenging scenarios, such as day-night changes. But it is not quite clear what accuracy is required for e.g. autonomous driving or augmented reality, or what accuracy is realistic to expect.

In the following section, I expand on some directions for future research I believe are promising regarding the visual localization problem.

1 Future work

1.1 Learned features

One approach which has recently caught on is to learn a feature detector and descriptor. I.e, to train a neural network to output a detection heatmap and corresponding dense features, or different variations thereof. In other words, train a detector which tends to trigger on points which are long-term stable and not sensitive to viewpoint differences. This might be corners of windows, rooftop corners, street signs, and so on (or any other kind of point which the system can reliably re-identify). Popular learned features include SuperPoint [20], D2-Net [21] and R2D2 [59]. One of the problems which may arise when training this form of feature, is that ground-truth correspondences will be needed, i.e., the system needs to know which features actually are stable such that it can learn from these. We believe our new datasets may be useful to other researchers in this regard.

Another related approach which seems promising, and is related in spirit to the semantic localization approaches in Papers I and II, is to perform localization by finding the pose which maximizes the consistency between the observed dense features extracted from an image using a neural network, and the features of 3D points downprojected onto the image from a hypothesized pose. As in paper I, such a method does not rely on explicit feature matches, and uses all contents of the image and scene to reason about the camera pose.

Such an approach is used e.g. by the GN-Net method [69]. This kind of method is also fairly similar to direct methods for visual odometry, in the sense that it is essentially correspondence free, but where the alignment error is based

on feature distance rather than photometric error. This approach would also make it possible to train a feature extraction network by directly minimizing the localization error.

1.2 Incorporate 3D information during feature matching

Another approach is to incorporate 3D information into the feature extraction or matching process. This is particularly relevant if a stereo setup is used, or if a well-performing monocular depth estimation network is available. This is very similar to the approach used in Paper III, but there is still room for improvement here. It might be possible to compute the viewpoint invariant descriptor without explicit unwarping of the flat surfaces, or perhaps to encode the surface shape for non-planar surfaces into the descriptor. If accurate surface normal estimates are available, then these may also provide constraints during the feature matching process.

1.3 Improved datasets

While we have provided three new datasets that we believe are useful for the community and a step in the right direction, the story is not over here, and there is still a need for even more comprehensive datasets to properly evaluate localization methods in the long-term scenario.

In particular, none of our datasets are particularly suited for evaluating sequential localization methods. It is not reasonable or realistic to expect single-image methods to work in all scenarios, and in most practical applications, such as autonomous driving or augmented reality, the pose estimate will be based on an entire history of data. As such, we believe future reasearch should focus more towards sequential methods. To evaluate these, large-scale cross-seasonal datasets captured at high frame-rates with corresponding IMU data would be valuable. As far as we are aware, no such datasets exist, at least publically.

1.4 Omnidirectional cameras

Lastly, to wrap up Part I, it seems that omnidirectional cameras would be very useful for the task of visual localization, but very few localization methods seem to focus on, or evaluate the performance for, this kind of camera. For instance, re-localizing a car driving down the same road as before, but in the opposite direction, is very challenging with a regular camera with a limited field of view, but for an omnidirectional camera, this scenario should pose no extra difficulty.

Omnidirectional cameras also seem especially suited for this task due to their full field of view: they will not as frequently encouter the situations often encountered for regular cameras, where there simply is not anything informative

inside the current image. However, how to develop neural networks for learning local features for omnidirectional cameras is, as far as we are aware, still an open problem.

Bibliography

[1] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. "Interesting Interest Points". In: *International Journal of Computer Vision* 97.1 (Mar. 2012), pp. 18–35.

[2] Adnan Ansar and Konstantinos Daniilidis. "Linear pose estimation from points or lines". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.5 (2003), pp. 578–589.

[3] Relja Arandjelovic and Andrew Zisserman. "All about VLAD". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2013, pp. 1578–1585.

[4] R. Arandjelović et al. "NetVLAD: CNN architecture for weakly supervised place recognition". In: *CVPR*. 2016.

[5] Relja Arandjelović and Andrew Zisserman. "Visual Vocabulary with a Semantic Twist". In: *ACCV*. 2014.

[6] Anurag Arnab et al. "Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 37–52.

[7] Erik Ask, Olof Enqvist, and Fredrik Kahl. "Optimal Geometric Fitting Under the Truncated L_2-Norm". In: *Conference Computer Vision and Pattern Recognition*. 2013.

[8] H. Badino, D. Huber, and T. Kanade. "Visual Topometric Localization". In: *Intelligent Vehicles Symposium (IV)*. Baden-Baden, Germany, June 2011.

[9] H. Badino et al. "Real-Time Topometric Localization". In: *ICRA*. 2012.

[10] Hernán Badino, D Huber, and Takeo Kanade. "Visual topometric localization". In: *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE. 2011, pp. 794–799.

[11] H. Bay et al. "SURF: Speeded Up Robust Features". In: *European Conference on Computer Vision* 110.3 (2008), pp. 346–359.

[12] J-C. Bazin et al. “Globally Optimal Line Clustering and Vanishing Point Estimation in Manhattan World”. In: *Conference Computer Vision and Pattern Recognition*. 2012.

[13] Duane C. Brown. “Close-range camera calibration”. In: *Photogramm. Eng* 37.8 (1971), pp. 855–866.

[14] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2018), pp. 834–848.

[15] O. Chum et al. “Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval”. In: *International Conference on Computer Vision*. 2007.

[16] Taco S Cohen et al. “Spherical CNNs”. In: 2018.

[17] Marius Cordts et al. “The cityscapes dataset”. In: *CVPR Workshop on the Future of Datasets in Vision*. Vol. 1. 2. 2015, p. 3.

[18] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[19] N. Dalal and B. Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *CVPR*. San Diego, USA, 2005, pp. 886–893.

[20] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Superpoint: Self-supervised interest point detection and description”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 224–236.

[21] Mihai Dusmanu et al. “D2-Net: A Trainable CNN for Joint Description and Detection of Local Features”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8092–8101.

[22] O. Enqvist and F. Kahl. “Robust Optimal Pose Estimation”. In: *European Conference on Computer Vision*. 2008.

[23] Olof Enqvist, Klas Josephson, and Fredrik Kahl. “Optimal correspondences from pairwise constraints”. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 1295–1302.

[24] Carlos Esteves et al. “Learning SO(3) equivariant representations with spherical CNNs”. In: *European Conference on Computer Vision*. 2018.

[25] Wolfgang Faig. “Calibration of close-range photogrammetric systems: Mathematical formulation”. In: *Photogrammetric engineering and remote sensing* 41.12 (1975).

[26] M. Fischler and R. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* (1981).

[27] Xiao-Shan Gao et al. “Complete solution classification for the perspective-three-point problem”. In: *IEEE transactions on pattern analysis and machine intelligence* 25.8 (2003), pp. 930–943.

[28] J. A. Grunert. “Das Pothenot’sche Problem in erweiterter Gestalt; nebst Bemerkungen über seine Anwendung in der Geodäsie”. In: *Grunert Archiv der Mathematik und Physik* (1841).

[29] Christian Hane et al. “Joint 3D scene reconstruction and class segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 97–104.

[30] C. Harris and M. Stephens. “A Combined Corner and Edge Detector”. In: *Alvey Vision Conference*. 1988.

[31] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[32] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[33] E. Hecht. *Optics*. Addison-Wesley, Reading, Mass., 1987.

[34] Herve Jegou, Hedi Harzallah, and Cordelia Schmid. “A contextual dissimilarity measure for accurate and efficient image search”. In: *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE. 2007, pp. 1–8.

[35] H. Jégou, M. Douze, and C. Schmid. “On the burstiness of visual elements”. In: *CVPR*. 2009.

[36] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications, Second Edition*. Artech House mobile communications series. Artech House, 2005.

[37] Tong Ke and Stergios I Roumeliotis. “An efficient algebraic solution to the perspective-three-point problem”. In: *arXiv preprint arXiv:1701.08237* (2017).

[38] A. Kendall, M. Grimes, and R. Cipolla. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization”. In: *International Conference on Computer Vision*. 2015.

[39] L Kneip, D Scaramuzza, and R Siegwart. “A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation”. In: *CVPR*. 2011.

[40] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. “Clebsch–Gordan nets: A fully fourier space spherical convolutional neural network”. In: *Advances in Neural Information Processing Systems*. 2018.

[41] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. “Closed-form solutions to minimal absolute pose problems with known vertical direction”. In: *Asian Conference Computer Vision*. 2010.

[42] Lubor Ladický et al. “Joint optimization for object class segmentation and dense stereo reconstruction”. In: *International Journal of Computer Vision* 100.2 (2012), pp. 122–133.

[43] Måns Larsson. “End-to-end Learning of Deep Structured Models for Semantic Segmentation”. PhD thesis. Department of Signals and Systems, Chalmers University of Technology, 2018.

[44] V. Larsson et al. “Outlier rejection for absolute pose estimation with known orientation”. In: *British Machine Vision Conference*. 2016.

[45] Tony Lindeberg. “Feature detection with automatic scale selection”. In: *International journal of computer vision* 30.2 (1998), pp. 79–116.

[46] Seppo Linnainmaa, David Harwood, and Larry S Davis. “Pose determination of a three-dimensional object using triangle pairs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.5 (1988), pp. 634–647.

[47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[48] D. Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[49] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[50] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[51] Stephanie Lowry et al. “Visual place recognition: A survey”. In: *IEEE Transactions on Robotics* 32.1 (2016), pp. 1–19.

[52] Will Maddern et al. “1 Year, 1000km: The Oxford RobotCar Dataset”. In: *The International Journal of Robotics Research* 36.1 (2017), pp. 3–15. eprint: `http://ijr.sagepub.com/content/early/2016/11/28/0278364916679498.full.pdf+html`.

[53] EL Merritt. “Explicit three-point resection in space”. In: *Photogrammetric Engineering* 15.4 (1949), pp. 649–655.

[54] Marius Muja and David G. Lowe. “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *Int. Conf. on Computer Vision Theory and Applications*. 2009.

[55] T. Naseer et al. “Semantics-aware Visual Localization under Challenging Perceptual Conditions”. In: *ICRA*. 2017.

[56] D. Nistér and H. Stewénius. “Scalable Recognition with a Vocabulary Tree”. In: *CVPR*. Vol. II. New York City, USA, 2006, pp. 2161–2168.

[57] J. Philbin et al. “Object Retrieval with Large Vocabularies and Fast Spatial Matching”. In: *Conference Computer Vision and Pattern Recognition*. 2007.

[58] L. Quan and Z. Lan. “Linear $N \leqslant 4$-Point Camera Pose Determination”. In: *PAMI* 21.8 (Aug. 1999), pp. 774–780.

[59] Jerome Revaud et al. “R2D2: Reliable and Repeatable Detector and Descriptor”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 12405–12415.

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[61] Torsten Sattler et al. “Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?” In: *CVPR*. 2017.

[62] Torsten Sattler et al. “Image Retrieval for Image-Based Localization Revisited”. In: *British Machine Vision Conference*. 2012.

[63] G. Schindler, M. Brown, and R. Szeliski. “City-Scale Location Recognition”. In: *CVPR*. 2007.

[64] Lukas Schneider et al. “Semantic stixels: Depth is not enough”. In: *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE. 2016, pp. 110–117.

[65] Johannes L. Schönberger and Jan-Michael Frahm. “Structure-From-Motion Revisited”. In: *CVPR*. 2016.

[66] Markus Schreiber, Carsten Knöppel, and Uwe Franke. “LaneLoc: Lane marking based localization using highly accurate maps”. In: *IV*. 2013.

[67] N. Snavely, S.M Seitz, and R. Szeliski. “Photo tourism: Exploring photo collections in 3D”. In: *ACM SIGGRAPH* (2006).

[68] Robert Spangenberg, Daniel Goehring, and Raúl Rojas. “Pole-based localization for autonomous vehicles in urban scenarios”. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE. 2016, pp. 2161–2166.

[69] Lukas von Stumberg et al. “Gn-net: The gauss-newton loss for multi-weather relocalization”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 890–897.

[70] L. Svärm et al. “City-scale localization for cameras with known vertical direction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.7 (2017), pp. 1455–1461.

[71] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, 2010.

[72] A. Torii et al. “24/7 Place Recognition by View Synthesis”. In: *CVPR*. 2015.

[73] A. Torii et al. “Visual Place Recognition with Repetitive Structures”. In: *CVPR*. 2013.

[74] Bill Triggs. “Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes”. In: *European Conference on Computer Vision*. Vol. 4. Prague, Czech, 2004, pp. 100–113.

[75] Tinne Tuytelaars and Krystian Mikolajczyk. “Local invariant feature detectors: a survey”. In: *Foundations and Trends in Computer Graphics and Vision* 3.3 (2008), pp. 177–280.

[76] Danfei Xu, Hernán Badino, and Daniel Huber. “Topometric localization on a road network”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 3448–3455.

[77] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *ICLR*. 2016.

[78] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. “Camera Pose Voting for Large-Scale Image-Based Localization”. In: *ICCV*. 2015.

[79] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22 (2000).

[80] Hengshuang Zhao et al. “Pyramid Scene Parsing Network”. In: *CVPR*. 2017.