Computer Vision and Image Understanding 186 (2019) 58–73

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Performance analysis of single-query 6-DoF camera pose estimation in self-driving setups☆

Junsheng Fu [a,*], Said Pertuz [a,b], Jiri Matas [c], Joni-Kristian Kämäräinen [a]

[a] Tampere University - Hervannan Campus, Department of Signal Processing, P.O. Box 553, FI-33101 Tampere, Finland
[b] Universidad Industrial de Santander, 680003 Bucaramanga, Colombia
[c] Czech Technical University in Prague, Faculty of Electrical Engineering, Technicka 2, 16627 Praha 6, Czech Republic

ARTICLE INFO

Communicated by: P Mordohai

Keywords:
Camera pose estimation
3D point cloud
Hybrid method
Photometric matching
Mutual information
Self driving car

ABSTRACT

In this work, we consider the problem of single-query 6-DoF camera pose estimation, i.e. estimating the position and orientation of a camera by using reference images and a point cloud. We perform a systematic comparison of three state-of-the-art strategies for 6-DoF camera pose estimation: feature-based, photometric-based and mutual-information-based approaches. Two standard datasets with self-driving setups are used for experiments, and the performance of the studied methods is evaluated in terms of success rate, translation error and maximum orientation error. Building on the analysis of the results, we evaluate a hybrid approach that combines feature-based and mutual-information-based pose estimation methods to benefit from their complementary properties for pose estimation. Experiments show that (1) in cases with large appearance change between query and reference, the hybrid approach outperforms feature-based and mutual-information-based approaches by an average increment of 9.4% and 8.7% in the success rate, respectively; (2) in cases where query and reference images are captured at similar imaging conditions, the hybrid approach performs similarly as the feature-based approach, but outperforms both photometric-based and mutual-information-based approaches with a clear margin; (3) the feature-based approach is consistently more accurate than mutual-information-based and photometric-based approaches when at least 4 consistent matching points are found between the query and reference images.

1. Introduction

Camera pose estimation is a fundamental technology for various applications, such as augmented reality (Taylor, 2016), virtual reality (Ohta and Tamura, 2014), and robotic localization (Castellanos and Tardos, 2012). The aim of 6 degrees of freedom (DoF) camera pose estimation is to find the 3-DoF location and 3-DoF orientation of the query image in a given reference coordinate system. In the literature, the classical approach for 6-DoF camera pose estimation is to register a 2D query image with previously acquired reference data, which often consist of a set of reference images and corresponding 3D point clouds. In practice, this is a fundamental yet challenging problem due to large displacements between the query and reference images, as well as image variations caused by changes in the appearance of the scenes, weather and lighting conditions (Maddern et al., 2017; Mishkin et al., 2015). Depending on how the 6-DoF pose estimation problem is solved, state-of-the-art methods can be divided into 2 main categories: *direct* and *indirect* approaches. In our scope, *direct* approach means that the 6-DoF camera pose is directly optimized by a cost function defined over the 6D pose space. For example, the 6-DoF camera pose can be computed by directly minimizing a cost function that compares the query image with a rendered synthetic view from a 3D point cloud (Pascoe et al., 2017; Tykkälä et al., 2013; Newcombe et al., 2011a,b).

In the *indirect* approach, the query image is registered to the 3D point cloud by matching point features extracted in the query image and the reference images (Mishkin et al., 2015; Song et al., 2016; Irschara et al., 2009; Kim et al., 2014), and the reference images and the 3D point cloud are defined in the same world coordinate system. Both *direct* and *indirect* approaches have shown good performance in previous works with different datasets and experimental settings (Pascoe et al., 2017; Mishkin et al., 2015; Song et al., 2016). However, the relative performance of the *direct* and *indirect* approaches have not been studied in the same working conditions with large-scale, realistic datasets.

Although both the *indirect* and *direct* approaches have been widely utilized for 6-DoF pose estimation, we have identified two important

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cviu.2019.04.009.

* Corresponding author.
E-mail address: junsheng.fu@tut.fi (J. Fu).

https://doi.org/10.1016/j.cviu.2019.04.009
Received 22 May 2018; Received in revised form 20 March 2019; Accepted 17 April 2019
Available online 25 April 2019
1077-3142/

questions that warrant further research: first, there is no consensus in the community about which strategies yield the best performance in real-life conditions, where the appearance of the reference and query images change significantly according to different weather, lighting and season conditions. Second, in the literature, pose estimation strategies are often assessed as a part of full pipelines that involve additional pre- or post-processing steps, *e.g.* the incorporation of information from previous poses in sequential data or global optimization strategies in simultaneous localization and mapping approaches. As a result, the contribution of pose estimation methods on the overall performance of the system, as well as their response to different imaging factors, remains unclear. In order to tackle the aforementioned problems, we implemented and studied three state-of-the-art camera pose estimation approaches for the estimation of 6-DoF camera pose of a single query image using reference images and a point cloud. Specifically, the three implementations are one *indirect* approach, a feature-based method in Kim et al. (2014), and two *direct* approaches: a photometric-based method (Tykkälä et al., 2013) and a mutual-information-based method (Pascoe et al., 2017). The motivation for studying the selected methods is that they are state-of-the-art, have good speed performance and can be conveniently implemented and tested in the same conditions (Pascoe et al., 2017; Tykkälä et al., 2013; Kim et al., 2014).

We perform a systematic and extensive experimental comparison of the studied approaches and analyze their performances. Based on the obtained results, we evaluate a hybrid approach that combines the feature-based and mutual information-based camera pose estimation methods, and present an architecture for computing the 6-DoF camera pose from rough 2-DoF spatial position estimates. As the **main contribution** of this work, we perform an extensive comparison and analysis of three strategies for 6-DoF camera pose estimation: a feature-based approach, a photometric-based approach, and a mutual-information-based approach. We find that the feature-based approach is more accurate than photometric-based and mutual-information-based approaches with as few as 4 consistent feature points between the query and reference images. However, the mutual-information-based approach is often more robust and can provide a pose estimate when the feature-based approach fails. We experimentally demonstrate that a hybrid approach, which combines the feature- and mutual-information-based approaches, outperforms both. All source code for camera pose estimation methods and their performance evaluation will be made publicly available.[1]

In addition, we study the performance of the hybrid approach with an architecture that allows computing camera pose with multiple reference images and allows to naturally integrate and refine pose priors in large uncertainty cases. For the experiments, we used two publicly available datasets: the KITTI dataset (Geiger et al., 2012) and Oxford RobotCar dataset (Maddern et al., 2017). The KITTI dataset provides 11 individual sequences with ground truth trajectories. The recently released Oxford RobotCar dataset (Maddern et al., 2017) contains many repetitions on the same route. RobotCar dataset provides different combinations of weather, traffic and pedestrians, with long-term changes such as construction and roadworks, which allows a more challenging evaluation in realistic conditions. Our comparison shows how the hybrid approach outperforms feature-based, photometric-based or mutual-information-based approaches. Furthermore, the experiments show that using multiple reference images improves the robustness of all pose estimation pipelines.

1.1. Related work

Camera pose estimation using vision has received significant attention in recent decades. We focus on the case of registering a single query image with one or several reference images and 3D point clouds. The approaches can be divided into 2 main categories: *indirect* approaches (Irschara et al., 2009; Kim et al., 2014; Klein and Murray, 2007; Geiger et al., 2011; Kitt et al., 2010) and *direct* approaches (Pascoe et al., 2017; Tykkälä et al., 2013; Newcombe et al., 2011a). It is important to notice that many of the above works introduce a Simultaneous Localization and Mapping (SLAM) method. Specifically, camera pose estimation discussed in this paper is only one component utilized within more complex SLAM methods. In our discussion we refer only to the camera pose estimation part of them.

The *indirect* approaches establish 2D-3D correspondences between the query image and the 3D point cloud. The reference images and the 3D point cloud are pre-registered, so the 2D-3D correspondences are indirectly obtained by establishing 2D-2D correspondences between the query image and the reference images. Specifically, the query image is registered with the reference images by utilizing feature detectors for finding salient image structures for localization, e.g. corners (Rosten and Drummond, 2006; Mikolajczyk and Schmid, 2004), blobs (Lowe, 1999; Bay et al., 2006; Kadir and Brady, 2001) or regions (Matas et al., 2004; Tuytelaars and Van Gool, 2000, 2004; Mori et al., 2004). Then feature descriptors (Calonder et al., 2010; Rublee et al., 2011; Leutenegger et al., 2011; Alahi et al., 2012; Lowe, 1999; Bay et al., 2006; Dalal and Triggs, 2005; Tola et al., 2010; Ambai and Yoshida, 2011) are used to provide a robust representation regardless of appearance changes due to different viewpoints, weather, lighting, etc. Given the set of 2D-3D correspondences, a Perspective-n-Point solver (Torr and Zisserman, 2000; Gao et al., 2003) and RANSAC (Fischler and Bolles, 1981; Torr and Zisserman, 2000) are applied to compute the relative 6-DoF camera pose between the query image and the reference 3D point cloud. Because different combinations of 2D-3D correspondences lead to different camera pose estimations, the *indirect* approach can be considered as a combinatorial optimization method. A few of the popular *indirect* methods are described as follows: PTAM (Klein and Murray, 2007) is a widely used featured-based monocular SLAM algorithm that allows robust state estimation in real-time. LIBVISO1 (Kitt et al., 2010) is a feature-based 6 DoF camera pose estimation method for a stereo camera, and it is extended into LIBVISO2 (Geiger et al., 2011) which supports monocular ego-motion estimation. Besides, 3D scene representation either from LIDAR or Structure-from-Motion pipelines can be utilized to estimate the camera pose. One work (Irschara et al., 2009) registers on-line images to a sparse 3D scene generated by Structure-from-Motion pipelines. Another work (Kim et al., 2014) estimates camera pose by using LIDAR point cloud and reference images.

The *direct* approaches compute the 6-DoF camera pose by minimizing a cost function directly in the 6D space of camera poses (Pascoe et al., 2017; Tykkälä et al., 2013; Newcombe et al., 2011a,b; Engel et al., 2014, 2018), and do not need to extract local features of images. One commonly used cost function is the photometric error between the query image and the reference view, where the reference view is generated from the reference 3D point cloud (Tykkälä et al., 2013; Newcombe et al., 2011a,b). The *direct* photometric-based methods usually have good speed performance. For example, LSD-SLAM (Engel et al., 2014) is a monocular SLAM which allows to build large-scale maps of the environment and runs in real-time on a CPU. The recent DSO (Engel et al., 2018) combines a fully direct probabilistic model with joint optimization of all model parameters and it can be achieved in real-time by omitting the smoothness prior used in other direct methods and instead sampling pixels evenly throughout the images. However, they are arguably less robust to real-world global illumination changes (Newcombe et al., 2011b). A recent work (Pascoe et al., 2017) utilizes a mutual-information-based cost function for *direct* 6-DoF camera pose estimation outperforming both the feature-based and photometric-based approaches in two challenging datasets with large image variations. This mutual-information-based approach has been tailored for the SLAM problem and it relies on a well-initialized reference image (Pascoe et al., 2017). However, it is still unclear what the performance of the mutual-information-based approach would be

[1] https://github.com/JunshengFu/camera-pose-estimation.

without accounting for the initialization problem, where single query image is to be registered with no prior on the pose.

Besides the *direct* and *indirect* approaches, a semi-direct visual odometry pipeline, the SVO, has been proposed by Forster et al. (2014). In SVO, feature-correspondences are an implicit result of direct motion estimation rather than of explicit feature extraction and matching. Thus, feature extraction is only required when the initial key frame is selected to initialize the construction of a new 3D point cloud. The advantage of this approach is its increased speed due to the lack of feature-extraction at every frame and increased accuracy through sub-pixel feature correspondence. After the feature correspondences and an initial estimate of the camera pose are established, the algorithm continues using only point-features. In this work, we are interested in the solution of the single-query pose estimation problem. The SVO approach is designed for solving the pose estimation problem in the context of multiple, sequential frames and has therefore not been considered in this work.

A recent work (Delmerico and Scaramuzza, 2018) compares visual-inertial odometry algorithms on different hardware configurations, but their focus is on monocular visual-inertial odometry methods. Another benchmark (Li et al., 2016) provides detailed performance analysis of open source visual SLAM pipelines on different datasets. However, their work is focused on comparing the performance of the whole visual SLAM pipeline instead of a single step such as the pose estimation. To the best of our knowledge, there is a lack of prior art comparing the stand alone performance of *direct* and *indirect* camera pose estimation approaches in this scenario.

1.2. Overview

Based on our literature review, we selected and implemented three state-of-the-art 6-DoF pose estimation methods: (1) an *indirect* feature-based method (Kim et al., 2014), (2) a *direct* photometric-based method (Tykkälä et al., 2013) and (3) a *direct* mutual-information-based method (Pascoe et al., 2017). We choose these 3 approaches because they provide good performance and can be adapted for the same experimental settings. The details of these methods are presented in Section 2. In order to conduct a rigorous and systematic analysis of their practical performance, the studied methods were compared in three different scenarios: the single-reference case, the multi-reference case and the large uncertainty case. Each one of the experimental setups for these 3 scenarios are described in Section 3. Experiments and results on real datasets are presented in Section 4. Based on the experimental results, we also evaluate a *hybrid approach* that combines *direct* and *indirect* methods for an improved performance. The conclusions and the implementation details of this work are presented in Section 5 and Appendices, respectively.

2. Evaluated pose estimation methods

The methods selected for comparison in this work are representative examples of *direct* and *indirect* approaches with state-of-the-art performance. In this section, we describe each one of the methods in the simplest scenario, where the inputs are a query image I_Q, and a single *reference tuple* (I_R, P_R) that is formed by a reference image I_R and its registered 3D point cloud P_R (see Fig. 1). The aim is to find the 6D pose of the query image I_Q.

2.1. Indirect feature-based (FB) pose estimation

Standard feature-based pose estimation can be divided into four main steps: (1) feature detection, (2) feature matching, (3) grouping of 2D-3D correspondences, and (4) Perspective-n-Point pose estimation. The block diagram of the feature-based (FB) method is shown in Fig. 2. In the first step, a feature detector and a feature descriptor are applied to both query and reference images to detect points – or regions – of interest and compute descriptors from pixels surrounding each point of interest. Secondly, based on the previously computed descriptors, 2D-2D point correspondences are sought between query and reference images by means of feature matching. Thirdly, since the 3D point cloud is registered with the reference image, the 2D-3D correspondences between the query image and the 3D point cloud can be established indirectly through the 2D-2D correspondences between points of interest in the query and reference images. Finally, a Perspective-n-Point solver (Gao et al., 2003) and RANSAC (Fischler and Bolles, 1981; Torr and Zisserman, 2000) are applied for computing the 6-DoF camera pose from these 2D-3D correspondences. The algorithm and implementation details of each stage of the feature-based pose estimation can be found in Appendix A.

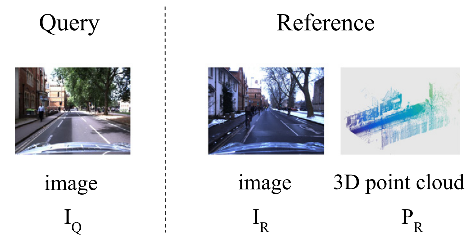


Fig. 1. Inputs for the pose estimation methods in the simplest scenario: a query image I_Q and a *reference tuple* (I_R, P_R), where I_R is a single reference image and P_R is the registered 3D point cloud associated to I_R. Both the point cloud P_R and the camera pose of the reference image I_R are defined in a common world coordinate system. The aim is to estimate the 6D pose of the query image I_Q.

2.2. Direct photometric-based (PB) pose estimation

The *direct* photometric-based approach (Tykkälä et al., 2013) is defined as a direct minimization of a cost function defined over the 6D space of camera poses. The pixel intensities of the query image and a rendered synthetic view from the 3D point cloud are directly compared in the cost function (Tykkälä et al., 2013). The photometric-based approach can be divided into three main steps: (1) synthetic image generation, (2) photometric matching, and (3) coarse-to-fine search.

The block diagram of this method is shown in Fig. 3. In summary the algorithm works as follows: firstly, for rendering purposes, a *colored 3D point cloud* is generated by projecting each 3D point of the cloud P_R to the reference image frame and then assigning the colors from the reference image at that location. Subsequently, we generate a synthetic image I_S by projecting the colored 3D point cloud into an image plane (see Appendix B.1), where the transformation matrix $\mathbf{M}$ of the reference image is used as the initial pose estimate. The goal is to find the transformation matrix that minimizes the photometric error between the synthetic view and the query image:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} RES(I_Q, I_S), \tag{1}$$

where $RES(\cdot,\cdot)$ is the residual function used to compute the photometric error.

In this work, we solve (1) by means of a coarse-to-fine grid search (see Appendix B.3). It should be noted that in common tracking applications where the transformation baseline is small, fast optimization can be implemented by using Jacobian and gradient-based optimization (Tykkälä et al., 2013). However, in the case of big appearance differences between the query and reference images, gradient-based optimization often fails to find global solutions, so we adopted a grid search in our experiments. A more detailed description of the photometric pose estimation method with implementation details can be found in Appendix B.

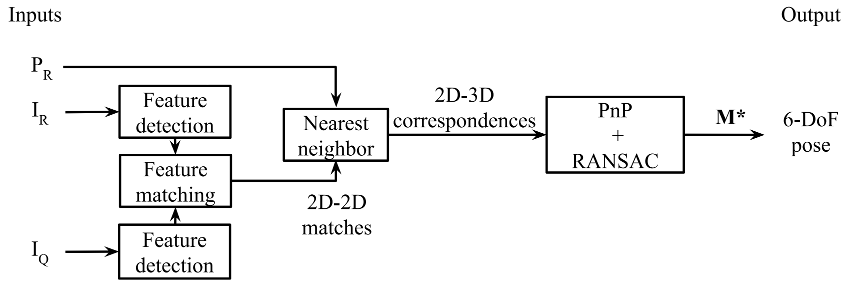


Fig. 2. Block diagram of feature-based camera pose estimation. I_Q is the query image. The reference image I_R and the 3D point cloud P_R are pre-registered and defined in the world coordinate system. $\mathbf{M}^*$ is the estimated transformation matrix. For the detailed descriptions of each step see Appendix A.

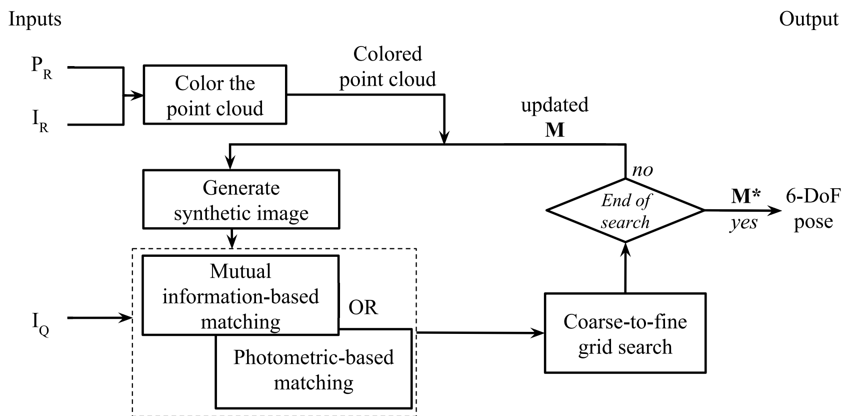


Fig. 3. Block diagram of direct photometric-based and mutual-information-based camera pose estimation. I_Q is the query image. The reference image I_R and the 3D point cloud P_R are pre-registered and defined in the world coordinate system. $\mathbf{M}^*$ is the estimated transformation matrix. For the detailed descriptions of each step see Appendix B.

2.3. Direct mutual-information-based (MI) pose estimation

The *direct* mutual-information-based approach is similar to the photometric-based approach presented in previous section with the main difference being that, in the cost function (1), the *normalized mutual information* (NMI) is used instead of the photometric error. Specifically, the mutual information-based pose estimation problem is formulated as the minimization problem:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} 1 - NMI(I_Q, I_S), \tag{2}$$

where $\mathbf{M}^*$ is the estimated camera pose, I_Q is the query image, I_S is the synthetic image; and the Normalized Mutual Information (NMI) is computed as (McDaid et al., 2011):

$$NMI(I_S, I_Q) = \frac{MI(I_S, I_Q)}{max(H(I_S), H(I_Q))} \tag{3}$$

with

$$MI(I_S, I_Q) = H(I_S) + H(I_Q) - H(I_S, I_Q)\ , \tag{4}$$

where $H(I_S, I_Q)$ is the joint entropy of I_S and I_Q, $H(I_S)$ and $H(I_Q)$ are the marginal entropies of I_S and I_Q, and $MI(I_S, I_Q)$ is the mutual information between I_S and I_Q.

2.4. Hybrid (HY) pose estimation

In our experiments, we also evaluate a combination of indirect and direct approaches for pose estimation. This approach is inspired by the strong empirical evidence in our experiments showing that: (1) the feature-based method is superior in accuracy if a sufficient number of matches can be found (see details in Sections 4.3 and 4.5); (2) the mutual-information-based approach can still provide a reasonable estimate in cases where the feature-based method fails to generate an estimate (no enough matched features found between the reference and query images). Therefore, our hybrid approach first executes the feature-based method and, if it fails to compute at least 4 consistent matching points between the query and reference images, then it switches to the MI-based method.

Specifically, given one query image I_Q and one *reference tuple* (I_R, P_R), a feature detector is firstly applied to both the query image I_Q and the reference image I_R, and then we apply feature matching to obtain 2D-2D matched features. Since the point cloud P_R is registered with the reference image I_Q, the 2D-3D correspondences can be found indirectly. Then a PnP solver (Gao et al., 2003) and RANSAC (Torr and Zisserman, 2000) are applied to the 2D-3D correspondences. For the PnP solver (Gao et al., 2003), at least 4 consistent 2D-3D correspondence pairs are required. If the camera pose of the query image cannot be estimated due to less than four 2D-3D correspondences (Torr and Zisserman, 2000; Gao et al., 2003), the *direct* mutual-information-based pose estimation is used to compute the camera pose. The block diagram of the hybrid approach is shown in Fig. 4.

3. Comparative methodology

In this work, we systematically compare camera pose estimation approaches in three scenarios: firstly, we compare the performance of different pose estimation methods for single query images in the simplest scenario of using only one *reference tuple*, as shown in Fig. 1. Secondly, we increase the number of reference images and evaluate the improvement in accuracy. Thirdly, we evaluate the different approaches with large spatial uncertainties, where the reference images can be far away from the query image. The three scenarios considered for comparison are described in more detail below.

3.1. Single-reference pose estimation

The aim of using a single reference image for different pose estimation methods is to compare their performance at the most basic

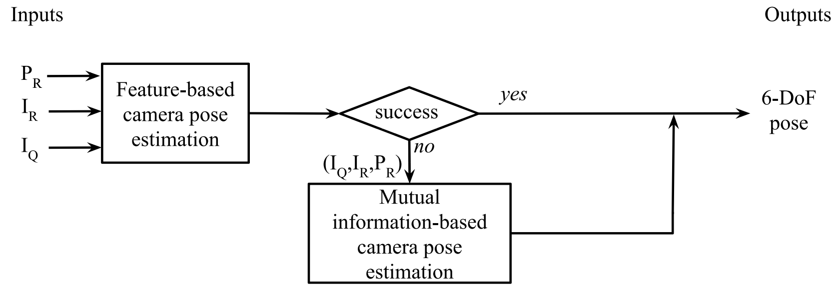


Fig. 4. Block diagram of the hybrid approach for camera pose estimation.

Fig. 5. Single-reference pose estimation. The actual location of the query image is marked with a purple dot, and a circle around the purple dot represents the initial uncertainty on the location of the query image. Within the uncertainty circle, one reference image is randomly selected among all possible candidates that are indicated with red markers from A to L. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

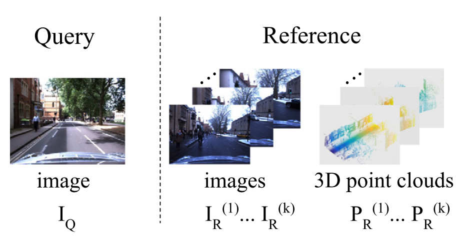


Fig. 6. Example of inputs for multi-reference case: one query image I_Q and multiple *reference tuples* $\{(I_R^{(1)}, P_R{}^{(1)}), \dots (I_R^{(k)}, P_R{}^{(k)})\}$ which consist of k reference images and k 3D point clouds. All the reference images and 3D point clouds are defined in an unified coordinate system.

level without pre- or post-processing steps. As illustrated in Fig. 5, the experiment starts by first defining a radius r around the actual location of the query image. The radius r represents the uncertainty in the location of the query image. The reference image is randomly selected in the region within the circle. The motivation of random selection is to evaluate how the studied algorithms respond to different overlaps between query and reference images. Increasing the radius reduces the potential overlap between query and reference images, which makes pose estimation more challenging. For *direct* methods, this can be considered as different initialization. After randomly selecting one reference image within the radius, the inputs of the single-reference case are the query image I_Q and a *reference tuple* (I_R, P_R), where I_R is a single reference image and P_R is its corresponding 3D point cloud. The quality of the estimated pose is then assessed in the terms of translation error and rotation error (see Section 4.2).

3.2. Multiple-reference pose estimation

In this section we explain the case of incorporating the information obtained from multiple reference images to estimate the camera pose of a single query image. In this case, the inputs are one query image and multiple *reference tuples* which consist of k pairs of reference images and their corresponding 3D point clouds, $\{(I_R^{(1)}, P_R{}^{(1)}), \dots (I_R^{(k)}, P_R{}^{(k)})\}$, as shown in Fig. 6. All the reference images and 3D point clouds are defined in an unified coordinate system. The aim of using multiple reference images is to leverage the additional information to improve accuracy of camera pose estimation.

In the prior art, Song et al. (2016) fuse multiple camera poses by: (1) averaging three rotation angles to compute the final rotation matrix; (2) minimizing a geometrical error term to estimate the final translation. However, 3D point clouds are not utilized in their approach, so from each reference image only a line where the camera pose of the query image should lie is obtained. In contrast, in our approach, each reference image together with a 3D point cloud are already sufficient to compute a unique 6-DoF camera pose for the query image. Therefore, we have considered 4 strategies, which can be easily adapted to different camera pose estimation methods.

1. Maximum number of matched features (*maxf*): we match the query image with all the available reference images, and select the reference image with the largest number of matched features after the feature matching stage. Then, we compute the camera pose of the query image with only the *reference tuple* that contains the selected reference image. The remaining processing steps are the same as in the camera pose estimation with a single *reference tuple*.
2. Simple average (*avg*): for each *reference tuple* in $\{(I_R^{(1)}, P_R{}^{(1)}), \dots (I_R^{(k)}, P_R{}^{(k)})\}$, we compute an individual candidate camera pose $\mathbf{M}_{(i)} = [\mathbf{R}_{(i)} \;|\; \mathbf{t}_{(i)}]$ where $\mathbf{R}_{(i)}$ and $\mathbf{t}_{(i)}$ are the rotation matrix and translation vector of the ith camera pose, and $i \in \{1, \dots, k\}$. As a result, k candidate camera poses will be obtained. Each 6-DoF camera pose consists of a rotation matrix and a translation vector. We average the k rotation matrices by firstly converting them to quaternions and then apply quaternion space interpolation (Markley et al., 2007). As a result, the final rotation matrix is obtained from the averaged quaternion representation, and the final translation vector can be computed by averaging all the translation vectors.
3. Weighted average (*wavg*): similar to *simple average*, this approach starts with k individual candidate pose estimates $\mathbf{M}_{(i)} = [\mathbf{R}_{(i)} \;|\; \mathbf{t}_{(i)}]$ obtained from each *reference tuple*. Then we take a weighted average of these k camera poses, and the weights $\mathbf{w}_{(i)}$ are computed according to the number of matched features between the query image and each reference image. The calculation of the final pose can be formulated as follows:

$$\mathbf{M}^* = \sum_i \mathbf{w}_{(i)} \mathbf{M}_{(i)}, \quad i \in \{1, 2 \dots, k\} \tag{5}$$

where the rotation matrix $\mathbf{R}_{(i)}$ in $\mathbf{M}_{(i)}$ is converted to quaternions and then we compute a quaternion-weighted average (Markley et al., 2007). Each weight value is computed as follows,

$$\mathbf{w}_{(i)} = \frac{m_{(i)}}{\sum m_{(i)}}, \quad i \in \{1, 2 \dots, k\} \tag{6}$$

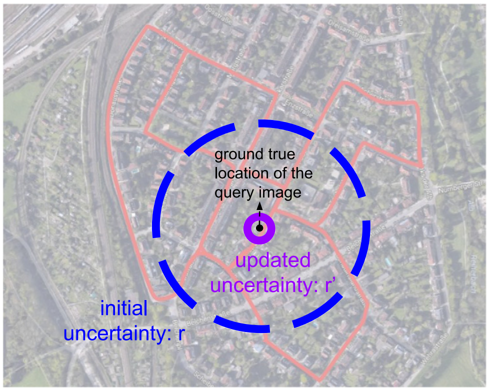


Fig. 7. Camera pose estimation with a large uncertainty. An image retrieval method is combined with a camera pose estimation method to reduce the large position uncertainty of the query image. The black dot represents the actual location of the query image, the big blue dashed circle shows the initial uncertainty and the small purple solid circle indicates the updated uncertainty in pose estimation. The red route marked in the background is one of the routes in the KITTI dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $m_{(i)}$ is the number of the matched features between the query image I_Q and the ith reference image $I_R^{(i)}$.

4. Robust weighted average (*r-wavg*): firstly we match the query image with all the available reference images and record the numbers of their matches. If the maximum number of matched features between the query image and reference images is K, we select those reference images with at least half of the maximum matches $K/2$. The weights for individual candidate camera poses are computed as follows:

$$w(i) = \begin{cases} 0, & \text{if } m_{(i)} < \frac{K}{2} \\ \frac{m_{(i)}}{\sum m_{(i)}}, & \text{if } m_{(i)} \geq \frac{K}{2} \end{cases} \tag{7}$$

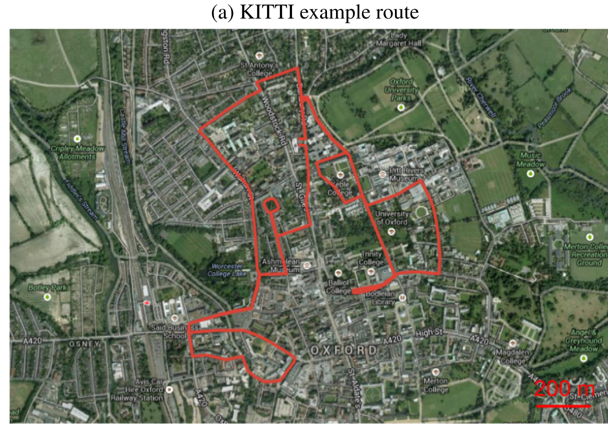
where K is the maximum number of matched features and it can be formulated as $K = \max\{m_{(i)}\}, i \in \{1, 2 \ldots, k\}$. In the end, we apply obtained weights to Eq. (5) to get the final camera pose.

3.3. Camera pose estimation with large uncertainties

In real-life applications, the query image may or may not have a GPS tag, and even with a GPS tag, the precision of the GPS can be poor (Linegar et al., 2016; Miura et al., 2015). Therefore, the initial uncertainty radius r of the query camera's location can be large (see Fig. 7). In the case of large uncertainties, choosing the reference image by random selection is not practical anymore, and the use of an image retrieval method becomes beneficial. Therefore, we compare the performance of the studied pose estimation methods with a large uncertainty, and evaluate how image retrieval improves their performance.

In image retrieval, methods such as Song et al. (2016), Philbin et al. (2007), Radenović et al. (2016) and Iscen et al. (2017) are used to effectively identify a few good reference images from a large reference database. In this work, we select the retrieval method (Philbin et al., 2007) which performs image retrieval from a large image set by quantizing low-level image features based on randomized trees and using an efficient spatial verification stage to re-rank the results returned from a bag-of-words model. We take up to 5 reference images with the highest scores from the retrieved ones, and then we perform single query camera pose estimation with multiple reference images.

(a) KITTI example route



(b) Oxford RobotCar route

Fig. 8. Sample routes for KITTI and Oxford RobotCar dataset with scales.

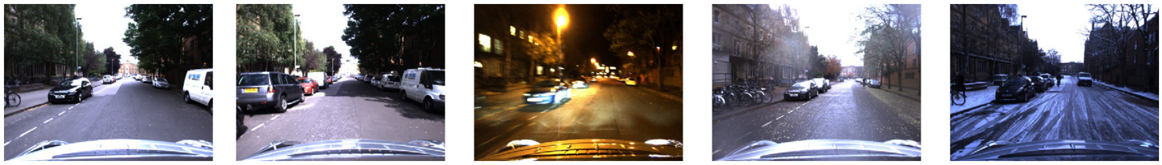
4. Experiments and results

4.1. Datasets

In this work, experiments were conducted using two public datasets: the KITTI Visual Odometry dataset (Geiger et al., 2012) and the Oxford RobotCar dataset (Maddern et al., 2017). The KITTI dataset was captured by driving around the mid-size city of Karlsruhe (Germany), in rural areas and on highways. The accurate ground truth is provided by a Velodyne laser scanner and a GPS localization system. There are 11 sequences in the KITTI dataset with ground-truth camera poses available, and we use all of them in our experiments. These sequences are summarized in Table 1. For each sequence, a 3D point cloud P_R is obtained from LIDAR, and both query image I_Q and reference image I_R are from one monochrome camera (according to the author the monochrome camera is less noisy). For illustration, one example route from the KITTI dataset is shown in Fig. 8a.

The recently released Oxford RobotCar dataset (Maddern et al., 2017) provides multiple traversals of the same route and allows a more challenging evaluation in changing weather and daylight conditions. 5 sequences of the Oxford RobotCar dataset with completely different environment conditions were selected for our experiments. The sequence route is shown in Fig. 8b and example images from 5 sequences are shown in Fig. 9. Similarly to the KITTI dataset, 3D point clouds are obtained from LIDAR. The reported GPS information is treated as the ground-truth for the camera location.

The Oxford RobotCar dataset includes images captured by a Bumblebee XB3 (1280 × 960 × 3, 16 Hz). In our experiment, we use the left image from the Bumblebee XB3 and, for efficiency, we reduced the number of images in each sequence by taking 1 out of every 10 images. Also we removed the beginning and ending frames of each sequence



(a) overcast (b) sun (c) night (d) rain (e) snow

Fig. 9. Appearance differences among the 5 sequences in the Oxford RobotCar dataset (the images are roughly from the same location).

Table 1
Overview of the 11 sequences in the KITTI dataset (Geiger et al., 2012).

Id	# images	Tag	Total length (km)	Mean distance between consecutive images (m)
00	4541	Urban	3.7	0.8
01	1101	Highway	2.5	2.2
02	4661	Urban	5.1	1.1
03	801	Urban	0.6	0.7
04	271	Urban	0.4	1.5
05	2761	Urban	2.2	0.8
06	1101	Urban	1.2	1.1
07	1101	Urban	0.7	0.6
08	4071	Urban	3.2	0.8
09	1591	Urban	1.7	1.1
10	1201	Urban	0.9	0.8

Table 2
Overview of 5 sequences with different environmental conditions in Oxford RobotCar dataset (Maddern et al., 2017).

Id	# images	Tag	Total length (km)	Mean distance between consequent images (m)
00	1916	Overcast	6.3	3.3
01	2873	Sun	8.6	3.0
02	2931	Night	9.1	3.1
03	2614	Rain	8.8	3.4
04	3019	Snow	8.7	2.9

where the car is usually parked, producing multiple instances of the same image. The resulting 5 sequences from Oxford RobotCar dataset are summarized in Table 2. For the Oxford RobotCar dataset, the query I_Q and reference I_R images are taken from different traversals of the route, and therefore give a much more demanding assessment of pose estimation performance in realistic conditions.

There are two main reasons why we used these specific datasets. One is the availability of ground truth from commercial-level Inertial and GPS navigation system. For example, KITTI dataset uses OXTS RT 3003 (Oxford-Technical-Solutions-Ltd, 2019), and Oxford RobotCar dataset uses NovAtel SPAN-CPT ALIGN (NovAtel-Inc., 2019). This type of ground truth information is very limited in other existing datasets. The other reason is that Oxford RobotCar dataset consist of the multiple traversals of the same route under changing weather and daylight conditions. However, since the both datasets are acquired by sensors on a car the main application field of our results is self-driving cars. This indicates certain limitations in the images, such as the small variation in viewpoint between consecutive frames.

4.2. Performance measures

We use translation error, maximum orientation error and the success rate of each method to compare the performance of the different approaches:

1. The translation error is the absolute translation between the ground-truth location and the estimated location of the query image.
2. Based on the rotation matrix between the ground-truth camera pose and the estimated camera pose of the query image, we convert the rotation matrix into 3 Euler angles. Then the maximum absolute Euler angle is used as the maximum orientation error.
3. The studied methods can fail to yield a camera pose estimate under some circumstances, for instance when there are not enough feature matches between the query and reference images in the *indirect* approach, or when grid search fails to converge in *direct* approaches. In this work, we define the *success rate* as the percentage of the processed images for which the estimated poses are within 10 m from ground truth, and this threshold is picked from the prior art (Pascoe et al., 2017).

4.3. Experiments with single reference image

In this section, we perform 12 sets of experiments for both KITTI and Oxford RobotCar datasets. Each set of experiments comprises hundreds of estimates for a pose estimation method at an uncertainty radius. The goal of these experiments was to compare the performance of different pose estimation methods under the single reference scenario, as described in Section 3.1. For the experiments, the uncertainty radius r was varied between 10 to 25 m. Since most of the photos are taken by a front-looking camera mounted in a car in the streets of an urban area, these search ranges were selected so that the reference and query images would have some overlap but not being too close to each other. The mean distance between two consequent images are from 0.7 to 3.4 m in the two evaluated datasets.

The experiments with the KITTI dataset tested the performance of different camera pose estimation methods under "ideal conditions", *i.e.* same time of the day, lighting and weather condition. For the KITTI dataset, all the 11 sequences listed in Table 1 have different routes. For this reason, each sequence was processed individually so that the query image and the reference images come from the same drive. In order to separate the query and reference images, we randomly selected 10% of the images in one sequence for queries, and the rest of images from the same sequence were used as references.

The experiments with the Oxford RobotCar dataset tested the performance of camera pose estimation methods in challenging conditions since the query and reference data capture large variation in appearance and structure of a dynamic city environment over long periods of time. For the Oxford RobotCar dataset presented in Table 2, each one of the 5 route traversals corresponds to different environmental conditions on the same route. The sequences were processed jointly in order to allow the query and reference images to come from the different sequences. For example, when the summer sunny sequence (01 in Table 2) was used for the reference images, the winter snow sequence (04 in Table 2) was used for the queries.

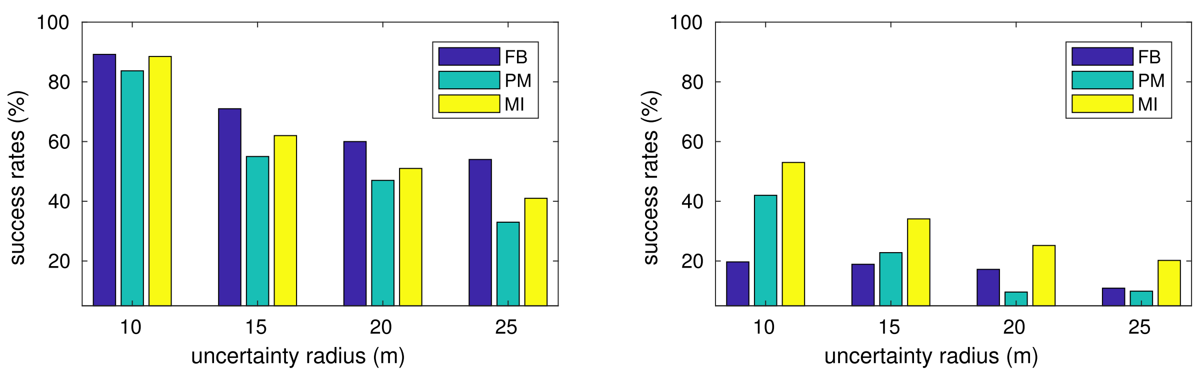
Table 3 summarizes the translation and orientation errors for the studied methods (FB, PB and MI) in the single-reference scenario. For a fair comparison of the performance measures, we decided to use only those images for which all methods are able to provide a pose estimate (regardless of accuracy). From Table 3a and b we observe the following:

1. By looking into each column, we find that as long as the feature-based approach is able to estimate the camera pose, its estimates have smaller translation errors than the other two methods in

Table 3
Translation error (in meters) and **maximum orientation error** (in degrees) using a single reference image. For the KITTI dataset, 454 images (random 10% of the whole sequence) in sequence 00 are used as queries, and the rest as the reference images. For the Oxford RobotCar dataset, summer sequence (01) is used as the reference and 302 images (random 10%) from the winter sequence (04) are used as the query images. The second row shows the number of images for which all methods are able to provide a pose estimate regardless of accuracy. The third row shows the percentage value. All the translation and orientation results are reported in median values.

(a) KITTI sequence: translation error (m)					(b) Oxford sequence: translation error (m)				
Uncertainty radius (m)	10	15	20	25	Uncertainty radius (m)	10	15	20	25
#images	406	328	282	259	#images	67	60	53	38
	(89%)	(72%)	(62%)	(57%)		(22%)	(20%)	(18%)	(13%)
FB (Kim et al., 2014)	**0.13**	**0.40**	**0.48**	**0.30**	FB (Kim et al., 2014)	**2.77**	**2.48**	**2.40**	**2.91**
PM (Tykkälä et al., 2013)	1.44*	6.66*	7.77*	14.85*	PM (Tykkälä et al., 2013)	10.44*	16.23*	20.09*	26.32*
MI (Pascoe et al., 2017)	1.56*	5.41*	6.15*	10.26*	MI (Pascoe et al., 2017)	8.71*	13.36*	16.27*	14.94*
(c) KITTI sequence: max orientation error (degree)					**(d) Oxford sequence: max orientation error (degree)**				
Uncertainty radius (m)	10	15	20	25	Uncertainty radius (m)	10	15	20	25
#images	406	328	282	259	#images	67	60	53	38
	(89%)	(72%)	(62%)	(57%)		(22%)	(20%)	(18%)	(13%)
FB (Kim et al., 2014)	1.76	3.83	5.42	3.33	FB (Kim et al., 2014)	**3.44**	**3.79**	2.72	3.25
PM (Tykkälä et al., 2013)	**1.07***	2.40*	**3.37***	3.12*	PM (Tykkälä et al., 2013)	3.48*	5.82	2.64	**1.88**
MI (Pascoe et al., 2017)	**1.07***	**2.30***	3.45*	**2.70***	MI (Pascoe et al., 2017)	6.16	4.00	**2.42**	1.93*

*Indicates a statistically significant difference at the $p < 0.05$ level computed with the Wilcoxon signed rank test (Gibbons and Chakraborti, 2011) against the Feature-based method (FB).



(a) KITTI (b) Oxford RobotCar

Fig. 10. Success rate comparison for three strategies with single reference image at different uncertainty ranges in two public datasets. (a): in the experiments with the KITTI sequence 00, a random 10% of the images are used as query image and the rest are used as references. (b): in the experiments with two sequences in Oxford RobotCar sequences, summer sequence (01) was used for the references and the snow sequence (04) was used for queries. Failure threshold was set to 10 m.

both the KITTI and Oxford RobotCar datasets. This result indicates that the feature-based approach is more accurate in pose estimation in both ideal environment conditions (KITTI dataset) and realistic environment conditions (Oxford RobotCar dataset) with random reference image selection.

2. By looking into each row, we find that the translation errors of both photometric-based and mutual-information-based approach increase with the increasing **uncertainty radius**, but the translation error of the feature-based approach does not vary much. This suggests that both the photometric and mutual-information-based approaches are more sensitive to the initialization.

Table 3c and d compare the orientation errors. Among the studied methods, the differences in their orientation errors are small. In other words, all these methods perform similarly in terms of orientation error for both KITTI and Oxford RobotCar datasets. The reason for these results might be that all the images are taken by a front-looking camera mounted on a car driving along the street, so the query images and the reference images may share similar viewpoints. Fig. 10 plots the *success rates* (see definition in Section 4.2) for the studied three strategies with a single *reference* at different uncertainty ranges. Fig. 10 shows the following:

1. The feature-based approach has higher success rate than the other two approaches in the KITTI dataset; however, the feature-based approach has the lowest success rate among all three approaches in the Oxford RobotCar dataset. The mutual-information-based approach has the highest success rate in Oxford RobotCar dataset. This suggests that the success rate of the feature-based approach is greatly influenced by the environmental conditions between the query and reference images. On the other hand, the mutual-information-based approach is the most robust in terms of the success rate under different environmental conditions.
2. When analyzing the same pose estimation method for different uncertainty radii, the *success rates* of all approaches decrease with the increase of the uncertainty radius.

Pascoe et al. (2017) claim that the mutual-information-based SLAM approach has higher success rate than state-of-the-art feature-based SLAM approaches (Mur-Artal et al., 2015). Our experiments in Fig. 10b lead to the same conclusion in the problem of 6-DoF camera pose estimation using single reference image and 3D point cloud. Interestingly enough, our experiments in Table 3 suggest that the feature-based approach can be more accurate as long as it is able to compute the camera pose.

The observations presented above lead us to use the hybrid (HY) method for pose estimation. Recall however that, for the results presented in Table 3, we selected images for which all the methods yield a pose estimate. As a result, the performance of the hybrid method (HY) in this setting is equivalent to the feature-based method (FB) since the photometric-based branch of the HY approach works only when the FB method fails. For this reason, we only include the HY approach in the large uncertainty scenario presented in Section 4.5.

4.4. Experiments with multiple reference images

In this experiment, we evaluated the performance of different methods in the multi-reference setting for the both KITTI and Oxford RobotCar datasets. The goal was to evaluate efficient ways to incorporate the information obtained from multiple reference images to improve the camera pose estimation.

Similarly to the single reference case of previous section, we consider the reference images within the uncertainty radius r around the actual location of the query image, and then randomly selected multiple *reference tuples*. Subsequently, we evaluated the 4 different methods to fuse camera poses from multiple *reference tuples*: maximum number of matched features (*maxf*), simple average (*avg*), weighted average (*wavg*) and the robust weighted average (*r-wavg*). The number of reference images was varied from one to five.

The results for different multi-reference pose estimation methods in the KITTI dataset are shown in Table 4. Fig. 11 compares the *success rates* for different camera pose estimation methods with multiple reference images using the *robust weighted average (r-wavg)* method in the both KITTI and Oxford RobotCar datasets. The *r-wavg* method was used in that figure since it yielded the best overall performance for all the pose estimation methods.

Table 4 summarizes the results for the experiments with multiple reference images. The results show that fusing the poses from multiple references improves the performance of the camera pose estimation results, and *robust weighted average* (*r-wavg*) outperforms the other approaches, especially with the increased number of reference images. Fig. 11 compares the *success rates* of the different approaches with multiple reference images using *robust weighted average* method in the both KITTI and Oxford RobotCar datasets. Fig. 11 tells us two things:

1. The success rates of each method show that the *success rate* increases with the increase of the number of reference images.
2. The three bars at each plot show that the feature-based approach has the highest success rate among different approaches in the KITTI dataset, but has the lowest success rate in the Oxford RobotCar dataset. In contrast, the mutual-information-based approach has the highest success rate in that dataset. In other words, mutual information is more robust than the two other approaches under changing environmental conditions. This finding is consistent with our results in the single reference scenario.

In the literature, camera pose estimation usually requires geometry verification (Sattler et al., 2016) which is very effective but requires extra computation. Interestingly enough, our results show that the *robust weighted average* method is a light approach and can be easily adapted with any pose estimation method with good results.

4.5. Experiments at large uncertainty

Based on the empirical results in Section 4.3, we evaluated a hybrid approach that leverages the advantages of both the feature-based and the mutual-information-based approaches. In this section, we tested these 4 camera pose estimation methods (feature-based, photometric-based, mutual-information-based, and hybrid approaches) with five reference images, under the large uncertainty condition.

In Section 3.3, we described the experimental setting for camera pose estimation under large location uncertainty. In the extreme case, no prior information on the location is available and the query image must be matched to the whole reference database. As a result, an image retrieval method is applied to find suitable reference images (Philbin et al., 2007). Among all retrieved reference images, up to 5 images with the highest scores are stored for further processing. In our experiments we restricted the uncertainty radius to 200 m for the KITTI and 50 m for the Oxford RobotCar dataset, and adopted the multi-references (up to 5 most similar reference images) pose-estimation approach to improve robustness of all the investigated methods. We conducted experiments in all the sequences of the KITTI dataset. In the Oxford RobotCar dataset, we performed a set of experiments where one sequence is used for the references and another sequence is used for the queries.

The results for the KITTI and Oxford RobotCar datasets are shown in Tables 5 and 6 respectively. In this two tables, we tag a pose estimate as a *failure* when the translation error is above 10 m. By looking at the *success rates* in Table 5, we see that the hybrid and feature-based approaches outperform other methods in cases where the query and reference images have been captured at similar imaging conditions (KITTI dataset). The hybrid approach performs similarly as the feature-based approach, which indicates that the evaluated hybrid method can retain good properties of the feature-based method. For the sequence 01, the hybrid method is superior. The plausible explanation for this is that the sequence 01 is captured from a highway (see Table 1) where there are less reliable features to be found than in urban scenes. In urban scenes, the hybrid and feature-based methods provide practically the same accuracy. Table 6 shows a confusion matrix summarizing the results in the Oxford RobotCar dataset. For that table, we repeated the experiments by using one sequence as reference and another one as query (a total of 5×5 different combinations). Therefore, in addition to a large spatial displacement, query and reference images have been acquired at very different imaging conditions. From that table, we conclude the following:

1. The mutual-information-based approach is more robust than the feature-based or photometric-based approaches, which is consistent with the findings in the single reference and multi-reference scenarios.
2. The hybrid approach outperforms all other approaches in success rate when the query and reference images have very different imaging conditions. This confirms that the hybrid method leverages complementary properties of the feature-based and mutual-information-based methods.

The results on the diagonal of Table 6 are consistent with previous experiments in the KITTI dataset in Table 5, i.e. in the ideal case when query and reference images come from the same sequence and imaging conditions. In this case, feature-based and our hybrid method outperform the other approaches. A remarkable result in Table 6 is that, even in the worst case scenario, the lowest success rate of the hybrid method is 13.2%. Recent results in the same dataset in similar conditions have reported *success rates* as low as 0% using SLAM (Pascoe et al., 2017). Notice that the experimental settings in that work (Pascoe et al., 2017) are different from ours, but this helps understanding the difficulty of pose estimation problem under real conditions.

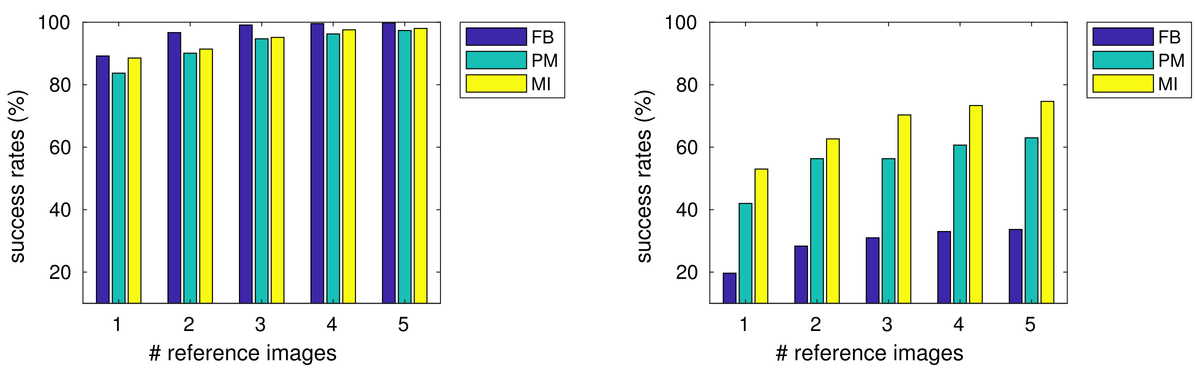
5. Conclusion

We performed systematic and extensive comparisons of three different strategies for 6-DoF camera pose estimation using reference images and 3D point clouds: an *indirect* feature-based approach, a *direct* photometric-based approach and a *direct* mutual-information-based approach. In our experiments the feature-based approach was more accurate than both the photometric-based and mutual-information-based approaches when as few as 4 consistent correspondent points were found between query and reference images. The mutual-information-based approach was more robust than the feature-based and photometric-based approaches which means that it can provide an estimate even in the cases when the other methods fail. As expected, the robustness and accuracy of all methods improved when multiple reference images were available. In the multi-reference scenario, the *robust weighted average* method outperformed other fusing methods for the estimation of the pose from multiple candidates. Based on the strong

Table 4
Performance in multi-reference pose estimation in the KITTI sequence 00. 10% images (454) from this sequence are used as query image and the rest are used as references. The uncertainty radius is $r = 10$ m. The reported results are computed from those images for which all methods are able to provide a pose estimate.

#reference images	1		2		3		4		5	
	Median (m)	Median (deg)	Median (m)	Median (deg)	Median (m)	Median (deg)	Median (m)	Median (deg)	Median (m)	Median (deg)
Feature-based (FB)										
avg	0.13	1.76	0.22*	2.07*	0.25*	2.20*	0.21*	1.80*	0.19*	1.61*
wavg	0.13	1.76	0.15*	**1.67***	0.15*	1.78	0.10*	1.22*	0.09*	1.11*
maxf	0.13	1.76	**0.11**	1.82	**0.09**	1.79*	0.06	1.21*	0.05*	1.03*
r-wavg	**0.13**	**1.76**	0.12	1.70	0.10	**1.59**	**0.06**	**1.13**	**0.04**	**0.93**
Photometric (PM)										
vg	1.44	1.07	2.29*	1.26*	2.15*	1.38*	2.08*	1.22*	1.90*	1.05*
avg	1.44	1.07	1.67*	1.00*	1.52	1.07*	1.28*	0.79*	1.12*	0.69*
axf	1.44	1.07	**1.34**	1.01	1.22	1.01*	1.19*	0.72*	1.07	0.66*
r-wavg	**1.44**	**1.07**	1.35	**0.95**	**1.21**	**0.86**	**1.12**	**0.68**	**0.99**	**0.58**
Mutual Information (MI)										
avg	1.56	1.07	1.65	1.24*	1.80*	1.43*	1.68*	1.23*	1.60*	1.12*
wavg	1.56	1.07	1.44*	1.10	1.37	1.20	1.16	0.77*	1.17	0.77*
maxf	1.56	1.07	**1.36***	1.07	1.29*	1.02*	1.16*	0.79*	1.12*	0.68*
r-wavg	**1.56**	**1.07**	1.38	**0.98**	**1.25**	**0.94**	**1.09**	**0.68**	**1.03**	**0.62**

*Indicates a statistically significant difference at the $p < 0.05$ level computed with the Wilcoxon signed rank test (Gibbons and Chakraborti, 2011) against the robust weighted average method (r-wavg).



(a) KITTI (b) Oxford RobotCar

Fig. 11. **Success rates** comparison for the studied methods with multiple reference images and *robust weighted average* method in two datasets. The failure threshold was set to 10 m.

empirical results and inspired by the complementary properties of the feature-based and mutual-information-based approaches, we evaluated a computationally cheap and easy-to-adapt hybrid approach that combines these two methods. In all experiments, the hybrid method was on par or superior to the single methods. This is particularly so in challenging scenarios such as the Oxford RobotCar dataset, where the hybrid approach outperforms feature-based and mutual-information-based approaches by an average increase in success rate of 9.4% and 8.7%, respectively.

In our experiments with multiple reference images (Section 4.4), we tested different fusion methods to compute the camera pose. The speed of the photometric and mutual-information methods could be greatly improved by utilizing GPU or thread programming. Based on the experimental results, we empirically fixed the number of the reference images to be 5 in the large uncertainty case. An interesting question to be addressed in the future work is to investigate the optimal number of images needed to achieve a certain accuracy and to compare different image retrieval approaches.

Acknowledgement

This work is supported by Academy of Finland under Grant No. 314180 and Czech Science Foundation under Grant GA18-05360S.

Appendix A. Indirect feature-based pose estimation

This appendix presents the detailed description of the four stages of the *indirect* feature-based pose estimation method presented in Section 2.1.

A.1. Feature detection and description

The first step of the system is to detect and extract features of salient locations in the query and reference images. Specifically, a feature detector is used for finding the salient points of an image, and a feature descriptor is used to describe the neighborhood surrounding that salient point.

Feature detectors can extract different types of image structures, e.g. corners (Rosten and Drummond, 2006; Mikolajczyk and Schmid, 2004), blobs (Lowe, 1999; Bay et al., 2006; Kadir and Brady, 2001) or regions (Matas et al., 2004; Tuytelaars and Van Gool, 2000, 2004; Mori et al., 2004). For reference purposes, a summary of invariance properties and performance analysis for some feature detectors are shown in Table A.7. In turn, feature descriptors can be divided into following categories: local binary descriptors (Ojala et al., 2002; Guo et al., 2010; Zhao and Pietikainen, 2007; Froba and Ernst, 2004; Calonder et al., 2010; Rublee et al., 2011; Leutenegger et al., 2011; Alahi et al., 2012), spectral descriptors (Lowe, 1999; Lienhart and Maydt, 2002; Bay et al., 2006; Dalal and Triggs, 2005; Tola et al., 2010; Ambai and Yoshida, 2011), basis space descriptors (Zahn and Roskies, 1972; Csurka et al., 2004), polygon shape descriptors (Matas et al., 2004; Belongie et al., 2001), 3D and volumetric descriptors (Klaser et al., 2008; Scovanner et al., 2007). In the literature, many feature

Table 5
Camera pose estimation results in a large uncertainty for all 11 KITTI sequences. The uncertainty radius was set to be 200 m and 5 best retrieved reference images were used for camera pose estimation. The failure threshold was set to 10 m. Note that the first three evaluated methods (Tykkälä et al., 2013; Pascoe et al., 2017) were originally designed for visual SLAM, but we modified the algorithms for the pose estimation problem.

#sequence ID	00			01			02			03		
	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Me-dian(deg)	%	Median (m)	Median (deg)
FB (Kim et al., 2014)	**99.8**	0.031	0.676	84.5	0.494	0.567	**99.8**	0.025	0.415	**100**	0.015	0.370
PM (Tykkälä et al., 2013)	98.2	0.603	0.423	76.4	1.208	0.343	92.9	0.550	0.324	98.8	0.342	0.279
MI (Pascoe et al., 2017)	97.8	0.633	0.415	60.0	0.980	0.353	97.6	0.475	0.327	98.8	0.270	0.223
HY (Combine FB and MI)	**99.8**	0.031	0.676	**89.1**	0.505	0.562	**99.8**	0.025	0.415	**100**	0.015	0.370
#sequence ID	04			05			06			07		
	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)
FB (Kim et al., 2014)	**100**	0.028	0.132	**100**	0.022	0.472	**100**	0.029	0.421	**100**	0.018	0.326
PM (Tykkälä et al., 2013)	96.3	0.783	0.222	97.8	0.514	0.360	98.2	0.382	0.308	97.3	0.505	0.336
MI (Pascoe et al., 2017)	**100**	0.495	0.177	97.1	0.537	0.352	96.4	0.551	0.332	98.2	0.500	0.319
HY (Combine FB and MI)	**100**	0.028	0.132	**100**	0.022	0.472	**100**	0.029	0.421	**100**	0.018	0.326
#sequence ID	08			09			10					
	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)			
FB (Kim et al., 2014)	**100**	0.018	0.383	99.4	0.019	0.356	**100**	0.019	0.420			
PM (Tykkälä et al., 2013)	97.3	0.499	0.329	95.0	0.548	0.321	94.2	0.634	0.355			
MI (Pascoe et al., 2017)	95.3	0.518	0.341	93.7	0.400	0.350	91.7	0.780	0.343			
HY (Combine FB and MI)	**100**	0.018	0.383	**100**	0.019	0.368	**100**	0.019	0.420			

Table 6
Large uncertainty pose estimation results for the 5 different sequences in Oxford RobotCar dataset. The uncertainty radius was set to 50 m and 5 best retrieved reference images are used for camera pose estimation. The failure threshold was set to 10 m. Note that the first three evaluated methods (Tykkälä et al., 2013; Pascoe et al., 2017) were originally designed for visual SLAM, but we modified the algorithms for the pose estimation problem.

		Overcast			Sun			Night			Rain			Snow		
		%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)	%	Median (m)	Median (deg)
Overcast	FB (Kim et al., 2014)	98.4	0.111	0.791	31.1	3.280	3.015	5.6	6.281	4.043	27.1	3.355	4.106	16.7	1.407	6.941
	PM (Tykkälä et al., 2013)	98.4	1.599	0.578	6.2	7.751	6.006	7.3	7.347	10.397	6.7	6.534	3.319	5.3	7.718	9.171
	MI (Pascoe et al., 2017)	99.0	1.551	0.716	16.3	4.648	3.991	15.5	7.479	8.625	12.0	7.363	11.716	18.2	6.902	6.556
	HY (Combine FB and MI)	**100.0**	0.112	0.788	**40.1**	3.398	3.103	**21.0**	6.966	4.486	**35.1**	3.828	5.029	**31.1**	4.251	6.687
Sun	FB (Kim et al., 2014)	33.3	2.604	1.993	98.3	0.121	0.706	2.9	4.642	9.733	12.1	2.275	3.963	15.2	2.877	3.527
	PM (Tykkälä et al., 2013)	10.7	6.242	2.135	96.2	1.919	0.585	9.2	7.412	13.885	8.2	6.968	9.364	5.0	7.080	4.104
	MI (Pascoe et al., 2017)	16.7	5.102	3.722	96.2	1.685	0.569	17.6	5.859	8.779	11.7	7.470	6.825	15.6	7.793	4.937
	HY (Combine FB and MI)	**40.0**	3.010	2.342	**99.7**	0.122	0.715	**20.1**	5.350	8.722	**21.8**	5.324	4.890	**26.2**	4.514	3.712
Night	FB (Kim et al., 2014)	4.9	3.332	2.647	1.8	5.337	7.200	89.4	0.217	0.744	1.2	2.879	4.171	2.3	5.788	8.191
	PM (Tykkälä et al., 2013)	5.9	8.255	12.956	3.2	8.437	3.251	90.8	2.303	0.543	4.3	8.725	8.275	2.3	6.973	4.625
	MI (Pascoe et al., 2017)	8.8	7.189	8.960	11.9	6.732	9.110	94.5	2.126	0.554	12.0	7.776	6.299	13.7	8.199	6.209
	HY (Combine FB and MI)	**13.7**	5.983	3.966	**13.3**	6.424	7.745	**96.9**	0.233	0.811	**13.2**	6.996	5.593	**15.0**	7.257	7.406
Rain	FB (Kim et al., 2014)	31.6	3.251	2.536	13.2	2.264	4.631	3.7	2.006	1.504	96.9	0.192	0.764	17.2	3.289	3.619
	PM (Tykkälä et al., 2013)	13.5	6.959	2.313	13.6	6.913	3.210	9.2	7.050	6.144	96.2	2.336	0.578	9.3	6.436	4.910
	MI (Pascoe et al., 2017)	9.4	6.847	9.182	13.9	6.631	5.906	11.7	7.731	9.125	95.0	1.915	1.067	17.5	6.135	5.652
	HY (Combine FB and MI)	**37.4**	3.295	2.908	**24.4**	3.724	5.779	**14.7**	6.447	7.147	**99.2**	0.200	0.773	**30.8**	4.286	4.577
Snow	FB (Kim et al., 2014)	9.9	2.521	3.625	10.1	2.310	7.945	2.2	5.733	13.984	10.8	2.260	4.811	97.7	0.145	0.834
	PM (Tykkälä et al., 2013)	5.4	7.192	33.412	2.4	8.353	20.601	4.8	7.529	5.721	3.6	5.899	14.798	95.7	2.026	0.553
	MI (Pascoe et al., 2017)	12.6	8.000	7.760	12.5	7.073	4.269	13.6	7.559	8.608	8.0	6.417	9.510	95.4	2.012	0.734
	HY (Combine FB and MI)	**19.8**	4.534	4.230	**20.9**	5.343	5.731	**15.0**	6.993	8.608	**18.4**	3.727	7.399	**100.0**	0.149	0.804

descriptors, such as SURF (Bay et al., 2006), BRISK (Leutenegger et al., 2011) and others, provide their own detector method along with the descriptor method. DoG (Lowe, 1999) and SURF (Bay et al., 2006) detectors were designed for efficiency and the other properties are slightly compromised. However, for most applications they are still more than sufficient (Tuytelaars et al., 2008).

A summary of the invariance properties of the detectors is in Table A.7. In two public datasets used in this work we use images taken by a front-looking camera mounted in the car, so those images have similar viewpoints which is along the road. In this work we have utilized SURF (Bay et al., 2006) for both feature detection and description due to its invariance properties, performance, and widespread use in

Table A.7
Invariance properties of feature detectors (Tuytelaars et al., 2008).

F-detector	Invariance		
	Rotation	Scale	Affine
Harris	✓		
Hessian	✓		
SUSAN	✓		
Harris–Laplace	✓	✓	
Hessian–Laplace	✓	✓	
DoG	✓	✓	
Salient regions	✓	✓	✓
SIFT	✓	✓	
MSER	✓	✓	✓
SURF	✓	✓	

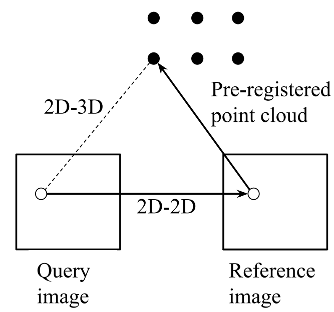


Fig. A.12. Build 2D-3D correspondences through the 2D-2D matched features and the pre-registered point cloud.

multiple applications. Another reason is that our evaluated *indirect* method (Kim et al., 2014) also uses SURF (Bay et al., 2006), and we would like to implement it in the same way.

A.2. Feature matching

Based on the previously computed feature descriptors, the aim of feature matching is finding 2D-to −2D correspondences between feature points in the query and reference image.

The popular approaches for feature matching are *exhaustive search*, *hashing* (Strecha et al., 2012), and *nearest neighbor techniques* (Friedman et al., 1977; Lowe, 2004; Muja and Lowe, 2009). *Exhaustive search* is achieved by minimizing pairwise distance measures between the feature vectors of the reference and query image. The *hashing* approach reduces the size of the descriptors by finding a more compact representation, e.g. binary strings (Strecha et al., 2012). In *nearest neighbor techniques*, KD-trees (Friedman et al., 1977) and their variants (Lowe, 2004; Muja and Lowe, 2009) are commonly used to quickly find approximate nearest neighbors in a relatively low-dimensional real-valued space. The algorithm works by recursively partitioning the set of training instances based on a median value of a chosen attribute (Friedman et al., 1977).

We use the exhaustive search approach and adopt a minimum Euclidean distance on the descriptor vector. For each feature point in one image, we find the nearest neighbor as its corresponding feature point in the other image. Besides, we reject some ambiguous matches by comparing the distance of the closest neighbor to that of the second-closest neighbor. In other words, correct matches need to have the closest neighbor significantly closer than the second closest match to achieve reliable matching (Lowe, 2004). The output of the feature matching steps are a set C of n 2D-to −2D correspondences between the query image I_Q and reference image I_R:

$$C = \{(\mathbf{p}_Q^{(1)}, \mathbf{p}_R^{(1)}), (\mathbf{p}_Q^{(2)}, \mathbf{p}_R^{(2)}), \dots, (\mathbf{p}_Q^{(n)}, \mathbf{p}_R^{(n)})\} \tag{A.1}$$

where $\mathbf{p}_Q^{(i)} = [u_Q^{(i)}, v_Q^{(i)}]^T$ and $\mathbf{p}_R^{(i)} = [u_R^{(i)}, v_R^{(i)}]^T$ are the ith 2D feature locations on reference and query images, respectively.

A.3. 2D-3D correspondences

The 2D-3D correspondences between the query image and the 3D point cloud are established by using the set C of 2D-2D matches and the point cloud P_R. Since the point cloud P_R and the reference image I_R are pre-registered and defined in the same world coordinate system, with the 2D-2D matched features, we could indirectly link the 2D-3D correspondences as illustrated in Fig. A.12.

However, if the matched 2D features at the reference image do not have associated 3D points from the pre-registered point cloud, we need to compute the 2D-3D correspondences by following steps: (1) project 3D point cloud onto the reference image, (2) compute the depth of the feature points, (3) find the corresponding 3D coordinates.

Firstly, we project the 3D point cloud $P_R = [\mathbf{P}_R^{(1)}, \mathbf{P}_R^{(2)}, \dots, \mathbf{P}_R^{(m)}]$ onto the reference image plane, and get a set of 2D projections $p = [\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(m)}]$, as shown in Fig. A.13. For the ith 3D point, $\mathbf{P}_R^{(i)} = [x^{(i)}, y^{(i)}, z^{(i)}, 1]^T$, we generate a 2D projection $\mathbf{p}^{(i)} = [u^{(i)}, v^{(i)}]^T$ on the reference image plane by:

$$\mathbf{p}^{(i)} = \mathbf{K} \quad \mathbf{M} \quad \mathbf{P}_R^{(i)} \tag{A.2}$$

where $\mathbf{M}$ is the world to camera transformation matrix and $\mathbf{K}$ is the intrinsic matrix of the reference image. $\mathbf{M}$ and $\mathbf{K}$ can be represented by (A.3) and (A.4):

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \tag{A.3}$$

where $\mathbf{R}$ is a 3×3 rotation matrix, and $\mathbf{t}$ is a 3×1 translation vector.

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

where f_x and f_y are focal lengths (in pixels) along the x and y axis directions; γ represents the skew coefficient between x and y axis and it is often 0; u_0 and v_0 represents the principal point which would ideally be in the center of the image. In the experiments of this paper, we assume the query image and the reference images share the camera intrinsic matrix, because the images from each dataset are captured with the same camera device.

Secondly, we use nearest-neighbor search (Friedman et al., 1977) to find the closest point among 2D projections p for each 2D feature point in C at the reference image. In particular, the jth feature point $\mathbf{p}_R^{(j)}$ in the reference image is associated to the kth point of the 2D projection set p by:

$$k = NN(\mathbf{p}_R^{(j)}, p), \quad k \in \{1, 2 \dots, m\} \tag{A.5}$$

Finally, we find the 3D coordinates for each 2D feature point. In particular, the kth depth value corresponding to $\mathbf{p}^{(k)}$, namely $z^{(k)}$, is then used to find the 3D coordinates in the reference image frame corresponding to $\mathbf{p}_R^{(j)}$ as:

$$\mathbf{P}^{(j)} = \begin{bmatrix} \mathbf{K}^{-1} \mathbf{p}_R^{(j)} z^{(k)} \\ z^{(k)} \end{bmatrix} \tag{A.6}$$

As a result, the final 2D-to −3D correspondences can be expressed as:

$$\hat{C} = \{(\mathbf{p}_Q^{(1)}, \mathbf{P}^{(1)}), (\mathbf{p}_Q^{(2)}, \mathbf{P}^{(2)}) \dots, (\mathbf{p}_Q^{(n)}, \mathbf{P}^{(n)})\} \tag{A.7}$$

where $\mathbf{p}_Q^{(i)}$ is the ith 2D feature location in the query image, and $\mathbf{P}^{(i)}$ is the ith corresponding 3D location in the reference image coordinate.

A.4. Perspective-n-point and RANSAC

The set of 2D-3D correspondences $\hat{C}$ establishes one-to-one correspondences between 2D points in the query image frame $\mathbf{p}_Q^{(j)}$, and 3D

(a) reference image (b) 3D point cloud (c) projections

Fig. A.13. An example of projecting the 3D point cloud into the reference image.

points in the reference image frame $\mathbf{P}^{(j)}$, for $j = 1, 2, \dots, n$. The last step is to apply the Perspective-n-Point solver (Gao et al., 2003) to compute the relative 6-DoF camera pose $\mathbf{M}$ between the query image and the reference image. For this purpose, two approaches are combined to solve the problem: the algebraic approach and the geometric approach. In the algebraic approach, we use Wu's zero decomposition method (Wen-Tsun, 1986) to find a complete triangular decomposition of a practical configuration for the P3P problem (Gao et al., 2003). We can obtain up to 4 solutions for the pose using 3 points, and in the geometric approach, we choose the solution that results in smallest squared re-projection error for the 4th point (Gao et al., 2003),

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{\forall i} \|\mathbf{p}_Q^{(i)} - \mathbf{KMP}^{(i)}\|, \quad i \in \{1, 2 \dots, n\} \tag{A.8}$$

where $\mathbf{M}$ is the sought world-to-camera transformation matrix, $\mathbf{M}^*$ is its best estimate, $\mathbf{K}$ is the intrinsic matrix, $\mathbf{p}_Q^{(i)}$ is the ith feature point at the query image and $\mathbf{P}^{(i)}$ is its corresponding 3D coordinate.

In reality, the set of 2D-3D correspondences $\hat{C}$ can be corrupted by outliers, so it is common to use a robust estimator together with PnP solvers. RANSAC (Fischler and Bolles, 1981) estimator is a popular choice, and in our work we use a generalization of the RANSAC estimator, MLESAC (Torr and Zisserman, 2000). MLESAC adopts the same sampling strategy as RANSAC to generate putative solutions, but chooses the solutions by maximizing the likelihood rather than just the number of inliers.

Finally, the 6-DoF camera pose can be obtained by means of the decomposition of $\mathbf{M}^*$ via (A.3).

Appendix B. Direct photometric-based camera pose estimation

This appendix explains the details of the three stages of the *direct* photometric-based camera pose estimation, namely, generation of synthetic views, direct photometric matching and coarse-to-fine search.

B.1. Generation of synthetic views

The reference 3D point cloud P_R does not have any color or intensity information, but this information can be retrieved from the reference image as follows. Firstly, we project 3D point clouds $P_R = [\mathbf{P}_R^{(1)}, \mathbf{P}_R^{(2)}, \dots, \mathbf{P}_R^{(m)}]$ onto the reference image plane using (A.2) and get a set of 2D projections, $p = [\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(m)}]$. This process is the same as Fig. A.13. Secondly, we use cubic interpolation to compute the intensity values for each 2D projection and assign the intensity values to the 3D point cloud as:

$$I(\mathbf{P}_R^{(i)}) \leftarrow f(\mathbf{p}_R^{(i)}, I_R), \quad I_R \in \mathbb{R}^2 \tag{B.1}$$

where I_R is the reference image, $\mathbf{p}^{(i)}$ is the ith 2D projection, $I(\mathbf{P}_R^{(i)})$ is the intensity value of the 3D point $\mathbf{P}_R^{(i)}$, and f is the cubic interpolation function. As a result, we assign intensity (or color) information to the 3D point cloud P_R.

Synthetic views can now be rendered by projecting the colored 3D point cloud using a transformation matrix $\mathbf{M}$ using (A.2), and the intensities of the synthetic view I_S can be obtained as:

$$I_S(\mathbf{KMP}_R^{(i)}) \leftarrow I(\mathbf{P}_R^{(i)}), \tag{B.2}$$

where $I(\mathbf{P}_R^{(i)})$ is the intensity value of the ith 3D point $\mathbf{P}_R^{(i)}$, $\mathbf{K}$ is the intrinsic matrix, $\mathbf{M}$ is the world-to-synthetic-view transformation, and $I_S(\mathbf{KMP}_R^{(i)})$ is the intensity value of the projection of the 3D point $\mathbf{P}_R^{(i)}$ at the synthetic frame. Synthetic views are quickly rendered by the standard computer graphics procedure of surface splatting (Zwicker et al., 2001).

B.2. Direct photometric matching

The *direct* photometric-based approach (Tykkälä et al., 2013) is defined as a direct minimization of the cost function in the space of 6D camera pose, and in the cost function it compares the pixel intensities of the query image I_Q and rendered synthetic view I_S from the colored 3D point cloud (Tykkälä et al., 2013). The task is to find the best relative camera transform $\mathbf{M}^*$ that minimizes the photometric error between query image I_Q and synthetic image I_S:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \mathrm{RES}(I_Q, I_S), \tag{B.3}$$

where, the photometric error is represented by a residual (RES) defined as

$$\mathrm{RES}(I_Q, I_S) = \frac{1}{\mu} \sum_{(u,v)\in I_S} (I_Q(u,v) - I_S(u,v))^2 \tag{B.4}$$

In (B.4) I_Q is the query image, the synthetic view I_S is generated by (B.2), and μ is the number of pixels in I_S.

To improve the robustness of the matching process, we smooth the query image I_S by using a Gaussian filter and then we use the smoothed version of query image in the image matching process. Moreover, we use the M-estimator to improve the matching process, since the M-estimator can be used for managing outliers when the residual vector is of sufficient length for statistical purpose (Huber, 2011). The main idea is to generate small weights for residual elements that are classified as outliers by analyzing the distribution of residual values. Inliers always have small residual values whereas outliers may have any error value. In our work, a median filter is used to find the median value among the residuals, $\mathrm{RES}(I_Q, I_S)$, then we give zero weights to all the residual values that are greater than the median value, and give normalized weights to the remaining residuals.

With the M-estimator, we can rewrite the residual (B.4) as the average of the weighted sum-of-square difference:

$$\mathrm{RES}(I_Q, I_S) = \frac{1}{\lambda} \sum_{\forall(u,v)} (E(u,v))^2 w(u,v) \tag{B.5}$$

where we apply the weights to the residual vector and compute the average of the weighted sum-of-square difference, and λ is the number of nonzero weights. The squared difference $E(u,v)$ and weights $w(u,v)$ are defined in (B.6) and (B.7) as follows:

$$E(u,v) = (I_Q(u,v) - I_S(u,v))^2, (u,v) \in I_S \tag{B.6}$$

where I_Q is the query image, I_S is the synthetic image, and E is the difference between the two images.

$$w(u,v) = \begin{cases} 0, & \text{if } E(u,v) > \theta \\ 1, & \text{otherwise} \end{cases} \tag{B.7}$$

where θ is the median value of $E(u,v)$ and $(u,v) \in I_S$.

B.3. Coarse-to-fine grid search

We use a two-step coarse-to-fine grid search to solve for the matrix $\mathbf{M}^*$ in (B.3). The coarse-to-fine grid search concatenates a search with a coarse step for the local minimum with a subsequent search with a finer step at the location of the previous minimum location. Given a reference image, we use the camera pose of the reference image as the starting point for grid search. The coarse-to-fine search is firstly applied to the translation, and based on the previous minimum, we then apply

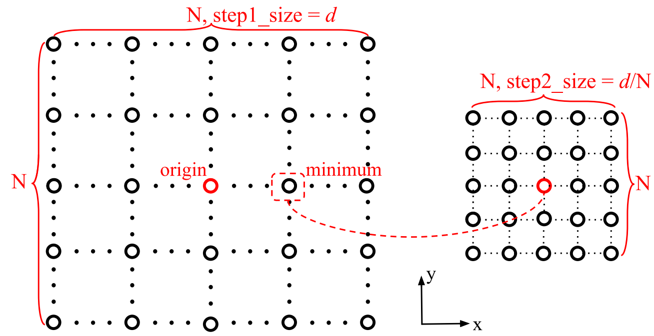


Fig. B.14. Coarse-to-fine grid search for translation. Grids are placed along x (toward to the right of the camera) and y (toward to the front of the camera) axis. Search the minimum within N steps of the step size d in a search grid, then apply a finer grid in the minimum point with another N steps of the size d/N.

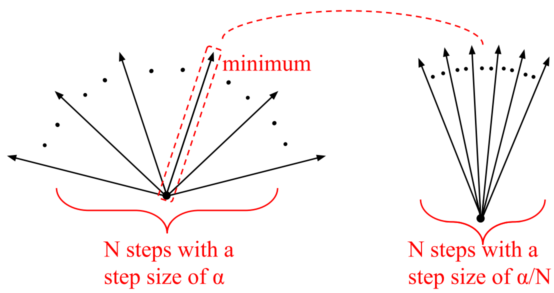


Fig. B.15. Coarse-to-fine grid search for orientation. For the selected axis (z axis, toward up of the camera), search by N steps of the angular size α, then refine the search by another N of the size α/N.

it to the orientation. The process of the coarse-to-fine grid search is illustrated in Figs. B.14 and B.15, and we describe the steps as follows:

Firstly, we take the orientation of the reference image for query image and start coarse-to-fine grid search for translation. There are 2 iterations in total. In the 1st iteration, we define a 2D grid along the x axis (towards the right of the camera) and y axis (toward the front of the camera) with a grid dimension of N and a step size of 10. A synthetic view I_S is generated for each grid point by (B.2), then we apply (B.5) to compute the residual value (RES) for this grid point. Then grid point with the minimum residual value is taken as the starting point for the 2nd iteration. In the 2nd iteration, we reduce the step size by 10 times and repeat the same procedure. In the end, we have estimated translation for query image. The above coarse-to-fine grid search for translation is illustrated in Fig. B.14.

Secondly, we fix translation of the query image and apply another coarse-to-fine grid search for orientation. We could search the optimal orientations along one or multiple axes. For our experiments, we search the optimal orientations along the z axis (toward up direction of the car), i.e. optimizing the yaw angle. The search procedure is similar to the one for translation. The process of the coarse-to-fine grid search for orientation is illustrated in Fig. B.15.

In our experiments, the both datasets consist of images captured by cameras mounted on cars and therefore there is mainly variation in the yaw angle for orientation. In our experimental setup, we choose to do orientation search only along the z axis. The full 6-DoF grid search would require a combination of the translation search (Fig. B.14) and three orientation searches (Fig. B.15).

In the process of generating a synthetic view I_S, a 3D point cloud is projected on a camera pose by (A.2). For each synthetic view in the grid search, we count the number of points projected inside the image frame. If the number of projected points is less than a threshold (100 in our experiments, see Table C.10), the synthetic view is considered as invalid. The invalid synthetic view is skipped in the grid search. If all synthetic views are invalid, the grid search fails to give a camera pose estimate.

Table C.8
Details of the test platform.

Processor	Intel i7CPU 2.70 GHz
OS	Ubuntu 16.04
Memory	32 GB
SW Env.	Matlab

Table C.9
Average time performance of the evaluated methods with a single query and a single reference image. Note these two original papers (Tykkälä et al., 2013; Pascoe et al., 2017) were designed for slam problem, but we modified the algorithms to adjust to our problem, and we implemented them in a laptop without utilizing GPU and multi-threads.

	KITTI	Oxford RobotCar
FB (Kim et al., 2014)	0.06 s	0.08 s
PM (Tykkälä et al., 2013)	1.23 s	4.82 s
MI (Pascoe et al., 2017)	1.34 s	5.15 s
HY (Combine FB and MI)	0.07 s	4.00 s

Appendix C. Implementation details and limitations

C.1. Platform and time performance

For reference purposes, we implemented and tested all the evaluated methods without utilizing GPU or multi-thread processing. The specifications of the platform and the programming language are shown in Table C.8. The average computing times are reported in Table C.9. In our implementation, the feature-based approach was the fastest one. The most time-consuming task for the photometric and mutual-information method was generation of synthetic views Appendix B.1. The computations are slower for the Oxford RobotCar dataset since point clouds are much larger. In addition, with the Oxford RobotCar dataset the feature-based method fails more frequently, and the HY method takes more mutual-information matching which is much slower.

C.2. Data preprocessing

With the KITTI Visual Odometry dataset (Geiger et al., 2012), we utilize the original 3D point clouds (LIDAR), ground truth pose data, and gray-scale images of each sequence. With the Oxford RobotCar dataset (Maddern et al., 2017), we also utilize the LIDAR scans, camera pose, and the left image from the trinocular stereo camera. However, the original 2D LIDAR data is saved as a single scan instead of an accumulated 3D point cloud as in the KITTI dataset. Therefore we applied two pre-processing steps to Oxford point clouds:

1. We converted the 2D LIDAR scans into a 3D point cloud by utilizing the toolkit provided by the authors.[2]
2. For efficiency, we reduced the number of images in each sequence by using every 10-th image and removed the start and final frames where the car was usually parked. The mean metric distance between two consecutive frames are shown in Table 2.

C.3. Parameters selection

The details of all parameters used in our experiments are shown in Table C.10.

[2] https://github.com/ori-drs/robotcar-dataset-sdk.

Table C.10
Method parameter values used in the experiments.

Feature-based (FB)	
Feature type	SURF
Photometric (PM)	
Min # of projected points	100
Mutual-information (MI)	
Min # of projected points	100
Grid search	
Translation	
Grid dimension	$d = 2 \times r$ meters, r is the uncertainty radius.
# of steps (1st iter)	10
Step length (1st iter)	$\frac{d}{10}$ meters
# of steps (2nd iter)	10
Step length (2nd iter)	$\frac{d}{100}$ meters
Orientation	
Search range	30 degrees
# of steps (1st iter)	10
Step size (1st iter)	3°
# of steps (2nd iter)	10
Step size (2nd iter)	0.3°

References

Alahi, A., Ortiz, R., Vandergheynst, P., 2012. Freak: Fast retina keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 510–517.

Ambai, M., Yoshida, Y., 2011. CARD: Compact and real-time descriptors. In: IEEE International Conference on Computer Vision. pp. 97–104.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features. In: European Conference on Computer Vision. Springer, pp. 404–417.

Belongie, S., Malik, J., Puzicha, J., 2001. Shape context: A new descriptor for shape matching and object recognition. In: Advances in Neural Information Processing Systems. pp. 831–837.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. Brief: Binary robust independent elementary features. In: European Conference on Computer Vision. Springer, pp. 778–792.

Castellanos, J.A., Tardos, J.D., 2012. Mobile robot localization and map building: A multisensor fusion approach. Springer Science & Business Media.

Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C., 2004. Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision in European Conference on Computer Vision, Vol. 1. Prague, pp. 1–2.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1. pp. 886–893.

Delmerico, J., Scaramuzza, D., 2018. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. Memory 10, 20.

Engel, J., Koltun, V., Cremers, D., 2018. Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. 40 (3), 611–625.

Engel, J., Schöps, T., Cremers, D., 2014. LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision. Springer, pp. 834–849.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Forster, C., Pizzoli, M., Scaramuzza, D., 2014. SVO: Fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation. pp. 15–22.

Friedman, J.H., Bentley, J.L., Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Software 3 (3), 209–226.

Froba, B., Ernst, A., 2004. Face detection with the modified census transform. In: IEEE International Conference on Automatic Face and Gesture Recognition. pp. 91–96.

Gao, X.-S., Hou, X.-R., Tang, J., Cheng, H.-F., 2003. Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. 25 (8), 930–943.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the KITTI vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition.

Geiger, A., Ziegler, J., Stiller, C., 2011. Stereoscan: Dense 3d reconstruction in real-time. In: Intelligent Vehicles Symposium. IEEE, pp. 963–968.

Gibbons, J.D., Chakraborti, S., 2011. Nonparametric statistical inference. In: International Encyclopedia of Statistical Science. Springer, pp. 977–979.

Guo, Z., Zhang, L., Zhang, D., 2010. Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recognit. 43 (3), 706–719.

Huber, P.J., 2011. Robust statistics. Springer.

Irschara, A., Zach, C., Frahm, J.-M., Bischof, H., 2009. From structure-from-motion point clouds to fast location recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2599–2606.

Iscen, A., Tolias, G., Avrithis, Y., Furon, T., Chum, O., 2017. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In: IEEE Conference on Computer Vision and Pattern Recognition.

Kadir, T., Brady, M., 2001. Saliency, scale and image description. Int. J. Comput. Vis. 45 (2), 83–105.

Kim, H., Lee, D., Oh, T., Lee, S.W., Choe, Y., Myung, H., 2014. Feature-based 6-dof camera localization using prior point cloud and images. In: Robot Intelligence Technology and Applications 2. Springer, pp. 3–11.

Kitt, B., Geiger, A., Lategahn, H., 2010. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: Intelligent Vehicles Symposium. IEEE, pp. 486–492.

Klaser, A., Marszałek, M., Schmid, C., 2008. A spatio-temporal descriptor based on 3d-gradients. In: British Machine Vision Conference. British Machine Vision Association, 275–1.

Klein, G., Murray, D., 2007. Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 225–234.

Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. BRISK: Binary robust invariant scalable keypoints. In: IEEE International Conference on Computer Vision. pp. 2548–2555.

Li, A.Q., Coskun, A., Doherty, S.M., Ghasemlou, S., Jagtap, A.S., Modasshir, M., Rahman, S., Singh, A., Xanthidis, M., OKane, J.M., et al., 2016. Experimental comparison of open source vision-based state estimation algorithms. In: International Symposium on Experimental Robotics. Springer, pp. 775–786.

Lienhart, R., Maydt, J., 2002. An extended set of haar-like features for rapid object detection. In: International Conference on Image Processing, Vol. 1.

Linegar, C., Churchill, W., Newman, P., 2016. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In: IEEE International Conference on Robotics and Automation. pp. 787–794.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision. pp. 1150–1157.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2), 91–110.

Maddern, W., Pascoe, G., Linegar, C., Newman, P., 2017. 1 year, 1000km: The oxford robotCar dataset. Int. J. Robot. Res. 36 (1), 3–15.

Markley, F.L., Cheng, Y., Crassidis, J.L., Oshman, Y., 2007. Averaging quaternions. J. Guidance Control Dyn. 30 (4), 1193.

Matas, J., Chum, O., Urban, M., Pajdla, T., 2004. Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. 22 (10), 761–767.

McDaid, A.F., Greene, D., Hurley, N., 2011. Normalized mutual information to evaluate overlapping community finding algorithms, arXiv preprint arXiv:1110.2515.

Mikolajczyk, K., Schmid, C., 2004. Scale & affine invariant interest point detectors. Int. J. Comput. Vis. 60 (1), 63–86.

Mishkin, D., Matas, J., Perdoch, M., 2015. Mods: Fast and robust method for two-view matching. Comput. Vis. Image Underst. 141, 81–93.

Miura, S., Hsu, L.-T., Chen, F., Kamijo, S., 2015. GPS Error correction with pseudorange evaluation using three-dimensional maps. IEEE Trans. Intell. Transp. Syst. 16 (6), 3104–3115.

Mori, G., Ren, X., Efros, A.A., Malik, J., 2004. Recovering human body configurations: Combining segmentation and recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2. II–II.

Muja, M., Lowe, D.G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration.. Int. Conf. Comput. Vis. Theory Appl. 2 (331–340), 2.

Mur-Artal, R., Montiel, J.M.M., Tardos, J.D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. 31 (5), 1147–1163.

Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A., 2011a. Kinectfusion: Real-time dense surface mapping and tracking. In: IEEE International Aymposium on Mixed and Augmented Reality. pp. 127–136.

Newcombe, R.A., Lovegrove, S.J., Davison, A.J., 2011b. DTAM: Dense tracking and mapping in real-time. In: IEEE International Conference on Computer Vision. pp. 2320–2327.

NovAtel-Inc., 2019. span gnss inertial systems, Accessed: 2019-03-01, https://www.novatel.com/products/span-gnss-inertial-systems/.

Ohta, Y., Tamura, H., 2014. Mixed reality: merging real and virtual worlds. Springer Publishing Company, Incorporated.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987.

Oxford-Technical-Solutions-Ltd, OXTS-RT3000, Accessed: 2019-03-01, https://www.oxts.com/products/rt3000/.

Pascoe, G., Maddern, W., Tanner, M., Pinies, P., Newman, P., 2017. NID-SLAM: Robust monocular SLAM using normalised information distance. In: IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI.

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A., 2007. Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Radenović, F., Tolias, G., Chum, O., 2016. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In: European Conference on Computer Vision.

Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection. European Conference on Computer Vision 430–443.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. In: IEEE International Conference on Computer Vision. pp. 2564–2571.

Sattler, T., Havlena, M., Schindler, K., Pollefeys, M., 2016. Large-scale location recognition and the geometric burstiness problem. In: IEEE Conference on Computer Vision and Pattern Recognition.

Scovanner, P., Ali, S., Shah, M., 2007. A 3-dimensional sift descriptor and its application to action recognition. In: ACM International Conference on Multimedia. pp. 357–360.

Song, Y., Chen, X., Wang, X., Zhang, Y., Li, J., 2016. 6-DOF image localization from massive geo-tagged reference images. IEEE Trans. Multimed. 18 (8), 1542–1554.

Strecha, C., Bronstein, A., Bronstein, M., Fua, P., 2012. Ldahash: Improved matching with smaller descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 34 (1), 66–78.

Taylor, A.G., 2016. Develop microsoft hololens apps now. Springer.

Tola, E., Lepetit, V., Fua, P., 2010. Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. 32 (5), 815–830.

Torr, P.H., Zisserman, A., 2000. MLESAC: A new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. 78 (1), 138–156.

Tuytelaars, T., Mikolajczyk, K., et al., 2008. Local invariant feature detectors: a survey. Found. Trends. Comput. Graph. Vision 3 (3), 177–280.

Tuytelaars, T., Van Gool, L.J., 2000. Wide baseline stereo matching based on local, affinely invariant regions.. In: British Machine Vision Conference, Vol. 412.

Tuytelaars, T., Van Gool, L., 2004. Matching widely separated views based on affine invariant regions. Int. J. Comput. Vis. 59 (1), 61–85.

Tykkälä, T., Comport, A.I., Kämäräinen, J.-K., 2013. Photorealistic 3D mapping of indoors by RGB-d scanning process. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1050–1055.

Wen-Tsun, W., 1986. Basic principles of mechanical theorem proving in elementary geometries. J. Automated Reason. 2 (3), 221–252.

Zahn, C.T., Roskies, R.Z., 1972. Fourier descriptors for plane closed curves. IEEE Trans. Comput. 100 (3), 269–281.

Zhao, G., Pietikainen, M., 2007. Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. 29 (6).

Zwicker, M., Pfister, H., Van Baar, J., Gross, M., 2001. Surface splatting. In: Conference on Computer Graphics and Interactive Techniques. ACM, pp. 371–378.