This paper has been accepted for publication at the
International Journal of Computer Vision (IJCV), 2020. ©Springer

Noname manuscript No.
(will be inserted by the editor)

Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis

Zichao Zhang · Torsten Sattler · Davide Scaramuzza

Received: date / Accepted: date

Abstract Visual Localization is one of the key enabling technologies for autonomous driving and augmented reality. High quality datasets with accurate 6 Degree-of-Freedom (DoF) reference poses are the foundation for benchmarking and improving existing methods. Traditionally, reference poses have been obtained via Structure-from-Motion (SfM). However, SfM itself relies on local features which are prone to fail when images were taken under different conditions, e.g., day/ night changes. At the same time, manually annotating feature correspondences is not scalable and potentially inaccurate. In this work, we propose a semi-automated approach to generate reference poses based on feature matching between renderings of a 3D model and real images via learned features. Given an initial pose estimate, our approach iteratively refines the pose based on feature matches against a rendering of the model from the current pose estimate. We significantly improve the nighttime reference poses of the popular Aachen Day-Night dataset, showing that state-of-the-art visual localization methods perform better (up to 47%) than predicted by the original reference poses. We extend the dataset with new nighttime test images, provide uncertainty estimates for our new reference poses, and introduce a new evaluation criterion. We will make our reference poses and our framework publicly available upon publication.

Z. Zhang and D. Scaramuzza
Robotics and Perception Group
University of Zurich
E-mail: zzhang@ifi.uzh.ch, sdavide@ifi.uzh.ch

T. Sattler
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
E-mail: torsten.sattler@cvut.cz

Keywords Visual localization, benchmark construction, learned local features

1 Introduction

Visual localization is the problem of estimating the camera pose, i.e., the position and orientation from which an image was taken, with respect to a known scene. Visual localization is a core component of many interesting applications such as self-driving cars [1] and other autonomous robots such as drones [2], as well as for augmented and virtual reality systems [3, 4].

Similar to other areas in computer vision, the availability of benchmark datasets such as [5–11] has served as a main driving force for research. Yet, there is a fundamental difference between visual localization and areas such as semantic segmentation and object detection in the way ground truth is obtained. For the latter, ground truth is provided by human annotations. However, humans are not able to directly predict highly accurate camera poses. Instead, ground truth is typically computed through a reference algorithm, e.g., Structure-from-Motion (SfM). Thus, localization benchmarks do not measure absolute pose accuracy. Rather, they measure to what degree methods are able to replicate the results of the reference algorithm. Given that the reference approach itself will produce inaccuracies and errors in the pose estimates, we use the term "reference poses" instead of "ground truth poses".

It is crucial that the reference algorithm generates poses with a higher accuracy than the actual localization methods evaluated on a benchmark. It is thus common to provide more data to the reference algorithm compared to what is made available to the localization approaches. For example, data from other sensors

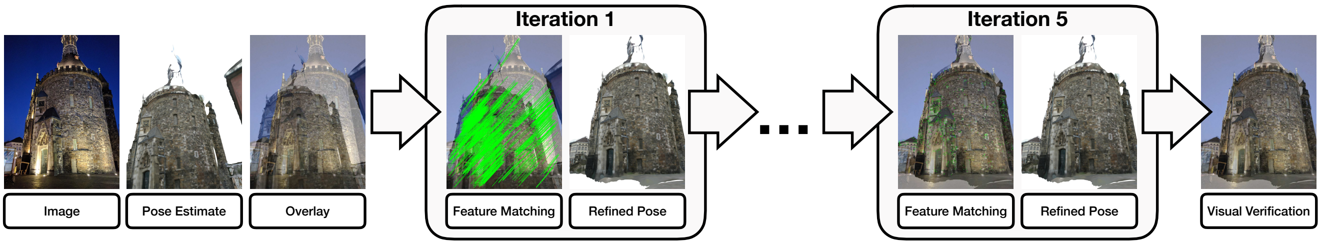


Fig. 1: Overview of our approach: Given an image, we render a synthesized view of a 3D model from the given initial pose estimate of the image. Superimposing the rendered image over the original image provides a visual cue on the accuracy of the pose estimate. We match features extracted from the actual image and the rendering (shown as green lines connecting the corresponding positions in the overlay of the two images). This provides 2D-3D correspondences between the image and the underlying scene model. These 2D-3D matches are then used to obtain a refined estimate. Iterating this approach leads to subsequently more accurate poses (as evident from the smaller lines caused by a more accurate overlay). The final pose estimate can also be verified visually.

such as depth [5,6], Lidar [11], an external motion capture system such as Vicon [12], or additional images not available to the localization methods [12] can be used if available. This paper considers the case where only images are available. In this case, SfM is typically used as the reference algorithm, i.e., the reference poses are obtained jointly from all test images whereas localization approaches typically localize a single image at a time. This should lead to more accurate reference poses compared to what can be obtained from a single image.

In particular, we are interested in reference pose generation in the context of long-term localization, which is the problem of localizing images taken under different conditions, e.g., day-night or seasonal changes, against a scene captured under a reference condition. Given that scenes change over time, long-term localization is an important problem in practice. The main challenge in this setting is data association, i.e., establishing feature matches between images taken under different conditions. Naturally, this causes problems for generating reference poses using SfM algorithms, which themselves rely on local features such as SIFT [13] for data association. In previous work, we thus relied on human annotations to obtain feature matches between images taken under different conditions [9]. However, this approach is not scalable. Furthermore, human annotations of feature positions in images tend to be inaccurate, as they can easily be off by 5-10 pixels or more.

This paper is motivated by the observation that the reference poses for the nighttime test images of the Aachen Day-Night dataset [8,9], obtained from human annotations, are not accurate enough to benchmark state-of-the-art localization methods. This paper thus proposes a semi-automated approach to reference pose generation. Our method is inspired by previous work on pose verification via view synthesis [14–16] and the observation that modern learned local features [17, 18] capture higher-level shape information. The latter allows feature matching between real images and 3D models, e.g., obtained via multi-view stereo [19]. As shown in Fig. 1, our approach starts with a given initial pose estimate. It renders the 3D scene model from the current pose estimate. Feature matches between the actual and the re-rendered image are then used to refine the pose estimate. This procedure is repeated for a fixed number of iterations. Detailed experiments, for multiple ways to obtain initial poses, show that our approach yields more accurate pose estimates.

Re-rendering the image from its estimate pose enables visual inspection of the accuracy of the estimate. Using this aid, we observe that even larger differences in pose of 20cm or more can have little impact on the rendered image. This is not particularly surprising as the uncertainty of a pose estimate depends on the distance to the scene. However, it also implies that using fixed position and rotation thresholds on the pose error to measure localization accuracy [5,9] is not appropriate if there are significant changes in scene depth between test images. As a second contribution, we thus discuss and evaluate multiple evaluation measures that (explicitly or implicitly) use per-image uncertainty measures rather than global thresholds on pose errors.

In detail, this paper makes the following contributions: (**1**) we propose an approach based on view synthesis and learned features that can be used to generate reference pose for long-term visual localization benchmarks. (**2**) we provide a detailed experimental analysis of our approach, including studying different initialization approaches, different strategies for rendering and different features. (**3**) we show that the existing nighttime reference poses of the Aachen Day-Night dataset are not accurate enough to evaluate state-of-the-art long-term localization approaches. We further use our approach to obtain refined reference poses and show that current localization approaches achieve a much higher (up to 47%) pose accuracy than indicated by the original reference poses. (**4**) we extend the Aachen Day-Night dataset by additional nighttime test images, effectively doubling the number of available test images. We evaluate state-of-the-art localization approaches on the extended dataset and will provide a benchmark at visuallocalization.net. (**5**) we discuss and experimentally study additional evaluation measures. (**6**) we will make source code for our approach and our evaluation measures publicly available to facilitate the creation of new benchmarks. (**7**) we provide a concise review of current trends in the area of visual localization.

2 Related work

Besides discussing related work on benchmark creation for visual localization and the use of view synthesis for pose estimation and verification, this section also aims at giving an interested reader a concise overview over main trends in the area of visual localization.

Visual localization. Traditionally, most visual localization algorithms have been based on a combination of local features and a 3D scene model [20–28]. In most cases, the underlying 3D model is a sparse 3D point cloud constructed using SfM [29, 30] or SLAM [31, 32]. Each point in this model has been triangulated from two or more local image features such as SIFT [13] or ORB [33]. Thus, each 3D point can be associated with one or more local image descriptors. 2D-3D correspondences between local features in a query image and 3D model points can be found using nearest neighbor search in the descriptor space. In turn, these 2D-3D matches can be used to estimate the camera pose of the query image by applying an n-point pose solver [34–40] inside a hypothesize-and-verify framework such as RANSAC [40] and its variants [41–43]. Research on such 3D *structure-based* methods has mostly focused on scalability, e.g., by accelerating the 2D-3D matching stage [2, 22–24, 27, 44–46] and the use of image retrieval [8, 14, 25, 47–49], by reducing memory requirements through model compression [4, 22, 50–52], or by making the pose estimation stage more robust to the ambiguities encountered at scale [23, 53–57].

Such approaches are computationally too complex for mobile devices with limited resources, e.g., robots and smart phones. In order to achieve real-time localization on such devices, non-real-time global localization against a pre-built map is combined with real-time local camera pose tracking [4, 27, 58–63]. To this end, results from the localization process (most often 2D-3D inliers) are integrated into visual(-inertial) odometry or SLAM to prevent drift in the local pose estimates.

Structure-based approaches rely on underlying 3D models, which are expensive to build at scale and costly to maintain [64]. Alternatively, the absolute pose of a query image can be estimated from the relative poses to database images with known poses [65, 66] or 2D-2D matches with multiple database images [67]. It can also be estimated using local SfM models computed on the fly [64].

Instead of explicitly using an underlying 3D model, *absolute pose regression* train a CNN to directly regress the camera pose from an input image [7, 68–77]. However, they are not consistently more accurate than simple image retrieval baselines [16, 78, 79] that approximate the pose of a query image by the poses of the top-retrieved database images [80]. Furthermore, these approaches need to be trained specifically per scene. The latter problem can be overcome by *relative pose regression* techniques [66, 81–84], which train CNNs to predict relative poses. In combination with image retrieval against a database of images with known poses, these relative poses can be used for visual localization. While recent work shows promising results [66, 82, 84], relative pose regression techniques do not yet achieve the same level of pose accuracy as methods explicitly based on 2D-3D matches.

Rather than learning the full localization pipeline, *scene coordinate regression* algorithms only replace the 2D-3D matching stage through a machine learning algorithm, typically either a random forest [5, 6, 85–91] or a CNN [90, 92–97]. For a given patch from an image, these methods predict the corresponding 3D point in the scene. The resulting in a set of 2D-3D matches can then be used for camera pose estimation. Scene coordinate regression techniques constitute the state-of-the-art in terms of pose accuracy in small scenes. However, they currently do not scale well to larger scenes. For example, ESAC [94], a state-of-the-art scene coordinate regression technique, localizes 42.6% of all daytime query images of the Aachen Day-Night dataset [8, 9] within errors of 25cm and 5°. In contrast, SIFT-based Active Search [44], a classical structure-based method, localizes 85.3% within the same error thresholds.

Learned local features. State-of-the-art approaches for long-term localization [14, 15, 17, 47, 98–102] are based on local features and explicit 3D scene models.[1] Classical handcrafted features such as ORB [33], SIFT [13], and SURF [103] struggle to match features between im-

[1] See also visuallocalization.net/benchmark/.

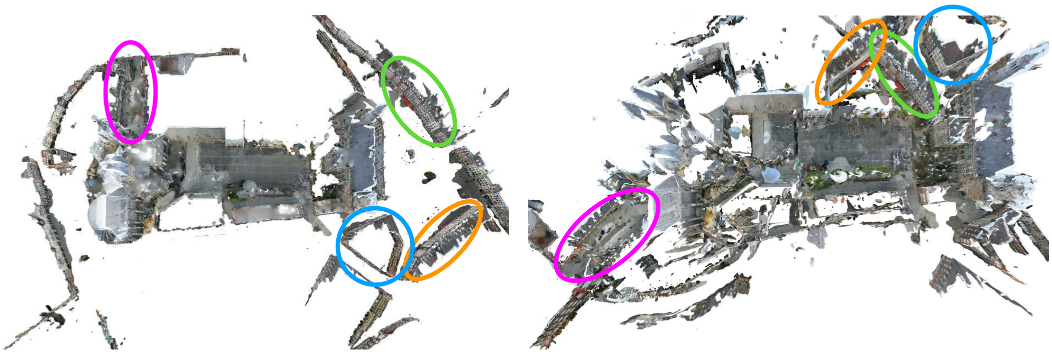


Fig. 2: Multi-View Stereo reconstructions obtained from SfM models of the Aachen dataset using SIFT (left) and D2-Net (right) features (top-down view). D2-Net features are more robust to changes in conditions, e.g., day-night and seasonal changes, than classic SIFT features, but also produce more false positive matches. This leads to connecting unrelated scene parts during the SfM process and ultimately in an incorrect 3D model. In contrast, SIFT correctly reconstructs the scene. Some wrong placements are illustrated through colored ellipses.

ages taken under strongly differing viewing conditions, e.g., day and night or seasonal changes. Thus, long-term localization approaches typically use machine learning, both for image retrieval [78,104,105] and for local features [17,101,102,104,106,107].

Traditionally, local feature learning has focused on learning feature descriptors [108–115]. However, it has been shown that the local feature detector often is the limiting factor [9, 14, 16, 98]. Thus, recent work trains feature detectors and descriptors jointly [101, 102, 104, 106,107,116], leading to state-of-the-art feature matching performance for images taken under strongly differing conditions. Interestingly, using deeper layers of neural networks pre-trained on ImageNet [117] to define both feature detector and descriptor leads to very competitive performance [17, 102]. Equally important, such features are very robust to changes in different conditions, even though this might come at a price of more false positives (cf. Fig. 2). We use this robustness to establish correspondences between real images and renderings of 3D models and the resulting 2D-3D matches to compute reference poses for benchmarking long-term visual localization. In addition, we benchmark state-of-the-art long-term localization approaches [17,18,47,98] based on local features our reference poses.

Semantic visual localization. Besides using learned features that are more robust to changes in viewing conditions, long-term localization approaches also use semantic image segmentation [15,99,100,118–126]. These methods are based on the observation that the semantic meaning of scene elements, in contrast to their appearance, is invariant to changes. Semantic image segmentations are thus used as an invariant representation for image retrieval [123, 126, 127], to verify 2D-3D matches [99, 100, 118, 124] and camera pose estimates [15,100,122,124], for learning local features [119, 120], and as an additional input to learning-based localization approaches [118,121,125].

View synthesis. As shown in Fig. 1, our approach iteratively renders a 3D model from a camera pose estimate and uses matches between the rendering and the actual image to refine the pose. Our approach takes inspiration from previous work on using view synthesis for pose estimation and verification. [128, 129] render detailed laser scans [128] respectively dense Multi-View Stereo point clouds [129] from new perspectives. They show that SIFT feature matching between the renderings and actual images is possible if both were taken from very similar poses. [16] shows that view synthesis from very similar viewpoints (obtained from depth maps) improves SIFT feature matching between day and night images. [130] learns features that can be matched between paintings and renderings of a 3D model. [131] first learns a randomized decision forest (RDF) and a hierarchical navigation graph using synthesized images (rendered from reconstructed scene models) and then uses the RDF and the graph for efficient and gradient-free localization of new query images. In these works, view synthesis is used to create novel viewpoints in a given scene in order to enable camera pose estimation at all. In contrast, this paper focuses on using view synthesis to refine an initial pose estimate and to use it for generating reference poses for a long-term localization benchmark. Thus, the contributions of this paper center around a detailed experimental evaluation of the use of view synthesis to improve pose accuracy rather than on proposing a new method.

[14,15] use view synthesis for automated pose verification. To this end, they render a dense laser scan point cloud from a set of given poses. They densely extract

descriptors from each rendering and compare each descriptor against a descriptor extracted at the same pixel in the original image to compute an image-level similarity score. This score is then used to select the pose that best explains the input image. In contrast, this paper uses view synthesis to refine the camera pose estimates. While [14,15] automate pose estimation, their approach still has room for improvement, even if additional information such as semantics is used [15]. Thus, we use the rendering for visual inspection of the poses rather than automating the verification process.

[132] utilizes synthesized views for improving an initial pose estimate with respect to a 2.5D street map (containing only the outline of the buildings). Given a rendered view of the 2.5D map and a semantic segmentation of the image, they combine two networks and a line search strategy to compute a 3DoF pose correction (horizontal translation and yaw) to the initial pose. Then the improved pose is used as the new input, and the correction procedure is applied iteratively. This paper uses a similar strategy of iterative synthesis and correction but differs in several aspects. [132] focuses on the geolocalization in urban environment, while we aim at providing a general tool for creating accurate visual localization benchmarks. This also results in the different choices of the input modality: we choose to use SfM models instead of 2.5D maps, which are specific to urban environments. Moreover, our method is more generic in that it is able to correct the poses in 6DoF instead of 3DoF.

Visual localization benchmarks. This paper considers the visual localization problem, i.e., the task of computing the full camera pose for a given image. Closely related is the visual place recognition problem of determining which place is visible in a given image, without necessarily estimating its camera pose. However, we will not discuss pure place recognition datasets that do not provide full 6DoF camera poses such as [16, 133–136].

Early localization benchmarks used SfM to reconstruct scenes from internet photo community collections such as Flickr. Query images were then obtained by removing some images from the reconstruction, together with all 3D points visible in only one of the remaining images [22]. Examples for this approach to benchmark creation are the Dubrovnik, depicting the old city of Dubrovnik (Croatia), Rome [22] and Landmarks 1k [23] datasets. The latter two datasets consists of individual landmarks in Rome respectively around the world. The same approach was later also used for images taken under more controlled conditions, e.g., the crowd-sourced Arts Quad [23,137] dataset, the scenes from the Cambridge Landmarks [7] benchmark, and the San Francisco SF-0 [23,64,138] dataset. Similarly, RGB-D SLAM algorithms [139,140] were used to obtain reference poses for the 7Scenes [5] and 12Scenes [6] datasets. Both depict small indoor scenes captured with RGB-D sensors.

Long-term localization benchmarks [9,141,142] typically use images captured under a reference condition to represent the scene while images taken under different conditions are used as query. SLAM and SfM algorithms depend on data association between images. Thus, they tend to fail if images were taken under too dissimilar conditions. Using image sequences and / or multi-camera systems can allow using SLAM and SfM algorithms under stronger viewing condition changes. The former exploits the fact that it is not necessary to find matches between each query image and a reference image. Rather, finding enough matches for some query images is sufficient to register an entire sequence. The latter exploit the fact that a larger field-of-view typically leads to more matches. Both the SILDa [142] and (extended) CMU Seasons [9,10] use sequences and multi-camera systems. SILDa depicts a single building block in London, UK under diferent conditions. The (extended) CMU Seasons dataset was constructed from images collected in and around Pittsburgh, US over the span of a year. For the (extended) CMU Seasons, additional humanly annotated matches were used in areas where cross-seasonal matching failed [9]. Human annotations were also used for the Mall [143] dataset to obtain initial pose estimates of test images with respect to a laser scan.

Manually annotated matches are often not very precise [64]. If available, additional sensors such as Lidar can be used to avoid the need for human annotations. The RobotCar Seasons [9, 11], depicting the city of Oxford, UK under various seasonal conditions, and the University of Michigan North Campus Long-Term Vision and LIDAR [141] datasets use Lidar data to obtain reference poses. However, human intervention might still be necessary if the scene geometry changes [9].

The Aachen Day-Night [8, 9] depicts the old inner city of Aachen, Germany. The 3D model of the scene was reconstructed from daytime images using SfM. Similarly, reference poses for daytime query images were also obtained using SfM. Since additional sensor data is not available and since SfM failed to provide reference poses [9], manual annotations were used for a set of nighttime query images. To this end, a daytime image taken from a similar viewpoint was selected for each nighttime query. The pixel positions corresponding to 10 to 30 3D points visible in the daytime image were then annotated manually. [9] estimated that the median mean position accuracy for the nighttime images is between 30cm and 40cm. However, in this paper, we show that the pose estimates are actually often worse. This observation motivates our approach for refining the original reference poses. We show that the refined poses are more accurate and are thus more suitable to measure the performance of state-of-the-art localization techniques. While this paper focuses on the Aachen Day-Night dataset, our approach is not specific to it and can be applied on other datasets as well. As described in Sect. 5.1, it uses the same information as required for building SfM models, which is available in many visual localization benchmarks. The scene models used in our approach, namely a SfM model and a dense mesh, can be generated using publicly available software packages (e.g., COLMAP in our setup).

3 Reference Pose Generation

Typically, a visual localization dataset provides a set of images $\mathcal{I}:\{I_i\}_{i=1}^{N}$ and the corresponding reference poses $\mathcal{T}:\{\mathtt{T}_i\}_{i=1}^{N}$ in a 3D model $\mathcal{M}$. Our goal is to know whether the poses $\mathcal{T}$ are accurate (verification) and get more accurate reference poses if necessary (refinement). Since each image in a visual localization dataset is usually treated individually, we consider a single image I and its (potentially inaccurate) pose $\mathtt{T}$ in this section. $\mathtt{T}$ represents the camera pose with respect to the model $\mathcal{M}$. More specifically, $\mathtt{T}$ is a 4×4 transformation matrix:

$$\mathtt{T} = \begin{bmatrix} \mathtt{R} & \mathbf{c} \\ \mathtt{0}_{1\times3} & 1 \end{bmatrix}, \tag{1}$$

and $\mathbf{p} = \mathtt{R}\cdot {}_c\mathbf{p} + \mathbf{c}$ converts point coordinates in the camera frame ${}_c\mathbf{p}$ to the coordinates in the model.

Given the 3D model $\mathcal{M}$, we first render a synthesized view I^r (or multiple rendered images) at pose $\mathtt{T}$ (Sect. 3.1). Then learned features are extracted and matched between the actual image I and the synthesized image I^r. By analyzing the matched features, denoted as $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ for the actual and rendered images respectively, we can determine whether the pose $\mathtt{T}$ is accurate (Sect. 3.2). Finally, we can back-project the 2D features from the rendered view $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ to the 3D model $\mathcal{M}$ to get a set of 3D points $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$. From the 2D-3D correspondences $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$, we can calculate a more accurate pose $\mathtt{T}^r$ for the actual image (Sect. 3.3). The aforementioned process is repeated several times to get more accurate poses (cf. Fig. 1). We also discuss different methods to quantify the uncertainties of the resulting poses (Sect. 3.4), which are useful for defining localization accuracy metrics (cf. Sect. 4.1).

For simplicity of presentation, we assume that all the 2D features $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ have a valid back-projection in $\mathcal{M}$ and all the 3D points $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$ are inliers in the refinement process. In practice, we remove 2D features with invalid depth (e.g., due to an incomplete the model $\mathcal{M}$) and reject outliers using LO-RANSAC [42]. For simplicity, we assume that the features are ordered based on matches: for a feature $\mathbf{u}_l$ in the real image, the corresponding matching feature in a rendering is $\mathbf{u}_l^r$.

3.1 Rendering Synthesized Views

There are different methods to render synthesized views from a pose $\mathtt{T}$ with respect to a scene model $\mathcal{M}$. In this work, we investigate view synthesis from two different scene models: a 3D point cloud with SIFT descriptors and a 3D mesh. In the process of generating reference poses using SfM, the scene is typically reconstructed as a 3D point cloud, where each point is associated with a descriptor, e.g., SIFT. A 3D mesh can be further generated using Multi-View Stereo. Therefore, these two models are readily available from the standard process for generating reference poses.

To render images from a 3D mesh, there are various off-the-shelf renderers that can be used. As for a point cloud with descriptors, we follow [144] and train a CNN to reconstruct the images from such a scene representation. The network uses a U-Net architecture [145]. The input to the network is a 3D tensor of size $h\times w\times 129$, where h and w are the height and width of the image to be synthesized. The 129 channels consists of a depth channel and one channel per byte in the SIFT descriptor (128 bytes). The input is constructed by finding and projecting the visible points in the point cloud to the pose to render and then filling the input tensor at the pixel coordinates where there is a projected 3D point. The output of the network is the synthesized image at a given pose. For details of the method (e.g., training and evaluation), we refer the reader to [144]. While each rendering technique alone is sufficient in certain cases, combining the two rendering methods utilizes the information from different scene models and results in the best performance in our experiment (cf. Sect. 5.4).

3.2 Matching Features with Synthesized views

To extract and match features between the real images I and the rendered images I_r, we choose to use learning-based local features. This is due to the fact that the rendered images usually have large appearance change compared with the real night images. Traditional features, such as SIFT, rely on low level image statistics and are not robust to day-night condition change and rendering artifacts. In particular, we choose to use the

D2-Net feature [17] in our pipeline, which uses a single CNN for joint feature detection and description and achieves state-of-the-art matching performance in challenging conditions.

For the images rendered using the two rendering techniques, we extract and match features between each rendered image and the real image individually. We then directly aggregate the feature matches obtained from both rendered images for the next step. Note that after obtaining the 2D feature matches, we can already verify whether there exists pose errors in the reference poses by checking the matching locations in the rendered and real images (cf. Fig. 3 and Fig. 8 for large and small pose errors respectively): if the real and rendered images are taken from the same pose, the two features $\mathbf{u}_l$ and $\mathbf{u}_l^r$ should be found at identical 2D positions (up to noise in the feature detection stage). Similarly, a large 2D distance $||\mathbf{u}_l - \mathbf{u}_l^r||_2$ is indicative for a significant difference in pose.

3.3 Refining Reference Poses

Given N_f matched features $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ between the real and rendered images, we first back-project the features in the rendered images to $\mathcal{M}$ to get the corresponding 3D points as $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$

$$\mathbf{p}_l^r = \pi^{-1}(\mathbf{u}_l^r, \mathtt{T}, K, D, \mathcal{M}), \tag{2}$$

where $\pi : \mathbf{p} \rightarrow \mathbf{u}$ is the camera projection function and π^{-1} the inverse. K and D are the intrinsics and distortion parameters respectively. In practice, we get the depth map at $\mathtt{T}$ in the process of rendering images from the 3D mesh, and the depth at $\mathbf{u}_l$ can be directly read from the depth map. After finding the 3D points, the refined reference pose $\mathtt{T}^r$ can be computed by solving a nonlinear least-squares problem

$$\mathtt{T}^r = \arg\min_{\mathtt{T}} \sum_{l=1}^{N_f} \|\pi(\mathbf{p}_l^r, \mathtt{T}, K, D) - \mathbf{u}_l\|^2. \tag{3}$$

We minimize (3) over the inliers of a pose obtained by LO-RANSAC. Note that it is possible to additionally refine K and D in the above optimization. This could help correct errors in the intrinsics and distortion parameters but potentially make the optimization problem less stable (e.g., when there are few feature matches). In addition, relatively accurate K and D values are required for rendering reasonable synthesized views for successful feature matching, regardless of whether they are refined in the optimization.

3.4 Uncertainty Quantification

To use the refined pose $\mathtt{T}^r$ for evaluating localization accuracy, it is also important to quantify the uncertainty of the refined pose for meaningful interpretation of the difference between the pose to evaluate and the reference pose. For example, if two poses to be evaluated are both within the uncertainty range of the reference pose, they should be considered equally accurate, regardless of their absolute differences with respect to the reference pose. It is thus a typical practice to consider the uncertainty of the reference poses in the accuracy evaluation of visual localization methods (cf. Sect. 4.1). Next, we introduce two commonly used methods for covariance estimation [146, Ch. 5] and a sampling strategy explored in this paper.

First Order Approximation For the nonlinear least squares problem (3), the covariance of $\mathtt{T}^r$ can be computed by $\mathbf{\Sigma}_{\mathtt{T}^r} = (\sum_{l=1}^{N_f} \mathtt{J}_l^\top \mathbf{\Sigma}_{\mathbf{u}}^{-1} \mathtt{J}_l)^{-1}$, where $\mathtt{J}_l = \partial \mathbf{u}_l / \partial \mathtt{T}$ is the Jacobian (evaluated at $\mathtt{T}^r$) of the lth landmark observation with respect to the camera pose [2], and $\mathbf{\Sigma}_{\mathbf{u}}$ is the covariance of the landmark observation $\mathbf{u}$. $\mathbf{\Sigma}_{\mathbf{u}}$ is usually assumed to be a diagonal matrix and the same for each observation. Then the uncertainty of the camera position and rotation, denoted as δ^c and δ^r, can be calculated from positions and rotations sampled from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathtt{T}^r})$:

$$\delta^c = \text{median}(\{|\delta \mathbf{c}_n|\}_{n=1}^{N_s}) \quad \delta^r = \text{median}(\{|\delta \mathbf{r}_n|\}_{n=1}^{N_s}), \tag{4}$$

where $|\cdot|$ denotes 2-norm. $\delta \mathbf{c}_n$ and $\delta \mathbf{r}_n$ are the sampled position and rotation respectively. We slightly abuse the notation and denote the number of samples as N_s in this Section.

Monte Carlo Estimation An alternative method is to use Monte Carlo method. In particular, from the 3D points $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$ and refined pose $\mathtt{T}^r$, we first calculate the ideal (i.e., no noise) observations $\{\bar{\mathbf{u}}_l\}_{l=1}^{N_f}$, where $\bar{\mathbf{u}}_l = \pi(\mathbf{p}_l^r, \mathtt{T}^r, K, D)$. A set of noisy measurements $\{\tilde{\mathbf{u}}_l\}_{l=1}^{N_f}$ can be simulated by $\tilde{\mathbf{u}}_l = \bar{\mathbf{u}}_l + \mathbf{n}_{\mathbf{u}}$, where $\mathbf{n}_{\mathbf{u}} \in \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{u}})$. We then can compute a camera pose estimate $\mathtt{T}^m$ using the method in Sect. 3.3 with the simulated feature locations $\{\tilde{\mathbf{u}}_l\}_{l=1}^{N_f}$ instead of the actual ones $\{\mathbf{u}_l\}_{l=1}^{N_f}$. After repeating the above process N_s times, the uncertainty of camera position σ_m^c and rotation σ_m^r can be computed as

$$\delta_m^c = \text{median}(\{\epsilon_n^c\}_{n=1}^{N_s}) \quad \delta_m^r = \text{median}(\{\epsilon_n^r\}_{n=1}^{N_s}), \tag{5}$$

[2] In solving (3), iterative methods (e.g., Gauss-Newton) are usually used, where $\mathtt{J}_l$ is actually the Jacobian with respect to a small perturbation around the current estimate.

where $\epsilon_n^c, \epsilon_n^r = \mathtt{T}^r \boxminus \mathtt{T}_n^m$ (cf. (7)).

Sampling Uncertainty We also explore a sampling strategy for uncertainty estimation in this paper. In particular, for a sampling ratio k (e.g., 50%), we first randomly sample from all the 2D-3D matches (i.e., all the inliers used in (3)). We then apply LO-RANSAC to the sampled subset and solve the nonlinear optimization problem (3) using the inliers returned by LO-RANSAC to get a pose $\mathtt{T}^s$. The sampling and solving process is repeated multiple times (typically 50 times in our experiment), resulting in multiple pose estimates $\{\mathtt{T}_n^s\}_{n=1}^{N_s}$. The sampling uncertainty for the camera position s_k^c and rotation s_k^r are calculated as

$$s_k^c = \text{median}(\{\epsilon_n^t\}_{n=1}^{N_s}) \quad s_k^r = \text{median}(\{\epsilon_n^r\}_{n=1}^{N_s}), \tag{6}$$

where $\epsilon_n^c, \epsilon_n^r = \mathtt{T}^r \boxminus \mathtt{T}_n^s$ (cf. (7)). We calculate the sampling uncertainties for different sampling ratios.

We would like to highlight the difference between the uncertainties computed from the above methods and the absolute uncertainties. The absolute uncertainties reflect the differences between the refined poses and the *unknown* ground truth, which cannot be calculated directly. The above uncertainties, on the other hand, evaluate the variance (by approximation or computing statistics from randomized sampling) with respect to the refined poses, which are essentially the local minima in the optimization problem (3). Therefore, these uncertainties tend to be smaller than the actual uncertainties, since the local minima can hardly be the actual ground truth poses. We will further discuss how to consider these uncertainties in the context of evaluation in Sect. 4.1.

3.5 Discussion

The method proposed in this section essentially estimates more accurate poses from some potentially inaccurate initial estimates. Yet, it can not only be used to verify and refine existing reference poses, but also to easily extend existing visual localization datasets. For example, to add more images to an existing localization dataset, one only needs to provide coarse initial poses for these images, which can be obtained by, for example, manually selecting the most similar images. This is useful especially for images with large appearance difference compared with the localization database (e.g., adding nighttime images to a localization database constructed from daytime images), where accurate poses cannot be reliably estimated using SfM directly.

4 Metrics for Localization Accuracy

The reference poses generated using SfM or our method are inherently subject to inaccuracies, which complicates the evaluation process. For example, the difference between the reference pose and a pose to evaluate is no longer a meaningful metric if the actual error (i.e., the difference between the pose to evaluate and the unknown ground truth) is comparable to the uncertainty in the reference pose. Therefore, it is a common practice to set certain thresholds for the reference poses based on their uncertainties, and measure whether the poses to evaluate lie within those thresholds. Unfortunately, quantifying the uncertainties in the reference poses is a highly non-trivial task in itself. The actual uncertainties depend on various factors, such as the depth of the scene and the accuracy of the local features. In this section, we first discuss several performance metrics based on directly considering the uncertainties in pose space. We then discuss a performance metric based on the re-projection of the scene points, which removes the necessity of directly quantifying the pose uncertainty.

4.1 Direct Pose Uncertainty-Based Measures

Direct pose uncertainty-based measures analyze the position and rotation error between the reference and estimated poses. Typically, given a reference pose $\mathtt{T}$ and a pose to evaluate $\hat{\mathtt{T}}$, the position and rotation error $\epsilon^t, \epsilon^r = \mathtt{T} \boxminus \hat{\mathtt{T}}$ are computed as [9]:

$$\epsilon^t = \|\mathbf{c} - \hat{\mathbf{c}}\|_2, \quad \epsilon^r = \arccos(\frac{1}{2}(\text{trace}(\mathtt{R}^{-1}\hat{\mathtt{R}}) - 1)). \tag{7}$$

To account for the uncertainties in the reference poses, we can either use a set of fixed thresholds for all the images in a dataset or define thresholds for each image individually.

Fixed error thresholds. We can define a set of N_e increasing error thresholds $E^{\text{fixed}} = \{\mathbf{e}_j^{\text{pose}}\}_{j=1}^{N_e}$, where $\mathbf{e}_j = (c_j, r_j)$ contains both position and orientation thresholds. These thresholds apply to all the images in a dataset. A pose is said to be below a threshold $\mathbf{e}_j$ if $\epsilon^c < c_j$ and $\epsilon^r < r_j$. The overall localization accuracy is the percentages of images that are localized within these thresholds, and higher values indicate better performance. For example, the error thresholds for Aachen night time images on visuallocalization.net are 0.5/1.0/5.0 m and 2.0/5.0/10.0 deg [3] , and the local-

[3] Note that these thresholds were used *before* we updated the reference poses with the poses generated using our method. A set of tighter thresholds, as in Table 1 and 2, are used now since the new reference poses are more accurate.

ization accuracy is reported as three percentages corresponding to the these categories.

Per image error thresholds Using the same thresholds for all the images in a dataset, however, has limitations. The uncertainties are image-dependent if, as in our case, the poses are calculated by minimizing the reprojection errors of 2D-3D correspondences. The position uncertainty is lower for images observing landmarks that are closer to the camera. Ideally, these uncertainties should be taken into consideration to choose the error thresholds *per image*. As shown in Sect. 3.4, there are different ways of computing the pose uncertainty for each image, which can be used as per image error thresholds. For the first order and Monte Carlo uncertainties, we can simply use (4) and (5) as thresholds. In terms of sampling uncertainties, we can choose a set of sampling uncertainties $E_i^{\text{sample}} = \{\mathbf{s}_k\}_{k=k_1,k_2,\ldots}$ as error thresholds, where $\mathbf{s}_k = \{s_k^c, s_k^r\}$ is the sampling uncertainty with sampling ratio k. For example, in our experiment, we use a set of thresholds calculated from sampling ratios of 50%, 30% and 10% respectively. However, as discussed before, the uncertainties in Sect. 3.4 tend to be lower than the (unknown) absolute uncertainties. Therefore, using these uncertainties as error thresholds tends to under-estimate the accuracy of localization algorithms (cf. Sect. 5.5).

4.2 Indirect Pose Uncertainty-Based Measures

To avoid the need to consider the uncertainties in 6 DoF poses (which is non-trivial as seen before), we follow the literature on object pose estimation and measure pose accuracy based on reprojections [147]. More precisely, we measure the difference between the reprojection of a set of 3D points in the reference and estimated poses. Intuitively, perturbations to the camera pose will result in the changes of the reprojected 2D locations of 3D points. Therefore, we can define certain thresholds around the reprojection of the 3D points as an *indirect* measure of the pose uncertainty. A key advantage of this approach is that the error thresholds can be defined on the image plane. While we use the same thresholds for all the images, this actually results in per-image uncertainty thresholds in pose space: the same change in reprojection error will typically result in a position error that increases with increasing distance of the camera to the scene. Formally, we define the following metric:

Maximum reprojection difference. The maximum distance between the projected points in the reference pose $\mathtt{T}_i^r$ and the estimated pose $\hat{\mathtt{T}}_i$ is used to measure the localization error:

$$r_i^\infty = \max_{l \in [1, N_f^i]} \|\pi(\mathbf{p}_l^r, \mathtt{T}_i^r) - \pi(\mathbf{p}_l^r, \hat{\mathtt{T}}_i)\|_2, \tag{8}$$

where the intrinsics and distortion parameters are omitted for simplicity. Similar to the pose error, a set of reprojection thresholds $E^{\text{rep}} = \{e_j^{\text{rep}}\}_{j=1}^{N_e}$ are selected, and the percentages of the images with r_i^∞ lower than these thresholds are used to indicate the overall accuracy on the dataset. We slightly abuse N_e here to denote the number of error thresholds in general.

5 Experimental Evaluation

To demonstrate the value of the proposed method, we first use our method to analyze the reference poses of the nighttime query images in the Aachen Day-Night dataset (Sect. 5.2). Then, we extend the dataset with new nighttime query images and generate the corresponding reference poses using our method (Sect. 5.3). We also compare our method against baseline methods of directly matching features (SIFT and D2-Net) and computing poses via SfM models. To understand the impact of the different parameters in our method, we perform an extensive ablation study regarding different learned features, different rendering techniques, and the stability of our reference poses (Sect. 5.4). Finally, we evaluate state-of-the-art localization methods on both the original and the extended Aachen Day-Night datasets based on the performance metrics discussed in Sect. 4 (Sect. 5.5).

In this paper, we focus on the Aachen Day-Night dataset [8, 148]. This is motivated by our observation that the reference poses for the nighttime images are the least accurate reference poses among the three datasets from [148]. At the same time, the dataset is becoming increasingly popular in the community, e.g., [17, 18, 46, 47, 94, 98, 101, 102, 111, 116, 122, 149, 150] have already been evaluated on the dataset. However, our approach is generally applicable and can be applied to other datasets as well. Note that we only consider the nighttime query images in this paper as SfM already provides accurate reference poses for the daytime queries of the Aachen Day-Night dataset. In contrast, the authors of the dataset reported in [9] that localizing the nighttime images directly against the SfM model built from daytime images resulted in highly inaccurate poses due to a lack of sufficiently many SIFT feature matches. This is due to the strong illumination changes between the nighttime and daytime images and the limited repeatability of the SIFT descriptor to such strong changes [151]. With insufficient feature matches, pose

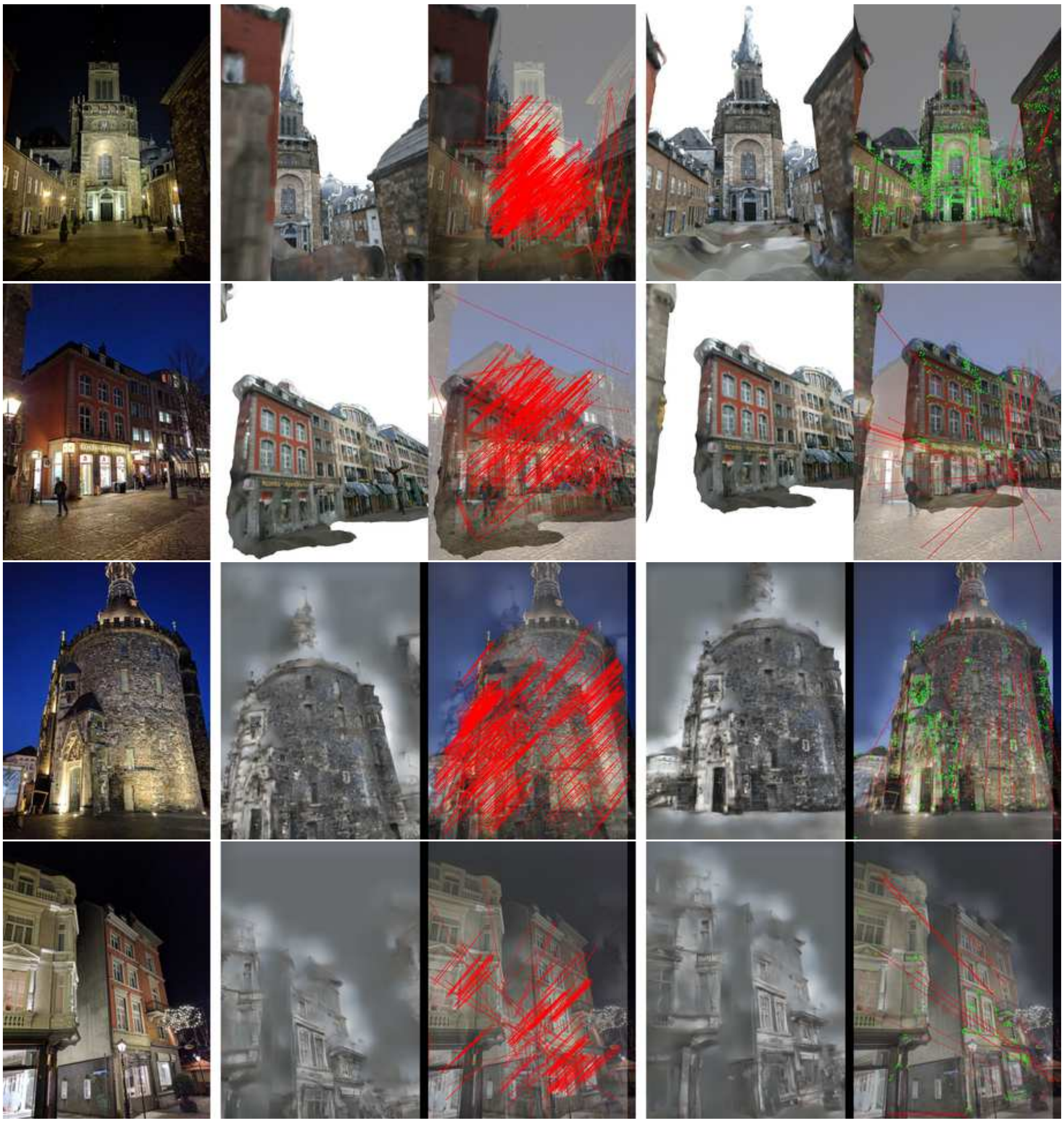


Fig. 3: Comparison of images rendered from the original and refined (ours) reference poses of the nighttime images in Aachen Day-Night dataset. **First column**: nighttime images; **Second column**: images rendered from the existing reference poses, overlay of the rendering and the image together with D2-Net matches between the two; **Third column**: images rendered from our refined poses and the corresponding overlays with D2-Net matches. The top two rows render a Multi-View Stereo (MVS) mesh and the bottom two use Structure-from-Motion inversion [144] (invSfM). The colored lines visualize D2-Net feature matches. Green is used to indicate that the 2D location difference between a feature in the real image and its match in the rendered image is below 20 pixel.

estimation using SIFT would either fail or have erroneous results due to ill-conditioned configurations (e.g., the feature matches may concentrate in a small, well illuminated region in the image, resulting in an inaccurate pose estimate). We observe similar failure cases when using SIFT to add new nighttime images to the Aachen Day-Night dataset (see Fig. 6 and the discussion in Sect. 5.3 for examples).

5.1 Experimental Setup and Data Acquisition

Additional data capture. To extend the Aachen Day-Night dataset, we captured another 119 nighttime images and 119 daytime images with the camera of a Nexus 5X smart phone in July 2017. The nighttime and daytime images form pairs of photos taken from very similar poses. Registering the daytime images against the reference SfM model provided by the Aachen Day-Night dataset then yields initial pose estimates for the new nighttime queries. Both the original and the newly captured nighttime images have a resolution of 1600 × 1200 pixels (the diagonal is thus of 2000 pixels).

Scene model generation. Our approach to refine camera poses requires an underlying 3D scene model. The Aachen Day-Night dataset provides a *reference SfM model* consisting of 4,328 database images and 1.65M 3D points triangulated from 10.55M SIFT features [148]. This publicly available reference model is a sub-model of a larger *base SfM model* that was reconstructed using COLMAP [29]. This base model also contains images from a set of videos as well as the daytime queries, resulting in a SfM model with 7,604 images and 2.43M 3D points triangulated from 17.75M features. This model was registered against the original Aachen SfM model from [8] to recover the scale of the scene. The reference model was obtained by removing the sequences and query images from the base model.

We started from the base model and created an *extended SfM model*. We registered the additional daytime images and an additional image sequence[4] against the base model while keeping the poses of the base model images fixed. The resulting model contains 12,916 images and 3,90M 3D points triangulated from 32.19M SIFT features. We used this extended base model when creating our new reference poses.

We removed all query images and the newly added sequence images from the extended base model to create an *extended reference SfM model* consisting of 6,697 images and 2.32M points triangulated from 15.93M SIFT features. This model will be used to benchmark localization algorithms on our extended Aachen Day-Night dataset. We will make this new reference model publicly available, but will withhold the base models and the reference poses for the query images. Instead, we will provide an evaluation service on visuallocalization.net. The motivation behind publishing this smaller dataset is to make sure that the reference poses were computed from additional data not available to localization algorithms. The inclusion of the original sequences is necessary as some of the newly added nighttime queries depict places not covered in the original reference model.

In addition to the extended models, we also created a colored 3D mesh of the scene. We used COLMAP's Multi-View Stereo pipeline [19] to obtain a dense point cloud. Screened Poisson surface reconstruction [152] of the point cloud then yields a colored mesh.

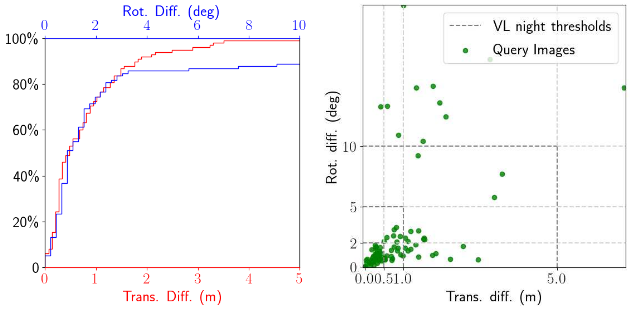


Fig. 4: Differences between the original reference poses and the refined reference poses (our method). **Left**: Cumulative distribution of position and rotation differences. **Right**: Distribution of the position and rotational differences. The position and rotation thresholds (0.5/1.0/5.0 m, 2/5/10 deg) used in [148] and visuallocalization.net (VL) are also shown for reference.

Rendering. Our method requires rendering the scene from estimated poses. For each pose, we generate two renderings: (1) we render the MVS *mesh*, (2) we use the SfM inversion approach (*invSfM*) from [144] to recover an image directly from a rendering of the extended base model. We use our own implementation of invSfM. Note that we only use the CoarseNet stage and skip the VisibNet and RefineNet. We use the MVS mesh to determine which points are visible instead of VisibNet. While skipping RefineNet reduces image quality, we found the results to be of sufficient quality. Moreover, as shown in [144], RefineNet mostly improves the color of the rendered image with respect to the CoarseNet. Since D2-Net feature used in our method is quite robust to such changes in the view condition, we do not expect skipping RefineNet would have a large impact on the performance of our method. Fig. 3 shows example renderings obtained from the mesh and invSfM.

Implementation details. If not mentioned otherwise, we extract D2-Net features [17] from both rendered images. The refinement process is repeated for 5 iterations. We use single scale features since the initial pose estimates are accurate enough such that multi-scale processing is not required. To determine whether our refinement succeeded, we only accept the refined pose when there are more than 10 effective inliers[5] found by LO-RANSAC [42,153] from the input 2D-3D matches, using the P3P solver from [39]. More precisely, we subdivide each image into a 50 × 50 grid and count at most one inlier per cell. The cell size and the inlier threshold are determined experimentally.

[4] Using one of the original videos and extracting images at a higher frame rate.

[5] The effective inlier count takes the spatial distribution of the matches in the image into account. It has been shown to be a better measure than the raw inlier count [25].

Table 1: Evaluation of state-of-the-art localization methods in the original Aachen nighttime images. We evaluate results submitted by the authors to visuallocalization.net on both the original and our refined poses. We compare the methods based on the Pose Error, i.e., the percentage of queries localized within fixed error thresholds of the reference poses. As can be seen, our more accurate reference poses yield a better measure of pose accuracy. For our poses, we also report results for two additional metric: the percentage of queries localized within sampling-based thresholds (Sampling) of the reference poses (cf. Sect. 4.1) and the percentage of queries with maximum reprojection errors within given error thresholds in pixels (Reprojection Diff.) (cf. Sect. 4.2).

	Original Poses	**Refined Poses**		
	Pose Error	Pose Error	Sampling	Reprojection Diff.
	0.25m,2°/0.5m,5°/5m,10°	0.25m,2°/0.5m,5°/5m,10°	(50%/30%/10%)	(10/20/50/100 px)
Active Search v1.1 [148]	13.3/27.6/56.1	39.8/49.0/64.3	2.0/4.1/11.2	28.6/39.8/52.0/62.2
D2-Net [17]	18.4/45.9/88.8	74.5/86.7/100.0	7.1/13.3/35.7	46.9/68.4/89.8/98.0
DELF [104]	14.3/40.8/85.7	54.1/75.5/96.9	4.1/5.1/14.3	28.6/56.1/78.6/88.8
DenseVLAD [16] + D2-Net [17]	14.3/39.8/74.5	66.3/75.5/84.7	7.1/8.2/24.5	45.9/65.3/77.6/82.7
Hierarchical Localization [47]	17.3/43.9/76.5	68.4/77.6/88.8	7.1/9.2/24.5	41.8/65.3/78.6/85.7
NetVLAD [78] + D2-Net [154]	17.3/43.9/85.7	84.7/90.8/96.9	8.2/11.2/40.8	51.0/75.5/92.9/95.9
R2D2 V2 20K [18]	19.4/48.0/88.8	75.5/90.8/100.0	8.2/14.3/36.7	51.0/70.4/92.9/95.9

5.2 Refining the Original Aachen Nighttime Poses

In a first experiment, we analyze the accuracy of the reference poses for the 98 original nighttime queries of the Aachen Day-Night dataset. We show that the original reference poses are inaccurate and that our refinement approach considerably improves the pose accuracy.

Our approach used the original poses for initialization. For 3 out of the 98 images, our method failed to find sufficiently many 2D-3D matches, mostly due to an incomplete mesh (see Sect. 5.4). For the failure cases, we simply kept the existing reference poses.

Qualitative evaluation. Fig. 3 visually compares the original reference poses with our refined poses. As can be seen, the existing reference poses, obtained from manual annotated 2D-3D matches, can be rather inaccurate. In contrast, our method generates reference poses such that the rendering from the refined pose is visually consistent with the actual image. Thus, features matching between the real and rendered images are found at the same positions (up to noise), as can be see from the (short) green lines. Fig. 3 shows selected examples where the original reference poses were rather inaccurate. Visual comparison between the original and our refined poses showed that our approach consistently produced more accurate poses for all nighttime queries.

It is also worth noting that D2-Net features can provide robust matches even though the rendered images (using a model reconstructed from daytime imagery) are visually very different from the actual images and contain non-trivial rendering artifacts.

Quantitative evaluation. To quantify the differences between the original and our reference poses, we computed the differences in camera position and orientation (see (7)). Fig. 4 shows the results of this comparison. It can be seen that there exists a non-trivial discrepancy between the original and refined reference poses.

[148] measures localization accuracy by the percentage of nighttime query poses estimated within (0.5 m, 2 deg), (1 m, 5 deg), and (5 m, 10 deg) of the reference poses. These thresholds are also shown in Fig. 4. As can be seen, the differences between the original and refined poses fall outside of the largest error threshold for 11 images ($\sim$ 11.2% of all the nighttime queries). Interestingly, the best results reported on visuallocalization.net register 88.8% of the nighttime queries within 5 m and 10 deg. Thus, state-of-the-art methods might actually be more accurate than the reference poses.

Finally, Table 1 evaluates several state-of-the-art localization methods using the existing and refined reference poses. As can be seen, the accuracy of the localization methods is indeed (significantly) under-estimated by the existing reference poses of the nighttime images in the Aachen Day-Night dataset. In contrast, our reference poses allow us to measure localization performance more accurately. Table 1 also provides results for additional evaluation measures for our new reference poses, which will be discussed in Sect. 5.5. Note that the improvement reported here is particular to the nighttime images in the Aachen Day-Night dataset. The result on a different dataset will depend on the quality of the existing reference poses in the dataset.

Summary. Our results clearly show that our new reference poses are more accurate than the original poses. We will integrate our new poses in the visuallocalization.net online benchmark, allowing us to easily update all results on the website.

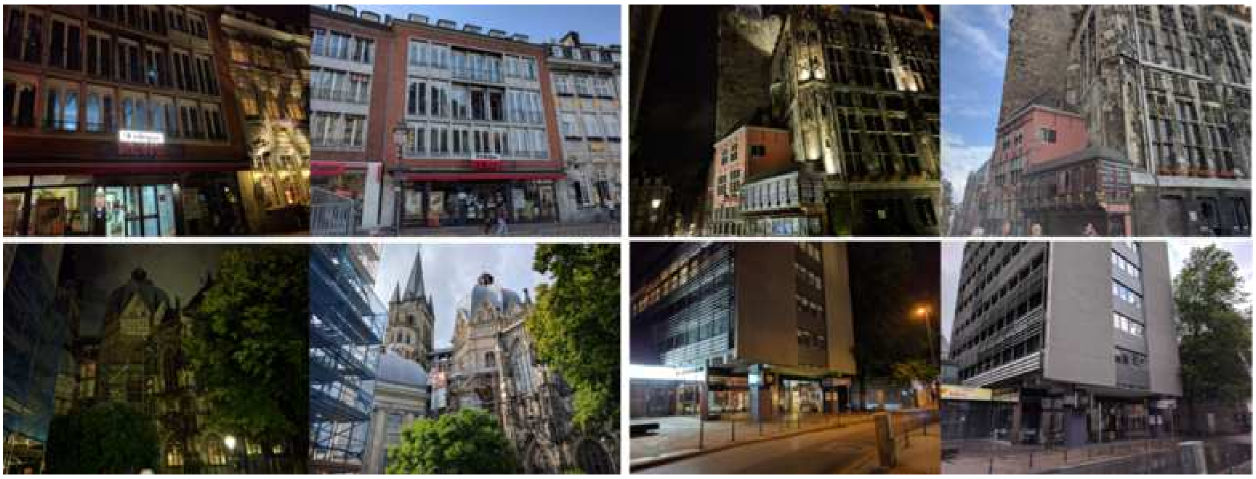


Fig. 5: Pairs of day-night images taken from similar poses. We obtain reference poses for the daytime images via SfM. The resulting poses are used to initialize our approach for generating poses for the nighttime images. For the SIFT and D2-Net registration baselines, an additional 20 daytime images that overlap with these images are selected from the base model for the daytime image in each pair.

5.3 Extending the Aachen Day-Night Dataset

Our approach is capable of estimating an accurate pose from a coarse initialization. Besides verifying and refining existing reference poses, our approach can also be used for generating reference poses for new images. In the next experiment, we thus extend the Aachen Day-Night dataset by additional nighttime queries. We compare our reference poses with two registration baselines using SIFT and D2-Net features, respectively.

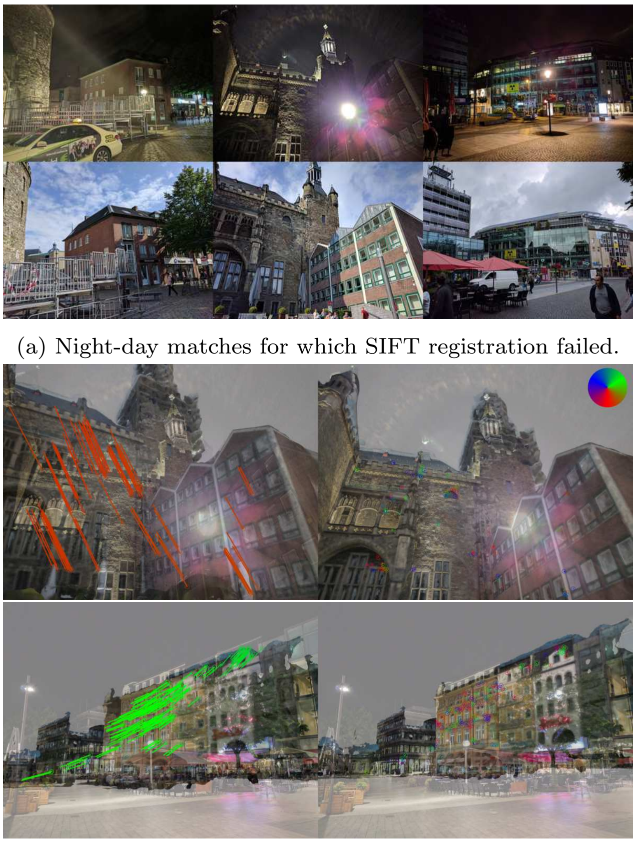
Reference pose generation. As shown in Fig. 5, we captured a daytime photo from a similar pose for each the 119 new nighttime images. The poses of these daytime images in the extended base model, obtained via SfM, then provide initial pose estimates for the nighttime queries that are subsequently refined by our approach. We excluded images for which our method resulted in less than 10 effective inliers to avoid unreliable reference poses. This results in reference poses for 93 out of the 119 images.

We compare our method with two baselines using SIFT and D2-Net features, respectively. Both baselines match features between the 93 new nighttime queries and a small set of images in the extended base SfM model. For a nighttime query, this set includes the corresponding daytime image $\mathcal{I}_D$ as well as the 20 images in the extended base model that share the most 3D points with $\mathcal{I}_D$. 2D-2D matches between the nighttime image and the daytime photos in the set then yield a set of 2D-3D correspondences based on the 3D points visible in the latter. COLMAP's image registration pipeline was then used to obtain the camera pose based on these matches. Note that for D2-Net features, we re-triangulated the extended base 3D model before day-night feature matching.

Robustness. Both the D2-Net baseline and our method are able to consistently estimate poses for challenging images for which the SIFT baseline fails. Fig. 6 shows such failure cases of SIFT. In each of the shown cases, there is a strong light source in the scene, causing significant appearance differences between the day and nighttime images. SIFT is not able to deal with these strong changes. In contrast, our method, as well as the D2-Net baseline, which relies on high level learned features, are able to handle these cases (cf. Fig. 6b).

[148] reported that the reference poses obtained via SfM and SIFT were unreliable. Interestingly, we observe the opposite for many images in our experiments. We attribute this to the inclusion of the corresponding daytime images: as shown in [16], SIFT features better handle day-night changes under small viewpoint changes. Note that daytime images taken from very similar poses are not available for the original nighttime queries.

Quantitative evaluation. Excluding the failure cases, we computed the pose differences between our method and two baselines. The results of this comparison are shown in Fig. 7. Interestingly, the poses from our method and the SIFT registration are very consistent. For the majority of the images, the pose difference is below 0.2 m and 0.5 deg. In contrast, we observe much larger difference between our poses and the D2-Net registration baseline. As there is no external reference poses that can be used to calculate the absolute pose accuracy, we resort to visual inspection based on the renderings.



(a) Night-day matches for which SIFT registration failed.

(b) Refinement process of our method (1st iteration).

Fig. 6: Typical failure cases of the SIFT registration baseline. **Top**: nighttime images where SIFT registration failed and the corresponding daytime images; **Bottom**: Visualization of the first iteration of our method (left: initial pose; right: refined pose). The differences between D2-Net features and the projection of the matching 3D points are color coded according to the direction in the image plane (cf. legend in the top-right).

Visual inspection. Fig. 8 analyses example poses obtained by the D2-Net baseline. Besides overlaying the real and rendered images, we also show D2-Net features matches between the two. For each match, we compute the 2D offset between the feature positions in the real and the rendered view. Following [12], we color-code the features based on the directions of these 2D offsets. As argued in [12], these directions should be randomly distributed for accurate pose estimates. Patterns of similar direction in the same region of an image indicate a shift between the two images and thus pose errors.

The D2-Net poses in Fig. 8 are visually more accurate than those in Fig. 3 and Fig. 6b. Still, we observe clear patterns in the distribution of the directions (e.g., the concentration of green color on one side and purple on the other), which indicates inaccuracies in the poses of the D2-Net baseline. We further used one iteration of our method to refine the D2-Net poses. As can be seen in Fig. 8, the refinement improves the distribution of directions. We conclude that our approach is able to provide more accurate than the D2-Net baseline.

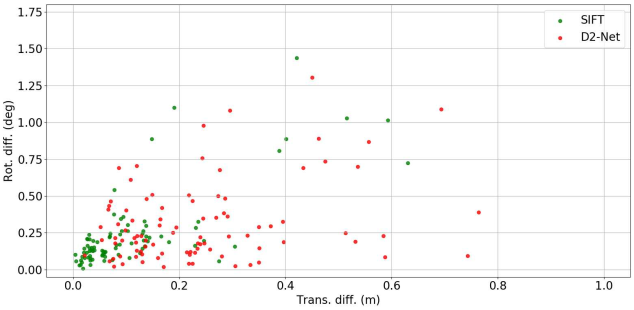


Fig. 7: Distribution of the pose difference between our method and the two registration baselines.

As can be seen from Fig. 7, the pose differences between our approach and the SIFT baseline are significantly smaller than the differences between our approach and D2-Net. Unlike for D2-Net poses, we did not see strong feature direction patterns for the SIFT poses. We therefore omit the corresponding visualizations. We observe that if the SIFT baseline is able to estimate a pose it is usually visually similar to the pose obtained with our approach (cf. Fig. 9a). There are images where the poses from our method seem to be visually more accurate than the SIFT registration and vice versa (shown in Fig. 9b and 9c, respectively). Yet, overall there are only 7 out of the 93 new nighttime queries for which we consider the SIFT poses to be visually more accurate than the poses provided by our method. For these images, we use the SIFT poses as reference poses. At the same time, SIFT failed to provide poses for 5 of the nighttime images due to a lack of sufficient matches.

Discussion and summary. It is interesting to see that SIFT poses are not necessarily more accurate than our poses. SIFT features are much more accurately localized in images than D2-Net features [17]. Thus, one might have expected that a few accurately localized SIFT matches are better than many less accurately localized D2-Net matches. Yet, finding more matches with D2-Net between the renderings and the real images seems to compensate for the inaccuracy of the D2-Net feature detections.

For the newly acquired nighttime images, we observe that our approach performs similar to SIFT in terms of accuracy. In this case, SIFT benefits from daytime images taken from similar viewpoints. As evident from the failure cases of SIFT on both the original and new queries, our approach is more robust than the SIFT

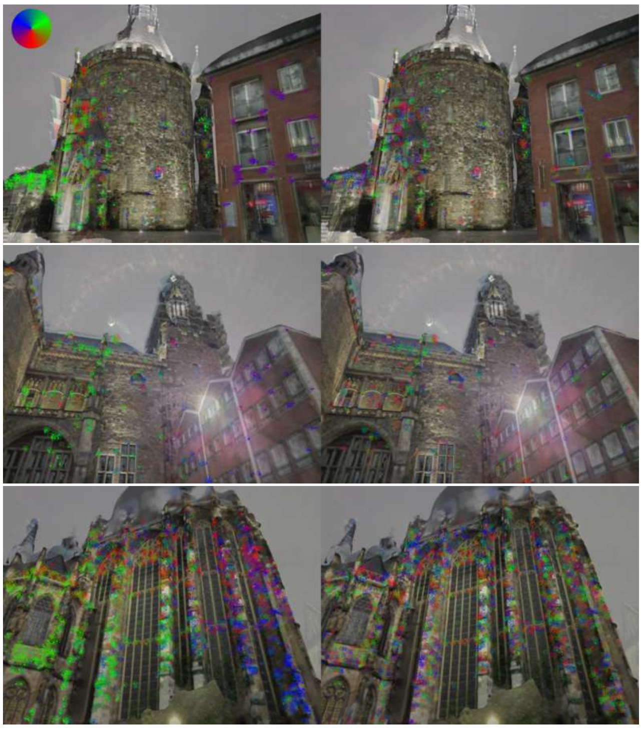
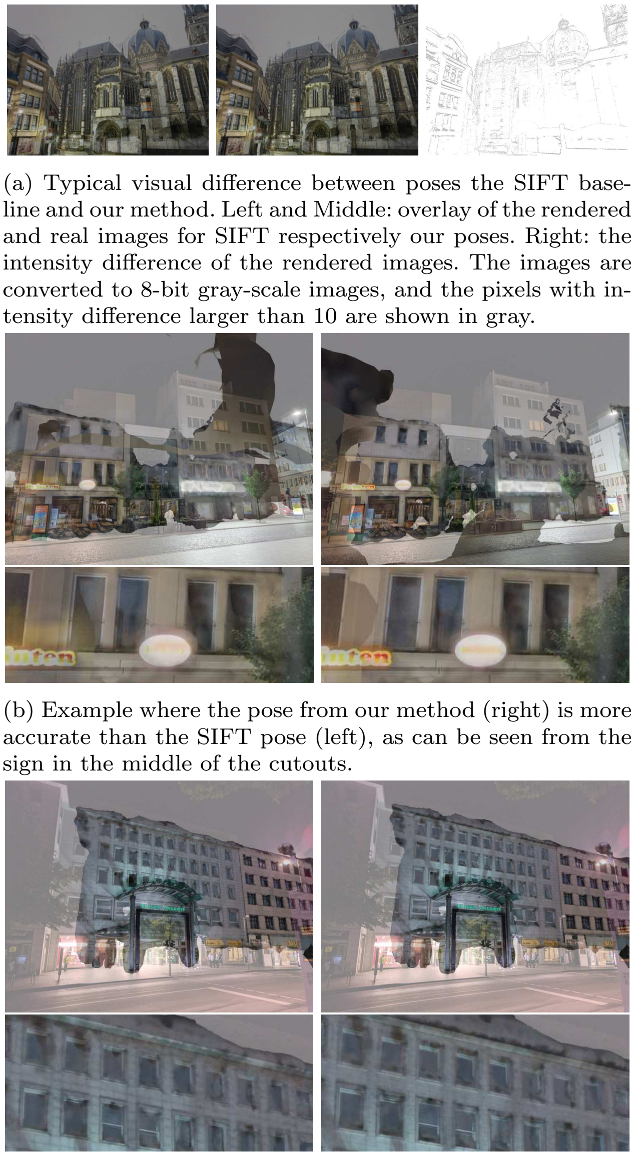


Fig. 8: Comparing the D2-Net baseline against our refinement. **Left**: overlay of real photos and images rendered with D2-Net poses. D2-Net features in the rendered images are connected to the matching locations in the real images (circles), and the color indicates the direction of the feature location differences in the two images (see legend in the top-left). **Right**: corresponding visualization using poses obtained by one iteration of our method (initialized with D2-Net poses). The patterns of the feature directions in the left images indicate the inaccuracy in the poses from D2-Net registration, which are corrected with our method (right images).

baseline. As a result, our approach is better suited to for reference pose generation for datasets that benchmark long-term visual localization algorithms.

Compared with the D2-Net baseline, the poses resulting from our method are more accurate. The main difference between the D2-Net baseline and our approach is the use of rendered images. The results thus validate our choice to iteratively render the scene from the current pose estimate and match features against the rendering. Moreover, as seen from the analysis of the D2-Net baseline, the ability of our method to verify and refine existing poses is also valuable when it is combined with other approaches.



(a) Typical visual difference between poses the SIFT baseline and our method. Left and Middle: overlay of the rendered and real images for SIFT respectively our poses. Right: the intensity difference of the rendered images. The images are converted to 8-bit gray-scale images, and the pixels with intensity difference larger than 10 are shown in gray.

(b) Example where the pose from our method (right) is more accurate than the SIFT pose (left), as can be seen from the sign in the middle of the cutouts.

(c) Example where the SIFT pose (left) is more accurate than our method (right), as can be seen from the windows and the edge of the roof in the cutouts.

Fig. 9: Visual comparison of images rendered from the poses obtained by our method and the SIFT baseline.

5.4 Ablation Study

Next, we present ablation studies to analyze our proposed approach. We first obtain an estimate for the stability of our reference poses. Next, we determine the impact of using different features and rendering techniques, which are the two key ingredients in our method. Finally, we show failure cases of our method.

Pose stability. To provide a quantitative measure of the uncertainties/stability of the reference poses obtained with our method, we compute the sampling un-

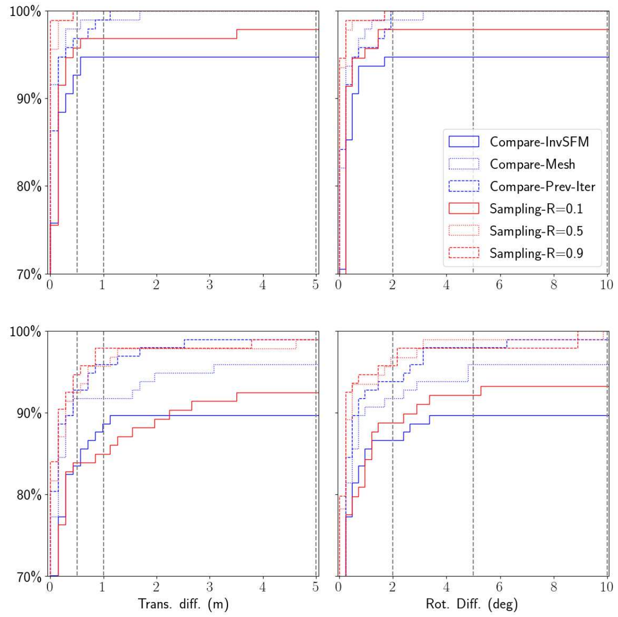


Fig. 10: Different uncertainties for the original (**top**) and additional (**bottom**) Aachen Night images. The vertical dash lines corresponds to the error thresholds proposed in [148] and used by the online benchmark.

certainties as described in Sect. 3.4 for both the original and additional nighttime images: we randomly sample a percentage of 2D-3D matches from the inliers used to estimate the reference poses. This sample is then used to obtain another pose estimate. The differences between these new and our poses provide a measure for the stability of the minima found by our approach.

We used three sampling rates that use 90%, 50%, and 10% of the inliers, respectively. For each rate, we drew 50 random samples and report the median position and orientation differences. In addition, since our method uses different rendering techniques and is an iterative process, we also computed the following for comparison:

- *Compare-InvSfM*: the differences between the refined poses using both types of rendered images and using InvSfM only;
- *Compare-Mesh*: the differences between the refined poses using both types of rendered images and mesh rendering only;
- *Compare-Prev-Iter*: the pose differences between the two last iterations of our refinement process.

The results of our comparisons are shown in Fig. 10. For the original images, more than 90% of the images are below the finest error threshold (0.5 m, 2 deg) of the visual localization benchmark, independently of which sampling rate and rendering is used. For the additional images, the uncertainties are higher. Still, more than 80% of the images fall in that threshold as well. The fact that the uncertainties of the additional images are overall higher than the original images indicates that the newly added images might be more challenging. Regarding the different rendering techniques, images rendered using the MVS mesh seem to provide more information for the final refined poses, as *Compare-Mesh* shows less uncertainty than *Compare-InvSfM*.

While it is difficult to quantify the absolute uncertainties, the uncertainties shown in Fig. 10 indicate that the reference poses generated using our method are at least stable solutions considering the available 2D-3D matches. This can be seen from the fact that even using as little as 10% of the available inlier matches leads to very similar pose estimates for nearly all images.

Different features. Instead of using D2-Net features, we also used SIFT and R2D2 [18] features to obtain matches between the rendered and real images.

Fig. 11 compares the results obtained with different types of features. As can be seen, SIFT failed to find enough matches in most cases for both the original and additional night images. This is not surprising considering SIFT relies on low-level image statistics, which are strongly impacted by imperfections in the MVS model and the invSfM rendering process. In contrast, both D2-Net and R2D2 features were able to find enough matches for most of the original Aachen night images. The success rate for both features drops on the additional Aachen night images, where the D2-Net feature performed better. Plotting the reprojection error (after nonlinear optimization) against the number of effective inliers, we observe a clear trend across different features: D2-Net recovers the most matches, followed by R2D2 and SIFT; while SIFT features were most accurately localized in the images, D2-Net has the largest reprojection errors.

To see how the number of effective inliers and reprojection error translates to the quality of the refined poses, we further computed the sampling uncertainties for D2-Net and R2D2, shown in Fig. 12. We excluded SIFT since it failed for most of the images. It can be seen that the refined poses from D2-Net features are more stable than the R2D2 poses for both the original and additional images.

The results validate our choice of using D2-Net features to match between real and rendered images as they better handle imperfections in the renderings.

Different rendering techniques. The experiments presented so far used both rendering types (using MVS mesh and the invSfM process). Next, we compare using

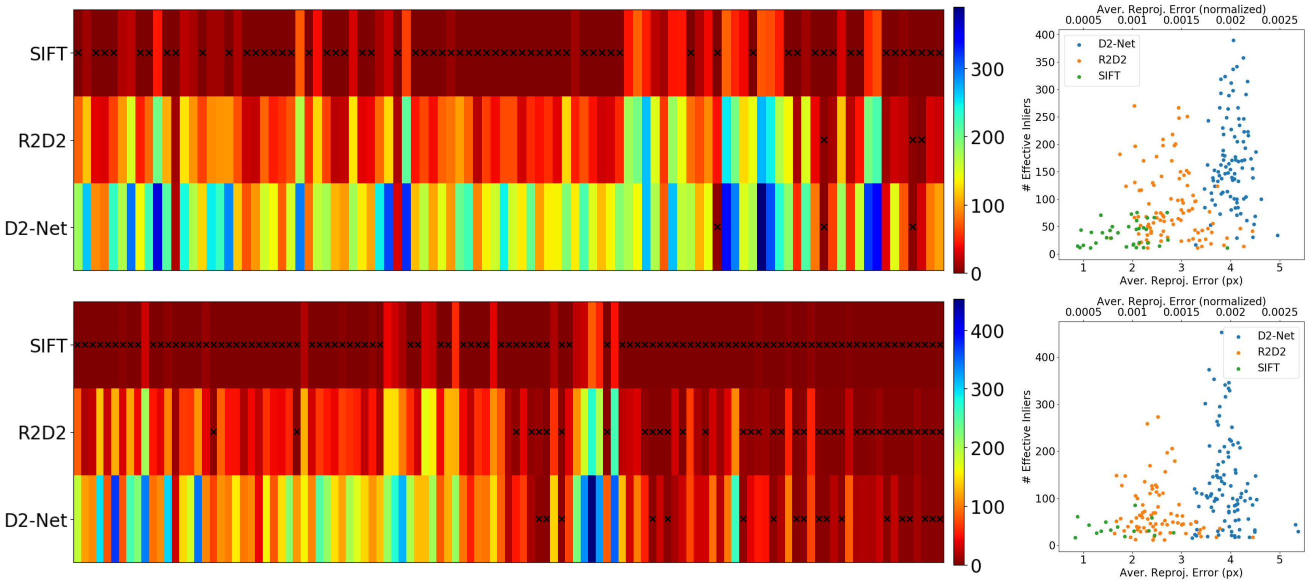


Fig. 11: Effect of using different features in our method. **Left**: the number of effective inliers for each image. Each block along the horizontal axis corresponds to one image. A black cross indicates there are less than 10 effective inliers, i.e., the pose is likely not reliable. **Right**: the number of effective inliers and the mean reprojection error (after nonlinear optimization) for different features. Failure cases (i.e., the black crosses) are excluded. The normalized reprojection error is normalized by the image diagonal length. The top row shows the result for the original Aachen nighttime images, and the bottom for additional images.

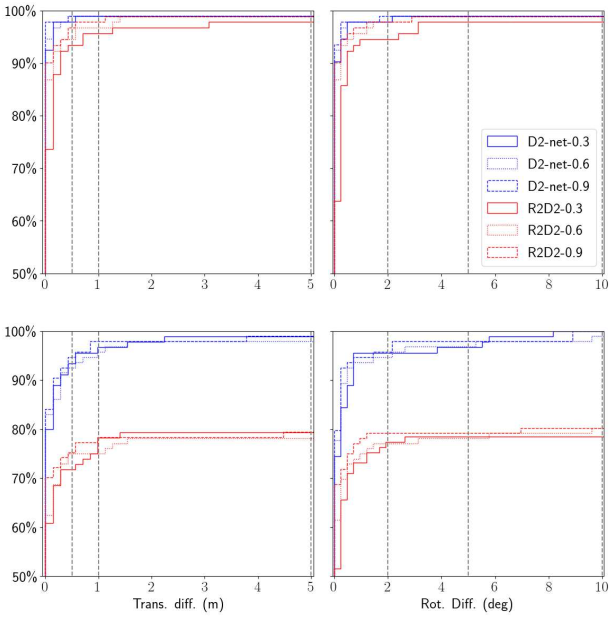
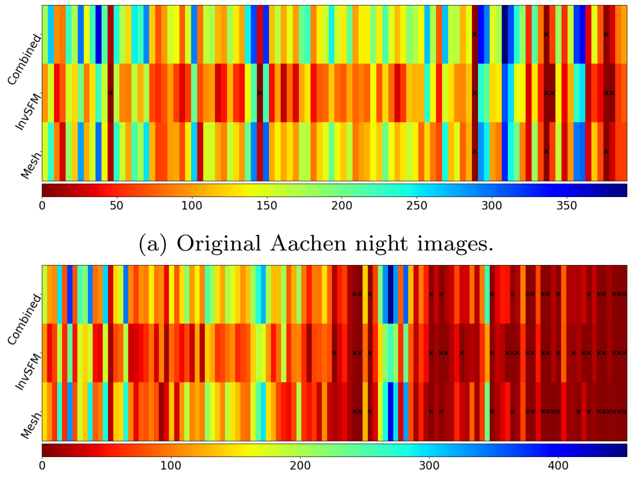


Fig. 12: Sampling uncertainties of the D2-Net and R2D2 poses for the original (**top**) and additional (**bottom**) Aachen night images. Median position and orientation errors over 50 random samples are shown.

both types against using only one of the two using the number of effective inliers.

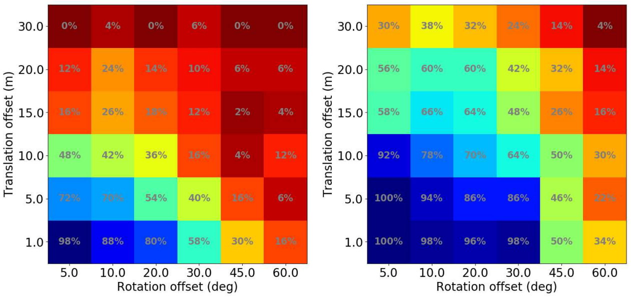


(a) Original Aachen night images.

(b) Additional Aachen night images.

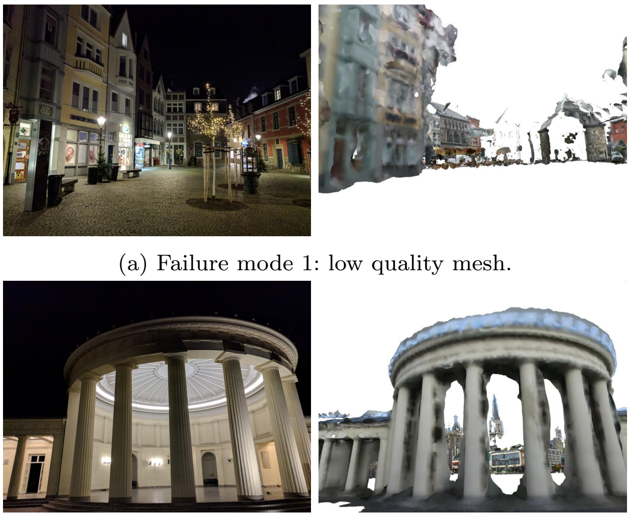
Fig. 13: The number of effective inliers for D2-Net features when different rendering techniques are used. The visualization is the same as Fig. 11.

As can be seen in Fig. 13, using renderings based on the MVS mesh in general resulted in more effective inliers compared to using invSfM for rendering. Accordingly, there are more images where our method could find sufficient effective inliers in the images rendered from mesh. This is also consistent with our results in



(a) After the 1st iteration (b) After the 5th iteration

Fig. 14: Success rates of our method in the presence of different initial error levels on the original Aachen nighttime images using D2-Net features. Each cell corresponds to the success rate of 50 trials of pose refinement using perturbed poses as input. A perturbed pose was generated by adding both random rotation and position offsets of certain magnitudes, as indicated on the x and y axes respectively, to a refined pose. The refinement of a pose is considered successful if the pose error after refinement is within 0.25 m and 1.0 deg.



(a) Failure mode 1: low quality mesh.

(b) Failure mode 2: scene with little texture.

Fig. 15: Typical failure cases of our method. Left: real nighttime images; Right: MVS mesh renderings from the initial pose.

Fig. 10, which show that the poses based only on mesh rendering are more accurate than those obtained using only invSfM. Yet, there are a few cases where mesh rendering fails while invSfm rendering succeeds. The corresponding nighttime images show parts of the model that are only sparsely covered by images and where the MVS reconstruction is thus incomplete. The invSfM process seems to be more stable for such cases.

Combining the 2D-3D matches obtained from both types of renderings increases the number of effective inliers. Note that the effective inlier count selects at most one inlier for each 50 pixels by 50 pixels region in an image. A higher effective inlier count thus indicates that the matches found by the two rendering types are somewhat complimentary as matches are found in different image regions. Moreover, there are a few cases (right part of Fig. 13b) for which using both rendering types is necessary to obtain sufficiently many inliers.

The results validate our choice of using both rendering techniques as they are (partially) complimentary.

Sensitivity to initialization. Our approach requires an initial pose estimate as input to the iterative refinement. Naturally, our approach will fail if the initial pose estimate is not accurate enough. To determine the sensitivity of our approach to the initial pose error, we randomly perturbed the refined poses by translations and rotations of different magnitudes and used the perturbed poses as input to our method. We then measured the sensitivity of our method to the initial pose error by the success rates at different perturbation levels, and the refinement is considered successful if the manually added error can be reduced to under 0.25 m and 1.0 deg, which is half of the smallest fixed error threshold used in our evaluation (cf. Sect. 5.5). The experiment was performed on the original Aachen nighttime images using D2-Net features. As can be seen in Fig. 14b, the success rate of our method is more than 50% for initial pose errors up to 30 deg and 10 m. The success rate is more than 90% for disturbances within 10 deg and 5 m. Moreover, the increase of the success rates from the 1st iteration (Fig. 14a) to the 5th iteration (Fig. 14b) indicates the necessity of performing the refinement iteratively multiple times. Therefore, our method is quite robust to the errors in initial poses and can be potentially used with systems that provide less accurate localization information (e.g., GPS) to get more accurate poses automatically.

Failure cases. Fig. 15 shows examples of two typical failure cases of our method. The first failure mode is when the nighttime image was taken in a part of the scene where the MVS mesh is of low quality, e.g., parts of the surface have not been reconstructed (cf. Fig. 15a). This could be overcome by using a more complete/higher quality mesh of the scene, but might require additional data capture. The second failure mode is caused by weakly textured scenes (cf. Fig. 15b). In the shown example, the rendered image is of reasonable quality visually. However, due to the lack of texture, our method failed to find enough matches between the rendered image and the real night image. Using contour edges as an additional feature type could help avoid this failure mode. However, edges are also typically harder to match than local features. Furthermore, care would need to be taken to handle protruding regions in the MVS model.

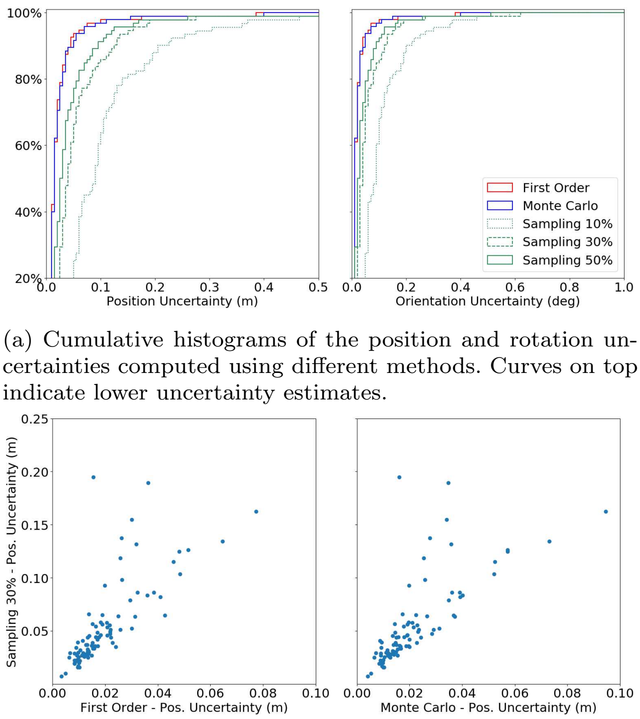
5.5 Evaluation of State-of-the-Art Methods

Table 1 evaluates published state-of-the-art localization methods using our new reference poses for the original nighttime images. The results were obtained by re-evaluating poses submitted to visuallocalization.net.[6] In the following, we present results for state-of-the-art methods on our new extended Aachen Day-Night dataset. Note that the extended dataset uses a larger reference SfM model than the original one and we thus cannot use results from the benchmark website.

Given that D2-Net and R2D2 features achieve state-of-the-art results in Table 1, we use two image retrieval-based approaches based on these features in our evaluation. Both approaches first re-triangulate the reference SfM model with feature matches between the reference images found by D2-Net respectively R2D2. Next, NetVLAD [78] is used to retrieve the 20 most similar reference image for each nighttime query. Feature matches between each query and its retrieved image yield a set of 2D-3D matches via the 3D points visible in the reference images. These 2D-3D matches are used for pose estimation against the reference model inside COLMAP.[7] For R2D2, we provide results for two variants that use at most 20k (R2D2-20k) respectively 40k (R2D2-40k) features per image.

Table 2 shows the results of our experiments using the evaluation measures discussed in Sect. 4. Similarly, Table 1 also shows results for all metrics for our new reference poses. Overall, the accuracy is lower when considering all nighttime queries compared to only focusing on the original night images, independent of the metric used. This indicates the newly added images might be more challenging. In the following, we discuss the results per evaluation metric.

Pose error with fixed thresholds. We first considered the three fixed error thresholds used in [148] and on the benchmark website, i.e., (0.5 m, 2 deg), (1 m, 5 deg), and (5 m, 10 deg). Based on this metric, the performance on the original and extended Aachen dataset seems saturated for certain algorithms (e.g., D2-Net and R2D2). However, these thresholds were originally chosen to take the uncertainties in the original nighttime reference poses into account. As shown in our previous experiments, our new reference poses are significantly more accurate. As such, using rather loose thresholds could lead to an overestimate in the localization accuracy, and we instead chose a set of tighter thresholds i.e., (0.25 m, 2 deg), (0.5 m, 5 deg), and (5 m, 10 deg) for evaluation. Furthermore, as discussed in Sect. 4, using the same thresholds for all images does not take into account that the uncertainty in the pose depends on the distance of the camera to the scene.



(a) Cumulative histograms of the position and rotation uncertainties computed using different methods. Curves on top indicate lower uncertainty estimates.

(b) Correlation between different position uncertainties. **Left**: 30% Sampling uncertainty with respect to the first order uncertainty. **Right**: 30% Sampling uncertainty with respect to the Monte Carlo uncertainty.

Fig. 16: Comparison of the different uncertainty definitions provided in Sect. 3.4 on the original Aachen nighttime images using the refined poses.

Per image error thresholds. The second metric aims at computing error thresholds on the camera pose per image. We first show the results using the sampling uncertainties (6) as error thresholds and then discuss the first order and Monte Carlo uncertainties (4) and (5). For each reference pose, we randomly sampled set containing 10%, 30% and 50% of the inliers of our method. For each sampling percentage, we drew 50 samples and computed the median position and orientation

[6] There results available at visuallocalization.net for methods that outperform the approaches used in Table 1. For our experiments, we limited ourselves to methods that have been published in peer reviewed conferences and journals. Updated results for the other methods will be available on the benchmark website once we update the reference poses.

[7] Based on code available at `https://github.com/tsattler/visuallocalizationbenchmark`.

Table 2: Localization accuracy using different metrics on the extended Aachen Day-Night dataset. We compare the methods based on the Pose Error, i.e., the percentage of queries localized within fixed error thresholds of the reference poses. As can be seen, our more accurate reference poses yield a better measure of pose accuracy. For our poses, we also report results for two additional metric: the percentage of queries localized within sampling-based thresholds (Sampling) of the reference poses (cf. Sect. 4.1) and the percentage of queries with maximum reprojection errors within given error thresholds in pixels (Reprojection Diff.) (cf. Sect. 4.2).

	Original Night Images			**All Night Images**		
	Pose Error	Sampling	Reprojection Diff.	Pose Error	Sampling	Reprojection Diff.
	0.25m,2°/0.5m,5°/5m,10°	(50%/30%/10%)	(10/20/50/100 px)	0.25m,2°/0.5m,5°/5m,10°	(50%/30%/10%)	(10/20/50/100 px)
D2-Net	84.7/90.8/98.0	11.2/19.4/43.9	56.1/80.6/92.9/95.9	71.7/90.6/97.9	6.3/11.0/30.9	36.1/73.8/91.1/96.9
R2D2-20k	82.7/90.8/95.9	7.1/11.2/38.8	54.1/76.5/89.8/93.9	73.3/88.5/96.3	5.2/7.9/29.8	40.8/72.8/91.6/94.8
R2D2-40k	80.6/91.8/98.0	7.1/13.3/44.9	56.1/76.5/92.9/95.9	73.3/88.5/97.9	5.8/8.9/33.0	41.9/73.3/91.6/95.8

difference between the poses obtained from the samples and the reference poses. These median differences were then used as the error thresholds. As can be seen from Table 1 and Table 2, the sampling uncertainties tend to under-estimate the localization performance of the different methods. This is due to the fact that our reference poses are rather stable under using a subset of the inlier matches (cf. Fig. 12). The sampling uncertainties reflect the stability of the local minimum reached in the refinement process, rather than the absolute uncertainties. Thus, this metric should not be used to evaluate localization performance.

As for the first order and Monte Carlo uncertainties, we found that they tend to be lower than the sampling uncertainties. As an example, a comparison of different uncertainties on the original Aachen nighttime images (using the refined pose) is shown in Fig. 16. From Fig. 16a, we can see that the uncertainties estimated by first order and Monte Carlo methods are in general lower than the sampling uncertainties, even for the highest sampling ratio of 50%. We also inspected the correlation between different uncertainties and visualized an example of the position uncertainties in Fig. 16b. It can be seen that different uncertainties show similar trend across images, but the sampling uncertainties tend to be higher than the uncertainties computed from the first order and Monte Carlo methods (notice different axis scales). Therefore, using the first order and Monte Carlo uncertainties as error thresholds will under-estimate the localization performance as well (even worse than the sampling uncertainties) and thus should not be used as accuracy metrics.

Maximum reprojection difference. Our reference poses are obtained by minimizing a reprojection error in image space, rather than an error in camera pose space. Thus, evaluating localization algorithms based on the quality of their reprojections seems a natural metric, especially if these algorithms compute poses by minimizing an image space error.

For each 3D point in the inlier 2D-3D matches of the reference poses, we compute a reprojection difference between the reference and an estimate pose. For each image, we report the maximum difference and we compute the percentages of images that have a maximum reprojection difference below 10, 20, 50 and 100 pixels. Since all nighttime images have a resolution of 1600×1200 pixels, these thresholds correspond to 0.5%, 1%, 2.5%, and 5% of the image diagonal.

Comparing the results with the pose error metric using fixed thresholds, we can see that although the top performing algorithms achieve approximately 90% in the finest pose error category, they only have $70-80\%$ of all the images that were localized within 20 pixel according to the maximum reprojection difference. Even less images are localized within 10 pixels. Since the accuracy of local features are typically below 5 pixel (cf. Fig. 11(right)), this indicates that there is still much room for improvement on our extended version of the Aachen Day-Night dataset. As such, we believe that the maximum reprojection error metric should be the metric of choice for this dataset.

6 Conclusion

In this paper, we have considered the problem of creating reference camera poses for long-term visual localization benchmark datasets. In this setting, classical features often struggle to obtain matches between images taken under strongly differing conditions. At the same time, human annotations are both time-consuming to generate and not necessarily highly accurate. Thus, we have presented an approach for refining reference poses based view synthesis and learned features that allow robust feature matching between real and rendered images. In addition, we have discussed multiple metrics for evaluating localization performance.

The main contribution of this paper is an extensive set of experiments. We have shown that the original nighttime reference poses of the Aachen Day-Night dataset are rather inaccurate. As a result, the localization accuracy of state-of-the-art methods is currently drastically under-estimate. Using our approach, we have created a more accurate set of reference poses. We will integrate these poses into the online evaluation service provided at visuallocalization.net as to provide better evaluations to the community. We also used our approach to create an extended version of the Aachen Day-Night dataset and showed that this dataset offers room for improvement. We will make the dataset available on the benchmark website. Furthermore, we will release the code for our approach as to allow other researchers to more easily build localization benchmarks.

One disadvantage of our approach is its rather slow run-time, taking about 10-20 seconds per iteration for a single image, where most of the time is spend for rendering and especially for the SfM inversion process. This is not an issue when creating reference poses for a benchmark, as these calculations only need to be done once and can be done offline. At the same time, our approach can be used as a post-processing step for any visual localization algorithm. An interesting research question is whether more efficient rendering techniques can be used to improve its run-time to a degree that enables online operation.

Acknowledgements. This work was supported by the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots), the Chalmers AI Research Centre (CHAIR) (VisLocLearn), the National Centre of Competence in Research (NCCR) Robotics, through the Swiss National Science Foundation and the SNSF-ERC Starting Grant, and the European Regional Development Fund under IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468. This article is part of the RICAIP project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857306. We thank Mihai Dusmanu and Martin Humenberger for contributing the D2-Net and R2D2 results, respectively.

References

1. L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler. Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System. In *International Conference on Robotics and Automation (ICRA)*, 2019.
2. H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-Time Image-Based 6-DOF Localization in Large-Scale Environments. In *CVPR*, 2012.
3. R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *ISWC*, 2008.
4. Simon Lynen, Torsten Sattler, Mike Bosse, Joel Hesch, Marc Pollefeys, and Roland Siegwart. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In *RSS*, 2015.
5. Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013.
6. J. Valentin, A. Dai, M. Niessner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. Learning to Navigate the Energy Landscape. In *International Conference on 3D Vision (3DV)*, 2016.
7. Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Int. Conf. Comput. Vis. (ICCV)*, pages 2938–2946, 2015.
8. Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *British Mach. Vis. Conf. (BMVC)*, 2012.
9. Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 8601–8610, 2018.
10. H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium (IV)*, 2011.
11. Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Research*, 36(1):3–15, 2017.
12. Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle Adjusted Direct RGB-D SLAM. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
13. David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, November 2004.
14. Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
15. Hajime Taira, Ignacio Rocco, Jiri Sedlar, Masatoshi Okutomi, Josef Sivic, Tomas Pajdla, Torsten Sattler, and Akihiko Torii. Is This The Right Place? Geometric-Semantic Pose Verification for Indoor Visual Localization. In *Int. Conf. Comput. Vis. (ICCV)*, 2019.
16. Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(2):257–271, 2018.
17. Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

18. Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019.
19. Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
20. Stephen Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
21. D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.
22. Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *ECCV*, 2010.
23. Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012.
24. S. Choudhary and P. J. Narayanan. Visibility probability structure from sfm datasets and applications. In *ECCV*, 2012.
25. A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009.
26. T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-Based Localization using Direct 2D-to-3D Matching. In *ICCV*, 2011.
27. Eagle S. Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. J. Robot. Research*, 30(4), April 2011.
28. B. Williams, G. Klein, and I. Reid. Real-Time SLAM Relocalisation. In *ICCV*, 2007.
29. Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.
30. N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 80(2):189–210, 2008.
31. A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, June 2007.
32. Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.*, 33(5):1255–1262, October 2017.
33. Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Int. Conf. Comput. Vis. (ICCV)*, pages 2564–2571, 2011.
34. R.M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994.
35. Z. Kukelova, M. Bujnak, and T. Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *ICCV*, 2013.
36. Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-Form Solutions to Minimal Absolute Pose Problems with Known Vertical Direction. In *ACCV*, 2010.
37. Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making Minimal Solvers for Absolute Pose Estimation Compact and Robust. In *ICCV*, 2017.
38. Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Rolling Shutter Absolute Pose Problem With Known Vertical Direction. In *CVPR*, 2016.
39. Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2969–2976, 2011.
40. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
41. O. Chum and J. Matas. Optimal Randomized RANSAC. *PAMI*, 30(8):1472 –1482, 2008.
42. Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *British Machine Vision Conference (BMVC)*, 2012.
43. Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A Universal Framework for Random Sample Consensus. *TPAMI*, 35(8):2022–2038, 2013.
44. T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 39(9):1744–1756, 2017.
45. M. Donoser and D. Schmalstieg. Discriminative Feature-to-Point Matching in Image-Based Locallization. In *CVPR*, 2014.
46. Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
47. Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
48. Liu Liu, Hongdong Li, and Yuchao Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map. In *ICCV*, 2017.
49. S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. In *CVPR*, 2013.
50. S. Cao and N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *CVPR*, 2014.
51. Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid Scene Compression for Visual Localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
52. M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale. The gist of maps - summarizing experience for lifelong localization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
53. Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *ICCV*, 2015.
54. L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *PAMI*, 39(7):1455–1461, 2017.
55. Carl Toft Viktor Larsson, Johan Fredriksson and Fredrik Kahl. Outlier Rejection for Absolute Pose Estimation with Known Orientation. In *BMVC*, 2016.
56. P. F. Alcantarilla, K. Ni, L. M. Bergasa, and F. Dellaert. Visibility learning in large-scale urban environment. In *ICRA*, 2011.
57. Dror Aiger, Haim Kaplan, Efi Kokiopoulou, Micha Sharir, and Bernhard Zeisl. General Techniques for Approximate Incidences and Their Application to the Camera Posing Problem. In *35th International Symposium on Computational Geometry (SoCG 2019)*, 2019.

58. R. Mur-Artal and J. D. Tardós. Visual-Inertial Monocular SLAM With Map Reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.
59. S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *ECCV*, 2014.
60. T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization. *IEEE Robotics and Automation Letters*, 3(3):1418–1425, 2018.
61. A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe. Keyframe-based visual-inertial online slam with relocalization. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
62. R. C. DuToit, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis. Consistent map-based 3d localization on mobile devices. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
63. Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Daniel Schmalstieg. Global Localization from Monocular SLAM on a Mobile Phone. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):531–539, 2014.
64. Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *CVPR*, 2017.
65. W. Zhang and J. Kosecka. Image based Localization in Urban Environments. In *International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2006.
66. Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To Learn or Not to Learn: Visual Localization from Essential Matrices. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
67. Enliang Zheng and Changchang Wu. Structure From Motion Using Structure-Less Resection. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.
68. Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.
69. Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *CVPR*, 2017.
70. Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
71. Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.
72. Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based Localization using Hourglass Networks. In *ICCV Workshops*, 2017.
73. T. Naseer, Gabriel Leivas Oliveira, Thomas Brox, and W. Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*, 2017.
74. Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep Multitask Learning For Semantic Visual Localization And Odometry. *RA-L*, 3(4):4407–4414, 2018.
75. Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep Auxiliary Learning For Visual Localization And Odometry. In *ICRA*, 2018.
76. Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-Based Localization Using LSTMs for Structured Feature Correlation. In *ICCV*, 2017.
77. Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
78. R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.
79. A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *Proceedings of the 2nd IEEE Workshop on Mobile Vision, with ICCV*, 2011.
80. Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019.
81. Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation using Neural Nets. In *The European Conference on Computer Vision (ECCV)*, September 2018.
82. Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera re-localization. In *ICCV*, 2019.
83. Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *ICCV Workshops*, 2017.
84. Soham Saha, Girish Varma, and C. V. Jawahar. Improved visual relocalization by discovering anchor points. In *BMVC*, 2018.
85. Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Luigi Di Stefano, and Philip H. S. Torr. On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation. In *CVPR*, 2017.
86. T. Cavallari, L. Bertinetto, J. Mukhoti, P. Torr, and S. Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online rgb-d camera relocalisation. In *3DV*, 2019.
87. T. Cavallari, S. Golodetz, N. Lord, J. Valentin, V. Prisacariu, L. Di Stefano, and P. H. S. Torr. Real-Time RGB-D Camera Pose Estimation in Novel Scenes using a Relocalisation Cascade. *TPAMI*, 2019.
88. Lili Meng, Jianhui Chen, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Backtracking regression forests for accurate camera relocalization. In *IROS*, 2017.
89. Lili Meng, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Exploiting points and lines in regression forests for RGB-D camera relocalization. In *IROS*, 2018.
90. D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. H.S. Torr. Random Forests versus Neural Networks - What's Best for Camera Relocalization? In *ICRA*, 2017.
91. Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip Torr. Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization. In *CVPR*, 2015.
92. Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017.
93. Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

94. Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019.
95. Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *arXiv:2002.12324*, 2020.
96. Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *ICCV*, 2019.
97. Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. KFNet: Learning Temporal Camera Relocalization using Kalman Filtering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
98. Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization. In *International Conference on 3D Vision (3DV)*, 2019.
99. Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
100. E. Stenborg, C. Toft, and L. Hammarstrand. Long-Term Visual Localization Using Semantically Segmented Images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
101. Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision. *arXiv:2001.07252*, 2020.
102. Assia Benbihi, Matthieu Geist, and Cedric Pradalier. ELF: Embedded Localisation of Features in Pre-Trained CNN. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
103. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.
104. Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Int. Conf. Comput. Vis. (ICCV)*, pages 3476–3485, 2017.
105. F. Radenović, G. Tolias, and O. Chum. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.
106. Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning Local Features from Images. In *Advances in Neural Information Processing Systems 31*, 2018.
107. Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.
108. Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016.
109. M. Brown, G. Hua, and S. Winder. Discriminative Learning of Local Image Descriptors. *TPAMI*, 2011.
110. Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond Cartesian Representations for Local Descriptors. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.
111. Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, 2017.
112. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning Local Feature Descriptors Using Convex Optimisation. *TPAMI*, 36(8):1573–1585, 2014.
113. Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.
114. Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
115. Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
116. Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. *arXiv:2004.13324*, 2020.
117. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.
118. Ignas Budvytis, Marvin Teichmann, Tomas Vojir, and Roberto Cipolla. Large Scale Joint Semantic Re-Localisation and Scene Understanding via Globally Unique Instance Coordinate Regression. In *BMVC*, 2019.
119. Sourav Garg, Niko Suenderhauf, and Michael Milford. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, 0(0), 2019.
120. J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. In *CVPR*, 2018.
121. Zachary Seymour, Karan Sikka, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware attentive neural embeddings for image-based visual localization. In *BMVC*, 2019.
122. T. Shi, S. Shen, X. Gao, and L. Zhu. Visual Localization Using Sparse Semantic 3D Map. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2019.
123. Carl Toft, Carl Olsson, and Fredrik Kahl. Long-term 3D Localization and Pose from Semantic Labellings. In *ICCV Workshops*, 2017.
124. Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *The European Conference on Computer Vision (ECCV)*, 2018.
125. P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The ApolloScape Open Dataset for Autonomous Driving and its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
126. X. Yu, S. Chaturvedi, C. Feng, Y. Taguchi, T. Lee, C. Fernandes, and S. Ramalingam. VLASE: Vehicle Localization by Aggregating Semantic Edges. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
127. Relja Arandjelović and Andrew Zisserman. Visual Vocabulary with a Semantic Twist. In *ACCV*, 2014.
128. D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. SIFT-Realistic Rendering. In *3DV*, 2013.

129. Qi Shan, Changchang Wu, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M. Seitz. Accurate geo-registration by ground-to-aerial image matching. In *3DV*, 2014.

130. M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (TOG)*, 33(2):14, 2014.

131. Julien Valentin, Angela Dai, Matthias Niessner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3D Vision (3DV)*, pages 323–332, 2016.

132. Anil Armagan, Martin Hirzer, Peter M. Roth, and Vincent Lepetit. Learning to align semantic segmentation and 2.5D maps for geolocalization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4590–4597, 2017.

133. Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian D. Reid, and Michael Milford. Deep Learning Features at Scale for Visual Place Recognition. *ICRA*, 2017.

134. Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In *RSS*, 2015.

135. A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual Place Recognition with Repetitive Structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

136. Michael J Milford and Gordon F Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, 2012.

137. D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

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

139. Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *IEEE ISMAR*, 2011.

140. Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017.

141. Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of Michigan North Campus long-term vision and lidar dataset. *IJRR*, 35(9):1023–1035, 2016.

142. Vasileios Balntas, Duncan Frost, Rigas Kouskouridas, Axel Barroso-Laguna, Arjang Talattof, Huub Heijnen, and Krystian Mikolajczyk. (SILDa): Scape Imperial Localisation Dataset, 2019.

143. Xun Sun, Yuanfan Xie, Pei Luo, and Liang Wang. A Dataset for Benchmarking Image-Based Localization. In *CVPR*, 2017.

144. Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. Revealing scenes by inverting structure from motion reconstructions. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2019.

145. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

146. Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2nd Edition.

147. Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *ACCV*, 2012.

148. Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 8601–8610, 2018.

149. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

150. Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *International Conference on Computer Vision (ICCV)*, 2019.

151. Hao Zhou, Torsten Sattler, and David W Jacobs. Evaluating local features for day-night matching. In *Proc. ECCV Workshops*, 2016.

152. Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. *ACM Trans. Graph.*, 32(3), July 2013.

153. Torsten Sattler et al. RansacLib - A Template-based *SAC Implementation, 2019.

154. Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019.