Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?

Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, Torsten Sattler

Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?

Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler

Abstract—Accurate visual localization is a key technology for autonomous navigation. 3D structure-based methods employ 3D models of the scene to estimate the full 6 degree-of-freedom (DOF) pose of a camera very accurately. However, constructing (and extending) large-scale 3D models is still a significant challenge. In contrast, 2D image retrieval-based methods only require a database of geo-tagged images, which is trivial to construct and to maintain. They are often considered inaccurate since they only approximate the positions of the cameras. Yet, the exact camera pose can theoretically be recovered when enough relevant database images are retrieved. In this paper, we demonstrate experimentally that large-scale 3D models are not strictly necessary for accurate visual localization. We create reference poses for a large and challenging urban dataset. Using these poses, we show that combining image-based methods with local reconstructions results in a higher pose accuracy compared to state-of-the-art structure-based methods, albeight at higher run-time costs. We show that some of these run-time costs can be alleviated by exploiting known database image poses. Our results suggest that we might want to reconsider the need for large-scale 3D models in favor of more local models, but also that further research is necessary to accelerate the local reconstruction process.

Index Terms—Visual Localization, Image-based Localization, Place Recognition, Pose Estimation, Image Retrieval.

1 Introduction

Determining the location from which a photo was taken is a key challenge in the navigation of autonomous vehicles such as self-driving cars and drones [1], robotics [2], mobile Augmented Reality [3], [4], and Structure-from-Motion (SfM) [5], [6], [7], [8]. In addition, solving the visual localization problem enables a system to determine the content of a photo. This can be used to develop interesting new applications, *e.g.*, virtual tourism [9] and automatic annotation of photos [10], [11].

Currently, approaches that tackle the visual localization problem can mainly be divided into two categories (*c.f.* figure 1 and table 1). *Visual place recognition* approaches [12], [13], [14], [15], [16], [17], [18] cast the localization problem as an image retrieval, *i.e.*, instance-level recognition, task and represent a scene as a database of geo-tagged images. Given a query photo, they employ **2D image-based localization** methods that operate purely on an image level to determine a set of database images similar to the query. The geo-tag of the most relevant retrieved photo then often serves as an approximation to the position from which the query was taken. *Image-based localization* methods [19], [20], [21], [22], [23], [24] cast the localization problem as a camera resectioning task. They represent scenes via 3D models, with image descriptors attached to the 3D points, which are obtained from SfM or by attaching local features/patches to 3D point clouds [25], [26]. **3D structure-based localization** algorithms then use these descriptors to establish a set of 2D-3D matches. In turn, these matches are used to recover the full 6DOF camera pose, *i.e.*, position and orientation, of the query image [27], [28].

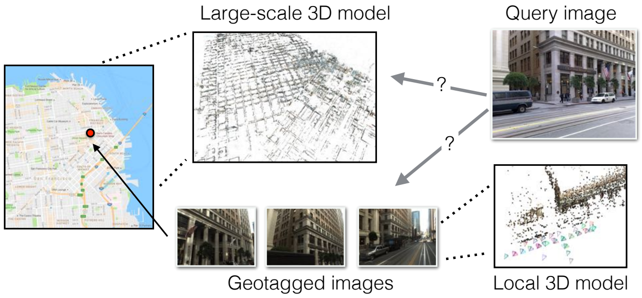


Fig. 1. **The state-of-the-art for large-scale visual localization.** 2D image-based methods (bottom) use image retrieval and return the pose of the most relevant database image. 3D structure-based methods (top) use 2D-3D matches against a 3D model for camera pose estimation. Both approaches have been developed largely independently of each other and never compared properly before.

A common perception is that 2D image-based approaches can be used by 3D structure-based methods to determine which parts of a scene might be visible in the query [22], [23], [29], [30]. Purely 2D-based techniques are considered unsuited for accurate visual localization due to only approximating the true camera position of the query. Consequently, 2D- and 3D-based localization methods are only compared in terms of place recognition performance [18], [23], [24]. However, this ignores the fact that a more accurate position, together with the camera orientation, can be computed if two or more related database images can be

- *M. Pollefeys is with the Department of Computer Sciene, ETH Zurich, Zurich, Switzerland and with Microsoft, Switzerland. E-mail: marc.pollefeys@inf.ethz.ch*
- *T. Sattler is with Chalmers University of Technology, Gothenburg, Sweden. E-mail: torsat@chalmers.se*
- *A. Torii, H. Taira, and M. Okutomi are with the Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology, Tokyo, Japan. E-mail: torii@sc.e.titech.ac.jp, htaira@ok.ctrl.titech.ac.jp, mxo@sc.e.titech.ac.jp*
- *J. Sivic is with the Inria, WILLOW project, Departement d'Informatique de l'Ecole Normale Superieure, ENS/INRIA/CNRS UMR 8548, PSL Research University. E-mail: Josef.Sivic@ens.fr*
- *J. Sivic and T. Pajdla are with the Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague. E-mail: pajdla@cvut.cz*

TABLE 1
System-level summary of visual localization approaches.

	2D image-based localization	3D structure-based localization
Scene representation	Database of geo-tagged images	3D points with associated image descriptors
Approach	Image retrieval	Descriptor matching followed by pose estimation
Output	Set of database images related to query, coarse position estimate	6DOF camera pose of the query image (position and orientation)
Advantage	Easy to maintain / update database	Directly provides pose estimates
Disadvantage	Requires extra post-processing to obtain 6DOF poses	Needs to construct a consistent 3D model

retrieved [31], [32], [33]. This naturally leads to the question whether 2D image-based localization approaches can achieve the same pose accuracy as 3D structure-based methods. This is a compelling question due to the way both types of methods represent scenes: especially for large-scale scenes, building and maintaining the 3D models required by structure-based techniques is a non-trivial task. At the same time, image-based techniques just require a database of geo-tagged images, which is significantly easier to maintain.

Contributions. In this paper, we want to answer whether large-scale 3D models are actually necessary for accurate visual localization or whether sufficiently precise pose estimates can already be obtained from a database of geo-tagged images. Our work makes the following contributions: i) We generate reference camera pose annotations for the query images of the San Francisco Landmarks dataset [12], resulting in the first city-scale dataset with such information. We make our reference poses together with all data and evaluation scripts required to reproduce our results or use our dataset for further research publicly available[1]. ii) We use this new dataset for the first comparison of 2D- and 3D-based localization approaches regarding their pose accuracy. To this end, we combine 2D image-based methods with a SfM-based post-processing step for pose estimation. Our results clearly show that 2D image-based methods can achieve a similar or even better positional accuracy than 3D structure-based methods. As such, our paper refutes the notion that purely 2D image-based approaches are inaccurate. iii) We demonstrate that the previously used strategy of evaluating localization methods via a landmark recognition task is unsuitable for predicting pose accuracy. Also, we show that pose precision results obtained on smaller landmark datasets do not translate to large-scale localization. Thus, our new benchmark closes a crucial gap in the literature and will help to drive research on accurate and scalable visual localization.

This paper is an extended version of [34] with a new extended version of the reference poses for the San Francisco dataset and detailed description of our approach to improve the reference poses (section 3.1). In detail, we provide **157 new reference poses** compared to the 442 poses provided in [34]. To register these images, we use another source of geo-registered images to fill the spatial gap of queries and original database images. The newly added reference poses thus correspond to more challenging images. We also measure the uncertainty of our reference poses, showing the limitations of our dataset. We perform additional experiments with variants of existing methods that use pose priors, add additional baselines, provide timing results, and use more evaluation measures compared to [34]. In particular, we propose modifications of SfM-based post processing that both reduce the run-time and increase pose accuracy. All the results in section 6 are renewed using the new reference poses.

2 Related work

Image-based approaches model localization as an image retrieval problem. They employ standard retrieval techniques such as Bag-of-Words (BoW) representations with inverted files [35], followed by fast spatial verification [36], [37], or more compact representations such as VLAD or Fischer Vectors [38], [39].

A more discriminative BoW representation can be constructed by only using informative features for each place [40]. Similarly, detecting and removing confusing features [41], *e.g.*, structures appearing in multiple places, or down-weighting their influence [15] improves performance as well. Arandjelović & Zisserman consider the descriptor space density to automatically weight the influence of image features [13]. Thus, features on repetitive structures have a smaller impact on the similarity score between images than features with unique local appearance.

One major challenge in visual localization is to handle large changes in illumination, *e.g.*, between day and night. To this end, Torii *et al.* create synthetic views from novel viewpoints by using the depth maps associated with street-view images to warp the original images [16]. Adding these images to the database lessens the burden on the feature detector to handle both viewpoint and illumination changes, resulting in a higher localization performance. Very recently, convolutional neural networks (CNNs) have been used to directly learn compact image descriptor suitable for place recognition [42], [43], [44], [45].

Another approach is to model visual localization as a classification task [17], [46], [47]. Such methods subdivide a scene into individual places and then learn classifiers, *e.g.*, based on a BoW representation [17], [46] or using CNNs [47], [48], to distinguish between images belonging to different places.

3D Structure-based localization methods assume that a scene is represented by a 3D model. Each 3D point is associated with one or more local descriptors. Thus, structure-based methods establish 2D-3D matches between features in a query image and the 3D points via descriptor matching. In a second stage, the camera pose can be estimated by employing a PnP solver [27], [28], [49] inside a RANSAC [50], [51] loop.

Descriptor matching quickly becomes a bottleneck and three (partially orthogonal) approaches exist to accelerate this stage: i) Prioritized search strategies [19], [21], [52] terminate correspondence search early on, ii) model compression schemes use only a subset of all 3D points [52], [53], [54], iii) retrieval-based approaches restrict matching to the 3D points visible in the top-ranked database images [22], [23], [29], [53], [55].

Lowe's ratio test [56], which measures the local density of the descriptor space, is commonly used to reject ambiguous matches. Larger 3D models induce a denser descriptor space, forcing the ratio test to reject more correct matches as ambiguous [20]. In order to handle the higher outlier ratios resulting from a relaxed test, large-scale, structure-based localization methods use

[1] http://www.ok.sc.e.titech.ac.jp/~torii/project/vlocalization/

co-visibility information [19], [23] or advanced pose estimation approaches [20], [24], [57], [58], [59].

Structure-based localization approaches naturally benefit from deep learning by replacing handcrafted feature detectors and descriptors such as SIFT [56] with learned alternatives [45], [60], [61], [62], [63]. Rather than replacing individual components, **learning-based localization** approaches aim to replace larger parts of the localization pipeline. *Camera pose regression* techniques such as PoseNet and its variants [64], [65], [66], [67] replace the full localization stack by a single convolutional neural network (CNN) that is trained to directly predict a 6DOF camera pose. However, these methods do not yet achieve the same pose accuracy as structure-based methods on outdoor scenes [67]. In addition, training them on larger datasets still seems to be an open problem [68], [69]. In this paper, we show that a state-of-the-art PoseNet variant [65] performs worse than a simple image-based baseline on a medium-scale dataset. Approaches that predict the poses of two subsequent images in a sequence [70], [71] could potentially lead to more accurate poses, but are not applicable to the single-image scenario considered in this paper.

Rather than replacing the complete localization pipeline, *scene coordinate regression* techniques [72], [73], [74], [75], [76], [77] only replace the 2D-3D matching stage while keeping the RANSAC-based pose estimation stage. These methods outperform structure-based approaches in terms of pose accuracy [73]. However, they currently are not able to handle larger scenes such as the ones considered in this paper [69], [73].

3 San Francisco Revisited

In this section, we first motivate our new pose dataset by reviewing the currently used evaluation protocols. Next, we review the San Francisco dataset before detailing how we generate reference poses for some of its query images.

Current evaluation protocols & their shortcomings. 3D structure-based localization approaches are typically evaluated by counting how many query images have an estimated pose with at least X inliers, where X is some threshold. This is based on the observation, made on smaller datasets, that wrong pose estimates are rarely supported by many inliers. However, this observation does not transfer to large-scale datasets [18], [23], [24]. At scale, repetitive structures and sheer size increase the chance of finding more wrong matches that are geometrically consistent [23], [24]. Simply counting the query images with at least X inliers thus overestimates the performance of structure-based methods. As such, it is necessary to also consider pose accuracy.

The datasets commonly used to evaluate the localization accuracy of structure-based approaches, 7 Scenes [76], Arts Quad [8], Cambridge Landmarks [64], Dubrovnik [52], and the recent Aachen Day-Night, RobotCar Seasons, and CMU Seasons benchmarks [68], all depict small- to medium-scale scenes with significant texture. Consequently, it is often possible to find many matches, which aids pose accuracy. Richly textured scenes are less frequent in urban environments due to the prevalence of reflecting or texture-less surfaces. This creates a need to also evaluate pose accuracy for truly large-scale datasets characterized by more ambiguous structures. Creating a benchmark for such a scene is one of the contributions of this paper.

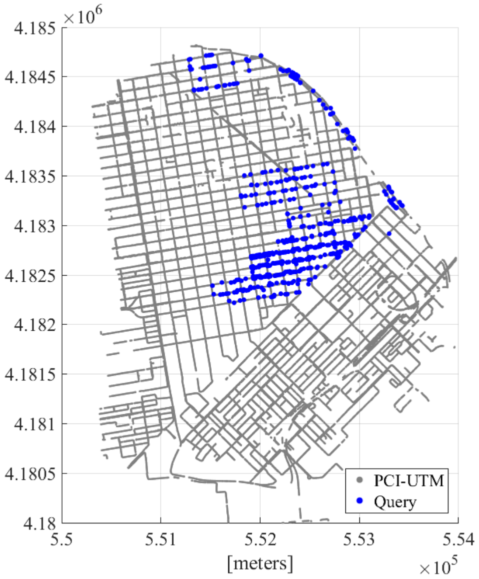


Fig. 2. **The San Francisco dataset with the reference poses of query images.** We provide the reference poses of query images (blue) which can be used as the ground truth for large-scale localization benchmarks on the SanFrancisco dataset.

2D image-based localization methods are mostly evaluated in the context of landmark or place recognition [12], [13], [14], [15], [16], [18], [32], [42], [78]. For landmark recognition, the goal is to retrieve at least one database image that depicts the same landmark or scene element as the query photo [12]. Vision is a long-range sensor and as such, a relevant database image might depict the same landmark while being taken tens or hundreds of meters away from the position of the query image. Thus, the geo-tag of such an image is not necessary a good approximation to the position of the query. Still, it might be possible to accurately determine this position through camera pose estimation (*c.f.* section 4). One of the contributions of this paper is to evaluate to what extend landmark recognition performance translates to accurate localization.

In terms of place recognition, image-based localization methods are tasked to find a database image whose geo-tag is within a certain radius of the query's GPS position [15], [32]. The fact that vision is a long-range sensor again causes problems in this setting as it is can be hard to distinguish between database images depicting the same part of the scene taken close to or far away from the query position [18]. In addition, the GPS positions associated with the query images can be rather inaccurate, especially in urban environments [12], requiring the use of a high threshold of tens or even hundreds of meters.

The San Francisco dataset. The publicly available San Francisco (SF) dataset, originally presented in [12], consists of $1,062,468$ street view images, cutout of panoramas using perspective projection from panoramas and denoted as PCI images, taken from the top of a car and 803 query images taken with cell phones by pedestrians. All photos depict downtown San Francisco (see the gray points in figure 2 for the distribution of the PCI images). Each PCI is associated with an accurate GPS position and a building ID, generated by back-projecting a 3D model of the city into the image [12]. Similarly, most query images have a GPS position and a list of IDs of the buildings visible in them. Unfortunately, the GPS coordinates of the query photos are not very precise and thus cannot be used as ground truth to measure localization accuracy.

There exist two SfM reconstructions of the San Francisco models [20]. The *SF-0* version of the dataset contains around 30M

3D points, associated with SIFT descriptors [56], reconstructed from $610,773$ images. To create the *SF-1* variant, the database images were histogram-equalized before extracting upright SIFT features, resulting in a model containing roughly 75M points reconstructed from $790,409$ images. For both 3D models, each 3D point can be associated with the building IDs from the database photos it was reconstructed from. However, both models do not provide reference poses for the query images. Thus, the SF dataset is commonly used to evaluate and compare structure- and image-based localization methods in the context of landmark recognition. Using the reference poses generated in this paper enables us to evaluate camera pose accuracy on the SF dataset.

3.1 Generating Reference Poses

Without any precise geo-tags, which are hard to obtain in downtown areas due to multi-path effects, the easiest way to obtain ground truth poses at scale is to use SfM algorithms. We follow this approach. Yet, instead of adding the query images to an existing model, which would require us to solve the localization problem, we generate local reconstructions around the queries which we subsequently geo-register. While we took great care to ensure the accuracy of our pose estimates, there is still a certain error in them. We thus use the term "reference poses" rather than "ground truth poses" to indicate that our poses are a rather precise reference and not a centimeter accurate ground truth.

Generating local reconstructions. The first step is to generate SfM reconstructions from the database images around the query images. Unfortunately, the GPS coordinates for the query images provided by the SF dataset are inaccurate with errors up to hundreds of meters [12]. Thus, we determine relevant PCI images for each query photo by exploiting the readily available building IDs. For each query, we perform feature matching against all database photos with a relevant building ID, followed by approximate geometric verification [15], [36]. We visually inspect the 20 images with the largest number of inliers, as long as they have at at least 5 inliers, and select the photo that is visually most similar to the query image for a later consistency check.

There is a strong change in viewpoint between the PCI (taken from the road) and query (taken mostly from sidewalks) images. Thus, finding sufficient matches to include the query image in the local reconstruction can be challenging. To increase the chance of finding enough matches, we thus include additional images that are not part of the original SF dataset. More precisely, we use Google Street View Time Machine (GSVTM) data [16], [17], [42] that covers the same area as the San Francisco dataset. We chose GSVTM instead of GSV images as they provide a denser spacial sampling. GSVTM provides panoramas of $13,312 \times 6,656$ pixels associated with geo-tag information, more precisely, longitude, latitude, and orientation (heading to the north)[2]. Similar to [12], [15], we cut 24 perspective images of $1,920 \times 1,440$ pixels, with a 60° field-of-view and 50% overlap to the neighboring views, from each panorama. For each query image, we use the 240 perspective images corresponding to the 10 GSVTM panoramas spatially closest to the most relevant PCI image.

We run SfM on the query, PCI images, and GSVTM perspective images. For redundancy, both COLMAP [6] and VisualSFM [79], [80] are used to obtain two SfM reconstructions.

Geo-registration with gravity constraint. In order to obtain the global positions and orientations of the cameras in each local reconstruction, we transformed the local model coordinate system to UTM coordinates. We first convert the GPS tags of the PCI and GSVTM images to UTM. Since the geo-tags do not include the height of the cameras above ground, we set it to zero. We then estimate the similarity transform between the camera positions in the model and their UTM coordinates.

A naive approach to geo-registration is to calculate a 7DOF similarity transform (scale, rotation, translation) between the SfM camera positions and the UTM coordinates of the database images [34]. However, this results in unstable estimates in degenerated camera configurations. A common degenerate configuration in our setting is that the cameras used for an SfM reconstruction align on a straight line since the car drives down a road.

This degenerated camera configuration problem can be addressed using the gravity direction computed from the SfM model. We first assume that all the PCI and GSVTM perspective images are aligned with respect to the gravity direction in UTM coordinates. Each camera pose in the SfM model and its pitch angle in UTM give a mapping of the gravity direction in UTM to the SfM coordinates. We determine the gravity direction in the SfM coordinates by taking a median of all the mapped gravity directions. Using this median gravity direction, we rotate the SfM model and finally compute a 5DOF similarity transform (1D scale, 1D rotation, and 3D translation), using LO-RANSAC with a 1 meter tolerance threshold[3].

Verification. Besides not being able to register the query image in the model, there are multiple ways a SfM reconstruction might provide an inaccurate estimate for a query's camera pose. For example, only few matches might be found or the correspondences might be in an unstable configuration, *e.g.*, all matches are situated in a small region of the query image. Consequently, we verify the poses after the registration process through consistency checks.

Let $\mathcal{Q}$ be the query image for which we want to verify the estimated pose and let $\mathcal{D}$ be the PCI image we selected for it. Using $\mathcal{D}$ and the SF-0 model, also registered to UTM coordinates, we generate a set of 2D-3D matches for the query image $\mathcal{Q}$. From the SF-0 model, we obtain a list of 3D points visible in $\mathcal{D}$. We project these 3D points into $\mathcal{D}$ to obtain 2D pixel positions, which we use to manually annotate the corresponding image positions in the query image. This results in a set of 2D-3D matches and, as a side product, also produces a set of 2D-2D correspondences between $\mathcal{Q}$ and $\mathcal{D}$. To obtain additional correspondences, we manually annotate 20 to 50 2D-2D matches between $\mathcal{D}$ and $\mathcal{Q}$. We use all these 2D-2D matches to compute the relative pose between the two images and use the 2D-3D matches to determine the scale of the translation. The resulting pose in UTM coordinates is then refined using bundle adjustment [81]. Ideally, this procedure should result in a precise estimate of $\mathcal{Q}$'s camera pose. However, it is hard to obtain accurate manually annotated pixel matches, resulting in some inaccuracy on the pose. We thus use it for a consistency check on the *absolute* camera pose. The check accepts a SfM pose if it is inside 10 meters of the position and within 15° of the view angle of the pose obtained from the manual matches.

A second consistency check employs the manually annotated

[2] GSVTM provides two location and orientation estimates, the original GPS information and geo-tags obtained via some alignment. We use the former as they are better aligned with the geo-tags of the PCI images.

[3] We experimented with three registration approaches based on the geo-tags of (1) only the PCI images, (2) only the GSVTM images, and (3) all images. All variants show a similar pose accuracy, but the last version registers the largest number of query images.

TABLE 2
Statistics on the consistency of the reconstructed SfM poses with our manual annotations.

Method \Consistency Test	Absolute	Relative	Both
COLMAP	311	553	306
VisualSFM	269	279	170
COLMAP & VisualSFM	228	245	142

2D-2D matches between $\mathcal{D}$ and $\mathcal{Q}$. From each of the two SfM models, we extract the essential matrix $\mathtt{E}$ describing the relative pose between the two images. For a given 2D-2D match $(\mathbf{x}_{\mathcal{Q}}, \mathbf{x}_{\mathcal{D}})$, we measure the pixel distances from the epipolar lines defined by $\mathtt{E}$ and $\mathtt{E}^{-1}$. $\mathtt{E}$ is considered to be consistent with the match if both errors are less than 3 pixels each. We consider a pose obtained by SfM to be consistent with this *relative* check if $\mathtt{E}$ is consistent with at least 10 of the manually annotated matches.

For each query image, a pose obtained by COLMAP or VisualSFM is accepted as a reference pose if it passes one of the two consistency checks. If poses from both COLMAP and VisualSFM pass this test, we select the one estimated by COLMAP.

Statistics. We created manual annotations for 687 out of the 803 query images from the SF dataset. Table 2 shows statistics on how many SfM poses, obtained with either COLMAP or VisualSFM, pass the two consistency checks. Finally, we obtain **598 reference poses** that are consistent with our manual annotations. For comparison, we only obtained 442 reference poses without using the additional GSVTM images.

Reference pose accuracy. In the next sections, we present the image- and structure-based localization methods that we evaluate using our reference poses in section 6. In order to draw valid conclusions, it is however necessary to understand the accuracy of these poses. We thus measure the uncertainty of the estimated poses as follows: Using RANSAC, we compute multiple poses from a subset of the 2D-3D matches used for each reference pose. From the resulting 100 poses, those supported by more than 80% of the 2D-3D matches are used to measure the differences to the reference pose. The mean median position and orientation errors are 1.01 meters and 2.19°.

The accuracy of any pose estimation approach, including SfM, that minimizes reprojection errors depends on the distance of the camera to the scene. More precisely, the uncertainty of the estimated pose grows roughly quadratically with the distance to the scene. On average, a query image is 10.5 meters away from its selected PCI image and 35.6 meters away from the 3D structure estimated during SfM. We thus consider the reference poses to be reasonably accurate.

We also measured the positional gap between the geo-tags of the PCI and the reconstructed PCI cameras registered in UTM coordinates. The mean average positional discrepancy is 0.39 meters, *i.e.*, the UTM coordinates estimated by SfM reconstruction and registration are consistent with the geo-tags of the PCI images.

4 2D Image-based Localization

The introduction posed the question whether 2D image-based localization approaches can achieve the same pose accuracy as structure-based methods. In other words, we are interested in determining whether an underlying 3D representation is necessary for high localization precision or whether a database of geo-tagged images can be sufficient.

In the following, we first review the 2D image-based methods that we chose for evaluation. We then explain different strategies to obtain camera poses of the query images using photos retrieved by the image-based methods.

Disloc [13], [18] is a state-of-the-art method based on the BoW paradigm and Hamming embedding [82]. During the voting stage of the retrieval pipeline, Disloc takes the density of the Hamming space into account to give less weight to features found on repeating structures while emphasizing unique features.

We also use the combination of Disloc with the geometric burstiness weighting scheme recently proposed in [18]. Given a list of spatially verified database images found by Disloc, the weighting strategy clusters these photos into places based on their geo-tags. It identifies features in the query image that are inliers to database photos coming from different places, *i.e.*, features found on repeating structures. Finally, the strategy performs a second re-ranking step where such features have less influence, which has been shown to improve landmark recognition performance.

DenseVLAD [16]. Disloc is based on the BoW paradigm and thus needs to store one entry per each image feature in an inverted file. This quickly leads to high memory requirements for large-scale scenes such as San Francisco. The DenseVLAD descriptor [16] is an example for a state-of-the-art localization algorithm based on compact image representations. Each image is represented by a single VLAD vector [38], [39], resulting in a more compact database representation. The DenseVLAD descriptor is constructed by aggregating RootSIFT [83] descriptors densely sampled on a regular grid in each image. As such, the method foregoes the feature detection stage, which has been shown to lead to more robust retrieval results, especially in the presence of strong illumination changes [16], [68].

NetVLAD [42]. The DenseVLAD descriptor is based on hand-crafted RootSIFT descriptors. In contrast, the NetVLAD representation uses a convolutional neural network to learn the descriptors that are aggregated into a VLAD vector. Training this representation in an end-to-end manner using a weakly supervised triplet loss has been shown to improve place recognition performance over DenseVLAD and other compact image descriptors.

4.1 Pose Estimation for 2D-based Approaches

Nearest neighbor (NN). Traditionally, most 2D image-based methods approximate the pose of the query image by the pose of the most relevant database image, *i.e.*, the database photo with the most similar BoW or VLAD descriptor. We use this strategy as a baseline and refer to it as *Nearest Neighbor (NN) pose*.

Spatial re-ranking (SR). Re-ranking the retrieved database images after spatial verification is known to improve image retrieval performance. As a second baseline, we use the pose of the best-matching database image after verification and refer to this strategy as *Spatial Re-ranking (SR) pose*. We perform spatial verification [36] for the top-200 retrieved images. For Disloc, we exploit the matches computed during the retrieval process while we extract and match RootSIFT features for both VLAD-based methods. For the VLAD-based methods, we re-rank based on the raw number of inliers. For Disloc, we also experiment with re-rank based on the geometric burstiness score.

SfM-on-the-fly (SfM). The previous two pose estimation strategies only consider the top-ranked database image. They ignore that each 2D-based approach typically retrieves multiple database images depicting the same place. In addition, the geo-tags of the database photos can also be used to identify a larger set of potentially relevant images. Inspired by [7], who generate a SfM model from a single photo by repeatedly querying an image database, we use small-scale SfM to obtain a local 3D model around the query image. Poses in the local model can then be converted into global poses by registering the SfM reconstruction into UTM coordinates based on the geo-tags of the database images. We refer to this strategy as *SfM-on-the-fly (SfM)*.

For 2D image-based methods, we generate a small subset from the top-200 retrieved images which are located within 25 meters from the pose obtained via the NN or SR strategy. We use COLMAP on the selected photos to obtain the 3D reconstruction. If COLMAP fails to recover the pose of a query camera, *e.g.*, when the reconstruction fails, we resort to the NN or SR pose.

A naive implementation of SfM-on-the-fly constructs a local model from scratch and ignores the fact that the poses of the database images are available. We thus also evaluate a version (*SfM init.*) that uses these known poses for initialization. This accelerates the reconstruction process and also makes it more robust. Compared to 3D-based methods, this approach achieves a higher pose accuracy.

Another approach to accelerate the local SfM process is to avoid exhaustive matching between all images. In order to reconstruct the query pose, database images with feature matches to the query image are most relevant. We thus also experiment with a *transitive matching strategy (trans.)*. We first match the query image against all retrieved database images. We then match two database images against each other only if each has a sufficient number of matches with the query image.

5 3D Structure-based Localization

This section reviews the two large-scale 3D structure-based localization methods used in this paper and justifies their selection.

Camera Pose Voting (CPV) [24]. Following [57], CPV assumes that the gravity direction, both in the local coordinate system of the camera and the global coordinate frame of the 3D model, is known together with a rough prior on the camera's height above the ground and its intrinsic calibration. In this setting, knowing the height of the camera directly defines the distance $\mathrm{dist}(p)$ of the camera to a matching 3D point p up to $\pm\varepsilon$, where ε is a small distance modeling the fact that the point might not re-project perfectly into the image. Thus, the camera's center falls into a circular band with minimum radius $\mathrm{dist}(p) - \varepsilon$ and maximum radius $\mathrm{dist}(p) + \varepsilon$ around p. As shown in [24], fixing the final[4] orientation angle of the camera also fixes the position of the camera inside the circular band.

The last observation directly leads to the camera pose voting scheme from [24]: Iterating over a set of discrete camera heights (defined by the coarse height prior) and a set of discrete camera orientations, each 2D-3D match votes for a 2D region[5] in which the camera needs to be contained. The matches voting for the cell receiving the most votes define a set of putative inliers and the position of the cell, together with the corresponding height and orientation, provides an approximation to the camera pose. This approximation is then refined by applying RANSAC with a 3-point-pose (P3P) solver on these matches. If available, a GPS prior can be used to further restrict the set of plausible cells and thus possible camera positions.

CPV was selected for our evaluation as [24] report state-of-the-art pose accuracy on the Dubrovnik dataset [52] and the state-of-the-art landmark recognition performance on the San Francisco dataset among structure-based localization methods.

Hyperpoints (HP) [23]. Rather than using Lowe's ratio test, which enforces *global uniqueness* of a match in terms of descriptor similarity, the HP method searches for *locally unique* matches [23]. It uses a fine visual vocabulary of 16M words [84] to define the similarity between the descriptor $\mathbf{d}(f)$ of a query image feature f and the descriptor $\mathbf{d}(p)$ of a 3D point p based on a ranking function: p has rank $\mathrm{r}(p, f) = i$ if $\mathbf{d}(p)$ falls into the i-th nearest visual word of $\mathbf{d}(f)$. The point's rank is $\mathrm{r}(p, f) = \infty$ if $\mathbf{d}(p)$ does not fall into any of the $k = 7$ nearest words of $\mathbf{d}(f)$. A 2D-3D match (f, p) is locally unique if there exists no other 3D point p' that is co-visible with p and has $\mathrm{r}(p', f) \leq \mathrm{r}(p, f)$. Two points are co-visible if they are observed together in one of the database images used to reconstruct the model.

Each locally unique 2D-3D match (f, p) votes for all database images observing p and the top-N images with the most votes are considered for pose estimation. Let $\mathcal{D}$ be one of these database images. All matches whose 3D point is visible in $\mathcal{D}$ as well as all matching points visible in nearby images are used for RANSAC-based pose estimation. Two images are considered nearby if they share at least one jointly observed 3D point in the SfM model. Considering points outside $\mathcal{D}$ increases the chance of obtaining more correct matches. Restricting the additional matches to nearby cameras avoids considering unrelated matches, thus avoiding high outlier ratios in RANSAC.

After computing a camera pose for each retrieved database image, the pose with the highest effective inlier count is selected. The effective inlier count takes both the number of inliers and their spatial distribution into account [22].

HP was selected as it represents a hybrid between 2D image-based and 3D structure-based localization methods. In addition, HP also outperforms other structure-based approaches employing retrieval techniques [22], [29], [53] at large scale.

Active Search (AS) [19]. CPV and HP have been designed to operate at large-scale. We compare their performance with Active Search, a state-of-the-art method for efficient localization at small-to-medium scale [19], [67]. AS relies on Lowe's ratio test to identify and reject ambiguous matches. As the ratio test rejects more and more correct matches at scale [20], we expect AS to localize significantly fewer images than CPV and HP. The comparison with AS thus serves to demonstrate the challenges encountered when scaling to larger, more complex scenes.

6 Experiments

This section uses our new reference poses to compare the localization accuracy of 2D image- and 3D structure-based methods. After describing the experimental setup and the evaluation protocol, we quantitatively evaluate the different approaches. We then discuss the results and their relevance.

Experimental setup. For Disloc [13], [18], DenseVLAD [16], and NetVLAD [42], we use source code provided by the authors for our evaluation. Disloc uses a visual vocabulary of 200k words trained on a subset of all database images. DenseVLAD uses a

[4] The other angles are already fixed by knowing the gravity direction.

[5] Regions account for the discretization of the pose parameters.

dictionary with 128 words also trained on the SF dataset, while NetVLAD uses 64 words. Unfortunately NetVLAD does not provide a version fine-tuned on San Francisco. Instead, we use the variant trained on the Pitts30k dataset [42]. Both DenseVLAD and NetVLAD generate 4,096 dimensional descriptors. For Hyperpoints (HP) [23], Camera Pose Voting (CPV) [24], and Active Search (AS) [19], we use poses estimated on the SF-0 dataset [20] as all methods use an SfM model to represent the scene. We run AS with vocabularies with 10k and 100k words, trained on the 3D point descriptors of the SF-0 dataset. We denote structure-based models by "(3D)" in the tables and legends.

Evaluation metric. We are mostly concerned with the pose accuracy achieved by the different methods. We measure the positional error in UTM coordinates since the local models used to construct the reference poses and the SF-0 reconstruction are registered to this coordinate system. However, the SF dataset only provides geodetic latitudes and longitudes of the cameras and not the altitudes. Thus, there is one degree of freedom in these registrations, namely the height above the plane defined by graticule. Accordingly, we measure the position error in 2D coordinates and evaluate how many images can be correctly localized by the different methods within a certain distance threshold.

In addition to the positional error, measured in meters, we also measure the orientation error. Given the reference camera orientation $\mathtt{R}_{\text{ref}}$ and the estimated query orientation $\mathtt{R}_Q$, both expressed as rotation matrices, we measure the angular error $|\alpha|$ between the two orientations as $2\cos(|\alpha|) = \text{trace}(\mathtt{R}_{ref}^\top \mathtt{R}_Q) - 1$ [85].

6.1 Quantiative Evaluation

We first evaluate the positional and orientational accuracy achieved by the 2D image-based methods. We compare the accuracy obtained when using the pose of the best-matching database image after retrieval (NN), the pose of the best-matching image after spatial verification (SR), and after local SfM reconstruction (SfM). The latter resorts to the NN (NN-SfM) and SR (SR-SfM) strategies if a pose cannot be estimated from the local model.

Figures 3 and 4 show results for BoW-based methods (a) and VLAD-based methods (b). As can be seen, spatial re-ranking (SR) increases the chance that the top-ranked database image is related to the query, *i.e.*, that the position and orientation of the retrieved database photo is close to the reference pose of the query. As a result, more query images can be correctly localized for larger distance thresholds. However, SR does not improve performance much in the high-accuracy regime. The reason is that the database images of the SF dataset were captured from a car driving on the road while the query photos were taken by pedestrians on the sidewalks. Thus, there is a certain minimal distance between their respective locations. A much better estimate can be obtained when using local SfM models (SfM), boosting the percentage of queries localized correctly within 5m from below 20% to about 60%. Similarly, local SfM improves the orientation accuracy by a large margin for smaller thresholds (0° to 20°). Interestingly, SR-SfM degrades the orientation accuracy compared to SR for angular errors above 20° (*c.f.* figure 4). This indicates that the orientation estimates provided by SfM can sometimes be rather inaccurate. As can be seen from the results in (c), using known poses for the database images (SfM init.) rater than computing them from scratch improves pose accuracy.

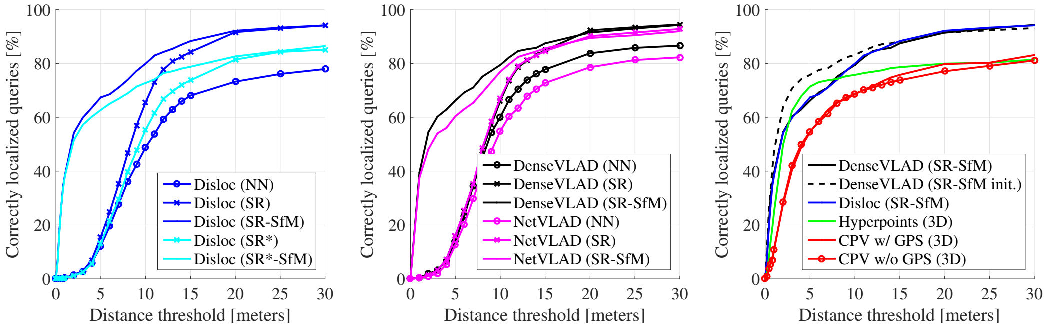
We observe that Disloc with inter-place geometric burstiness [18] (Disloc (SR*)) shows no additional improvement in comparison with the original Disloc (SR) [13] in this dataset. Disloc (SR*-SfM) uses relatively smaller subsets of images because Disloc (SR*) clusters relevant images by their geo-tags. This changes the quality of local reconstructions. This is an interesting result as [18] showed that accounting for geometric burstiness leads to a better location recognition performance. Our result thus indicates that better location recognition performance does not automatically translate to better camera pose accuracy.

For the VLAD-based representations, we notice that NetVLAD with the NN strategy performs worse than DenseVLAD (NN). DenseVLAD has the advantage that its vocabulary was trained on SF, while NetVLAD was trained on another dataset. However, their performance is virtually the same in combination with spatial re-ranking and local SfM. In the following, we focus on DenseVLAD and do not use NetVLAD for further experiments.

Figures 3(c) and 4(c) compare the positional and orientational accuracy of the best-performing 2D-based approaches with the two structure-based methods, Hyperpoints (HP) and Camera Pose Voting (CPV). As can be seen, both Disloc and DenseVLAD perform as good as HP for queries with an smaller error (2m and 5°) when using *SfM* as a post-processing step. 2D-based approaches are able to localize more images overall. If a pose cannot be estimated via local SfM, the 2D-based methods resort to reporting the position of the highest-ranking database image. The overall lower percentage of localized images observed for HP and CPV comes from such cases. For these images, their 2D-3D matching stage fails to produce enough matches for pose estimation. The interesting implication is that it is still possible to find relevant database images even when pose estimation itself fails due to a lack of matches.

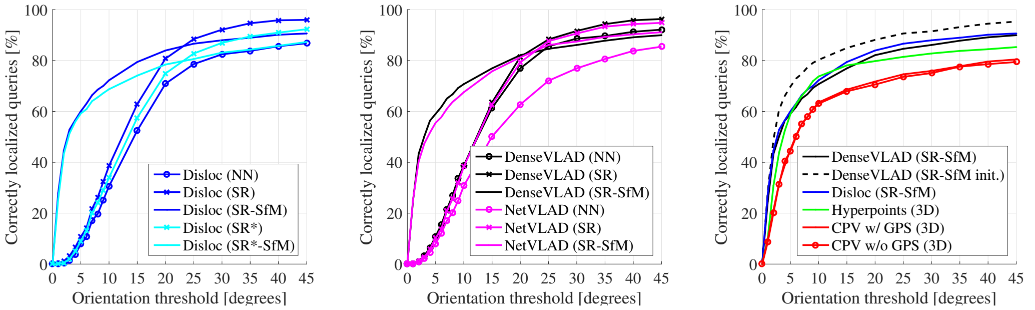
Many interesting applications, *e.g.*, self-driving cars, require highly accurate poses. In order to better understand the behavior of 2D-based and 3D-based methods in the high-precision regime, we compare their performance on two subsets of our reference poses. The first subset, containing 334 poses, is constructed from all reference poses for which either COLMAP or VisualSFM provides a pose that passes both consistency checks explained in section 3.1. This subset represents the more accurate among all of our reference poses. The second subset contains all 142 poses where both reconstructed poses pass both tests, thus containing the reference poses most likely to be highly accurate. Figure 5 depicts the performance of the different methods on both subsets. We again observe that HP performs better in the error range 2.5m to 9.0m than DenseVLAD (SfM) and Disloc (SfM). Yet, the best performance is again obtained using the SfM init. strategy. This clearly demonstrates that using known database poses for initialization increases the robustness of the SfM process. Based on the results, we conclude that large-scale 3D models are not really necessary for highly accurate visual localization.

The previous experiment considered the orientation and position errors separately. Following [68], [76], we next jointly consider both errors. Table 3 shows the percentage of query images that are localized within certain thresholds on the position and orientation error. The best results are clearly obtained by DenseVLAD (SR-SfM init.). The transitive matching strategy accelerates the local SfM process, but also decreases pose accuracy slightly. For more relaxed thresholds (25-30 meters, 20 degrees), 2D-based methods (DenseVLAD / Disloc (SR-SfM)) show better a recall than 3D-based methods (HP and CPV). Overall, image-based method can achieve a similar or higher performance than structure-based methods without a single consistent 3D model.



(a) 2D (BoW) (b) 2D (VLAD) (c) 2D vs. 3D

Fig. 3. **Evaluation of the positional localization accuracy** for BoW-based methods (a), VLAD-based approaches (b), and when comparing 2D- and 3D-based methods (c). Each plot shows the fraction of correctly localized queries (y-axis) within a certain distance to the reference pose (x-axis). As can be seen, using local SfM reconstructions (SfM) to estimate the camera poses allows 2D-based methods (Disloc, DenseVLAD) to achieve a positional accuracy similar or superior to 3D-based methods (Hyperpoints (HP), Camera Pose Voting (CPV)).



(a) 2D (BoW) (b) 2D (VLAD) (c) 2D vs. 3D

Fig. 4. **Evaluation of the orientational localization accuracy** for BoW-based methods (a), VLAD-based approaches (b), and when comparing 2D- and 3D-based methods (c). Each plot shows the fraction of correctly localized queries (y-axis) within a certain angular distance to the reference orientation (x-axis). Using local SfM reconstructions (SfM) to estimate the camera poses also allows 2D-based methods (Disloc, DenseVLAD) to achieve a orientational accuracy similar or superior to 3D-based methods (Hyperpoints (HP), Camera Pose Voting (CPV)).

TABLE 3
Localization performance depending on the positional and orientational errors. For each pair of thresholds, we provide the percentage of queries that are localized within the thresholds by each method.

	Time	Thresholds [meters, degrees]					
Method	[sec]	5, 5	10, 5	15, 10	20, 10	25, 20	30, 20
DenseVLAD (SR-SfM)	18.38	57.02	58.03	68.56	69.73	80.43	80.77
DenseVLAD (SR-SfM init.)	9.08	**66.89**	**67.39**	**77.76**	**78.43**	**85.79**	**86.12**
+trans.	7.26	63.71	64.55	**77.26**	77.93	85.62	85.95
Disloc (SR-SfM)	18.38	57.19	58.53	69.06	70.40	81.94	82.11
AS, 10K w/o GPS (3D)	0.62	27.42	27.76	33.44	33.44	35.79	35.79
AS, 10K w/ GPS (3D)	0.66	35.79	36.62	43.81	43.98	44.98	45.15
AS, 100K w/o GPS (3D)	0.09	29.43	29.93	35.95	36.12	37.46	37.46
AS, 100K w/ GPS (3D)	0.12	34.28	35.28	42.64	42.64	43.81	43.81
Hyperpoints (3D)	∼3	55.85	57.53	72.24	72.41	76.76	76.92
CPV w/ GPS (3D)	∼3	38.63	43.14	62.21	62.71	70.23	70.74
CPV w/o GPS (3D)	∼3	38.63	42.98	61.20	62.04	69.06	69.23

We next evaluate positional and orientational accuracy in a single unit, *i.e.*, pose accuracy, by computing reprojection errors. As described in section 3.1, the accuracy of any approach that estimates a camera pose by minimizing a reprojection error depends on the scene. Consequently, we can expect larger errors if the image is taken rather far away from the scene. In contrast, the mean reprojection error does not depend the distance to the scene.

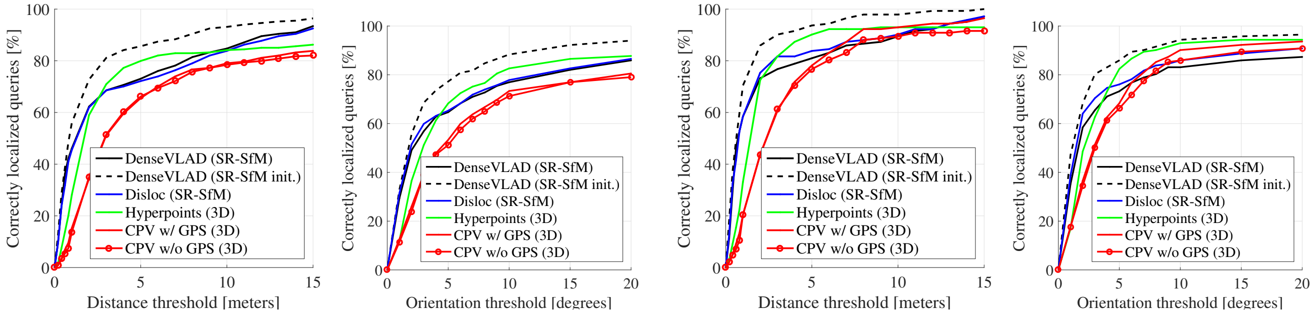
To compute the reprojection errors, we retrieve 2D-3D correspondences associated with the query reference pose. The reprojection errors are calculated by projecting 3D points to the image plane of the estimated query pose and computing the distances to the reference 2D points. Table 4 shows the mean reprojection error for each method. Not surprisingly, the patterns of results are similar to the positional and orientational accuracy evaluations. The higher errors of DenseVLAD / Disloc (SR-SfM) for the 75% quantile result from resorting to image retrieval if local SfM fails.

TABLE 4
Quantiles for mean reprojection errors.

Method	25%	50%	75%
DenseVLAD (SR)	100.09	155.75	233.56
DenseVLAD (SR-SfM)	10.92	33.78	165.30
DenseVLAD (SR-SfM init.)	**9.04**	**22.35**	**85.43**
DenseVLAD (SR-SfM init.)+trans.	9.90	24.03	86.95
Disloc (SR)	100.09	157.22	246.18
Disloc (SR-SfM)	10.46	38.60	160.02
Hyperpoints (3D)	14.93	32.21	124.82
CPV w/ GPS (3D)	21.05	46.87	116.99
CPV w/o GPS (3D)	21.18	44.74	122.36

Using positional priors. At large scale, it is often reasonable to assume that some coarse positional prior is given, *e.g.*, via GPS / WiFi localization. This prior can then be used to simplify the localization problem by restricting the search space. For example, image-based methods can restrict the search for relevant images to a certain radius around the positional prior of a given query [12]. Similarly, CPV can use such a regional prior to restrict the voting space [24]. We extend AS to use a position prior by restricting 2D-to-3D matching to points within 200 meters of the prior.

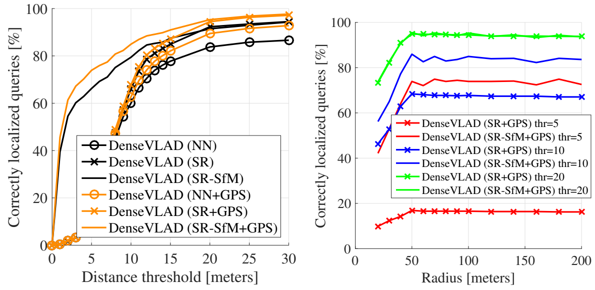
As can be seen in tables 3 and 4, using a pose prior for CPV has



(a) 334 subset.

(b) 142 subset.

Fig. 5. **Localization accuracy for subsets of the reference poses**, selected to include more accurate camera poses: (a) reference poses from either COLMAP or VisualSFM passing both consistency checks (334 reference poses) and (b) reference poses where both reconstructions pass both checks (142 poses). For each subset, we evaluate both positional (left) and orientational (right) accuracy for 2D- and 3D-based localization methods.



(a)

(b)

Fig. 6. **2D image-based localization with and without positional priors.** (a) Each plot shows the percentage of query images (y-axis) localized within a certain distance to the reference pose (x-axis). "+GPS" indicates restricting the search to a 100 meter radius around the given GPS prior. (b) The percentage of queries localized within 10, 20, 30 meters of the reference position (y–axis) obtained by DenseVLAD when varying the search radius (x–axis) around the GPS prior.

little impact on camera pose accuracy. This is an interesting observation and its relevance will be discussed in detail in section 6.3. In contrast, AS clearly benefits from the prior, which is due to its use of Lowe's ratio test. The prior allows AS to ignore some 3D points during matching. This results in a sparser descriptor space and thus decreases the chance that the ratio test rejects correct matches. Still, AS localizes significantly fewer images than CPV, even with the prior.

The original GPS measurements provided with the SF dataset are rather noisy, with errors of up to 150 meters [12]. These measurements were obtained with rather old hardware and software. We thus generate a more accurate positional prior by randomly sampling a position from a region which centers our reference pose and has a radius of 50 meters. We then incorporate the GPS priors into the 2D image-based methods. Figure 6(a) evaluates the localization performances for a search radius of 100 meters. As can be seen from the plots, using a GPS prior improves the localization performance of DenseVLAD.

We next evaluate the robustness of the localization to the search radius. In figure 6(b), both DenseVLAD (SR+GPS) and DenseVLAD (SR-SfM+GPS) show the best performances with the 50 meters radius, which is equal to the GPS uncertainty. This result is in line with the observation reported by Chen *et al.* [12]. DenseVLAD (SR-SfM) robustly retains its localization rate at the larger searching radius as it accurately estimates the pose of the query image. In contrast, we notice a significant drop in performance for DenseVLAD (SR) as it approximates the pose of the query through the pose of the top-retrieved database image.

Memory requirements. When dealing with large-scale datasets, the memory required to represent the scene is no longer a negligible issue. In the following, we summarize the memory requirements of each method. Here, we focus on the amount of data that needs to be kept in main memory during processing. For example, the image-based methods need access to the original images for the local SfM stage. However, these images could be read from disk after the initial retrieval step.

Storing 4096-dimensional VLAD [16], [42] for all 1,062,468 database images requires 17.4GB, and the requirements can be reduced further, *e.g.*, using product quantization [86], with negligible loss in performance [16]. For comparison, the DisLoc implementation from [18] requires about 20GB, although its memory footprint could be reduced to about 9.4GB by storing quantized feature geometry [87]. As reported in [23], the Hyperpoints approach requires 4.9GB to store the 3D model information and the visual vocabulary. In contrast, camera pose voting [24] uses all 149.3M SIFT descriptors of the SF-0 model for matching, thus requiring more than 18GB of memory. The memory footprint could be reduced to about 4.9GB by storing a single mean descriptor for each of the 30M 3D points, although this might reduce the localization performance. The two best-performing methods (Disloc and Hyperpoints) show similar localization accuracy but Hyperponts has five times smaller memory footprint. AS requires about 15GB of storage space and the difference in memory requirements for different vocabulary sizes is negligible.

Timings. Table 6 shows timings for the online components of the different algorithms, evaluated on Dubrovnik dataset. As can be seen, DenseVLAD (SR-SfM) is significantly slower than all other methods. This is unsurprising as it avoids the need for generating and maintaining a single 3D model by computing small 3D models on the fly. It thus trades flexibility for run-time. The corresponding run-times for the San Francisco dataset are shown in table 3. Note that local SfM is less efficient on the Dubrovnik dataset due to larger image resolutions, resulting in more extracted features.

Computing the DenseVLAD and NetVLAD descriptors for Dubrovnik's 6044 database images took 2.4h and 0.85h, respectively. While we use existing 3D models for Dubrovnik and SF-0, we expect that reconstructing the datasets from scratch takes less than 1 day and about 1-2 weeks, respectively. HP requires about 3s per image for online processing on SF-0.

Table 5 provides more detailed timings for the different

TABLE 5
Impact of different variations of SfM-on-the-fly on timings and performance. We provide the average computation time for each component of SfM-on-the-fly, and the percentage of queries that are localized within the thresholds by each method.

	Time [sec]			Thresholds [meters, degrees]					
Method	feature extracton	feature matching	reconstruction	5, 5	10, 5	15, 10	20, 10	25, 20	30, 20
DenseVLAD (SR-SfM)	1.13	2.23	15.03	57.02	58.03	68.56	69.73	80.43	80.77
DenseVLAD (SR-SfM init.)	1.13	2.23	5.73	66.89	67.39	77.76	78.43	85.79	86.12
DenseVLAD (SR-SfM init.)+trans.	1.13	1.80	4.33	63.71	64.55	77.26	77.93	85.62	85.95

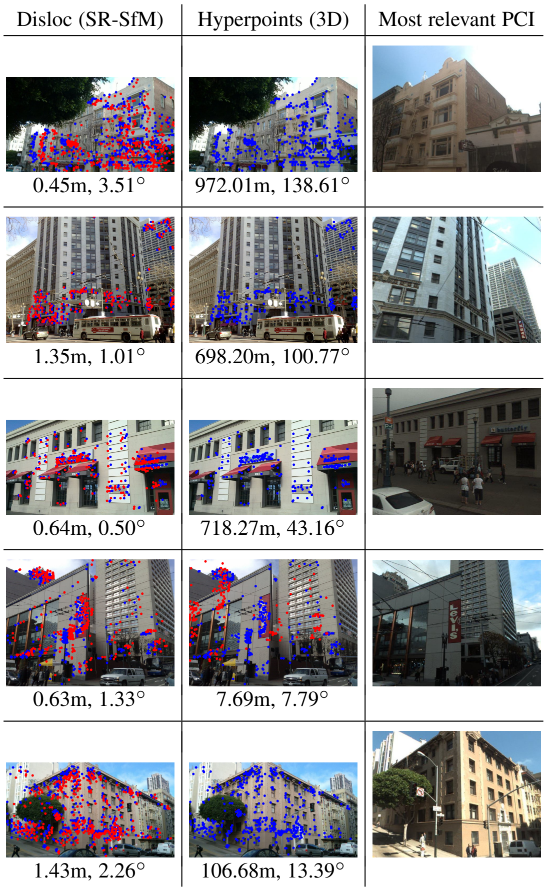


Fig. 7. **Examples of query images localized within 5m of the reference poses by a 2D image-based method (Disloc (SR-SfM)) (left) but not by a 3D structure-based method (Hyperpoints) (middle).** Colored dots are the reference 2D points used for computing the reference pose (blue) and the 3D points, associated to the reference 2D points, reprojected at the pose estimated by each method (red). The numbers below the images show the positional and orientational errors. The right column shows manually selected database PCI images that are most relevant to the queries.

variants of SfM-on-the-fly discussed in section 4.1, together with the impact of the modifications on pose accuracy. These timings were obtaining for COLMAP, using a GPU for feature extraction and parallelization for matching and reconstruction. As can be seen, most of the time is spent on the incremental reconstruction process. Using known database poses for initialization decreases the reconstruction time by a factor of about 3 while also increasing localization performance. Using transitive matching improves the matching times at a slight reduction of pose accuracy at the stricter thresholds. Feature extraction could be accelerated at the price of memory by pre-extracting features for the database images.

6.2 Qualitative Results.

We next show some qualitative examples to visually investigate when and how 2D image- and 3D structure-based methods work.

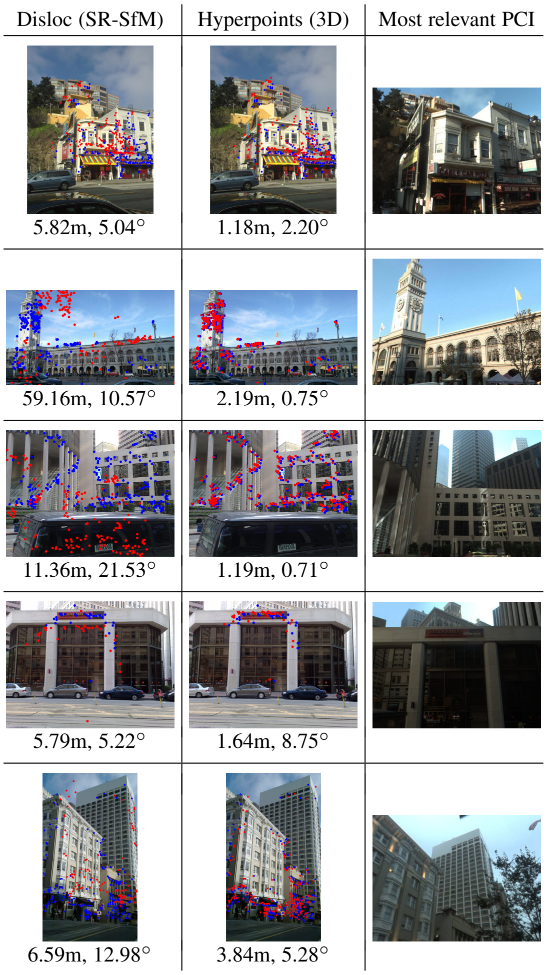


Fig. 8. **Examples of query images localized within 5m of the reference position by a 3D structure-based method (Hyperpoints) (middle) but not by a 2D image-based method (Disloc (SR-SfM)) (left).** See caption of figure 7 for details.

Based on the quantitative results, we choose the methods for 2D image- and 3D structure-based localization as Disloc (SR-SfM) and Hyperpoints (HP). We chose (SR-SfM) rather than (SR-SfM init.) to illustrate potential failure cases and since the former only requires coarse geo-tags for the database images.

Figures 7 and 8 show examples of query images correctly localized within 5m of the reference position by Disloc (SR-SfM) but not HP, respectively localized within 5m by HP but not by Disloc. Similarly, figures 9 and 10 show examples for which Disloc respectively HP provide pose estimates within 30m whereas the other method is less accurate.

HP uses quantized decriptors for matching. As a result, it has problems handling scenes with dominantly repetitive and similar structural elements. In contrast, Disloc (SR-SfM) uses the full feature descriptors and thus handles these scenes better.

However, Disloc (SR-SfM) has problems with accurately lo-

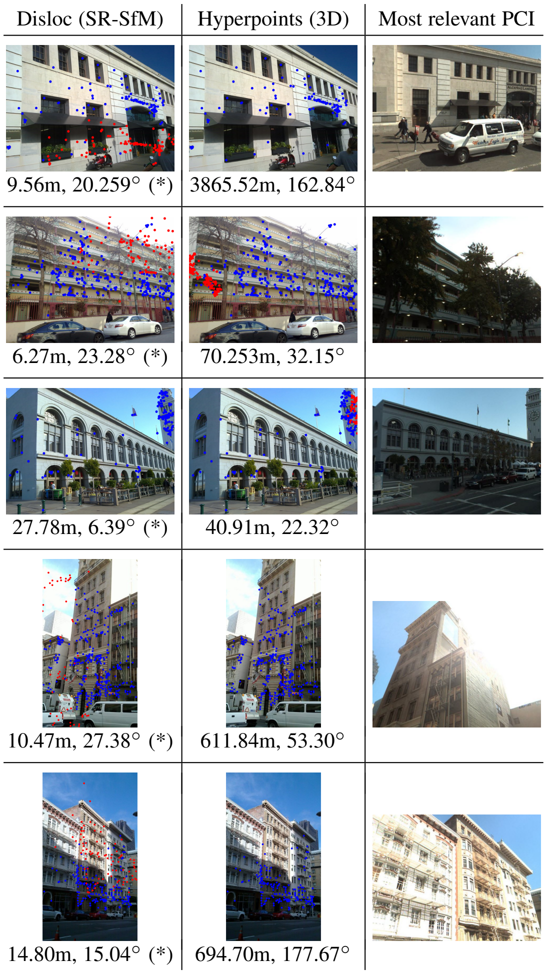


Fig. 9. **Examples of query images localized within 30m of the reference position by a 2D image-based method (Disloc (SR-SfM)) (left) but not by a 3D structure-based method (Hyperpoints) (middle).** The right column shows manually selected database PCI images that are most relevant to queries. "(*)" besides the results for Disloc (SR-SfM) indicate that local SfM fails so the results are the same as Disloc (SR). See also the caption of figure 7.

calizing images taken rather far away from the scene. This problem is compounded if the scenes are weakly textured. In this scenario, the local 3D models build from a few PCI images via SfM are less precise than the global model build used by HP. This reflects in the localization accuracy of Disloc (SR-SfM). If the viewpoint change between the retrieved PCI images and the query image is too large, the local SfM reconstruction process often fails to register the query image. In these cases, Disloc (SR-SfM) defaults to the pose provided by the (SR) strategy.

6.3 Relevance of the Results.

To put the results obtained at large scale with our references poses into context, we provide results on the medium-scale Dubrovnik dataset [52]. The 3D model consists of 1.9M 3D points reconstructed from 6044 database images.

Table 6 compares the DenseVLAD variants with CPV and AS. In addition, we also provide results for PoseNet [64], [65], a learning-based approach. HP is not applicable for this dataset as it was designed for larger-scale scenes where memory consumption and matching quality are issues. On the Dubrovnik dataset, the fine vocabulary of 16M words used by HP already requires more memory than the complete dataset.

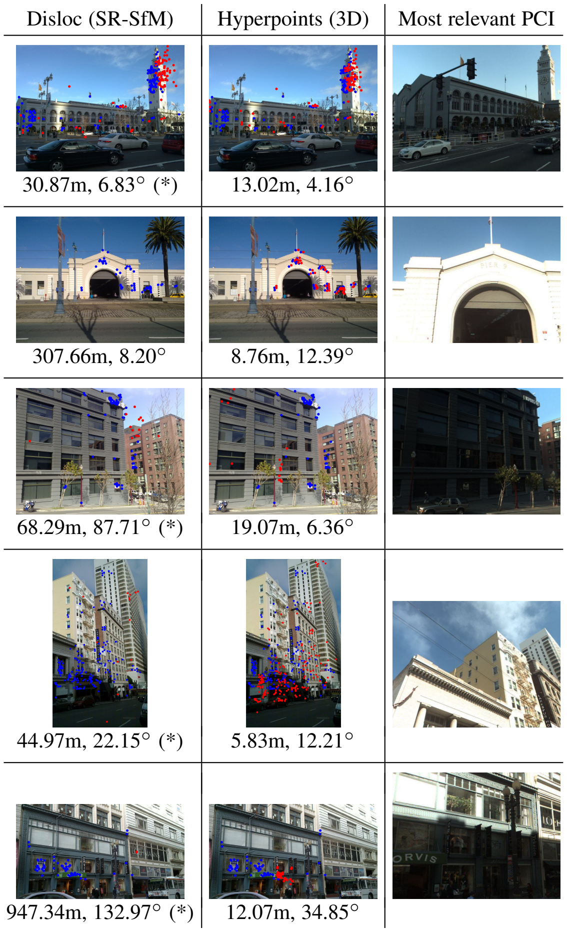


Fig. 10. **Examples of query images localized within 30m of the reference position by a 3D structure-based method (Hyperpoints) (middle) but not by a 2D image-based method (Disloc (SR-SfM)) (left).** See the caption of figure 9 for details.

As can be seen from table 6, combining DenseVLAD with local SfM results in a localization accuracy comparable to Active Search but worse than CPV. The opposite is the case for the larger SF-0 model, where DenseVLAD (SR-SfM) is clearly more precise. The reason is that finding good matches is easy on the smaller and well-textured Dubrovnik dataset while it is extremely challenging for the significantly larger SF-0 model. This is evident when comparing CPV's median positional accuracy on Dubrovnik (0.56m) and SF-0 (>2m). The matching step of local SfM is able to recover matches lost by CPV, enabling more accurate poses at large scale. The pose accuracy of DenseVLAD (SR-SfM) strongly depends on the quality of the local 3D models. Here, the SF-0 model is better suited due to the regular spatial distribution of its database images. In contrast, the spatial density of Dubrovnik's database photos varies strongly, making it harder to obtain good local models for some query images.

An interesting observation can be made from the relative performance between HP and CPV on SF-0. Previously, the SF dataset was used to evaluate the performance of structure-based localization methods in a landmark recognition scenario [20], [23], [24]. In this scenario, an image was considered correctly localized if it observed the correct building as specified by the building IDs provided by the SF dataset. Methods are evaluated based on their recall@95% precision, *i.e.*, based on the percentage of correctly localized images if the algorithm is allowed to make a mistake in 5% of all cases. In this scenario, CPV achieves a recall of

TABLE 6
Additional comparison on the Dubrovnik dataset [52].

Method	Time [sec]	Quantile errors [m] 25%	50%	75%
DenseVLAD [16] (NN)	1.42	1.4	3.9	11.2
DenseVLAD [16] (SR)	1.43	0.9	2.9	9.0
DenseVLAD [16] (SR-SfM)	∼200	0.3	1.0	5.1
DisLoc [13] (NN)	11.28	1.1	3.7	11.1
DisLoc [13] (SR)	11.29	0.9	2.9	8.9
DisLoc [13] (SR-SfM)	∼200	0.5	1.9	9.4
Camera Pose Voting (CPV) (3D) [24]	3.78	**0.19**	**0.56**	**2.09**
Active Search (3D) [19]	0.16	0.5	1.3	5.0
PoseNet [64], [65]	∼0.005	-	7.9	-

67.5% and 74.2% without and with a GPS prior, respectively. In contrast, HP only obtains a recall of 63.5%. This shows that good performance on the landmark recognition task does not necessarily translate to pose accuracy.

Another interesting observation can be made from the results obtained with CPV and a pose prior. In [24], including a pose prior improved landmark recognition performance. Yet, we observe no improvement in pose accuracy. This behavior is due to a peculiarity of the landmark recognition protocol: In order to increase the recall@5% precision, a large margin between the scores of correct and incorrect results is desirable. CPV uses the GPS prior to restrict the camera pose voting space, thus reducing the number of votes for incorrect poses. This, in turn, allows CPV to better distinguish between correct and incorrect poses. As such, our new dataset closes a crucial gap in the literature as it enables measuring pose accuracy at a large scale.

Regarding the performance of PoseNet, we observe that simply approximating the pose of a query image via the pose of the top-retrieved database image (DenseVLAD (NN)) already provides more accurate pose estimates. This shows that pose regression techniques such as PoseNet currently do not scale well. This is in line with reports from other work, which reports problems when trying to train such methods in more complex scenes [63], [68], [69].

Comparing the results obtained with Active Search on SF-0 (table 3) and Dubrovnik (table 6), we observe that AS scales well in terms of run-time. Part of this is due to the fact that the query images for Dubrovnik contain about five times more features on average, which compensates for the larger model size of SF-0. Yet, AS localizes significantly fewer images on the larger dataset. This demonstrates the need to also consider pose accuracy when evaluating the scalability of localization methods.

7 Conclusion

In this paper, we have presented the first comparison of 2D image-based and 3D structure-based localization methods regarding their localization accuracy at a large scale. To facilitate this comparison, we have created reference poses for some query images from the San Francisco dataset [12].

Our results show that purely 2D-based methods achieve the lowest localization accuracy. However, they offer the advantage of efficient database construction and maintenance and can localize images even if local feature matching fails. In contrast, 3D-based methods offer more precise pose estimates at the price of significantly more complex model construction and maintenance. Feature matching becomes harder at large-scale and finding fewer matches results in a lower pose quality. Combining 2D-based methods with local SfM reconstruction takes the advantages of both worlds, simple database construction and high pose accuracy, and results in state-of-the-art results for large-scale localization. However, this comes at the price of longer run-times during the localization process. Still, there is potential to accelerate this stage, *e.g.*, by caching local reconstructions. Alternatively, one could represent the scene using multiple smaller modes instead of a single large 3D model.

To the best of our knowledge, ours is the first dataset that can be used to measure the pose estimation accuracy on a large, complex dataset. Our results show that our dataset closes a crucial gap in the literature as this case is not covered by previous benchmarks and evaluation protocols. Our results show that there is still room for improvement in terms of pose precision. We make our reference poses, as well as all data required for evaluation, publicly available to facilitate further research on this topic.

Acknowledgments

This work was partly supported by EU-H2020 project LADIO No. 731970, JSPS KAKENHI Grant Number 15H05313, 17J05908, ERC grant LEAP (no. 336845), CIFAR Learning in Machines & Brains program and European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468), Grant Agency of the CTU in Prague project SGS18/104/OHK3/1T/37, and Google Tango.

References

[1] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele, “Real-Time Image-Based 6-DOF Localization in Large-Scale Environments,” in *Proc. CVPR*, 2012.

[2] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Trans. on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[3] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, “Scalable 6-DOF Localization on Mobile Devices,” in *Proc. ECCV*, 2014.

[4] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, “Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization,” in *RSS*, 2015.

[5] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, “Building Rome in a day,” in *Proc. ICCV*, 2009.

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

[7] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm, “From Single Image Query to Detailed 3D Reconstruction,” in *Proc. CVPR*, 2015.

[8] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher, “Discrete-continuous optimization for large-scale structure from motion,” in *Proc. CVPR*, 2011.

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

[10] T. Weyand and B. Leibe, “Discovering Details and Scene Structure with Hierarchical Iconoid Shift,” in *Proc. ICCV*, 2013.

[11] S. Gammeter, T. Quack, and L. Van Gool, “I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps,” in *Proc. ICCV*, 2009.

[12] 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 *Proc. CVPR*, 2011.

[13] R. Arandjelović and A. Zisserman, “DisLocation: Scalable descriptor distinctiveness for location recognition,” in *Proc. ACCV*, 2014.

[14] A. Torii, Y. Dong, M. Okutomi, J. Sivic, and T. Pajdla, “Efficient localization of panoramic images using tiled image descriptors,” *IPSJ Transactions on Computer Vision and Applications*, vol. 6, pp. 58–62, 01 2014.

[15] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, “Visual place recognition with repetitive structures,” *IEEE PAMI*, vol. 37, no. 11, pp. 2346–2359, 2015.

[16] A. Torii, R. Arandjelovìc, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," *IEEE PAMI*, vol. 40, no. 2, pp. 257–271, 2018.

[17] P. Gronat, J. Sivic, G. Obozinski, and P. Tomas, "Learning and calibrating per-location classifiers for visual place recognition," *IJCV*, vol. 118, no. 3, pp. 319–336, 2016.

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

[19] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE PAMI*, vol. 39, no. 9, pp. 1744–1756, 2017.

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

[21] S. Choudhary and P. J. Narayanan, "Visibility probability structure from sfm datasets and applications," in *Proc. ECCV*, 2012.

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

[23] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, "Hyperpoints and fine vocabularies for large-scale location recognition," in *Proc. ICCV*, 2015.

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

[25] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt, "SIFT-Realistic Rendering," in *3DV*, 2013.

[26] M. Aubry, B. C. Russell, and J. Sivic, "Painting-to-3d model alignment via discriminative visual elements," *ACM Trans. on Graphics (TOG)*, vol. 33, no. 2, p. 14, 2014.

[27] Z. Kukelova, M. Bujnak, and T. Pajdla, "Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length," in *Proc. ICCV*, 2013.

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

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

[30] S. Cao and N. Snavely, "Graph-Based Discriminative Learning for Location Recognition," in *Proc. CVPR*, 2013.

[31] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Proc. 3DPVT*, 2006.

[32] A. R. Zamir and M. Shah, "Accurate Image Localization Based on Google Maps Street View," in *Proc. ECCV*, 2010.

[33] E. Zheng and C. Wu, "Structure From Motion Using Structure-Less Resection," in *Proc. ICCV*, 2015.

[34] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla, "Are large-scale 3D models really necessary for accurate visual localization?" in *Proc. CVPR*, 2017.

[35] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *Proc. ICCV*, 2003.

[36] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. CVPR*, 2007.

[37] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval," in *Proc. ACCV*, 2016.

[38] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE PAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.

[39] R. Arandjelović and A. Zisserman, "All about VLAD," in *Proc. CVPR*, 2013.

[40] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. CVPR*, 2007.

[41] J. Knopp, J. Sivic, and T. Pajdla, "Avoding Confusing Features in Place Recognition," in *Proc. ECCV*, 2010.

[42] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE PAMI*, vol. 40, no. 6, pp. 1437–1451, 2018.

[43] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN Image Retrieval with No Human Annotation," *IEEE PAMI*, pp. 1–1, 2018.

[44] H. J. Kim, E. Dunn, and J. Frahm, "Learned contextual feature reweighting for image geo-localization," in *Proc. CVPR*, 2017.

[45] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. ICCV*, 2017.

[46] S. Cao and N. Snavely, "Graph-based discriminative learning for location recognition," in *Proc. CVPR*, 2013.

[47] T. Weyand, I. Kostrikov, and J. Philbin, "PlaNet - Photo Geolocation with Convolutional Neural Networks," in *Proc. ECCV*, 2016.

[48] P. H. Seo, T. Weyand, J. Sim, and B. Han, "CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps," in *Proc. ECCV*, 2018.

[49] M. Bujnak, Z. Kukelova, and T. Pajdla, "New efficient solution to the absolute pose problem for camera with unknown focal length," in *Proc. ACCV*, 2010.

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

[51] T. Sattler, C. Sweeney, and M. Pollefeys, "On Sampling Focal Length Values to Solve the Absolute Pose Problem," in *Proc. ECCV*, 2014.

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

[53] S. Cao and N. Snavely, "Minimal Scene Descriptions from Structure from Motion Models," in *Proc. CVPR*, 2014.

[54] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler, "Hybrid Scene Compression for Visual Localization," in *Proc. CVPR*, 2019.

[55] L. Liu, H. Li, and Y. Dai, "Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map," in *Proc. ICCV*, 2017.

[56] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[57] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-Scale Localization for Cameras with Known Vertical Direction," *IEEE PAMI*, vol. 39, no. 7, pp. 1455–1461, 2017.

[58] F. Camposeco, T. Sattler, A. Cohen, A. Geiger, and M. Pollefeys, "Toroidal Constraints for Two Point Localization Under High Outlier Ratios," in *Proc. CVPR*, 2017.

[59] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic Match Consistency for Long-Term Visual Localization," in *Proc. ECCV*, 2018.

[60] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *Proc. CVPR*, 2017.

[61] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *Proc. ECCV*, 2016.

[62] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *NIPS*, 2017.

[63] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic Visual Localization," in *Proc. CVPR*, 2018.

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

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

[66] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard, "Semantics-aware visual localization under challenging perceptual conditions," in *Proc. Intl. Conf. on Robotics and Automation*, 2017.

[67] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. ICCV*, 2017.

[68] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions," in *Proc. CVPR*, 2018.

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

[70] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-Aware Learning of Maps for Camera Localization," in *Proc. CVPR*, 2018.

[71] A. Valada, N. Radwan, and W. Burgard, "Deep Auxiliary Learning for Visual Localization and Odometry," in *Proc. Intl. Conf. on Robotics and Automation*, 2018.

[72] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC-differentiable RANSAC for camera localization," in *Proc. CVPR*, 2017.

[73] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *Proc. CVPR*, 2018.

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

[75] D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. H. Torr, "Random Forests versus Neural Networks - What's Best for Camera Relocalization?" in *Proc. Intl. Conf. on Robotics and Automation*, 2017.

[76] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. CVPR*, 2013.

[77] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr, "Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization," in *CVPR*, 2015.

[78] A. R. Zamir and M. Shah, "Image Geo-Localization Based on Multiple-Nearest Neighbor Feature Matching Using Generalized Graphs," *IEEE PAMI*, vol. 36, no. 8, pp. 1546–1558, 2014.

[79] C. Wu, S. Agarwal, B. Curless, and S. Seitz, "Multicore bundle adjustment," in *Proc. CVPR*, 2011.

[80] C. Wu, "Towards Linear-time Incremental Structure From Motion," in *Proc. 3DV*, 2013.

[81] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[82] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. ECCV*, 2008.

[83] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. CVPR*, 2012.

[84] A. Mikulík, F. Radenović, O. Chum, and J. Matas, "Efficient image detail mining," in *Proc. ACCV*, 2014.

[85] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation Averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, 2013.

[86] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE PAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[87] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total Recall II: Query Expansion Revisited," in *Proc. CVPR*, 2011.

Akihiko Torii received a Master degree and PhD from Chiba University in 2003 and 2006. He then spent four years as a post-doctoral researcher in Czech Technical University in Prague. Since 2010, he has been with Tokyo Institute of Technology, where he is currently an assitant professor in the Department of Systems and Control Engineering, School of Engineering. His research interests include 3D reconstruction, image and feature matching, and generating evaluation dataset.

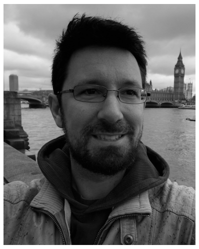
Hajime Taira recieved the B.E. and M.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Systems and Control Engineering, School of Engineering. He is a Research Fellow with the Japan Society for the Promotion of Science. His current research interest is in visual localization, consisting of image retrieval, feature matching, and 3D reconstruction.

Josef Sivic received MSc degree from the Czech Technical University in Prague, PhD from the University of Oxford and Habilitation from Ecole Normale Superieure in Paris. He currently holds a joint senior researcher position at Inria in Paris and Czech Technical University in Prague, where he leads a newly created team on Intelligent Machine Perception spanning both institutions. He has published more than 60 scientific publications and his papers have been awarded the Longuet-Higgins prize (CVPR'07) and the Helmholtz prize (ICCV'03 and ICCV'05) for fundamental contributions to computer vision that withstood the test of time. He has served as an area chair for major computer vision conferences and as a program chair for ICCV'15. In 2013, he has received an ERC starting grant.

Marc Pollefeys is a Professor of Computer Science at ETH Zurich and Director of Science at Microsoft working on HoloLens and Mixed Reality. He is best known for his work in 3D computer vision, having been the first to develop a software pipeline to automatically turn photographs into 3D models, but also works on robotics, graphics and machine learning problems. Other noteworthy projects he worked on with collaborators at UNC Chapel Hill and ETH Zurich are real-time 3D scanning with mobile devices, a real-time pipeline for 3D reconstruction of cities from vehicle mounted-cameras, camera-based self-driving cars and the first fully autonomous vision-based drone. Most recently his academic research has focused on combining 3D reconstruction with semantic scene understanding. He received a master of science in electrical engineering and a PhD in computer vision from the KU Leuven in Belgium in 1994 and 1999 respectively. He became an assistant professor at the University of North Carolina in Chapel Hill in 2002 and joined ETH Zurich as a full professor in 2007. Marc is an IEEE Fellow.

Masatoshi Okutomi received the B.Eng. degree from the Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, Japan, in 1981, and the M. Eng. degree from the Department of Control Engineering, Tokyo Institute of Technology, Tokyo, in 1983. He joined the Canon Research Center, Canon Inc., Tokyo, in 1983. From 1987 to 1990, he was a Visiting Research Scientist with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He received the Dr. Eng. degree from the Tokyo Institute of Technology, in 1993, for his research on stereo vision. Since 1994, he has been with the Tokyo Institute of Technology, where he is currently a professor with the Department of Systems and Control Engineering, School of Engineering.

Tomas Pajdla received the MSc and PhD degrees from the Czech Technical University in Prague. He works in geometry and algebra of computer vision and robotics with emphasis on nonclassical cameras, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, noncentral camera models generated by linear mapping, generalized epipolar geometries, to developing solvers for minimal problems in structure from motion and to solving image matching problem. He coauthored works awarded prizes at OAGM 1998 and 2013, BMVC 2002 and ACCV 2014. He is a member of the IEEE. Google Scholar: http://scholar.google.com/citations?user=gnR4zf8AAAAJ



Torsten Sattler received his PhD degree from RWTH Aachen University in 2014. From 2013 to 2018, he was a postdoc and senior researcher in the Computer Vision and Geometry Group at ETH Zurich. From 2016 to 2018, he was Marc Pollefeys' deputy, effectively managing the group. Since January 2019, he is an associated professor at Chalmers University of Technology in the Department of Electrical Engineering. His research interests center around visual localization and 3D mapping.