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

Torsten Sattler[1] Akihiko Torii[2] Josef Sivic[3,5]
Marc Pollefeys[1,4] Hajime Taira[2] Masatoshi Okutomi[2] Tomas Pajdla[5]
[1]Department of Computer Science, ETH Zürich [2]Tokyo Institute of Technology
[3]Inria* [4]Microsoft, Redmond [5]Czech Technical University in Prague

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 6DOF 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 pose accuracy similar to the state-of-the-art structure-based methods. Our results suggest that we might want to reconsider the current approach for accurate large-scale localization.

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 [28], robotics [30], mobile Augmented Reality [31, 32], and Structure-from-Motion (SfM) [2, 14, 42, 43]. 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 [46] and automatic annotation of photos [16, 52].

Currently, approaches that tackle the visual localization problem are divided into two categories (*c.f.* Fig. 1 and Tab. 1). *Visual place recognition* approaches [6, 12, 17, 37, 48, 49] 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 [13, 19, 26, 36, 38, 57] 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 [7, 44]. **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 [18, 25].

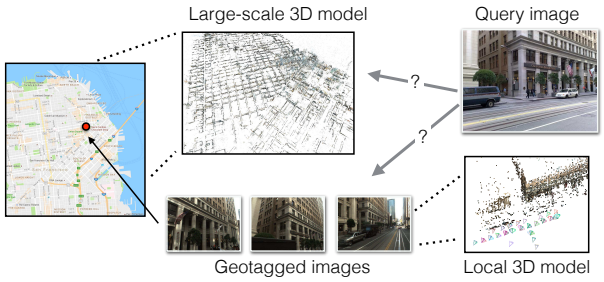


Figure 1. **The state-of-the-art for large-scale visual localization.** 2D image-based methods (bottom) use image retrieval and return the pose of the most relevant database image. 3D structure-based methods (top) use 2D-3D matches against a 3D model for camera pose estimation. Both approaches have been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

A common perception is that 2D image-based approaches can be a part of 3D structure-based methods to determine which parts of a scene might be visible in the query [9, 19, 36, 40]. 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 [36,

*WILLOW project, Departement d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548, PSL Research University.

	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

Table 1. System-level summary of visual localization approaches.

37, 57]. 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 retrieved [55,58]. This naturally leads to the question whether 2D image-based localization approaches can achieve the same pose accuracy as 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 easy to generate and 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 structure-based methods. As such, our paper refutes the notation that purely 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.

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 [45], fast spatial verification [34], or more compact representations such as VLAD or Fischer Vectors [5, 21].

A more discriminative BoW representation can be constructed by only using informative features for each place [41]. Similarly, detecting and removing confusing features [24], *e.g.*, structures appearing in multiple places, or down-weighting their influence [49] improves performance as well. Arandjelović & Zisserman consider the descriptor space density to automatically weight the influence of image features [6]. 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 [48]. 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 [3, 35].

Another approach is to model visual localization as a classification task [10, 17, 51]. Such methods subdivide a scene into individual places and then learn classifiers, *e.g.*, based on a BoW representation [10,17] or using CNNs [51], to distinguish between images belonging to different places.

3D structure-based localization. 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 [8, 18, 25] inside a RANSAC [15, 39] loop.

Descriptor matching quickly becomes a bottleneck in a localization pipeline and three (partially orthogonal) approaches exist to accelerate this stage: i) Prioritized search strategies [13,27,38] terminate correspondence search early on, ii) model compression schemes use only a subset of all 3D points [11, 27], iii) retrieval-based approaches restrict matching to the 3D points visible in the top-ranked database images only [11, 19, 36, 40].

Lowe's ratio test [29], which measures the local density

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

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 [26]. In order to handle the higher outlier ratios resulting from a relaxed test, large-scale, structure-based localization methods use co-visibility information [36, 38] or advanced pose estimation approaches [26, 47, 57].

Rather than explicitly estimating a camera pose from 2D-3D matches, recently proposed CNN-based approaches directly learn to regress a 6DOF pose from images [22,23,50]. However, as shown in [50] and our own experiment results, such methods do (not yet) achieve the same localization accuracy as 3D structure-based algorithms.

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 [36, 37]. Repetitive structures and sheer size increase the chance of finding more wrong matches that are geometrically consistent [36, 57]. 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, Dubrovnik [27] and Arts Quad [14], both mostly depict scenes with significant texture. Consequently, it is often possible to find many matches, which aids pose accuracy. Such scenes become 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 more complex datasets.

2D image-based localization are mostly evaluated in the context of landmark or place recognition [3, 6, 12, 37, 49, 49, 55, 56]. 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.* Sec. 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 [49, 55]. 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 [37]. 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 taken from the top of a car and 803 query images taken with cell phones. All photos depict downtown San Francisco (see the gray points in Fig. 2 for the distribution of the database images). Each database image 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 [26]. The *SF-0* version of the dataset contains around 30M 3D points, associated with SIFT descriptors [29], 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. Thus, the SF dataset is commonly used to evaluate and compare structure- and image-based localization methods in the context of landmark recognition.

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 vision-based 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 poses estimates, there is still a certain (hard-to-quantify) error in them. We thus use the term "reference poses" rather than "ground truth poses" to indicate that our poses should be considered as a rather precise reference rather than a centimeter accurate ground truth.

In the following, we detail the steps of our process.

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. Thus, we determine relevant database images 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 [34]. 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. Using the accurate geo-tag of this database photo, we run SfM on the query image and database photos within 50m of the selected one. For redundancy, we use both COLMAP [42] and VisualSFM [53, 54] to obtain two SfM reconstructions.

Geo-registration. 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 database images to UTM, where the height of each camera is set to zero. We then estimate the similarity transform between the camera positions in the model and their UTM coordinates.

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 a set of consistency checks.

Given the database image $\mathcal{D}$ selected above 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 [1]. 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 degrees of view angle of the pose obtained from the manual matches.

Method \Consistency Test	Absolute	Relative	Both
COLMAP	195	258	125
VisualSFM	139	263	134
COLMAP & VisualSFM	76	110	45

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

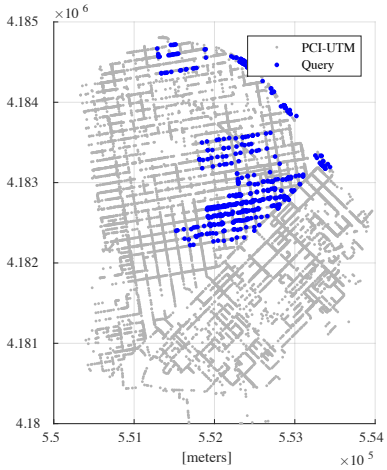


Figure 2. **The San Francisco dataset with the reference poses of query images.** We provide the reference poses of query images (blue) which can be used as the ground truth for large-scale localization benchmarks on the SanFrancisco dataset.

A second consistency check employs the manually annotated 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 correspondences.

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 684 out of the 803 query images from the SF dataset. Tab. 2 shows statistics on how many SfM poses, obtained with either COLMAP or VisualSFM, pass the two consistency checks. Based on the results, we obtain **442 reference poses** that are consistent with our manual annotations.

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 and then explain how the different strategies we used to obtain their camera poses.

We evaluate the performance of three 2D image-based approaches that differ in the way they solve the image retrieval problem inherent to 2D-based methods.

Disloc [6, 37]. Disloc is a state-of-the-art method based on the BoW paradigm and Hamming embedding [20]. 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 the impact of unique features.

We use the combination of Disloc with the geometric burstiness weighting scheme recently proposed in [37]. 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 the overall performance.

DenseVLAD [48]. Disloc is based on the BoW paradigm and thus needs to store an entry for each image feature in an inverted file. This quickly leads to large memory requirements for large-scale scenes such as San Francisco. The DenseVLAD descriptor [48] 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 [5, 21], resulting in a more compact database representation. The DenseVLAD descriptor is constructed by aggregating RootSIFT [4] 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 [48].

NetVLAD [3]. The DenseVLAD descriptor is based on hand-crafted RootSIFT descriptors. In contrast, the NetVLAD representations uses a convolutional neural network to learn the descriptors that are aggregated into a VLAD descriptor. 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 localization 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 pose* (NN).

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 pose* (SR). We perform spatial verification [34] 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 Disloc, we re-rank based on the geometric burstiness score while re-ranking based on the number of inliers for DenseVLAD and NetVLAD.

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 [43], who generate a SfM model from a single phot 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.

For DenseVLAD and NetVLAD, 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. For Disloc with geometric burstiness, we exploit the place clusters it computes [37]. We use those among the top-200 retrieved images that come from the same place as the top-retrieved photo. We use VisualSFM on the selected photos to obtain the 3D reconstruction. If VisualSFM fails to recover the pose of a query camera, *e.g.*, when the reconstruction fails, we resort to the NN or SR pose.

5. 3D Structure-based Localization

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

Camera Pose Voting (CPV) [57]. Following [47], 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 $\text{dist}(p)$ 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 $\text{dist}(p) - \varepsilon$ and maxi-

mum radius $\text{dist}(p) + \varepsilon$ around p. As shown in [57], fixing the final[2] 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 [57]: 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[3] 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 [57] report state-of-the-art pose accuracy on the Dubrovnik dataset [27] and the state-of-the-art recognition performance on the San Francisco among structure-based localization methods.

Hyperpoints (HP) [36]. 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 [36]. It uses a fine visual vocabulary with 16M words [33] 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 $\text{r}(p, f) = i$ if $\mathbf{d}(p)$ falls into the i-th nearest visual word of $\mathbf{d}(f)$. The point's rank is $\text{r}(p, f) = \infty$ 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 $\text{r}(p', f) \leq \text{r}(p, f)$. 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 one nearby image 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. Unlike the number of inliers of a pose, the effective inlier count takes both the number of inliers and their spatial distribution into account [19].

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

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

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 [11, 19, 40].

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.

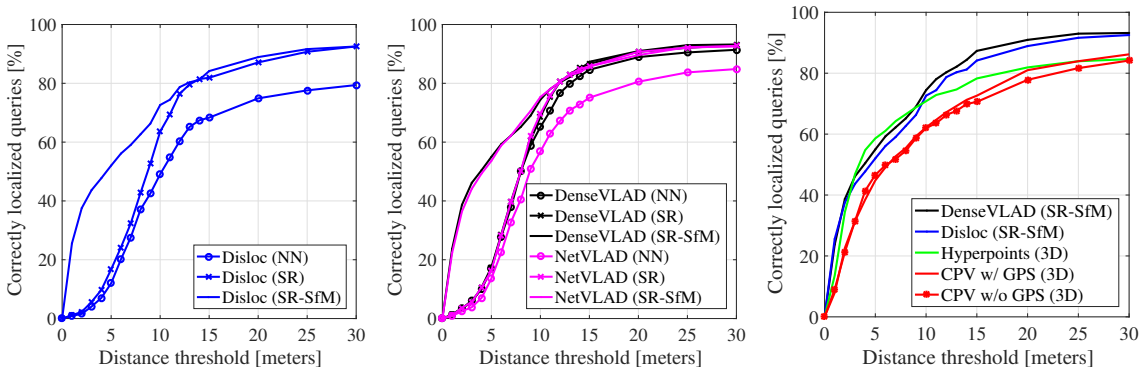
Experimental setup. For Disloc [6, 37], DenseVLAD [48], and NetVLAD [3], we use source code provided by the authors for our evaluation. Disloc uses a visual vocabulary of 200k words trained on a subset of all database images. DenseVLAD uses a dictionary with 128 words also trained on the SF dataset, 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 [3]. Both DenseVLAD and NetVLAD generate 4,096 dimensional descriptors. For Hyperpoints (HP) [36] and Camera Pose Voting (CPV) [57], we use poses estimated on the SF-0 dataset [26].

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 GPS coordinates and not the heights of the cameras. Thus, there is one degree of freedom in these registrations, namely the height above the plane defined by the GPS coordinates. 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.

Our reference poses provide both a position and an orientation estimate for the query images. However, we follow the common protocol in image-based localization and only evaluate the positional accuracy [13, 26, 27, 38, 57].

Quantitative evaluation. We first evaluate the positional accuracy achieved by the different 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.

Fig. 3 shows results for BoW-based methods (a) and VLAD-based methods (b). Spatial re-ranking (SR) increases the chance that the top-ranked database image is related to the query, *i.e.*, that the position of the retrieved database photo is close to the reference pose of the query.



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

Figure 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 (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, Camera Pose Voting).

As a result, more query images can be correctly localized for larger distance thresholds. However, SR does not improve performance for thresholds of 5m or less. 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 position estimate can be obtained when using local SfM reconstructions (SfM), boosting the percentage of queries localized correctly within 5m from below 20% to over 40%. We observe that Disloc with inter-place geometric burstiness re-ranking performs better than without, which is to be expected as it was shown to be superior to re-ranking based on the number of inliers in [37]. 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.

Fig. 3(c) compares the positional 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 error of 2m or less. While HP outperforms all other methods for the error range 2m to 10m, 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 fails.

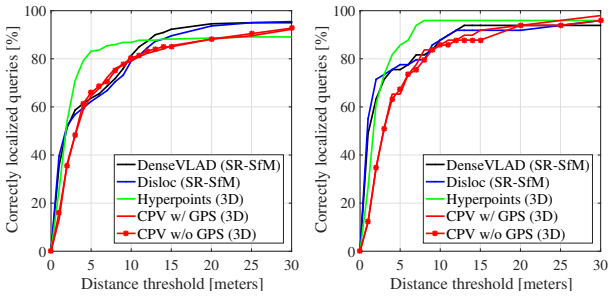


Figure 4. **Localization accuracy for subsets of the reference poses**, selected to include more accurate camera poses: (left) reference poses from either COLMAP or VisualSFM passing both consistency checks (214 reference poses) and (right) reference poses where both reconstructions pass both checks (45 poses).

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 214 poses, is constructed from all reference poses for which either COLMAP or VisualSFM provides a pose that passes both consistency checks explained in Sec. 3.1. This subset represents the more accurate among all of our reference poses. The second subset contains all 45 poses where both reconstructed poses pass both tests, thus containing the reference poses most likely to be highly accurate. Fig. 4 depicts the performance of the different methods on both subsets. On the first subset (Fig. 4, left), we again observe that HP performs better in the error range 2m to 12m, while both DenseVLAD and Disloc localize more images overall. An interesting observation can be made on the second subset containing the 45 reference poses passing the strictest consistency checks. DenseVLAD, Disloc, and HP performed equally well for small distance thresholds (<2m)

in the previous experiments. Yet, Fig. 4(right) shows a clear and substantial improvement from 20% (HP) to 50% (DenseVLAD, Disloc) localized images within 1m. Based on the results, we conclude that large-scale 3D models are not really necessary for highly accurate visual localization. However, pre-built 3D models can help to improve the accuracy for some images that are not accurately localized otherwise.

Relevance of the results. To put the results obtained with our references poses into context, we provide results on the Dubrovnik dataset [27]. The 3D model consists of 1.9M 3D points reconstructed from 6k database images.

Tab. 3 compares the DenseVLAD variants with CPV. In addition, we also provide results for Active Search [38], an efficient structure-based method using prioritization, and PoseNet [22, 23], 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. Interestingly, just performing retrieval without any pose estimation (DenseVLAD (NN)) yields more accurate poses than learning to regress poses via PoseNet.

As can be seen from Tab. 3, 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 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.

Another 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 [26, 36, 57]. 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 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. As such, our new dataset closes a crucial gap in the literature as it enables measuring pose accuracy at a large scale.

Method	Time [sec]	Quantile errors [m] 25%	50%	75%
DenseVLAD [48] (NN)	1.42	1.4	3.9	11.2
DenseVLAD [48] (SR)	1.43	0.9	2.9	9.0
DenseVLAD [48] (SR-SfM)	∼200	0.3	1.0	5.1
Camera Pose Voting (CPV) [57]	3.78	0.19	0.56	2.09
Active Search [38]	0.16	0.5	1.3	5.0
PoseNet [22, 23]	∼0.005	-	7.9	-

Table 3. Additional comparison on the Dubrovnik dataset [27].

Timings. Tab. 3 shows timings for the online components of the different algorithms. Computing the DenseVLAD and NetVLAD descriptors for Dubrovnik's 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 takes less than 1 day and about 1-2 weeks, respectively. HP requires about 5s per image on SF-0.

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 combines 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 significantly longer run-times during the localization process.

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. At the same time, our results suggest 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.

Acknowledgements. This work was partly supported by EU-H2020 project LADIO No. 731970, JSPS KAKENHI Grant Number 15H05313, ERC grant LEAP (no. 336845), CIFAR Learning in Machines & Brains program and ESIF, OP Research, development and education project IMPACT No. CZ.02.1.01/0.0/0.0/15_003/0000468, and Google Tango.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 4

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Proc. ICCV*, 2009. 1

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 2, 3, 5, 6

[4] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 5

[5] R. Arandjelović and A. Zisserman. All about VLAD. In *Proc. CVPR*, 2013. 2, 5

[6] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *Proc. ACCV*, 2014. 1, 2, 3, 5, 6

[7] M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM Trans. on Graphics (TOG)*, 33(2):14, 2014. 1

[8] 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. 2

[9] S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. In *Proc. CVPR*, 2013. 1

[10] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. In *Proc. CVPR*, 2013. 2

[11] S. Cao and N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *Proc. CVPR*, 2014. 2, 6

[12] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, H. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod. Residual enhanced visual vectors for on-device image matching. In *Asilomar*, 2011. 1, 2, 3, 8

[13] S. Choudhary and P. J. Narayanan. Visibility probability structure from sfm datasets and applications. In *Proc. ECCV*, 2012. 1, 2, 6

[14] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, 2011. 1, 3

[15] M. Fischler and R. Bolles. Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Commun. ACM*, 24:381–395, 1981. 2

[16] 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. 1

[17] P. Gronat, J. Sivic, G. Obozinski, and P. Tomas. Learning and calibrating per-location classifiers for visual place recognition. *IJCV*, 118(3):319–336, 2016. 1, 2

[18] R. 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. 1, 2

[19] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *Proc. CVPR*, 2009. 1, 2, 6

[20] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008. 5

[21] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE PAMI*, 34(9):1704–1716, 2012. 2, 5

[22] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, 2017. 3, 8

[23] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proc. ICCV*, 2015. 3, 8

[24] J. Knopp, J. Sivic, and T. Pajdla. Avoding Confusing Features in Place Recognition. In *Proc. ECCV*, 2010. 2

[25] 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. 1, 2

[26] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *Proc. ECCV*, 2012. 1, 3, 6, 8

[27] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *Proc. ECCV*, 2010. 2, 3, 6, 8

[28] 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. 1

[29] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 2, 3

[30] 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*, 32(1):1–19, 2016. 1

[31] 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. 1

[32] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *Proc. ECCV*, 2014. 1

[33] A. Mikulík, F. Radenović, O. Chum, and J. Matas. Efficient image detail mining. In *Proc. ACCV*, 2014. 6

[34] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007. 2, 4, 5

[35] F. Radenović, G. Tolias, and O. Chum. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *Proc. ECCV*, 2016. 2

[36] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proc. ICCV*, 2015. 1, 2, 3, 6, 8

[37] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-Scale Location Recognition and the Geometric Burstiness Problem. In *Proc. CVPR*, 2016. 1, 3, 5, 6, 7

[38] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE PAMI*, 2016 (accepted). 1, 2, 3, 6, 8

[39] T. Sattler, C. Sweeney, and M. Pollefeys. On Sampling Focal Length Values to Solve the Absolute Pose Problem. In *Proc. ECCV*, 2014. 2

[40] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *Proc. BMVC.*, 2012. 1, 2, 6

[41] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proc. CVPR*, 2007. 2

[42] J. L. Schönberger and J.-M. Frahm. Structure-From-Motion Revisited. In *Proc. CVPR*, 2016. 1, 4

[43] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. From Single Image Query to Detailed 3D Reconstruction. In *Proc. CVPR*, 2015. 1, 5

[44] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. SIFT-Realistic Rendering. In *3DV*, 2013. 1

[45] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. ICCV*, 2003. 2

[46] N. Snavely, S. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 80(2):189–210, 2008. 1

[47] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE PAMI*, 2016 (accepted). 3, 5

[48] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proc. CVPR*, 2015. 1, 2, 5, 6, 8

[49] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual Place Recognition with Repetitive Structures. *IEEE PAMI*, 37(11):2346–2359, 2015. 1, 2, 3

[50] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based Localization with Spatial LSTMs. *arXiv*, 1611.07890, 2016. 3

[51] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *Proc. ECCV*, 2016. 2

[52] T. Weyand and B. Leibe. Discovering Details and Scene Structure with Hierarchical Iconoid Shift. In *Proc. ICCV*, 2013. 1

[53] C. Wu. Towards Linear-time Incremental Structure From Motion. In *3DV*, 2013. 4

[54] C. Wu, S. Agarwal, B. Curless, and S. Seitz. Multicore bundle adjustment. In *Proc. CVPR*, 2011. 4

[55] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *Proc. ECCV*, 2010. 2, 3

[56] A. R. Zamir and M. Shah. Image Geo-Localization Based on MultipleNearest Neighbor Feature Matching Using Generalized Graphs. *IEEE PAMI*, 36(8):1546–1558, 2014. 3

[57] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *Proc. ICCV*, 2015. 1, 3, 5, 6, 8

[58] W. Zhang and J. Kosecka. Image based localization in urban environments. In *Proc. 3DPVT*, 2006. 2