This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

A Dataset for Benchmarking Image-based Localization

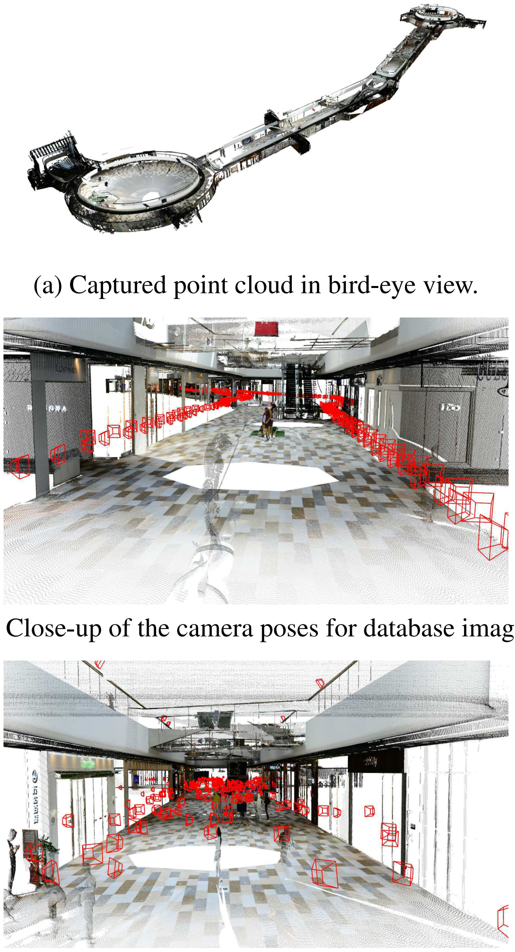
Xun Sun[*], Yuanfan Xie[*], Pei Luo, Liang Wang
Baidu Autonomous Driving Business Unit[†]
{sunxun,xieyuanfan,luopei,wangliang18}@baidu.com

Abstract

A novel dataset for benchmarking image-based localization is presented. With increasing research interests in visual place recognition and localization, several datasets have been published in the past few years. One of the evident limitations of existing datasets is that precise ground truth camera poses of query images are not available in a meaningful 3D metric system. This is in part due to the underlying 3D models of these datasets are reconstructed from Structure from Motion methods. So far little attention has been paid to metric evaluations of localization accuracy. In this paper we address the problem of whether state-of-the-art visual localization techniques can be applied to tasks with demanding accuracy requirements. We acquired training data for a large indoor environment with cameras and a LiDAR scanner. In addition, we collected over 2000 query images with cell phone cameras. Using LiDAR point clouds as a reference, we employed a semi-automatic approach to estimate the 6 degrees of freedom camera poses precisely in the world coordinate system. The proposed dataset enables us to quantitatively assess the performance of various algorithms using a fair and intuitive metric.

1. Introduction

Image-based Localization (IBL) methods estimate the 6 degrees of freedom (6DOF) pose of a camera with respect to a 3D representation of the scene. The input of an IBL algorithm is typically one or multiple images and the visual information of which is used to recover the camera pose. This technique has gained attention recently from both academia and industry since it can potentially enable many novel applications, such as location recognition [31, 16], navigation [37], autonomous driving, localization-based augmented reality, and various location-based services [17].



(a) Captured point cloud in bird-eye view.

(b) Close-up of the camera poses for database images.

(c) Groundtruth camera poses for the query images.

Figure 1. A glance of the captured point cloud and cameras in our dataset. This figure is best viewed in color.

Typical IBL algorithms assume that there is an existing 3D model produced by Structure from Motion (SfM) or Simultaneous Localization and Mapping (SLAM) [11, 29, 30, 31]. Usually, the 3D model consists of sparse 3D point clouds as well as their associated 2D feature descriptors. The IBL problem can then be formulated as a 2D-3D matching problem, i.e., with image features extracted

[*] Indicates equal contributions and joint first authors.

[†] This work was done when the authors were with Baidu Research's Institute of Deep Learning.

from query images, establishing 2D-3D correspondences such that the camera pose can be recovered via n-point-pose solvers [10, 15].

Much of the progress over the past few years has been driven by the availability of standard test images and benchmarks [6, 16, 11, 31, 32, 14, 39]. Existing IBL datasets, although seminal, are limited in the accuracy of ground truth camera pose and their evaluation methodologies. As summarized in Table 1, for the 6DOF localization tasks, the poses of query images rely heavily on SfM. The camera pose determined by SfM can be error-prone when the scene contains insufficient textures or repetitive structures. In addition, lack of absolute scale of the reconstructed 3D model makes it difficult to evaluate the recovered pose in a meaningful metric coordinate system. For these reasons, most SfM-based datasets use a heuristic "PnP-inlier-count" metric to assess localization accuracy [28].

In order to allow for objective comparisons between various IBL methods, new datasets are urgently needed. This paper addresses existing limitations of IBL datasets by introducing a novel benchmark dataset. We choose the ground level of a shopping mall which is over 5000 square meters with many challenging features for IBL such as reflective materials, transparent windows, moving people and repetitive structures etc. Different from previous datasets which commonly utilize internet photos and SfM-based 3D reconstruction, a high precision LiDAR scanner (millimeter accuracy) is used to model the environment. We capture multiple images using DSLR cameras as training set, and collect over 2000 cell phone photos taken by different users as query images. All training images are semi-automatically registered to the coordinate system defined by the LiDAR scanner. Our dataset provides accurate metric camera poses which can enable a fair comparison of different algorithms. The preliminary evaluation results reveal that high-quality training data is important to localization accuracy. By using the LiDAR point clouds which are much denser than SfM, the n-point-pose problem can be solved robustly with a large number of 2D-3D matching candidates.

The second contribution of the paper is a comparative study of representative IBL algorithms on our dataset. First are image retrieval based methods using state-of-the-art visual features [3, 5, 18, 23] and geometry verification [26]. Secondly, we experiment with more advanced image retrieval techniques such as RootSIFT [1], Query Expansion (AQE) [1, 8, 7], Feature Augmentation (SPAUG) [1], Selective Voting [31], and database side augmentation using virtual view synthesis. Thirdly, we test the direct matching algorithm [30] and compare it against image retrieval based methods. Lastly, we discuss some limitations and future works. This quantitative study is unique in the IBL literature and can facilitate research of a broad community in the field of visual place recognition and localization.

2. Related work

2.1. Previous datasets.

We briefly summarize the differences between the proposed dataset with previous arts commonly used in visual place recognition or localization in table 1). First of all, instead of relying on SfM, we adopt a high-precision LiDAR system to model the environment. The point cloud data in our dataset is defined in a metric world coordinate frame, and its precision as well as density is much higher than traditional SfM methods. Secondly, in contrast to our metric evaluation, most existing datasets assess the camera pose using the inlier number of RANSAC based n-point-pose solver. In practice, however, it is known that this measure is not reliable and can easily accept erroneous results [28]. Thirdly, another common problem for existing localization datasets is that the training and query images are collected from user-generated contents such as internet photos. The number of training images in previous datasets in general significantly exceeds the number of query images. The training data in our dataset is generated in a more systematic way. The cameras were mounted on a cart and the data collection process was accomplished by professional people. In contrast to using a large portion of the images to build the scene model, we pay more attention on the variation of query photos and provide more test data for researchers to validate their methods. Our 2296 query images were taken by amateurs with different cellphones. They sampled capture artifacts such as sensor noise, distortion, motion blur and lighting variations etc. Compared to existing RGB-D SLAM datasets [34], our query images are sparse and non temporal continuous. Test images were captured at random places with small field of view overlaps between training images to provide spatial diversity and difficulty. Lastly, our work also has significant differences in comparison with a recently published dataset for indoor IBL [37], which requires the 3D floor plan as input.

2.2. Image retrieval based approaches

Methods reviewed in this section take an intermediate step, i.e., image retrieval, to return a short list of candidates from database that are most relevant to the queries. In general, retrieval based approaches scale well to large datasets [11]. Since features in the database images are associated with 3D points, 2D-3D correspondences can be established by matching the query image with returned database images. Then camera pose can be estimated via the PnP solvers. As a key step, image retrieval itself is also a popular topic that has been extensively studied for decades. From the content representation point of view, image retrieval can be divided into two main categories: local feature based methods [24, 25], and global feature based methods such as Fisher vectors and vectors of locally ag-

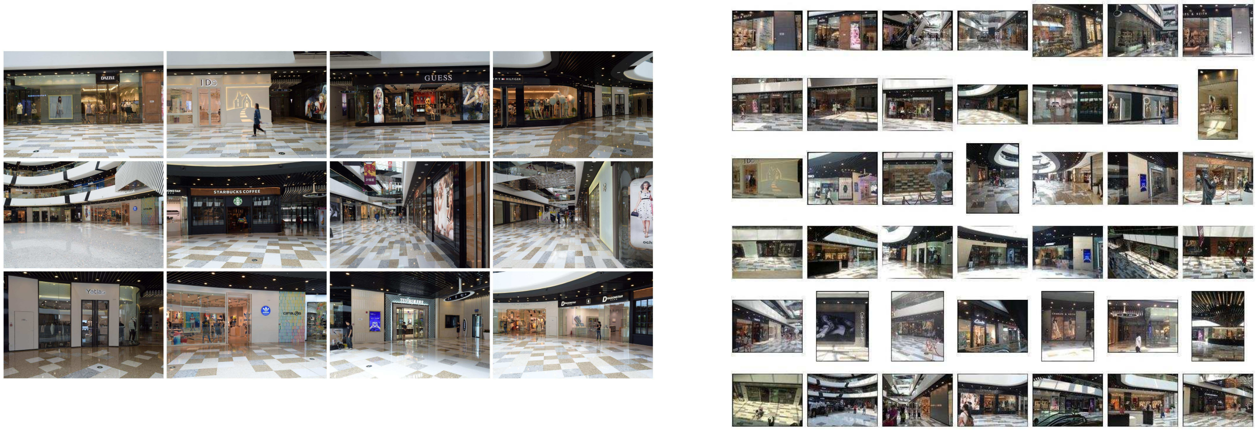


Figure 2. Illustration of our captured image data. Left: our database images where reflectance and repetitive structures are ubiquitous. Right: some query images from cell phones. Best viewed in color.

Datasets	Database	# Query	Ground truth	Evaluation
SanFrancisco PCI [6]	1.06M training images	803	building ID	recall@N
Pittsburgh [36]	254K training images	24K	GPS location	recall@N
Tokyo[35]	75984 training images	1125	GPS location	recall@N
SanFrancisco SF$-$0 [6]	610K training images	803	points from building	inliers on building
Dubrovnik [16]	1.9M 3D points, 6044 images	800	SfM pose	inlier count
Rome [16]	4.1M 3D points, 15K images	1000	SfM pose	inlier count
Vienna [11]	1.1M 3D points, 1324 images	266	SfM pose	inlier count
Aachen [31]	1.5M 3D points, 3047 images	369	SfM pose	inlier count
Proposed dataset	67M 3D points, 682 images	2296	LiDAR registered pose	metric evaluation

Table 1. Comparison of representative state-of-the-art visual place recognition (first four rows) and IBL datasets.

gregated descriptors [13, 2]. For the scope of this paper, we focus on local feature based methods because they naturally enable the searching of 2D-3D matches.

Local feature based methods represent the visual content with a bag-of-words (BoW) model, and then search for the most relevant database images using visual vocabulary [21]. In this way, the similarity between two images can be measured by the inner product of BoW vectors. Scalable indexing can be realized with inverted-files and TF-IDF weighting [21]. Due to the vocabulary quantization errors and lack of spatial constraints, a geometric verification step is usually needed to further improve image ranking. In addition, Philbin et al. built a flat k-means tree by fast approximate nearest-neighbor search [26] which performs better than hierarchical vocabulary tree [21]. Since then, progresses were continuously made in feature representation [9, 6, 1], quantization errors [12] and feature enhancement [8, 7, 1].

2.3. Direct matching based approaches

Methods fall into this category try to seek 2D-3D matches by searching nearest neighbors among all 3D points in a database (either from 2D features to 3D points [29, 30], or from 3D to 2D [16, 17]). In order to improve searching efficiency, various prioritized searching schemes were proposed [16, 29, 30]. In [16], the visibility cue was exploited to prioritize matching order and find seed points. Sattler et al. proposed a vocabulary tree prioritized scheme in which the number of features in a visual word bin is used to prioritize the matching order and the search process terminates once sufficient matches are found [29]. Recent works proposed to combine the 2D-to-3D and 3D-to-2D searching [30, 17] to further improve the registration rate.

3. Dataset construction

3.1. Sensors and data acquisition

In order to enable accurate camera registration, precise 3D modeling of the scene is essential. Traditional passive 3D reconstruction methods such as SfM are error-prone to sensor noise and scene textures therefore we chose the Riegl VZ$-$400 3D laser scanner to acquire the 3D point clouds for our dataset. It has a sensing range up to 600 meters with a measurement precision up to 5mm. For the training images, we capture them using two DSLR cameras with a resolution of 2992×1000 with a fixed focal length. Specifically, to account for large viewing angle variation, we captured the training data with two types of motion: *walking mode* with the forward motion along the corridor, and *wall mode* in which people move sideways facing the facade of stores. For query images, we collected them using cell phones. In total, 7 different cell phones were used including Apple iPhone 4S, 5S, 6S and Samsung S4 etc. Different from the training set collection process, when capturing query images, the volunteers were allowed to take photos at random positions to simulate real user scenarios. At each position, a group of 4 photos with different camera viewing angles were taken to provide more comprehensive appearance information for localization algorithms. Moreover, in order to include moderate appearance changes, query images were taken at a different time compared to the training data.

3.2. Data pre-processing

Sensor calibration. In our setup, all intrinsic parameters, i.e., focal lengths as well as principal points of each camera were pre-calibrated using the method described in [40]. For each training image, radial distortion was removed offline .
Point cloud pre-processing. To model a large scene that is over 5000 square meters, we collected multiple LiDAR scans by placing the scanner at 7 different positions and aligning the data using ICP registration. In detail, we incrementally registered the per-frame point cloud in a frame-to-model manner. Starting from two point clouds with the largest overlap, a set of point-to-point correspondences between them were manually selected. A rigid transform is coarsely initialized from selected correspondences and then refined with generalized ICP algorithm [33]. These two point clouds formed the initial 3D model and remaining frames were iteratively registered to the global model in a similar manner. Once all 7 point clouds were merged, we filtered the final 3D model with a statistical filter [27] to reduce scanning noises.

3.3. Ground truth

To help readers better understand the data generation process, we in this section introduce the method we used to register the cameras to the LiDAR 3D model. To achieve this with moderate supervision, we developed a three-step scheme that works in a semi-automatic manner. Firstly, by labelling 2D-3D correspondences manually, we localized a few "seed" cameras by evenly sampling the training images along the camera trajectories. Secondly, we used a progressive algorithm to automatically estimate the poses of remaining cameras. Lastly, all the estimated camera poses were manually examined by human and unsatisfactory results were picked and later processed manually using the 2D-3D labeling tool.

To generate the ground truth poses for "seed" cameras, we arranged several annotators to manually label the 2D-3D correspondences and perform quality reviews. The "seed" cameras are chosen in approximately constant steps along the capturing path to ensure their even distribution, and their percentage over all cameras is about 20%.

Algorithm 1 Progressive Ground truth Pose Generation

```
Input:
    Target image set $U = \{I_i\}$, reference image set $K = \{I_j\}$,
    initial known poses $P = \{P_j\}$ for $I_j \in K$, matching table
    $M = \{M_{i,j}\}$ for image pair $\{I_i, I_j\}$
Output:
    Enlarged known pose set $P^{'}$ with added images from $U$
Initialize a history table $H$ for reliable 2D-3D matches.
while (stop $\neq$ true & $U \neq \emptyset$) do
    Sort $U$ in descending order of a 2D-3D matches score.
    if the $1^{st}$ $I_i \in U$ is with more than 20 matches then
        Solve $P_i$ by a RANSAC based 3-point-pose solver
        if the $P_i$ are with an inlier number $\geq 12$ then
            Move $I_i$ from $U$ to $K$, add $P_i$ to $P$
            Update the history 2D-3D matches in $H$
        else
            Clean up records in $\{M, H\}$
            Continue
        end if
    else
        stop = true
    end if
end while
return $P^{'} = P$
```

We move on to localize the remaining cameras. Conceptually, the idea is to recursively exploit the available 2D-3D matches from already located images so far to guide the localization of unknown cameras. Our progressive localization algorithm is listed in Algorithm 1, and we briefly summarize it as follows. Starting from a matching table $M = \{M_{i,j}\}$ initialized with SIFT-matching results between all the unknown-"seed" image pairs, where each entry of M records the matched features' locations and indexes, the algorithm proceeds in the inner loop by:

- **Find the next camera to add**. Next camera $I_i \in U$ is picked out via a 2D-3D matches based ranking. Using M, we pair each 2D feature in unknown images with a 3D point if it has matches from $\{I_j\}$ where $I_j \in K$. Duplicated 3D points are stacked for each 2D feature, and the one with minimal mean re-projection error is selected in the end[*].
- **Camera pose calculation**. The camera pose is calculated with a RANSAC based 3-point-pose solver [22].
- **Update recorded information**. If the 3-point-pose

[*] Note that the inconsistency of 2D-3D matches among $\{I_j\}$ stems from the characteristic of our dataset: the 3D point is obtained by re-projecting on a quasi-dense depth map. The depth map is generated by projecting scene model onto a registered view point with Z-buffering occlusion handling, and the depth values are then bilaterally interpolated with a 11×11 window.

solver succeed to calculate pose P_i, we perform a non-linear pose optimization using the inliers. To suppress the noises, only the inliers when solving P_i are used to update the records in $\{M, H\}$.

For the "*Find the next camera to add*" step, we found that the pose ranking strategy proposed in [28] brings noticeable improvements comparing to the simple criterion with 2D-3D matches number. The intuition is that a major uncertainty for pose estimation arises from the case that many features lie in a small image region. Based on this observation, we sort the candidates using a covering area based score : suppose each 2D feature point covers a circle of radius r (equals to the re-projection error threshold used in 3-point-pose solver), then the ranking score is calculated by the total covering area of circles on a candidate image.

3.4. Evaluation metric

In this section, the covariance of our ground truth camera poses is calculated with a Monte Carlo method [19] in order to model the error of labelled poses. According to that, a standard error threshold in our metric evaluation is proposed.

For each query image data, a set of 3D scene points $\mathbf{P}_i(i = 1 \dots 20)$, are randomly sampled from all visible points inside its viewing frustum. By projecting these points with ground truth camera pose, a set of "perfect" 2D-3D matches $(\mathbf{m}_i, \mathbf{P}_i)$ are generated. Considering sensor noise and feature positioning error, a two-dimensional Gaussian noise is added to each $\mathbf{m}_i$ with a zero mean and covariance matrix $[\eta^2, 0; 0, \eta^2]$, where η is set to be $\sqrt{2}$ (in pixels and modeling the labeling errors). Every $\mathbf{P}_i$ is modeled as a three dimensional Gaussian distribution whose mean is at $\mathbf{P}_i$. The covariance of $\mathbf{P}_i$ is calculated as the covariance of all visible 3D points whose projections are in a small neighborhood of $\mathbf{m}_i$. Here we use a circle neighborhood whose radius equals to re-projection error in the 3-point-pose solver. For each query image, we repeat this simulation for 1000 rounds. Then the accuracy of our ground truth poses can be measured by the trace of the pose covariance matrices. As a result, we observe that for our ground truth poses the mean standard deviation of position and rotation is $0.098m$ and $0.6°$. In this sense, we choose $1m$ and $5°$ (position and orientation) as a standard error thresholds.

4. Experiments

We present the first qualitative evaluation on IBL algorithms using our proposed dataset. The chosen algorithms for this evaluation are popular image retrieval based and direct matching based methods. To demonstrate the advantages of our dataset, extended evaluation using a simulated SfM point cloud is also introduced. Our testing platform is a PC equipped with Intel i7 3.4GHz CPU and 16GB memories. All the implementation code is written in C/C++.

Image retrieval based method. Following the scheme proposed in [1], in training stage we resize the database image data to long side 1000 pixels before feature extraction. A flat k-means tree with $500k$ centers is then built with publicly available FLANN library [20]. When the tree is ready, all the features from database go through it to build the BoW vectors [21]. Meanwhile, the inverted indexes to database images are recorded in the tree leaves. Note that all 2D features are used in retrieval step, but only features associated with 3D points are used in localization step. In our set up, the 3D points are generated by back-projecting 2D features using depth maps. In the testing stage, features from every query image goes through the tree to search for candidate database images. Geometric verification is employed to rerank the top 10 results. For efficiency consideration, we only keep top 4 candidates in our experiments. Finally, with the 2D-3D correspondences obtained by feature matching between the query and retrieved images, the camera pose result for the query image is calculated with a RANSAC based 3-point-pose solver [22].

Taking the above approach as baseline, we further test the effects of integrating various techniques in literatures into this pipeline. First of all, we consider several features commonly used in outdoor IBL context, including BRIEF [5], SURF [3], SIFT [18] and affine covariant features(COV) [23]. For each kind of features, the visual words are trained separately. Specially for BRIEF feature, we cluster the binary centers in a slightly different way: every bit for each center is updated by the median at the same bit among all features falling into this node. The distance between binary center and feature is measured with Hamming distance. The results are presented in Table 2. As we expected, COV feature outperforms other features due to its robustness to affine variation in either outdoor or indoor environments. In addition, we notice that although binary feature descriptor BRIEF is much faster than other features, but there is a noticeable performance drop.

For the next test, we evaluate several enhanced technologies using the COV feature. All these techniques are claimed to greatly boost the image retrieval step. As shown in Table 3, here we present the registration rate gains from RootSIFT [1], averaging query expansion (AQE) [1, 8, 7], database-side feature augmentation (SPAUG) [1] and selective voting(SV) [31]. In our evaluation, not all these techniques show improvements on localization results. Query expansion, which is designed to improve recall [31], shows little effect on registration rate. SPAUG, which is thought to be complementary to AQE, also has a very limited positive impact. This could be explained by the fact that our dataset provides a richer representation model (with over $10\times$ more 2D-3D matches per query image) than traditional datasets,

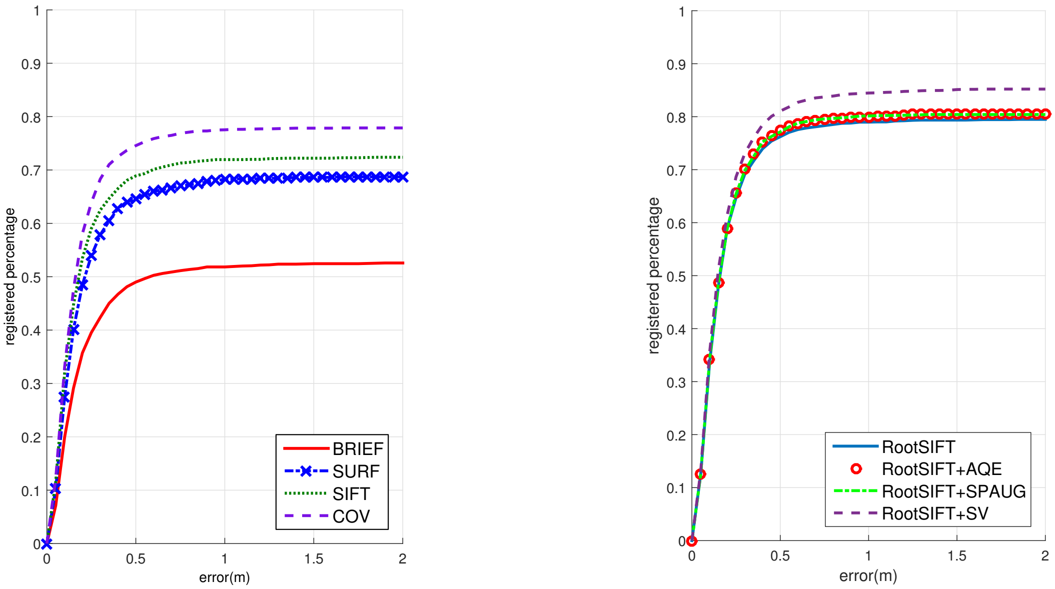


Figure 3. Per-query registration rate with different features on image retrieval based baseline (left) and enhanced techniques added to baseline (right). Varying positional and fixed angular (5°) error thresholds are used.

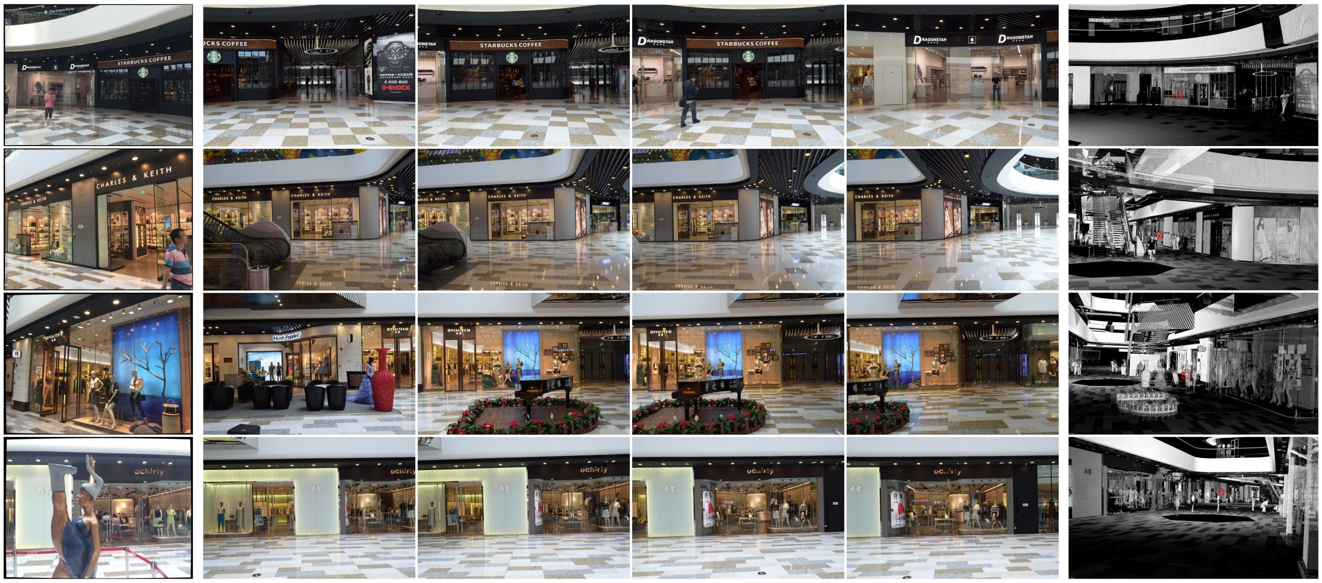


Figure 4. Example results on image retrieval based approach using our dataset. From left to right in each row: cell phone query image, returned top 4 database images and camera pose results visualized in point cloud. Best viewed in color.

hence the method to enrich the model such as AQE brings limited benefits. Last but not the least, similar to indoor scene datasets, SV shows a great benefit in registration rate (2.4% per image).

As the last test for image retrieval based method, we implemented a virtual view synthesis algorithm (VVS) to 10× enlarge our training set. To synthesize the virtual view at a particular view point, two spatially nearest source views to the novel view point are chosen. Then the two views are projected onto the novel view and angle-weighted blended to produce the virtual view result. The depth maps are used as the geometry proxy for the two source views. Thanks to our high-quality point cloud from laser scanner, we found that this simple forward warping method [4] works well in practice. Moreover, we generated the position and orientation for each virtual camera by sampling on an approximate "eye plane" (parallel to ground plane at the averaging height of database cameras). Around each database image, we ran-

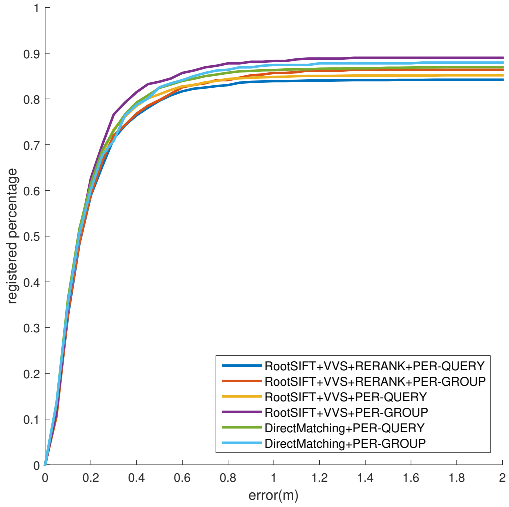


Figure 5. Registration rates for various configurations on retrieval based methods and direct matching.

Table 2. Results for different features on retrieval based baseline.

Features	BRIEF	SURF	SIFT	COV
Registration rate	47.8%	46.2%	59.2%	74.2%

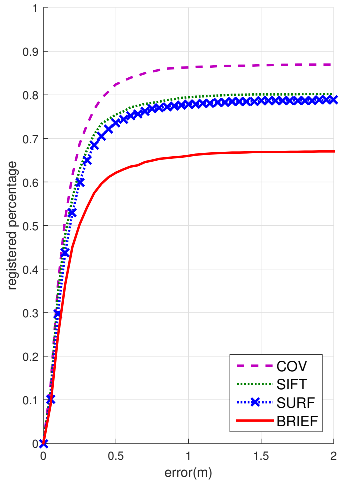


Figure 6. Results for different features on direct matching. Registration rate with varying position error thresholds are used.

domly sampled 10 virtual cameras using an offset within $2m$ and relative angle smaller than 30°.

As shown in Figure 5, the results on virtual view synthesis show consistent improvements on per image registration rate (improved by 9.4% comparing to the baseline). This can be explained by the robustness to view point changes introduced by introducing virtual views, which is made available by our dataset with high-quality (on density and accuracy) laser point cloud. For the final results, we found that a very simple configuration, i.e., "RootSIFT + VVS" provides a good trade-off on accuracy and speed: it achieves 84.8% registration rates, and costs 0.8 seconds per query.

Table 3. Results for various configurations on retrieval based approach and direct matching method.

Retrieval Method	**Registration rate**
COV + RootSIFT	75.4%
COV + RootSIFT + AQE	75.4%
COV + RootSIFT + SPAUG	75.8%
COV + RootSIFT + SV	77.8%
RootSIFT + VVS + RERANK	83.9%
RootSIFT + VVS	84.8%
Direct Matching	83.3%

Direct matching method. We implemented a direct matching method based on the active search scheme [30]. In training stage, all the image features (COV + RootSIFT) associated with 3D points are indexed in a kd-tree. In testing time, for each feature on a query image, its approximate k nearest neighbors are found as a 2D-3D match. Since in our set up the 3D points are obtained by re-projection with depth maps, we perform a "cross check" to conduct the ratio test. Specifically, for each feature from query images, assuming $\mathbf{P}_i$ is the returned i^{th} 3D point, $\mathbf{P}_i$'s associated camera is $\mathbf{C}_i$ and its feature on $\mathbf{C}_i$ is f_i. Project $\mathbf{P}_i$ to $\mathbf{C}_j$ where $j \neq i$, if the projected position is close to f_j under a threshold r (set as the re-projection error in 3-point-pose solver), then $\mathbf{P}_j$ is assumed to be a duplicate of $\mathbf{P}_i$. To perform the ratio test, we linearly search in the returned list and skip the duplicates until we find two different points. The 3D-to-2D searching is also enabled to find more correspondences. Finally, all the 2D-3D matches are used to calculate the camera pose results.

The results on direct matching are presented in Figure 5 and table 3. Comparing to retrieval based method, direct matching method delivers better registration rate with a slightly slower speed (about 1 seconds for each query image in average). The reasonable results from both the two approaches can serve as a good starting point for further researching efforts with our proposed dataset.

Extended tests. Without loss of generality, we use the retrieval based pipeline to further test two interesting scenarios. Firstly, in addition to standard tests using precise intrinsics of pre-calibrated phone cameras, we also evaluate the performance when camera intrinsics are unknown. This case is generally more common in practice. We replace the 3-point-pose solver with a 6-point-pose solver in the RANSAC framework. The results show an obvious performance drop: the registration rate decreases by 16.4% with the previously best configuration, i.e., "COV + VVS + RERANK". This result indicates that the uncalibrated scenario is more challenging for IBL.

Secondly, we also test the SfM method to build the point cloud for our dataset. Unfortunately, we found that simply

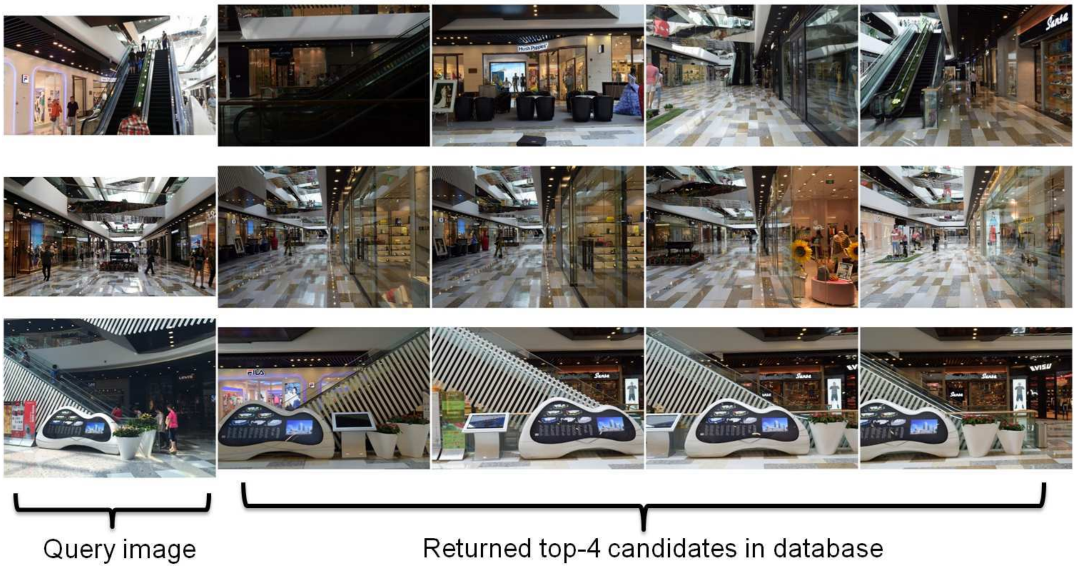


Figure 7. Illustration for failure cases. Left-most column is the mobile phone queries, and the right 4 columns are corresponding top-4 retrieved database images.

inputting our database images to the state-of-the-art SfM software *VisualSFM* [38] delivers noisy results with drastic drift. The main reason is that the widely used feature matching (e.g.SIFT) in traditional SfM can hardly handle the large amount of similar structures in our database. In viewing this, we port to use our groundtruth camera poses as a prior, and simulate the point cloud via triangulating matched features in neighboring view points. The epipolar line constraint and consistency among nearest 3 cameras are used to filter out suspicious matches. Using this method, we obtain a point cloud which is sparser than point from laser scanner. It leads to in average 80% less 2D-3D matches in localization step. As a result, the registration rate of a COV feature based baseline drops by 15%. The result supports the authors' claim that the point cloud density and accuracy is also a key to good performance of IBL algorithms.

Failure case analysis. Here we briefly summarize several typical failure cases using image retrieval results as shown in Figure 7. In short, the challenges which frequently cause localization failures are repetitive structures (elevator and white wall in first row, which have nearly identical structures from different locations), unstable depth values in forward motion (middle row, remember that our 2D-3D matches are obtained by interpolating depth values via projecting point cloud, thus larger depth range means higher uncertainty on 3D points), and misleading sign post (last row). All these cases lead to severe degrade in localization algorithms, and are common in man-made scenarios. Another typical failure case is due to the drastic appearance change between database and query images since they are recorded at different time. All these issues together leave open questions for future works on IBL algorithms.

5. Conclusion

This paper tries to shed some light on image-based localization by providing a novel dataset with LIDAR point clouds and precise 6DOF ground truth camera poses defined in the world coordinate system. The proposed dataset can be employed to qualitatively evaluate the performance of various visual localization algorithms. In addition to the dataset contribution, we also conduct a comparative study on state-of-the-art image-based localization algorithms and report our experimental results in the paper. In the future, we plan to gradually enlarge our dataset with both indoor and outdoor scenes. We hope that publishing our dataset will foster future research in the field of image-based place recognition and localization. The full dataset is available for downloading at: http://research.baidu.com/institute-of-deep-learning/ibl-dataset/.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012. 2, 3, 5

[2] R. Arandjelovic and A. Zisserman. All about VLAD. In *CVPR*, 2013. 3

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *ECCV*, pages 404–417, 2006. 2, 5

[4] C. Buehler, M. Bosse, L. Mcmillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001. 6

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *ECCV*, pages 778–792, 2010. 2, 5

[6] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR*, pages 737–744, 2011. 2, 3

[7] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, pages 889–896, 2011. 2, 3, 5

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *CVPR*, pages 1–8, 2007. 2, 3, 5

[9] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. 3

[10] R. Hartley and A. Zisserman. Multiple view geometry in computer vision, 2004. Cambridge University Press, 2nd edition. 2

[11] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, pages 2599–2606, 2009. 1, 2, 3

[12] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010. 3

[13] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010. 3

[14] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *ECCV*, 2010. 2

[15] V. Lepetit, F. Morenonoguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 2

[16] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, pages 791–804, 2010. 1, 2, 3

[17] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, pages 15–29, 2012. 1, 3

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2, 5

[19] I. Manno. Introduction to the monte-carlo method, 1999. 5

[20] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014. 5

[21] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006. 3, 5

[22] D. Nistér and H. Stewénius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007. 4, 5

[23] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, pages 9–16, 2009. 2, 5

[24] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, pages 1–8, 2007. 2

[25] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010. 2

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1–8, 2007. 2, 3

[27] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *ICRA*, pages 1–4, 2011. 4

[28] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale localtion recognition. In *ICCV*, pages 2102–2110, 2015. 2, 5

[29] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, pages 667–674, 2011. 1, 3

[30] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, pages 752–765, 2012. 1, 2, 3, 7

[31] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, pages 7–18, 2012. 1, 2, 3, 5

[32] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, pages 1–7, 2007. 2

[33] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems*, volume 2, 2009. 4

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. pages 573–580, 2012. 2

[35] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, pages 1808–1807, 2015. 3

[36] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, pages 883–890, 2013. 3

[37] S. Wang, S. Fidler, and R. Urtasun. Lost shopping! monocular localization in large indoor spaces. In *ICCV*, pages 2695–2703, 2015. 1, 2

[38] C. Wu. Visualsfm : A visual structure from motion system. http://ccwu.me/vsfm/. 8

[39] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, pages 255–268, 2010. 2

[40] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, 1999. 4