# Large-scale Localization Datasets in Crowded Indoor Spaces

Donghwan Lee[1,*], Soohyun Ryu[1,*], Suyong Yeon[1,*], Yonghan Lee[1,*], Deokhwa Kim[1], Cheolho Han[1], Yohann Cabon[2], Philippe Weinzaepfel[2], Nicolas Guérin[2], Gabriela Csurka[2], and Martin Humenberger[2]

[1]NAVER LABS, [2]NAVER LABS Europe

[1,2]{donghwan.lee, soohyun.ryu, suyong.yeon, yh.l, deokhwa.kim, cheolho.han, yohann.cabon, philippe.weinzaepfel, nicolas.guerin, gabriela.csurka, martin.humenberger}@naverlabs.com

## Abstract

*Estimating the precise location of a camera using visual localization enables interesting applications such as augmented reality or robot navigation. This is particularly useful in indoor environments where other localization technologies, such as GNSS, fail. Indoor spaces impose interesting challenges on visual localization algorithms: occlusions due to people, textureless surfaces, large viewpoint changes, low light, repetitive textures, etc. Existing indoor datasets are either comparably small or do only cover a subset of the mentioned challenges. In this paper, we introduce 5 new indoor datasets for visual localization in challenging real-world environments. They were captured in a large shopping mall and a large metro station in Seoul, South Korea, using a dedicated mapping platform consisting of 10 cameras and 2 laser scanners. In order to obtain accurate ground truth camera poses, we developed a robust LiDAR SLAM which provides initial poses that are then refined using a novel structure-from-motion based optimization. We present a benchmark of modern visual localization algorithms on these challenging datasets showing superior performance of structure-based methods using robust image features. The datasets are available at:* https://naverlabs.com/datasets

## 1. Introduction

Visual localization estimates the 6 degrees of freedom (DoF) camera pose of a given query image. It is an important task in robotics, autonomous driving, and augmented reality (AR) applications. Modern visual localization methods that utilize both, deep learning and multi-view geometry, show promising results in outdoor environments, even in the presence of dramatic changes in scene appearance [12, 31, 60, 48], and interesting real-world AR applications, *e.g*. navigation, have been developed [20]. When considering challenges for visual localization, indoor environments differ from outdoor scenes in important aspects. For example, while the lighting conditions in indoor spaces tend to be more constant, indoor spaces are typically narrower than outdoor environments, thus, small camera movements can cause large changes in viewpoint. Dynamic objects, *e.g*., people, trolleys, flat screens, textureless areas, and repetitive patterns also impose challenges for visual localization and tend to occur more often indoors. High-accuracy indoor visual localization is crucial, especially for applications such as robot navigation, where a 10cm error in position could already cause a robot falling down a stair case. Even if in practice such extreme cases are avoided

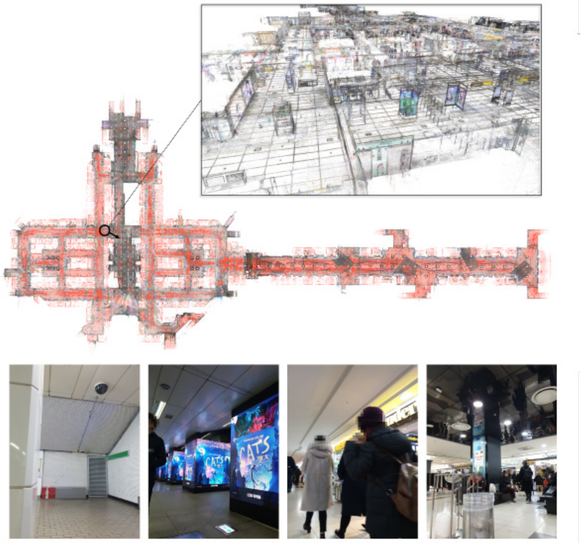


Figure 1: Illustration of the proposed NAVER LABS indoor visual localization datasets. *Top:* Point clouds from LiDAR SLAM (red) and dense reconstruction [51] (color). *Bottom:* Images showing different challenges; from left to right: textureless areas, changing appearance, crowdedness/occlusions, dynamic environment.

[*] These authors contributed equally.

with proper redundancy in the on-board sensor suite [18, 6], we want to push the capabilities of visual localization to its limits. Large-scale indoor datasets with accurate ground truth poses are thus required in order to develop and benchmark high performing (*e.g*. error below 10cm and 1°) visual localization algorithms.

Numerous datasets have been proposed for visual localization [27, 30, 49, 48, 49, 31, 25, 25, 12, 60, 34, 4, 19, 57] and Table 1 provides an overview. In outdoor environments, structure-from-motion (SFM) is employed as an effective tool to generate the ground truth poses. Unfortunately, due to the aforementioned challenges, large-scale SFM is more difficult to apply for indoor spaces [62, 56]. Thus, indoor datasets often rely on additional depth sensors such as RGB-D cameras [19, 58] and laser scanners (LiDAR) [56]. The depth range provided by RGB-D cameras is often limited to about 10m [24]. As a result, most of the existing datasets constructed using RGB-D cameras capture relatively small indoor spaces [3, 14, 11, 19, 57] and LiDARs [56] or dedicated visual 3D mapping devices [58, 11] have been used in order to create larger indoor datasets. Such devices generate panoramic images or colored point clouds that capture the structure of indoor spaces with high spacial resolution [17, 38]. In addition to high precision 3D scans, a good visual localization dataset may also consist of many images covering a large variety of views and situations. Thus, additional training and realistic query images often have to be recorded separately and they need to be registered within the generated point clouds in order to compute their ground truth poses. One solution for this, as already done for existing indoor visual localization datasets [56], is utilizing semi-automatic annotation algorithms with a significant amount of manual workload. As a consequence, the area covered by existing datasets (see Table 1) is sparsely sampled by the provided images [19, 57] and this causes difficulties when, *e.g*., training deep neural networks for visual localization [50]. We conclude that datasets that are large-scale, accurate, and densely sampled are still missing.

This motivated us to contribute another step towards closing this gap. In this paper, we introduce five new localization datasets acquired in challenging real-world indoor environments. The datasets were captured in a large shopping mall and a large metro station in Seoul, South Korea and include images and point clouds obtained from a dedicated mapping platform comprising ten cameras and two 3D laser scanners. Compared to existing indoor datasets, the proposed datasets provide dense image sampling and cover many challenges of indoor visual localization (see Figure 1). In addition, the database and the query sets were captured with time intervals of up to 128 days, thus, the datasets contain various changes over time. Since estimating precise ground truth (6DoF) camera poses for such large indoor spaces using SFM, to the best of our knowledge, was not possible so far (*e.g*. [62, 56] report failure cases), we developed a novel, automated pipeline that utilizes both, LiDAR SLAM and SFM algorithms to solve this problem. The proposed pipeline applies LiDAR-based pose-graph SLAM to estimate the trajectories of the mapping platform. The LiDAR SLAM poses are spline-smoothed and used as a prior information for the subsequent SFM process that obtains the camera poses jointly for multiple trajectories within the same space. As shown in Figure 1, the 3D models from both LiDAR SLAM and SFM are well aligned. The datasets are referred to as *NAVER LABS localization datasets* and are available at:
https://naverlabs.com/datasets

Our primary contributions are summarized as follows. *(i)* We introduce five new large-scale datasets for challenging real-world indoor environments (Section 2.4). The proposed datasets provide dense image sampling with ground truth poses as well as accurate 3D models. *(ii)* We propose a novel, fully-automated pipeline based on LiDAR SLAM and SFM to obtain ground truth poses (Section 3). *(iii)* We present detailed evaluations of modern visual localization methods of relevant algorithmic groups showing their limits on these challenging datasets (Section 4). We hope to open new interesting research directions to further improve visual localization.

## 2. The NAVER LABS localization datasets

This section first describes the characteristics of the proposed datasets as well as the considered indoor spaces, it then introduces the mapping platform we used to acquire the data, and finally, it relates our datasets with existing work.

### 2.1. Dataset description

Our datasets consist of multiple sequences captured over four months in five different floors in a department store and a metro station (see Table 1). All places were publicly accessible while scanning, thus many moving objects such as people, trolleys, luggage, as well as flat screens and store fronts were captured. Each floor is different in terms of size, statistics, and characteristics: The captured surface size varies from 5,250$m^2$ to 20,879$m^2$. The entire dataset contains over 130k images, each individual floor contains between 14k and 41k images. The 3D models, reconstructed with the proposed SFM pipeline, consist of 1.3M to 3.0M 3D points, triangulated from 4.8M to 22.7M local features (see Table 1).

The *Dept.* datasets were captured in one of the largest shopping malls in South Korea. It consists of three different floors: *Dept. B1* contains areas with restaurants, supermarkets, and cafes, *Dept. 1F* and *4F* contain cosmetic, jewelry, and clothing shops. *Dept. 1F* and *4F* were captured in good lighting conditions but include lots of reflective surfaces. *Dept. B1* was captured under low-light conditions. During

| | Datasets | Environments | Images | | Space | 3D SFM Model | | Condition | |
|---|---|---|---|---|---|---|---|---|---|
| | | | DB | Query | $(m^2)$ | Points | Observations | Interval | Crowdedness |
| Outdoor | Cambridge [27] | 5 smaller scenes | 6,848 | 4,081 | | 1.89M | 17.68M | - | - |
| | Dubrovnik [30] | large city area | 6,044 | 800 | | 1.89M | 9.61M | - | - |
| | Aachen Day-Night [48, 49, 65] | medium city area | 4,328 | 922 | | 1.65M | 10.55M | - | - |
| | Landmarks [31] | 1k landmarks | 204,626 | 10,000 | | 38.19M | 177.82M | - | - |
| | Rome [30] | 69 landmarks | 15,179 | 1000 | | 4.07M | 21.52M | - | - |
| | Vienna [25] | 3 city zones | 1,324 | 266 | | 1.12M | 4.85M | - | - |
| | San Francisco [12, 31, 60] | very large city area | 1M | 803 | | 75M | 300M | - | - |
| | RobotCar Seasons [48, 34] | 49 urban zones | 20,862 | 11,934 | | 6.77M | 36.15M | - | - |
| | CMU Seasons [48, 4] | 17 sub-urban zones | 7,159 | 75,335 | | 1.61M | 6.50M | - | - |
| Indoor | Stanford [3] | 270 small rooms | 71,909 | | 6,020 | laser scanner | | - | - |
| | ScanNet [14] | 707 small rooms | 2,492,518 | | 34,453 | RGB-D camera | | - | - |
| | Matterport3D [11] | 2,056 small rooms | 194,000 | | 46,561 | RGB-D camera | | - | - |
| | 7-scenes [19] | 7 small rooms | 26,000 | 17,000 | - | RGB-D camera | | - | $< 0.1\%$ |
| | TUM Indoor [22] | Univ.bldg. (7 floors) | 41,888 | 7,086 | 16,341 | laser scanner | | 26 days | 2.1% |
| | TUM-LSI [62] | Univ. bldg. (5 floors) | 875 | 220 | 5,575 | laser scanner | | - | - |
| | InLoc [57] | Univ. bldg. (5 floors) | 9,972 | 356 | 10,370 | laser scanner / RGB-D | | months | $< 0.1\%$ |
| | Baidu [56] | mall | 682 | 2,296 | 9,179 | laser scanner | | - | 3.75% |
| **NL (Ours)** | Dept. B1 | mall | 22,726 | 10,757 | 8,513 | 1.7M | 17.1M | 128 days | 6.87% |
| | Dept. 1F | mall | 21,600 | 5,323 | 10,046 | 1.5M | 10.6M | 128 days | 5.47% |
| | Dept. 4F | mall | 12,421 | 7,515 | 8,348 | 1.3M | 8.7M | same day | 3.45% |
| | Metro St. B1 | mall & turnstiles | 23,795 | 17,638 | 20,879 | 3.1M | 13.6M | 17 days | 12.9% |
| | Metro St. B2 | platform | 6,768 | 8,240 | 5,250 | 1.3M | 4.8M | same day | 7.73% |

Table 1: Visual localization datasets. **Interval** represents the time interval between the capture of the database and the query sets. **Crowdedness** is the percentage of pixels belonging to humans. We compute this with an off-the-shelf semantic segmentation network [13] followed by manual inspection. NL stands for NAVER LABS.

data acquisition, *Dept. 4F* was under construction, thus, a large number of images containing texture-less temporary walls are included. Rapid rotations of the mapping platform often caused motion blur in the images.

*Metro St.* was collected in one of the most crowded metro stations in Seoul and is intended for evaluating robustness of visual localization algorithms in scenes with many moving objects. The images in this dataset show the largest proportion of human pixels among the existing indoor localization datasets, including ours. Furthermore, this dataset is the largest in size and includes a variety of scenes with shops, turnstiles, and stairs. *Metro St. B2* was captured at the platforms of the metro station. The two platforms in this dataset were built with highly similar designs and internal structures as shown in Figure 5. This dataset is intended for benchmarking visual localization algorithms in repetitive and symmetric scenes. In addition, digital signage and platform screen doors, which change appearance over time, introduce new challenges.

## 2.2. Dataset format

To easily evaluate an image-based localization technique, each dataset is provided in the *kapture*[1] format [23] which supports timestamps, shared camera parameters, multi-camera rigs, reconstruction data, and other sensor data. The kapture toolbox enables to convert data into other formats such as OpenMVG [36] and COLMAP [52]. As can be seen in Table 1, our datasets consist of database images for mapping, and query images for testing. In addition, we split the query images into test and validation sets that represent different areas with little overlap (see Figure 2). The ground truth poses of the validation set are released publicly, the poses of the test set are retained to build a visual localization benchmark.

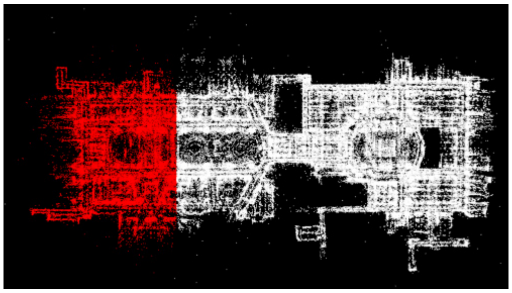
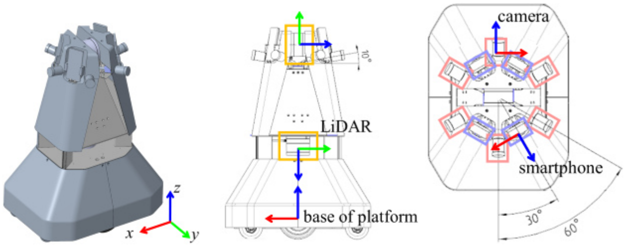


Figure 2: Example of validation (red) and test (white) splits for one of our datasets (Dept. 4F).



(a) (b) (c)

Figure 3: The mapping platform: (a) appearance, (b) side view (c), top view.

## 2.3. Mapping platform

We collected data by using the dedicated mapping platform shown in Figure 3. The mapping platform is equipped with differential drive consisting of two wheels and four

[1] https://github.com/naver/kapture

| Type | Amount | Model | Specification |
|---|---|---|---|
| 3D LiDAR | 2 | Velodyne VLP-16 | FoV: 360°×30°, angular resolution: 0.2°×2.0°, rotation rate: 10Hz, |
| camera | 6 | Basler acA2500-20gc | FoV: 79.4°×63.0°, image resolution: 2592×2048, frame rate: 2.5Hz, global shutter |
| smartphone (as camera) | 4 | Galaxy S9 | FoV: 77.0°(D), image resolution: 2160×2880, frame rate: 1Hz, rolling shutter |
| magnetic rotary position sensor (as wheel encoder) | 2 | ams AS5047 | resolution: 1024 pulses/rotation |

Table 2: Sensors of the platform.

casters. Each wheel has an encoder to obtain odometry data. The center of the virtual axle connecting two wheels is considered as the origin of the platform, the $x$-axis defines the forward direction, and the $z$-axis defines the opposite of gravity.

The platform is equipped with two LiDARs with complementary roles: One LiDAR is positioned horizontally at the center of the platform because horizontal orientation of a LiDAR sensor maximizes the information required to estimate poses in indoor spaces. The other LiDAR is positioned perpendicularly at the top of the platform to construct the dense 3D point clouds via push broom scanning.

Furthermore, the mapping platform is equipped with two different types of cameras; 6 industrial cameras (Basler acA2500-20gc) and 4 smartphone (Galaxy S9) cameras. The six industrial cameras are mounted at intervals of 60 degrees. They have a horizontal field of view (FoV) of more than 70 degrees, thus, a set of images can horizontally cover 360 degrees. Each camera is facing 10 degrees upwards in order to capture not too much ground surface. The four smartphones are mounted slightly above the industrial cameras to better reflect real-world applications where the capture device at test time will be different from the mapping platform. The platform and the sensors are synchronized using a network time protocol (NTP) server. Note that illumination can change easily with camera movement in indoor spaces, thus, we set the exposure time to automatic for all cameras. Detailed specifications of each sensor are given in Table 2.

For extrinsic calibration between the two LiDARs, we estimate the relative pose using the iterative closest point (ICP) algorithm [7, 53] using geometric features in their shared field of view. For calibration between the base and the cameras, we employ an online self-calibration via SFM rather than an offline method. This is motivate by the fact that, on the mapping platform, the smartphone cameras have irregular time delays when compared to the industrial cameras caused by automatic exposure time control and SW synchronization of Android OS. As a result, an online calibration method, which can deal with irregular time offsets, is better suited. Additional details and results are described in Section 3.2.

### 2.4. Related datasets

To complete the description of the NAVER LABS datasets, in this section we discuss a list of related work. Existing datasets often used for visual localization can be divided in indoor and outdoor (see Table 1). While both groups share the fundamental challenges of visual localization (Section 1), outdoor datasets often contain images including environmental changes, *e.g.*, seasons, weather, illumination, and indoor datasets often include images that represent occlusions, dynamic objects, and repetitive structures. Popular indoor datasets such as Stanford [3], ScanNet [14], and Matterport3D [11] target semantic understanding rather than visual localization, thus, they do not provide database and query splits. However, even indoor datasets [19, 58, 56] that are more focused on localization tasks were captured in a comparably small indoor spaces which makes them less relevant for real-world applications. Another important aspect of indoor localization is crowdedness in the sense of image regions covered by people. This is interesting because it effects both, training and testing (*i.e.* mapping and localization). As can be seen in Table 1, crowdedness has not been considered so far. In summary, large-scale datasets for visual localization are more often available outdoors, in addition, indoor datasets rarely contain people as inherent part of the environments they cover. The proposed datasets target on closing these gaps. They are captured in highly dynamic and crowded indoor spaces and contain a large variety of symmetric, non-Lambertian, and texture-less areas. Furthermore, since these datasets were captured during business hours, they better reflect real-world applications.

## 3. Ground truth generation

The ground truth poses for all images are generated using our LiDAR SLAM (Section 3.1) and SFM algorithms (Section 3.2). First, LiDAR SLAM produces poses of the mapping platform for each sequence. Then it merges sequences taken in the same space such that the sequences have a unified coordinate system. Note that the point clouds and camera poses can then be computed based on the platform poses and calibration parameters. Finally, SFM is used to refine the camera poses. This process allows us to obtain extrinsic camera calibration of the mapping platform without the need of a manual calibration process.

### 3.1. LiDAR SLAM

In order to estimate the platform poses, we developed a LiDAR SLAM pipeline based on pose-graph optimization [33]. In the pose-graph, a node represents a platform

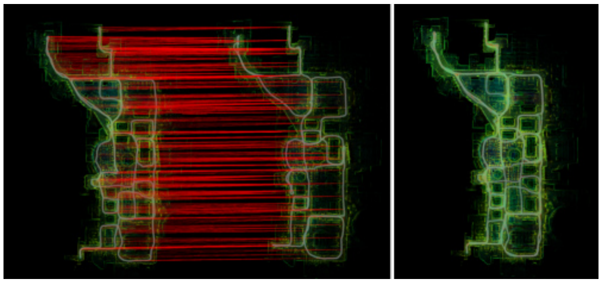


Figure 4: Pose-graph for different sequences acquired at *Dept. B1* before (left) and after (right) merging and optimization. The thick white line represents platform poses, the green dots represent the accumulated point cloud according to the poses, and the red line represents a connection between spatially adjacent nodes in different sequences.

pose and an edge between two nodes represents a relative pose constraint associated with the two platform poses.

**Undistorted point cloud generation.** To increase the amount of point cloud data used in one node, we merge point clouds collected from the two LiDARs. We transform point clouds represented in different LiDAR coordinate systems into a platform coordinate system (using the extrinsic offline calibration of the LiDARs) and concatenate them. In addition, in order to collect point clouds efficiently and evenly, we acquire point clouds by moving the platform at speeds of up to 1.0 $m/s$. The scanning LiDAR characteristics, which are similar to those of a rolling shutter camera, cause distortion of point clouds while being in motion. We employed wheel odometry to compensate for that. After calculating the transforms between platform poses at different times from wheel odometry, they are applied to the points to execute the compensation. Here, the position of a point in a point cloud in the platform coordinate system is calculated as

$$ {}^{B_t}\tilde{\mathbf{p}} = \tilde{\mathbf{R}}_{B_tB_{t+\Delta t}}(\mathbf{R}_{BL}{}^{L_{t+\Delta t}}\tilde{\mathbf{p}} + \mathbf{t}_{BL}) + \tilde{\mathbf{t}}_{B_tB_{t+\Delta t}}, \quad (1) $$

where $B_t$ and $L_t$ define the base of the mapping platform and the LiDAR at time $t$, respectively, and $[\tilde{\mathbf{R}}_{AB}|\tilde{\mathbf{t}}_{AB}]$ is an odometry measurement by the wheel encoder from $A$ to $B$. Thus, (1) transforms the LiDAR point ${}^{L_{t+\Delta t}}\tilde{\mathbf{p}}$ at time $t+\Delta t$, to the platform point ${}^{B_t}\tilde{\mathbf{p}}$.

**Pose-graph SLAM.** Since the wheel odometry is comparably inaccurate, we set the edges of the graph as relative poses estimated from ICP using undistorted sequential point clouds. However, it is difficult to compensate for accumulated pose errors using the information between sequential nodes only. To overcome this, additional edges between spatially-adjacent but time-distant nodes are calculated using ICP and added to the pose-graph. We acquire these edges by following a coarse-to-fine approach. Using a distance threshold, we first select candidates that we quickly verify with a rough ICP. Then we refine the verified candidates using a precise ICP. These edges act as additional bridges to increase robustness and enable loop closing. The platform poses are then estimated by optimizing this graph.

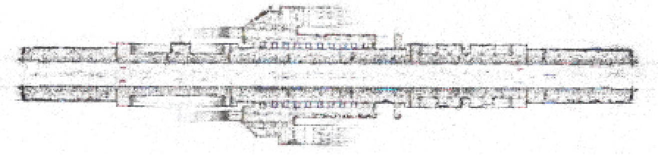


Figure 5: 3D reconstruction of Metro St. B2.

**Graph merging.** Platform poses estimated by the above procedures are represented in an independent coordinate system for each sequence. To unify the different coordinate systems, we perform a merging process between graphs from each sequence (Figure 4). As we start to scan a new sequence roughly at the same starting position like the other ones, the graphs are already coarsely aligned and loop closing (via ICP) can be performed throughout all the graphs (we needed to manually correct only a few initial poses). By optimizing the graph, we obtain the platform poses represented in a unified coordinate system.

## 3.2. Structure-from-motion

In this section, we describe the SFM pipeline used to refine the initial image poses from LiDAR SLAM and to reconstruct the 3D feature map aligned to the LiDAR point cloud. The proposed bundle adjustment utilizes the prior LiDAR SLAM poses and employs incremental mapping of the COLMAP library [51] to perform accurate and efficient large-scale indoor SFM. The intrinsic and rotational extrinsic parameters were also auto-calibrated for each dataset during the SFM process.

**Local feature matching.** We extracted R2D2 [43] features for each image. Then, we matched the features efficiently using only the spatially close image pairs. More precisely, we use the initial image poses from the LiDAR SLAM process to effectively select most of the co-visible image pairs by thresholding the distance and the viewing angle difference between two images. We then filter the initial feature matches using cross-validation and reject outliers using geometric verification.

**Map optimization.** Although traditional SFM libraries, such as COLMAP [51] and Visual SFM [64], could generally be applied to construct and optimize a 3D point cloud map, we found that their standard bundle adjustment implementation was not suited for our purpose, *i.e.*, refinement of LiDAR SLAM poses. To perform SFM precisely and efficiently in such large indoor spaces, it is crucial to embed precise prior pose information and rigid sensor configuration constraints. In the existing SFM optimizers, however, this is typically not possible if images or prior poses are not time-synchronized. Thus, we cannot apply them because

| Dataset | Images | Points | Avg. obs. | Rep. err. |
|---|---|---|---|---|
| Dept. B1 | 33,483 | 1.7M | 510.8 | 1.3614 |
| Dept. 1F | 26,923 | 1.5M | 394.5 | 1.3436 |
| Dept. 4F | 19,936 | 1.3M | 436.5 | 1.4695 |
| Metro St. B1 | 41,433 | 3.1M | 329.2 | 1.0373 |
| Metro St. B2 | 14,958 | 1.3M | 320.6 | 0.9900 |

Table 3: Mean reprojection error (Rep. err.) and mean feature observation count in the images (Avg. obs.) per dataset.

our datasets contain various types of sensors, *i.e.*, industrial cameras, smartphone cameras, and LiDARs, which are hard to synchronize.

For this reason, we developed a novel bundle adjustment algorithm which can optimize image poses, feature points, and calibration parameters, jointly. This is done by utilizing poses from the continuous trajectories of the platform that are estimated using LiDAR SLAM, as priors. To account for timestamp difference between sensors, we represented each platform trajectory as a continuous cubic spline in SE(3) [37]. With the platform trajectory spline $\mathbf{T}_{WB}(t)$ and the extrinsic pose for camera $C_i$, $\mathbf{T}_{BC_i}$, the pose error can be written as

$$\mathbf{e}_i^{spline} = \mathbf{T}_{WC_i} \ominus (\mathbf{T}_{WB}(t_i)\ \mathbf{T}_{BC_i}). \tag{2}$$

Here, $\ominus$ denotes generalized minus defined in SE(3) manifold. $\mathbf{T}_{WC_i}$ is the image pose and $t_i$ the corresponding timestamp. This continuous representation of the platform trajectory enables efficient and elegant usage of asynchronous prior poses coming from LiDAR SLAM. More importantly, rigid configuration constraints between the cameras and the platform can be embedded even though the images of two different types of cameras were not captured time-synchronously. The proposed pose error via spline can be optimized with feature reprojection error

$$\mathbf{e}_{ij}^{proj} = \tilde{\mathbf{z}}_{ij} - \pi_{C_i}(\mathbf{T}_{C_iW}\ {}^{W}\mathbf{p}_j) \tag{3}$$

to minimize the joint cost function

$$\mathcal{L} = \sum_{i \in \mathcal{I}} \left[ \rho(\|\mathbf{e}_{ij}^{proj}\|^2)\ +\ \|\mathbf{e}_i^{spline}\|^2 \right], \tag{4}$$

where $\rho$ denotes the Cauchy loss [61]. By this modified bundle adjustment method, image poses and triangulated feature points can be refined and auto-calibration of intrinsic and extrinsic parameters can be conducted.

**SFM pipeline.** We generated the SFM models by iterating our bundle adjustment and COLMAP's incremental triangulation [51], and for each iteration we decreased the outlier filtering thresholds. The initial reprojection errors using poses from LiDAR SLAM vary between 50 and 200 pixels. As shown in Table 3, our SFM achieves very low reprojection errors under 1.5 pixels. Additionally, our SFM pipeline could accurately register images with low visual saliency or crowded people by incorporating prior LiDAR SLAM results. An example of our sparse and dense 3D reconstruction models are shown in Figure 5 and Figure 1, respectively.

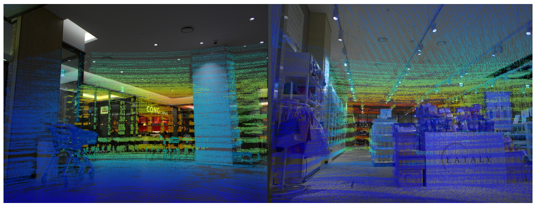


Figure 6: LiDAR scan data projected to images.

**Auto-calibration.** As mentioned above, the camera calibration parameters are automatically estimated while reconstructing the 3D model of each dataset via our SFM pipeline. To be more precise, only the intrinsic parameters and the rotational part of the extrinsic (camera to platform) parameters were auto-calibrated, the translation part of the extrinsic parameters is obtained from the CAD model. Motions of the mapping platform were much smaller (up to $1m/s$), when compared to the scale of target indoor spaces (up to $50m$). Experimentally, we found that the translation motion of the mapping platform showed a small impact on the reprojection loss term, which leads to unstable convergence of extrinsic parameters. Contrary, if we fix the translation part of extrinsic parameters, all camera parameters converge better.

## 4. Visual localization benchmark

In this section, we benchmark state-of-the-art visual localization algorithms on the proposed datasets. We selected relevant approaches representing the different algorithmic groups namely structure, image retrieval, scene coordinate regression, and absolute pose regression based methods.

### 4.1. Evaluated methods

**Structure-based methods** use an SFM model [47, 55, 26, 51, 39] as a representation of the environment in which new images can be registered in order to localize them. This is done with local feature matching using descriptors such as SIFT [32], R2D2 [43, 44], D2-Net [16] or SuperPoint [15, 46]. Image retrieval using global image representations such as DenseVLAD [59], NetVLAD [2], APGeM [42], or DELG [10] can be used to select the image pairs to match [60, 58, 45] rather than brute force matching of the entire reconstruction. Structure-based methods perform very well on a large variety of datasets but come with the burden of constructing and maintaining large 3D maps.

For our experiments, we chose a custom pipeline consisting of two stages: mapping and localization. For mapping, we start by selecting the image pairs to match using the global image representations mentioned above, we

| **Test set - *Galaxy* images** | Dept. B1 | | | Dept. 1F | | | Dept. 4F | | | Metro St. B1 | | | Metro St. B2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Algorithm / Accuracy th.** | 0.1m,1° | 0.25m,2° | 1m,5° | 0.1m,1° | 0.25m,2° | 1m,5° | 0.1m,1° | 0.25m,2° | 1m,5° | 0.1m,1° | 0.25m,2° | 1m,5° | 0.1m,1° | 0.25m,2° | 1m,5° |
| *structure-based methods* | | | | | | | | | | | | | | | |
| APGeM [42]+SIFT [32] | 69.8 | 75.7 | 81.3 | 74.0 | 78.8 | 83.2 | 83.4 | 89.2 | 91.5 | 40.8 | 59.6 | 69.5 | 39.2 | 57.8 | 62.8 |
| DELG [10]+SIFT [32] | 71.2 | 77.6 | 84.0 | 79.1 | 83.7 | 88.3 | 81.2 | 87.0 | 89.2 | 40.7 | 60.3 | 70.2 | 42.1 | 60.3 | 65.6 |
| DenseVLAD [59]+SIFT [32] | 72.2 | 77.9 | 84.5 | 82.6 | 86.9 | 92.3 | 84.6 | 89.8 | 92.3 | 40.8 | 60.4 | 69.7 | 41.8 | 59.2 | 64.2 |
| NetVLAD [2]+SIFT [32] | 72.0 | 78.2 | 84.8 | 82.2 | 86.2 | 90.3 | 86.5 | 92.0 | 94.9 | 38.6 | 55.8 | 66.3 | 41.9 | 59.9 | 65.3 |
| SIFT [32]+vocab. tree (COLMAP [52]) | 67.6 | 75.0 | 80.7 | 80.8 | 84.9 | 88.9 | 68.7 | 73.9 | 76.2 | 38.0 | 57.2 | 67.0 | 39.0 | 56.7 | 62.5 |
| APGeM [42]+D2-Net [16] | 73.7 | 79.3 | 87.2 | 78.0 | 82.8 | 88.0 | 84.2 | 89.8 | 92.0 | 44.2 | 64.8 | 75.7 | 41.3 | 60.1 | 65.0 |
| DELG [10]+D2-Net [16] | 75.3 | 81.7 | 91.6 | 84.5 | 89.2 | 94.3 | 81.8 | 87.2 | 90.0 | 44.3 | 65.4 | 76.0 | 40.8 | 60.5 | 65.9 |
| DenseVLAD [59]+D2-Net [16] | 75.7 | 81.5 | 90.5 | 85.2 | 89.7 | 95.9 | 85.0 | 90.5 | 92.9 | 44.4 | 64.8 | 74.6 | 42.3 | 60.5 | 65.1 |
| NetVLAD [2]+D2-Net [16] | 75.5 | 82.3 | 91.4 | 84.5 | 88.9 | 94.2 | 87.0 | 92.4 | 94.9 | 42.9 | 61.9 | 71.3 | 41.7 | 61.5 | 66.3 |
| APGeM [42]+R2D2 [43] | 75.2 | 80.3 | 87.6 | 80.6 | 84.3 | 89.4 | 85.3 | 91.0 | 93.1 | 46.5 | 66.8 | 76.7 | 43.2 | 62.0 | 66.3 |
| DELG [10]+R2D2 [43] | **77.1** | **82.8** | **91.9** | 86.4 | 90.7 | 95.9 | 83.6 | 89.1 | 91.8 | 46.0 | 66.6 | **77.3** | **44.0** | **62.9** | **68.0** |
| DenseVLAD [59]+R2D2 [43] | 76.4 | 81.9 | 90.7 | **87.2** | **91.7** | **97.0** | 85.7 | 91.0 | 93.6 | **46.5** | **67.0** | 76.8 | 43.8 | 61.5 | 65.8 |
| NetVLAD [2]+R2D2 [43] | 76.2 | 82.1 | 90.7 | 86.2 | 90.3 | 95.2 | **88.0** | **93.2** | **95.4** | 44.0 | 62.7 | 72.7 | 43.8 | 62.6 | 67.6 |
| *ESAC [9]* | | | | | | | | | | | | | | | |
| 1 expert | 0.0 | 0.0 | 0.5 | 0.0 | 0.4 | 9.1 | 0.0 | 0.5 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 10 experts | 0.9 | 2.9 | 6.5 | 21.1 | 46.0 | 62.1 | 20.1 | 39.5 | 62.2 | 4.8 | 17.9 | 35.8 | 5.9 | 19.2 | 33.9 |
| 20 experts | 1.7 | 3.8 | 7.8 | 34.9 | 62.6 | 78.0 | 30.5 | 47.7 | 65.7 | 7.5 | 22.0 | 38.1 | 9.3 | 23.6 | 37.5 |
| 50 experts | 3.5 | 8.2 | 12.6 | 43.3 | 66.3 | 77.0 | 45.2 | 62.5 | 73.1 | 11.4 | 31.1 | 44.7 | 11.2 | 26.1 | 39.9 |
| *PoseNet [27]* | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 0.4 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 |

Table 4: Results of various visual localization methods on the 5 NAVER LABS datasets, with the percentages of successfully localized test images within three thresholds for each datasets. The best method is shown in bold, the second best is underlined.

then extract the local keypoints and match them using cross validation and geometric verification, and finally, we compute the 3D points using COLMAP's [52] point triangulator. For localization, for each query image we first retrieve the 20 most similar database (map) images, we then extract the local keypoints and match the 20 image pairs again using cross validation and geometric verification. Since the database images are associated with 3D points from the map, we can use the resulting 2D-3D matches to compute the camera pose. For this, we use COLMAP's image triangulator. We also evaluated a traditional pipeline, fully based on COLMAP [52], with SIFT [32] features and vocabulary tree based matching. For both pipelines, we use less strict bundle adjustment parameters to maximize the percentage of localized images, even if this means that some of them might be significantly wrong (see the supplementary material for these parameters). We denote as *GLOBAL+LOCAL* the method using *GLOBAL* features for retrieval and *LOCAL* features for matching.

**Scene point regression methods** establish the 2D-3D correspondences between image pixel locations and the 3D space using random forests [54, 35] or deep neural networks [8]. The latter (notably DSAC++ [8]) report high pose accuracy on small scale environments but fail when scaled up to larger scenes [63]. To overcome this limitation, [9] proposes ESAC. The idea is to use the mixture of experts strategy to separate large scenes into smaller scenes, called experts. A gating network is used to first select a subset of experts, and second to apply each of them (*i.e.*, a DSAC++ like network) to estimate pose hypotheses. The pose hypothesis with maximum sample consensus is selected as final pose. Similarly, [29] proposes a coarse-to-fine strategy within the neural network to increase the size of environments scene point regression can be successfully used. For large-scale datasets however, [29] suggests to use image retrieval as additional conditioning.

For our experiments, we chose ESAC [9] with various numbers of experts (1, 10, 20 and 50). Note that using 1 expert is similar to DSAC++ [8] (minor differences are listed in [9]). To initialize the experts, pseudo-depth maps are required and are generated from an SFM model, here we use APGeM+R2D2. This is simply used to initialize the experts, they are then refined using a reprojection loss which does not depend on these pseudo depth maps.

**Absolute pose regression methods** use deep neural networks to directly estimate the 6DoF pose from an input image [27, 28, 5, 50]. Similar to scene point regression, these methods do not scale up well to large scenes and perform similar to pose approximation methods [50]. For our experiments, we chose PoseNet [27]. We follow [21] and train the model for 300 epochs with Adam.

## 4.2. Evaluation metric

Following the well established evaluation protocol from [48], we compute the percentage of successfully localized images within three position and orientation thresholds representing high (0.1m, 1°), medium (0.25m, 2°), and low (1m, 5°) accuracy. We chose the highest position accuracy being 10cm because, according to our experience, this is the minimum accuracy necessary for real-world applications.

## 4.3. Discussion

Table 4 shows the results of state-of-the-art visual localization methods on the Galaxy images from the test set of the 5 NAVER LABS datasets. We also provide the results on other splits (validation set and/or Basler images) in the supplementary material, as well as plots showing the percentage of localized images for varying thresholds.

First, we observe that structure-based methods significantly outperform coordinate point regression and absolute pose regression approaches. For instance, DELG+R2D2 successfully localizes more than 75% (resp. 90%) of the

| Test set<br>Galaxy images | # low freq.<br>images | performance for<br>low freq. images | performance for<br>other images |
|---|---|---|---|
| Dept. B1 | 96/2539 | 41.7 | 84.4 |
| Dept. 1F | 43/975 | 74.4 | 91.4 |
| Dept. 4F | 520/2411 | 67.1 | 95.2 |
| Metro St. B1 | 217/4334 | 49.8 | 67.5 |
| Metro St. B2 | 153/1786 | 49.7 | 64.2 |

Table 5: Impact of the absence of high frequencies (low freq. images) for DELG+R2D2. We report the percentage of successfully localized images within 25cm and 2°.

| Test set<br>Galaxy images | # crowded<br>images | performance for<br>crowded images | performance for<br>other images |
|---|---|---|---|
| Dept. B1 | 35/2539 | 77.1 | 82.9 |
| Dept. 1F | 6/ 975 | 100.0 | 90.6 |
| Dept. 4F | 10/2411 | 70.0 | 89.3 |
| Metro St. B1 | 565/4334 | 50.1 | 69.1 |
| Metro St. B2 | 169/1786 | 56.2 | 63.6 |

Table 6: Impact of crowdedness for DELG+R2D2. We report the performance as the percentage of successfully localized images within 25cm and 2°.

images at the high (resp. low) accuracy thresholds on the 3 Dept. datasets, while ESAC performs worse (below 50% at the high accuracy thresholds), and PoseNet simply fails. The performance on the 2 Metro St. datasets is lower but the overall ranking remains similar. Hypotheses for the lower performance on Metro St. include the narrower scene, the crowdedness, the symmetry of the scene (in B2, see Figure 5), and the screens changing their content regularly.

In terms of global features for image retrieval, DELG, DenseVLAD, and NetVLAD have overall similar performance, APGeM performing slightly worse, confirming the finding of [40] for the indoor dataset [56]. For local features, R2D2 often performs slightly better than D2-Net and SIFT, confirming its robustness to visual localization challenges reported in [40, 43].

For scene point regression methods, the performance with 1 expert is low. This is not surprising as it is equivalent to DSAC++ which is known to not scale to large environments [63, 29]. Using more experts allow to better handle larger datasets like ours, but the performance remains behind structure-based methods. Further increasing the number of experts does not necessarily increase the performance due to the unbalance of the clustering applied to 3D points to separate the experts' areas. The performance of ESAC is particularly low on Dept. B1, which can be led back to the blur in the images due to the low lighting conditions and the repetitive textures as well as the dynamics in food court.

Absolute pose regression does not perform well at all, due to limitations of such approaches, see [50].

In the supplementary material we report results for Basler images and we obtain significantly higher performances. This is not surprising since these images are taken with the same cameras as the training images. In particular, we observe that the performance of ESAC on the Basler test set significantly increases on 4 out of 5 datasets. This suggests that scene point regression might not be sufficiently robust to varying cameras between training and test time. We also report the performance on the validation set in the supplementary material and draw the same conclusion.

To better evaluate the impact of some specific challenges, in Table 5 and Table 6, we report DELG+R2D2 performance when considering only images without high frequency content (*i.e.*, blurry and/or with large textureless area) and crowded images, respectively. We define images without high frequency, following [41, 1]: we compute the mean absolute difference between the original image and its version without high frequencies, which is done in the Fourier domain, and threshold it at 20. From Table 5, we observe that the performance is significantly worse for low frequency images, for instance from 95.2% successfully localized images within 25cm and 2°drops down to 67.1% on Dept. 4F, the dataset with the most low frequency images due to large textureless temporary walls on a construction site. The main reason for this decrease in performance is the fact that local features used for matching mainly rely on high frequency signals both for detecting and describing the keypoints. For analyzing crowdedness, we use images where at least 20% of the pixels belong to the human class, according to a semantic segmentation approach. The results reported in Table 6 show that it also represents a significant challenge. For instance, on Metro St. B1 (largest number of crowded images) we observe a drop of performance (successfully localized images within 25cm and 2°) from 69.1% to 50.1%. This is due to the fact that humans introduce irrelevant data both for image retrieval and local feature matching. Despite these two challenges, the performance of structure-based method remain quite high, with about 50% of the images successfully localized within 25cm and 2°.

## 5. Conclusion

We introduce the NAVER LABS datasets that are first to propose large-scale densely-sampled indoor environments with challenges like dynamic objects, moving crowds, and changing scenes. We capture the data with a specific mapping platform equipped with LiDARs, cameras, and wheel odometry and propose a new pipeline based on LiDAR SLAM and SFM to generate ground truth poses. Finally, we provide a benchmark of existing methods on these datasets. We believe the NAVER LABS datasets will foster new research directions in visual localization, image retrieval, and local feature extraction and may also be used for other tasks such as depth estimation or completion, LiDAR place recognition, and map change detection.

## References

[1] https://www.pyimagesearch.com/2020/06/15/opencv-fast-fourier-transform-fft-for-blur-detection-in-images-and-video-streams/. 8

[2] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomáš Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *CVPR*, 2016. 6, 7

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 2, 3, 4

[4] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *IEEE Intelligent Vehicles Symposium*, 2011. 2, 3

[5] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation Using Neural Nets. In *ECCV*, 2018. 7

[6] Taylor E Baum, Kelilah L Wolkowicz, Joseph P Chobot, and Sean N Brennan. Negative obstacle detection using lidar sensors for a robotic wheelchair. In *Dynamic Systems and Control Conference*, 2018. 2

[7] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Trans. PAMI*, 1992. 4

[8] Eric Brachmann and Carsten Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 7

[9] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 7

[10] Bingyi Cao, André Araujo, and Jack Sim. Unifying Deep Local and Global Features for Efficient Image Search. *arXiv*, 2001.05027, 2020. 6, 7

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 2, 3, 4

[12] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale Landmark Identification on Mobile Devices. In *CVPR*, 2011. 1, 2, 3

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 2, 3, 4

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised Interest Point Detection and Description. In *CVPR*, 2018. 6

[16] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: a Trainable CNN for Joint Description and Detection of Local Features. In *CVPR*, 2019. 6, 7

[17] GeoSLAM. ZEB Discovery. https://geoslam.com/solutions/zeb-discovery. 2

[18] Muhammad Fahmi Abdul Ghani and Khairul Salleh Mohamed Sahari. Detecting negative obstacle using kinect sensor. *International Journal of Advanced Robotic Systems*, 2017. 2

[19] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. In *IEEE International Symposium on Mixed and Augmented Reality*, 2013. 2, 3, 4

[20] Google. Google AR & VR. /arvr.google.com/ar/. 1

[21] João F. Henriques and Andrea Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. In *CVPR*, 2018. 7

[22] Robert Huitl, Georg Schroth, Sebastian Hilsenbeck, Florian Schweiger, and Eckehard Steinbach. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In *ICIP*, 2012. 3

[23] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv 2007.13867*, 2020. 3

[24] Intel. Depth Camera D435. www.intelrealsense.com/depth-camera-d435/. 2

[25] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009. 2, 3

[26] Heinly Jared, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *CVPR*, 2015. 6

[27] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: a Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, 2015. 2, 3, 7

[28] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *ICCV Workshop*, 2017. 7

[29] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 7, 8

[30] Yunpeng Li, Noah Snavely, and Dan Huttenlocher. Location Recognition Using Prioritized Feature Matching. In *ECCV*, 2010. 2, 3

[31] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012. 1, 2, 3

[32] David G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *IJCV*, 2004. 6, 7

[33] Feng Lu and Evangelos Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 1997. 4

[34] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 2017. 2, 3

[35] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip H.S. Torr. Random Forests versus Neural Networks - What's Best for Camera Relocalization? In *ICRA*, 2017. 7

[36] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. OpenMVG. https://github.com/openMVG/openMVG. 3

[37] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and David Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Trans. on Robotics*, 2018. 6

[38] NavVis. NavVis M6. https://www.navvis.com/m6. 2

[39] Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 2017. 6

[40] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking Image Retrieval for Visual Localization. In *3DV*, 2020. 8

[41] Liu Renting, Li Zhaorong, and Jia Jiaya. Image partial blur detection and classification. In *CVPR*, 2008. 8

[42] Jérôme Revaud, Jon Almazan, Rafael Sampaio de Rezende, and Cesar Roberto de Souza. Learning with Average Precision: Training Image Retrieval with a Listwise Loss. In *ICCV*, 2019. 6, 7

[43] Jérôme Revaud, Philippe Weinzaepfel, César De Souza, and Martin Humenberger. R2D2: Reliable and Repeatable Detectors and Descriptors. In *NeurIPS*, 2019. 5, 6, 7, 8

[44] Jérôme Revaud, Philippe Weinzaepfel, César De Souza, Noé Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Reliable and Repeatable Detectors and Descriptors for Joint Sparse Keypoint Detection and Local Feature Extraction. *CoRR*, (arXiv:1906.06195), 2019. 6

[45] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*. 6

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020. 6

[47] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 6

[48] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DoF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018. 1, 2, 3, 7

[49] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012. 2, 3

[50] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, 2019. 2, 7, 8

[51] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *CVPR*, 2016. 1, 5, 6

[52] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 3, 7

[53] Haehnel Dirk Segal, Aleksandr V. and Sebastian Thrun. Generalized-ICP. In *RSS*, 2009. 4

[54] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *CVPR*. 7

[55] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008. 6

[56] Xun Sun, Yuanfan Xie, Pei Luo, and Liang Wang. A Dataset for Benchmarking Image-Based Localization. In *CVPR*, 2017. 2, 3, 4, 8

[57] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 2, 3

[58] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. 2019. 2, 4, 6

[59] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomáš Pajdla. 24/7 Place Recognition by View Synthesis. In *CVPR*, 2015. 6, 7

[60] Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler. Are large-scale 3d models really necessary for accurate visual localization? *IEEE Trans. PAMI*, 2019. 1, 2, 3, 6

[61] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, 1999. 6

[62] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *ICCV*, 2017. 2, 3

[63] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019. 7, 8

[64] Changchang Wu. Visualsfm: A visual structure from motion system. *URL: http://homes.cs.washington.edu/˜ ccwu/vsfm*, 2011. 5

[65] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis. *arXiv*, 2005.05179, 2020. 3