HPointLoc: Point-Based Indoor Place Recognition Using Synthetic RGB-D Images

Dmitry Yudin[1,4](✉), Yaroslav Solomentsev[1,2], Ruslan Musaev[1], Aleksei Staroverov[1,4], and Aleksandr I. Panov[1,3,4]

[1] Moscow Institute of Physics and Technology, Moscow Region, Dolgoprudny 141700, Russia
{yudin.da,panov.ai}@mipt.ru, {solomentsev.yak,musaev.rv}@phystech.edu
[2] LLC Integrant, Moscow 127495, Russia
[3] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow 117312, Russia
[4] AIRI (Artificial Intelligence Research Institute), Moscow 105064, Russia

Abstract. We present a novel dataset named as HPointLoc, specially designed for exploring capabilities of visual place recognition in indoor environment and loop detection in simultaneous localization and mapping. The loop detection sub-task is especially relevant when a robot with an on-board RGB-D camera can drive past the same place ("Point") at different angles. The dataset is based on the popular Habitat simulator, in which it is possible to generate photorealistic indoor scenes using both own sensor data and open datasets, such as Matterport3D. To study the main stages of solving the place recognition problem on the HPointLoc dataset, we proposed a new modular approach named as PNTR. It first performs an image retrieval with the Patch-NetVLAD method, then extracts keypoints and matches them using R2D2, LoFTR or SuperPoint with SuperGlue, and finally performs a camera pose optimization step with TEASER++. Such a solution to the place recognition problem has not been previously studied in existing publications. The PNTR approach has shown the best quality metrics on the HPointLoc dataset and has a high potential for real use in localization systems for unmanned vehicles. The proposed dataset and framework are publicly available: https://github.com/metra4ok/HPointLoc.

Keywords: Visual Place Recognition · Indoor Localization · Synthetic Image · RGB-D Image · Deep Learning · Dataset

1 Introduction

Place recognition based on camera images is an important task for navigation of a robot or an unmanned vehicle [28]. This can significantly reduce the cost and simplify the determining of the agent spatial pose in a 3D scene.

This work was supported by the Russian Science Foundation (Project No. 20-71-10116).

M. Tanveer et al. (Eds.): ICONIP 2022, LNCS 13625, pp. 471–484, 2023.
https://doi.org/10.1007/978-3-031-30111-7_40

Generally, visual localization has three key stages. The first stage is retrieving for a given query image the most similar image from a previously known database images based on global embeddings [3,10,22]. At the second stage, key points are extracted on the query image and on the retrieved image and matching them. Some modern methods combine these procedures into a single model [30]. Camera pose optimization is performed in the third step so that key points on the retrieved image coincide in location with the key points on the query image. Sometimes this is done in 2D [15,18], sometimes directly in 3D [38,41].

The modern trend is learning neural network-based methods that allow these stages to be performed. For them, it is important to have a diverse and large-scale dataset. It should contains, in addition to images, information about the exact camera position with 6 degrees of freedom (6DoF), and also data about distances corresponding to each pixel (depth map).

This paper focuses on the development of the dataset for indoor localization that can later be used for robot's intelligent navigation in the photorealistic simulator Habitat [27]. This will help to investigate quantitatively and qualitatively loop detection methods of a robot's movement when it enters the vicinity already visited place ("Points", see Fig. 1).

Another contribution is the development of a new modular approach named as PNTR. It first performs an image retrieval with the Patch-NetVLAD method, then extracts keypoints and matches them using R2D2, LoFTR or SuperPoint with SuperGlue, and finally performs a camera pose optimization step with TEASER++.

2 Related Work

Image Retrieval. The search problem for the closest image in the database can be reformulated as a ranking problem. The solution requires finding informative and compact local and global descriptors of the images.

The common "classical" approaches obtain global features (embeddings) by aggregating the local descriptors using the bag of words (BoW) scheme (DBoW2, DBoW3, FBoW) or vectors of locally aggregated descriptors (VLAD).

In the last few years, new approaches based on deep neural networks have been released: NetVLAD [3], distilled model HF-Net [25], Ap-Gem [22] approach with differentiable rank loss function, graph-based approach GraphVLAD [40]. They had surpassed the classical ones by feature learning for the specific problems. The similar image candidates can be also re-ranked by analyzing statistics of geometrically correct matchings of local descriptors for image patches as in Patch-NetVLAD [11].

To improve the matching of embeddings, some approaches utilize semantic information. They demonstrate good performance on popular benchmarks [20,21,37]. However, in spite of notable achievements of these neural networks, there still is a problem with the extraction of invariant semantic descriptors of images which have only low-quality semantic/instance segmentation. Such segmentation is often inherent in real-time neural networks that generate a lot of noise in the resulting masks.

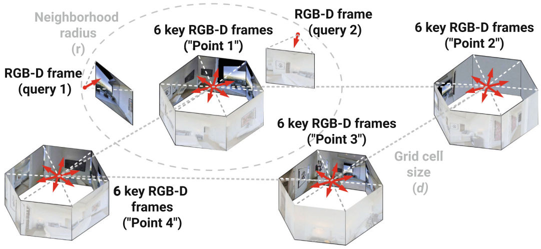
Neighborhood radius (r)
RGB-D frame (query 1)
6 key RGB-D frames ("Point 1")
RGB-D frame (query 2)
6 key RGB-D frames ("Point 2")
6 key RGB-D frames ("Point 3")
Grid cell size (d)
6 key RGB-D frames ("Point 4")

Fig. 1. Illustration of the localization problem by RGB-D query image. It is necessary to determine the pose of the corresponding camera relative to the camera poses from known database, consisting of a regular grid of groups of 6 cameras ("Points")

Local Feature Extraction and Matching. In the next stage of visual localization, we should find (detect and describe) common local features (keypoints) on pairs of images. The difficulties of this problem include getting informative and compact local descriptors and providing geometrical verification of detected keypoints.

The classical local features extraction approaches utilize scale-invariant feature transform based on local gradients [17], BoW [10], etc. These methods have achieved remarkable performance, but analogously to image retrieval methods, modern neural network approaches have outperformed them by finding more robust local features. The recent approaches demonstrate good results due to appropriate loss functions, neural network architectures (CNN, Transformers), and training schemes (Siamese networks). For example, the advantage of end-to-end training allows us to simultaneously train keypoint detectors and descriptors: SuperPoint [8], R2D2 [23], D2-Net [9], etc.

The goal of the feature matching is to find the same keypoint descriptors in both query and retrieved images. Metric learning is the most common training scheme for this purpose. Models using attention mechanisms (LoFTR [30]) and graphs (SuperGlue [26]) provide highly accurate keypoint matching. In the inference mode, models can use the k-nearest neighbors algorithm for fast and robust matching (Faiss [12]).

Camera Pose Optimization. is the final stage of the visual localization. The main problem is to find such transformation (SE3 rotation matrix and translation vector), which minimizes the pairwise distance between point clouds. The most popular methods for this task are the classical optimization approaches such as easy PnP + Ransac [18], graph-based g2o [15], 3D point cloud-based ICP [41], reliable to outliers TEASER++ [38], neural network-based methods PoseNet [13], DCP [34], etc.

Table 1. Open datasets for visual place recognition and localization

Dataset	Year	Scene	Synth.	Pose info.	RGB	Depth info.	Sem. segm.	Inst. segm.	Frames
Pittsburgh [19]	2015	outdoor		GPS	×				278k
Landmarks [19]	2016	outdoor		Label	×				10k
Google-Landmarks [19]	2017	outdoor		GPS	×				1.2M
Nordland [19]	2018	outdoor		GPS	×				143k
CMU-Seasons [1]	2018	outdoor		6DoF Pose	×	laser scan			82.5k
Aachen Day-Night [1]	2018	outdoor		6DoF Pose	×	stereo			7.5k
RobotCar Seasons [1]	2018	outdoor		6DoF Pose	×	laser scan			38k
Tokyo 24/7 [19]	2018	outdoor		GPS	×	stereo			2.8M
Argoverse Stereo [7]	2019	outdoor		6DoF Pose	×	stereo/laser scan			6,6k
NuScenes-lidarseg [5]	2020	outdoor		6DoF Pose	×	laser scan	×		40k point clouds
Waymo Perception [31]	2020	outdoor		6DoF Pose	×	stereo/laser scan			390k
KITTI360 [36]	2020	outdoor		6DoF Pose	×	laser scan	×	×	4×83k
Mapillary SLS [19]	2020	outdoor		6DoF Pose	×				1.68M
TUM Indoor [16]	2012	indoor		6DoF Pose	×	laser scan			7k
7-scenes [19]	2013	indoor		6DoF Pose	×	RGB-D cam			17k
Baidu [39]	2017	indoor		6DoF Pose	×	laser scan	×		2k
Matterport3D [6]	2017	indoor	×	Need to gen	×	3D Models	×	×	194k
TUM-LSI [16]	2017	indoor		6DoF Pose	×	laser scan			220
ScanNet [16]	2017	indoor		6DoF Pose	×	RGB-D cam	×	×	2.4M
2D-3D-Semantics [4]	2017	indoor		6DoF Pose	×	RGB-D cam	×		70k
Gibson [35]	2017	indoor	×	Need to gen	×	3D Models			572 3D Scenes
Inloc [1]	2018	indoor		6DoF Pose	×	3D point clouds			10k
Replica [29]	2019	indoor	×	Need to gen	×	3D Models	×	×	18 3D Scenes
ROI10 [33]	2020	indoor		6DoF Pose	×	RGB-D cam			250k
HM3D [2]	2021	indoor	×	Need to gen	×	3D Models			1000 3D scenes
Naver Labs [16]	2021	indoor		6DoF Pose	×	laser scan	×		100k
HPointloc (our)	2021	indoor	×	6DoF Pose	×	RGB-D cam	×	×	76k

Complex Localization Approaches. One of the most popular three-stage approach using neural networks is Hierarchical-Localization [25]. HF-Net [25] is an image retrieval method in this pipeline, SuperPoint [8] is a keypoint extracting method, SuperGlue [26] is used for keypoint matching and PnP [14] with RANSAC [18] are for pose optimization. Another state-of-the-art method on long-term visual localization benchmark [1] is Kapture framework using Ap-GeM [22] and R2D2 [23].

One of the most popular classic non-neural network visual localization method is the combination of ORB [24] and DBoW2 [10], which is widely used in popular SLAM methods, in particular, ORB-SLAM2, OpenVSLAM, etc.

Datasets. There are many known datasets used to solve the visual localization problem (see Table 1). The most popular of them are presented in the long-term visual localization challenge [1], as well as in the survey publications [16, 19]. One of the drawbacks of most of them is the lack of data on instance or semantic segmentation, or depth maps, while this information is very important for improving existing localization approaches.

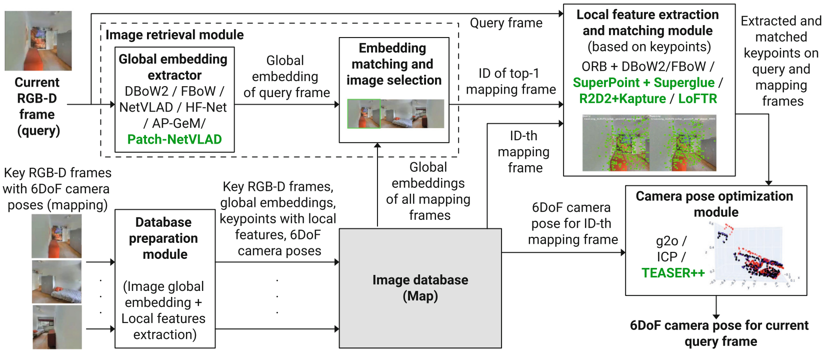
Current RGB-D frame (query)
Image retrieval module
Global embedding extractor
DBoW2 / FBoW / NetVLAD / HF-Net / AP-GeM/
Patch-NetVLAD
Global embedding of query frame
Embedding matching and image selection
Query frame
ID of top-1 mapping frame
Local feature extraction and matching module (based on keypoints)
ORB + DBoW2/FBoW /
SuperPoint + Superglue /
R2D2+Kapture / LoFTR
Extracted and matched keypoints on query and mapping frames
ID-th mapping frame
Global embeddings of all mapping frames
Key RGB-D frames with 6DoF camera poses (mapping)
Database preparation module
(Image global embedding + Local features extraction)
Key RGB-D frames, global embeddings, keypoints with local features, 6DoF camera poses
Image database (Map)
6DoF camera pose for ID-th mapping frame
Camera pose optimization module
g2o /
ICP /
TEASER++
6DoF camera pose for current query frame

Fig. 2. Scheme of the proposed framework for indoor localization using RGB-D images. Green text refers to the methods used in the PNTR approach. (Color figure online)

If we take the InLoc dataset [32] as an example, then it includes not dense depth maps, but sparse point clouds. Using them we can obtain an estimate of the depths for the found keypoints, but in our formulation of the problem, we do not consider such a case.

Of particular note are datasets that contain 3D models of a scene, rather than separate frames (Matterport3D [6], Gibson [35], Replica [29], HM3D [2]). They are usually integrated into simulators of the movement of intelligent agents, for example, into the popular and computationally efficient Habitat [27]. At the same time, to obtain a reproducible and useful result based on them, it is necessary to have sufficiently large and diverse samples from these datasets containing frames and the corresponding 6DoF camera poses.

In our work, we propose filling the gap among such datasets, specialized in the study of a specific problem—localization of an intelligent agent in the vicinity of some key point belonging to a regular grid in a 3D indoor environment.

3 The Point-Based Approach to Indoor Place Recognition

In this paper, we solve the problem of developing an approach to estimating the 6DoF pose P_q of an intelligent agent from the image I_q of its RGB-D camera (query) in the vicinity of the poses of the cameras P_{db}^i from the database (see Fig. 1), forming a regular grid of groups of 6 cameras (named as "Points"), the images I_{db}^i of which cover 360° environment. This formulation required the development of a new special dataset, named as HPointLoc. The dataset was generated automatically based on the Habitat simulator [27]. This will provide the perspective to use the results for navigation and learning the behavior of an intelligent agent (robot) in real time in this photorealistic environment. On such a "Point"-based dataset, it is advisable to evaluate the quality of the state-of-the-art methods of visual place recognition. For their study, we proposed the modular PNTR approach, which is based on trainable methods. Its modules are described in more detail later in the paper.

Image Retrieval Module. When developing the approach, various classical image retrieval models based on a bag of words and neural network approaches are investigated. They form global feature vectors (embeddings) E_q and E_{db}^i for the query image I_q and images from the database I_{db}^i. Based on embeddings, the most similar image in the sense of the similarity metric S is determined:

$$top1 = \underset{i}{\operatorname{argmin}} \left(S \left(E_q, E_{db}^i \right) \right). \tag{1}$$

We consider the classic approaches DBoW2 [10] and FBoW [42] based on the formation of global descriptors by image keypoints extracted with ORB method. The main attention is paid to neural network-based approaches of image retrieval: NetVLAD [3], AP-GeM [22], HF-Net [25] and the Patch-NetVLAD [11] (its speed up configuration (s) is included in our PNTR approach).

Local Feature Extraction and Matching Module. At the next stage of extraction and matching of local features, k matches of keypoints on the query image F_q^j and the most similar from the base F_{top1}^j are found, $j = 1..k$. For each of these points, the distance to the camera in meters (depth) is known: D_q^j and D_{top1}^j, $j = 1..k$ respectively.

We examined the extraction and matching of image keypoints based on the classic popular approaches ORB with DBoW2 [10] or FBoW [42] and the popular neural networks (used in our PNTR approach) SuperPoint [8] with SuperGlue [26], R2D2 [23] with feature matching based on the Kapture toolbox, as well as monolithic LoFTR [30].

Database Preparation Module. The database used to solve the localization problem is an array of $i \in [1, N]$ camera poses P_{db}^i, the corresponding images I_{db}^i, depth maps D_{db}^i, and segmentation masks M_{db}^i (the latter may not be used in the approach). In addition, extracted and pre-calculated global image embeddings E_{db}^i and information about local features F_{db}^j on them are additionally entered into the database.

Camera Pose Optimization Module. Further, the final estimation of the 4×4 matrix of the camera pose P_q is carried out. This pose corresponds to the query image I_q. In our case, the 4×4 matrix of the camera pose P_{db}^{top1} is known for retrieved image I_{db}^{top1} from database. An optimization problem is solved for calculating the relative pose of the camera P_{top1}^q (also a 4×4 matrix) based on minimizing some functional L that takes into account the 2D coordinates F and the depth D of matched keypoints in two images:

$$P_{top1}^q = \underset{P_{top1}^q, j \in [1,k]}{\operatorname{argmin}} \left(L \left(F_q^j, D_q^j, F_{top1}^j, D_{top1}^j \right) \right). \tag{2}$$

The final pose is defined as $P_q = P_{db}^{top1} P_{top1}^q$.

The widely used classical approaches g2o [15], ICP [41] and the newer and highly accurate Teaser++ [38] (used in the proposed PNTR approach) are investigated as methods for optimizing pose in our framework.

Table 2. Summary for HPointLoc Dataset.

	Points	Poses	Categories	Instances	Maps
HPointLoc-Val	23	1088	33	3266	1
HPointLoc-All	1757	86678	41	488717	49

Table 3. Statistics for semantic instances in the proposed HPointLoc-Val and HPointLoc-All datasets

Category	Instances per categ.		Category	Instances per categ.		Category	Instances per categ.	
	Val	All		Val	All		Val	All
appliances	0	1077	cushion	15	5600	shelving	18	4174
bathtub	11	621	door	1174	44500	shower	4	2190
beam	0	981	fireplace	0	1022	sink	6	1924
bed	28	5025	floor	0	52274	sofa	6	3871
blinds	0	162	furniture	8	376	stairs	15	3557
board_panel	3	145	gym equipment	10	105	stool	0	7660
cabinet	26	5383	lighting	184	8797	table	15	11886
ceiling	249	28143	mirror	16	1752	toilet	3	271
chair	48	27946	misc	180	47346	towel	17	482
chest_of_drawers	8	2123	objects	67	28902	tv_monitor	10	1190
clothes	0	191	picture	115	18945	void	10	14445
column	0	4255	plant	2	4330	wall	601	107291
counter	0	1521	railing	0	8157	window	110	19041
curtain	11	8129	seating	6	2927			

To quantitatively evaluate the quality of the overall framework, we use Recall metric with different thresholds. It is calculated as the fraction of query images which localization errors do not exceed the specified threshold, respectively for distance (translation) $\epsilon_t \in \{0.25\,\mathrm{m}, 0.5\,\mathrm{m}, 1\,\mathrm{m}, 5\,\mathrm{m}\}$ and rotation $\epsilon_r \in \{2^\circ, 5^\circ, 10^\circ, 20^\circ\}$. Such thresholds were chosen to assess the prospects for solving the problem of loop detection and place recognition in selected indoor environment using the developed approach.

4 HPointLoc Dataset

The dataset consists of 49 scenes from the Matterport3D dataset and is intended for training computer vision algorithms or testing them. It was formed according to the following algorithm.

For each scene, a regular grid with N key poses was generated $(x_{id}, y_{id}, z_{id}, yaw_{id}, pitch_{id}, roll_{id})$ with a distance of 2 m in x and y axes for one scene. A key pose has a unique id and field $status = 1$. An example of a key pose is marked in red in Fig. 3a. For each key pose, the two steps were taken:

1. Generation of five more key pose orientations with 60° step, simulating a 360° key pose camera. The view angle of each frame is 90°. An example of six key images is shown in Fig. 3d.

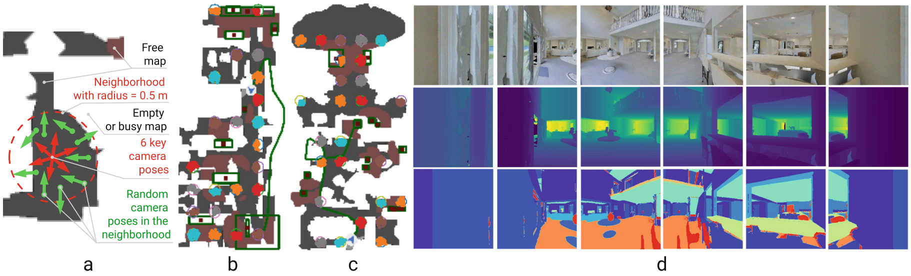
Free map
Neighborhood with radius = 0.5 m
Empty or busy map
6 key camera poses
Random camera poses in the neighborhood
a
b
c
d

Fig. 3. Explanations to the HPointLoc dataset content: a - the generation of query and database camera poses based on maps from the Matterport3D dataset in the Habitat environment; b - scene map in the HPointLoc-Val dataset; c - another scene map in the HPointLoc-All dataset, the positions of the camera centers are shown by colored dots in the circles corresponding to the "Points"; d - examples of 6 RGB-images (upper row), depth map (middle row) and map of object instance segmentation (lower row), corresponding to one "point" of scene map of the HPointLoc dataset. Six RGB-images from the "point" cover overlapping areas with a 360-degree view (Color figure online)

2. Generation of $M = 50$ random poses in a radius $r = 0.5\,\text{m}$ relative to the key pose (x_{id}, y_{id}, z_{id}) with a random orientation (shown in green in Fig. 3a). If the generated pose does not fall into the free area, then it is discarded; otherwise we add a new record to the dataset.

Each item in the dataset contains the following data: RGB and depth images, instance segmentation of the frame with 41 classes (see Table 3) and GPS with Compass data (orientation of the camera is described with a quaternion).

RGB images have 256 × 256 size and camera viewing angle 90°. Gaussian noise model was also added with factor intensity of 0.02, a mean value of 0 and a sigma parameter of 1.

Depth images also have 256 × 256 size. The depth values lie in the range from 0 to 1, where 0 is a minimum possible depth (0 m) and 1—maximum possible (10 m). Such maximum depth is chosen because most of the commercial RGB-D cameras, such as Intel RealSense, and ZED, have similar restrictions. The RGB-D camera stands at 1.25 m high above the floor.

Examples of top-view scene maps with marked frame poses are shown in Fig. 3b and 3c. Thus, in each spherical "Point" with a radius of 0.5 m, there are 6 key images with the prior known data for localization methods (images in the database) and several dozen images (query images), which need to be localized. It is assumed in the experiments that the pose of query images by which it is necessary to localize is not known to the localization method, but all other data, including the depth map, are known. A summary of the dataset contents is shown in Table 2. The dataset is split into two parts: the validation HPointLoc-Val, which contains only one scene, and the complete HPointLoc-All dataset, containing all 49 scenes, including HPointLoc-Val. We use these datasets only to validate the proposed approaches, not for training.

Table 4. Localization quality based on the pose of top-1 retrieved image (without pose optimization) on the HPointLoc-Val dataset

Image retrieval method	Embedding size	(5m,20°)	(1m,10°)	(0.5m,5°)	(0.25m,2°)	(5m)	(1m)	(0.5m)	(0.25m)
FBoW+ORB	∼1000000	0.527	0.269	0.133	0.029	0.911	0.864	0.864	0.476
DBoW2+ORB	∼1000000	0.542	0.269	0.133	0.028	0.916	0.877	0.877	0.477
AP-GeM	2048	0.559	0.282	0.138	**0.032**	0.964	0.893	0.893	0.493
NetVLAD	32768	0.584	0.291	**0.141**	**0.032**	0.971	0.919	0.919	0.502
HF-Net	4096	**0.600**	0.291	0.138	**0.032**	0.975	0.934	0.934	0.505
Patch-NetVLAD(s)	512×100 patches	0.590	**0.296**	0.140	0.030	**0.983**	**0.963**	**0.963**	0.515
Patch-NetVLAD(p)	4096×300 patches	0.576	0.292	0.138	0.028	0.979	0.961	0.961	**0.517**

5 Experimental Results

Localization Quality Estimation. The experiments were held on the HPointLoc dataset. All the methods we used were already pre-trained. According to Fig. 2 and Eq. 1, the image retrieval method searches for the closest similar image. After that, we optimize the relative pose of the retrieved image to query image (Eq. 2). Knowing the absolute pose of the image from the database, we get the final pose of the query. The quality metrics we used are generally recognized and observable on the benchmark [1]. Accuracy metrics of true localizations with the optimization step on the HPointLoc-Val dataset are given in Table 4. Pose optimization in this case is not accomplished: in other words, the result pose is equivalent to the database image that was accepted as the most similar. From Table 4, we can see that at all translation error thresholds the best method is Patch-NetVLAD with speed up configuration (s). The angle error should be taken into account secondary since the frame pose is not optimized.

The localization results of different methods on the HPointLoc-Val dataset are given in the Table 5. SuperPoint is abbreviated as SP. As we can see, the highest quality for image retrieval, keypoints matching, camera pose optimization is for the proposed PNTR approach with the R2D2 feature extraction and matching based on procedure form the Kapture tool (symbol ‘K’).

The localization results of different methods on the HPointLoc-All are given in Table 6. We can see that all metrics are worse in comparison with the evaluation on the HPointLoc-Val dataset. This directly follows from the degradation of the image retrieval methods.

In the course of the experiments, we detect problems that sometimes arise in all of the considered image retrieval methods. In Fig. 4, they are shown using the NetVLAD method as an example and demonstrate the need to explicitly use semantics and the complexity of choosing a top-1 image containing the same scene as the query image, but the corresponding camera is located at a considerable distance from the camera for images from the database.

Time Performance Estimation. Average execution time of the main stages of localization implemented in the proposed framework is shown in Table 7. The performance was evaluated on a workstation with NVidia RTX2080Ti 11 Gb GPU, AMD Threadripper 1900X (8 cores, 3.8 GHz) CPU, 64 Gb RAM.

Table 5. Quality of various visual localization methods on the HPointLoc-Val dataset

Approach	(5m,20°)	(1m,10°)	(0.5m,5°)	(0.25m,2°)	(5m)	(1m)	(0.5m)	(0.25m)
DBoW2+ORB+g2o	**0.844**	**0.812**	**0.78**	**0.726**	**0.905**	**0.857**	**0.841**	**0.796**
FBoW+ORB+g2o	0.801	0.766	0.719	0.64	0.903	0.826	0.78	0.719
AP-GeM+R2D2(K)+ICP	**0.961**	0.862	0.805	0.733	**0.965**	0.871	0.828	0.777
AP-GeM+R2D2(K)+ TEASER++	0.939	**0.904**	**0.881**	**0.877**	0.958	**0.906**	0.882	**0.880**
Ap-GeM+LofTR+g2o	0.907	0.868	0.837	0.746	0.963	0.902	**0.891**	0.861
Ap-Gem+LofTR+ICP	0.903	0.846	0.825	0.762	0.961	0.877	0.862	0.807
AP-GeM+LoFTR+ TEASER++	0.892	0.854	0.831	0.766	0.964	0.899	0.884	0.823
NetVLAD+SP+SuperGlue+g2o	0.917	0.891	0.874	0.845	0.967	0.919	0.906	0.887
NetVLAD+SP+SuperGlue+ICP	0.944	0.893	0.868	0.794	0.969	0.909	0.893	0.842
NetVLAD+SP+SuperGlue+TEASER++	0.924	0.892	0.872	0.854	0.966	0.918	0.906	0.888
NetVLAD+R2D2(K)+ICP	**0.968**	0.901	0.855	0.788	0.969	0.907	0.877	0.827
NetVlad+R2D2+TEASER++	0.941	**0.916**	**0.907**	**0.905**	0.967	0.918	0.911	**0.908**
NetVLAD+LoFTR+g2o	0.921	0.891	0.876	0.78	0.968	**0.92**	**0.916**	0.892
NetVLAD+LoFTR+ICP	0.917	0.875	0.858	0.799	**0.971**	0.9	0.892	0.843
NetVLAD+LoFTR+ TEASER++	0.908	0.881	0.865	0.806	**0.971**	0.919	0.914	0.857
HF-Net+SP+SuperGlue+g2o	0.925	0.903	0.881	0.852	0.972	0.936	0.92	0.896
HF-Net+R2D2(K)+ICP	**0.974**	0.917	0.869	0.791	**0.975**	0.925	0.894	0.834
HF-Net+R2D2(K)+ TEASER++	0.953	**0.936**	**0.923**	**0.92**	0.973	**0.938**	**0.926**	**0.923**
Patch-NetVLAD(s)+SP+SuperGlue+ g2o	0.946	0.924	0.908	0.883	**0.982**	**0.96**	0.947	0.924
Patch-NetVLAD(s)+ R2D2(K)+ICP	**0.98**	0.94	0.896	0.808	0.984	0.952	0.925	0.852
PNTR with SP+SuperGlue	0.947	0.925	0.899	0.882	0.975	0.945	0.934	0.914
PNTR with R2D2(K)	0.964	**0.957**	**0.952**	**0.945**	0.98	0.958	**0.953**	**0.951**
PNTR with LoFTR	0.942	0.919	0.900	0.834	0.977	0.953	0.948	0.892

Table 6. Quality of various visual localization methods on the HPointLoc-All dataset

Approach	(5m,20°)	(1m,10°)	(0.5m,5°)	(0.25m,2°)	(5m)	(1m)	(0.5m)	(0.25m)
DBoW2+ORB (top-1 db)	0.592	0.303	0.150	0.033	0.903	0.882	0.881	0.498
FBoW+ORB2 (top-1 db)	0.547	0.285	0.142	0.032	0.825	0.799	0.797	0.464
NetVLAD (top-1 db)	0.643	0.317	0.158	**0.034**	0.957	0.877	0.876	0.487
AP-GeM (top-1 db)	0.635	0.311	0.156	**0.034**	0.944	0.813	0.812	0.452
HF-Net (top-1 db)	**0.646**	0.318	0.158	**0.034**	0.955	0.879	0.878	0.487
Patch-NetVLAD(s) (top-1 db)	0.644	**0.320**	**0.159**	**0.034**	**0.969**	**0.944**	**0.942**	**0.516**
DBoW2+ORB2+g2o	**0.876**	**0.857**	**0.648**	**0.404**	**0.901**	**0.870**	**0.661**	**0.419**
FBoW+ORB2+g2o	0.776	0.748	0.571	0.357	0.820	0.769	0.590	0.377
NetVLAD+SP+SuperGlue+ g2o	0.925	**0.867**	0.647	0.406	0.950	**0.877**	0.656	0.414
NetVLAD+SP+SuperGlue+ICP	0.935	0.853	0.803	0.426	**0.956**	0.876	0.844	0.476
NetVLAD+R2D2(K)+ICP	**0.944**	0.842	0.748	0.341	**0.956**	**0.877**	0.846	0.474
NetVLAD+LoFTR+ICP	0.934	0.863	**0.829**	**0.458**	**0.956**	**0.877**	**0.849**	**0.480**
NetVLAD+LoFTR+TEASER++	0.933	0.865	0.829	0.445	**0.956**	**0.877**	**0.849**	0.479
Ap-Gem+LofTR+ICP	**0.907**	0.797	**0.765**	**0.423**	**0.943**	**0.813**	**0.787**	**0.445**
AP-GeM+LoFTR+ TEASER++	**0.907**	**0.799**	**0.765**	0.412	**0.943**	**0.813**	**0.787**	**0.445**
HF-Net+SP+SuperGlue+g2o	0.924	**0.868**	0.647	0.405	0.949	0.878	0.656	0.413
HF-Net+SP+SuperGlue+ICP	0.94	0.861	0.817	0.439	**0.955**	**0.879**	0.85	0.479
HF-Net+SP+SuperGlue+TEASER++	0.935	0.854	0.803	0.425	**0.955**	0.877	0.845	0.475
HF-Net+R2D2(K)+ICP	**0.943**	0.845	0.752	0.342	**0.955**	**0.879**	0.848	0.475
HF-Net+LoFTR+TEASER++	0.933	0.867	**0.831**	**0.445**	**0.955**	**0.879**	**0.851**	**0.480**
Patch-NetVLAD(s)+SP+SuperGlue+g2o	0.956	0.934	0.693	0.433	0.967	0.942	0.701	0.440
Patch-NetVLAD(s)+R2D2(K)+ ICP	**0.964**	0.931	0.876	0.464	0.968	**0.944**	0.911	0.507
PNTR with SP+SuperGlue	0.952	0.919	0.863	0.451	0.968	0.942	0.905	0.502
PNTR with R2D2(K)	0.961	**0.938**	**0.906**	**0.503**	**0.969**	0.943	**0.913**	**0.509**
PNTR with LoFTR	0.959	0.936	0.895	0.472	**0.969**	**0.944**	**0.913**	0.508

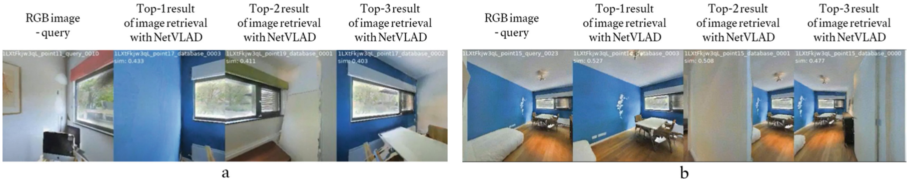
RGB image - query
Top-1 result of image retrieval with NetVLAD
Top-2 result of image retrieval with NetVLAD
Top-3 result of image retrieval with NetVLAD
1LXtFkjw3qL_point11_query_0010
1LXtFkjw3qL_point17_database_0003
sim: 0.433
1LXtFkjw3qL_point19_database_0001
sim: 0.411
1LXtFkjw3qL_point17_database_0002
sim: 0.403
a
RGB image - query
Top-1 result of image retrieval with NetVLAD
Top-2 result of image retrieval with NetVLAD
Top-3 result of image retrieval with NetVLAD
1LXtFkjw3qL_point15_query_0023
1LXtFkjw3qL_point14_database_0003
sim: 0.527
1LXtFkjw3qL_point15_database_0001
sim: 0.508
1LXtFkjw3qL_point15_database_0000
sim: 0.477
b

Fig. 4. Typical problems that sometimes arise in all image retrieval methods (for example, the NetVLAD) on the HPointLoc Dataset: a — the selection of an image that looks like a query occurs with an error caused by the lack of explicit usage of semantics (information about the presence of objects, about their color, etc.); b — the selected top-1 image contains the same scene as the query image, but is more than 1 m away from it and generates a localization error.

Table 7. Average execution time of different localization stages, sec

Image retrieval			Camera Pose Optimization	
Method	Embedding extr	Embedding match	Method	Avg. time
NetVLAD	0.00698	-	g2o	0.00074
AP-GeM	0.01542	0.00013	ICP	0.00158
HF-Net	0.09049	-	Teaser++	0.00826
Patch-NetVLAD(s)	0.01843	0.1039		
Patch-NetVLAD(p)	0.26415	2.28863		
Local feature extraction and matching			Overall PNTR approach	
Method	Feature extr	Feature match	Method	Avg. time
ORB+DBoW2	0.00407	0.00013	PNTR with SuperPoint+SuperGlue	0.13565
ORB+FBoW	0.00092	0.00007	PNTR with R2D2	0.19207
SuperPoint+SuperGlue	0.0025	0.00256	PNTR with LoFTR	0.15189
R2D2(K)	0.0585	0.00298		
LoFTR	0.0213	-		

It should be noted that the Patch-NetVLAD (s) configuration is more than 20 times faster than the Patch-NetVLAD (p) configuration, and their quality metrics are almost the same. NetVLAD leads in speed, which is 10 times faster than the almost identical in quality HF-Net method and than the significantly better Patch-NetVLAD (s).

Among the considered methods of feature extraction, the ORB FBoW method is the leader, which is almost five times faster than the classical ORB + DBoW2 method, but is significantly inferior in quality. LoFTR, the one-stage method for extracting and matching points, has the highest performance among neural network approaches, which slightly exceeds the total performance of the SuperPoint+SuperGlue method combination. Among the considered optimization methods, the fastest is the g2o method, which is twice as fast as ICP and almost 10 times faster than TEASER++. The total performance of the most accurate groups of methods, except the slow Patch-NetVLAD (p), is 5–10 FPS, which indicates the potential for their practical use in a parallel global localization stream to solve the problem of loop closure in SLAM methods.

6 Conclusions

In this work, we have shown that the proposed HPointLoc dataset allows us to visually assess the quality and performance of various approaches for solving the place recognition problem in a photorealistic indoor environment using RGB-D images. This is achieved primarily by creating a regular grid of "Points" during dataset preparation. This can be done for any real indoor environment whose model can be imported into the Habitat simulator.

On the small subset HPointLoc-Val, localization methods exhibit similar behavior, so it is sufficient to use it for quick evaluation and comparison of algorithms. The experiment results have revealed the limitations of the considered localization methods. It leads to the need of explicitly taking into account the semantics of the scene and the importance of a correct interpretation of the localization error when using one or the other image retrieval method.

State-of-the-art results were obtained for the proposed PNTR approach, which showed the highest quality on the developed dataset, but is rather slow and provides an image processing speed of 5 FPS. Nevertheless, it can be still used for loop detection in SLAM algorithms in a parallel stream to the main visual tracking procedure.

References

1. Long-Term Visual Localization. https://www.visuallocalization.net/
2. Habitat matterport dataset (2021). https://aihabitat.org/datasets/hm3d/
3. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition (2016)
4. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2D-3D-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
5. Caesar, H., et al.: nuScenes: a multimodal dataset for autonomous driving. arXiv:1903.11027 (2019)
6. Chang, A., et al.: Matterport3D: learning from RGB-D data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
7. Chang, M.F., et al.: Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8748–8757 (2019)
8. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: self-supervised interest point detection and description (2018)
9. Dusmanu, M., et al.: D2-Net: a trainable CNN for joint detection and description of local features. arXiv preprint arXiv:1905.03561 (2019)
10. Gálvez-López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Trans. Rob. **28**(5), 1188–1197 (2012)
11. Hausler, S., Garg, S., Xu, M., Milford, M., Fischer, T.: Patch-NetVLAD: multi-scale fusion of locally-global descriptors for place recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
12. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. arXiv preprint arXiv:1702.08734 (2017)

13. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DoF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
14. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: CVPR 2011, pp. 2969–2976 (2011). https://doi.org/10.1109/CVPR.2011.5995464
15. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2O: a general framework for graph optimization. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3607–3613 (2011). https://doi.org/10.1109/ICRA.2011.5979949
16. Lee, D., et al.: Large-scale localization datasets in crowded indoor spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3227–3236 (2021)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
18. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). https://doi.org/10.1145/358669.358692
19. Masone, C., Caputo, B.: A survey on deep visual place recognition. IEEE Access **9**, 19516–19547 (2021)
20. Neubert, P., Schubert, S., Schlegel, K., Protzel, P.: Vector semantic representations as descriptors for visual place recognition. In: Proceedings of Robotics: Science and Systems (RSS) (2021)
21. Peng, G., Yue, Y., Zhang, J., Wu, Z., Tang, X., Wang, D.: Semantic reinforced attention learning for visual place recognition. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 13415–13422. IEEE (2021)
22. Revaud, J., Almazan, J., de Rezende, R.S., de Souza, C.R.: Learning with average precision: training image retrieval with a listwise loss (2019)
23. Revaud, J., et al.: R2D2: repeatable and reliable detector and descriptor (2019)
24. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571 (2011). https://doi.org/10.1109/ICCV.2011.6126544
25. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale (2019)
26. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: learning feature matching with graph neural networks. In: CVPR (2020)
27. Savva, M., et al.: Habitat: a platform for embodied AI research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
28. Staroverov, A., Yudin, D.A., Belkin, I., Adeshkin, V., Solomentsev, Y.K., Panov, A.I.: Real-time object navigation with deep neural networks and hierarchical reinforcement learning. IEEE Access **8**, 195608–195621 (2020)
29. Straub, J., et al.: The replica dataset: a digital replica of indoor spaces. arXiv:1906.05797 (2019)
30. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: detector-free local feature matching with transformers (2021)
31. Sun, P., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2446–2454 (2020)

32. Taira, H., et al.: InLoc: indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7199–7209 (2018)
33. Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F.: Beyond controlled environments: 3D camera re-localization in changing indoor scenes. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 467–487. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_28
34. Wang, Y., Solomon, J.M.: Deep closest point: learning representations for point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3523–3532 (2019)
35. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson ENV: real-world perception for embodied agents. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
36. Xie, J., Kiefel, M., Sun, M.T., Geiger, A.: Semantic instance annotation of street scenes by 3D to 2D label transfer. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
37. Xue, F., Budvytis, I., Reino, D.O., Cipolla, R.: Efficient large-scale localization by global instance recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17348–17357 (2022)
38. Yang, H., Shi, J., Carlone, L.: Teaser: fast and certifiable point cloud registration. IEEE Trans. Rob. **37**(2), 314–333 (2020)
39. Yu, H., Yang, S., Gu, W., Zhang, S.: Baidu driving dataset and end-to-end reactive control model. In: 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE (2017)
40. Zhang, C., Budvytis, I., Liwicki, S., Cipolla, R.: Lifted semantic graph embedding for omnidirectional place recognition. In: 2021 International Conference on 3D Vision (3DV), pp. 1401–1410. IEEE (2021)
41. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vis. **13**(2), 119–152 (1994)
42. Zhao, R., Mao, K.: Fuzzy bag-of-words model for document representation. IEEE Trans. Fuzzy Syst. **26**(2), 794–804 (2017)