VirtualLoc: Large-Scale Visual Localization using Virtual Images

YUAN XIONG and JINGRU WANG, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, P.R.China
ZHONG ZHOU*, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, P.R.China and Zhongguancun Laboratory, P.R.China

Robust and accurate camera pose estimation is fundamental in computer vision. Learning-based regression approaches acquire 6 degree-of-freedom (DoF) camera parameters accurately from visual cues of an input image. However, most are trained on street-view and landmark datasets. These approaches can hardly be generalized to overlooking use cases, such as the calibration of the surveillance camera and unmanned aerial vehicle (UAV). Besides, reference images captured from the real world are rare and expensive, and their diversity is not guaranteed. In this paper, we address the problem of using alternative virtual images for visual localization training. This work has the following principle contributions: First, we present a new challenging localization dataset containing 6 reconstructed large-scale 3D scenes, 10594 calibrated photographs with condition changes, and 300k virtual images with pixel-wise labeled depth, relative surface normal, and semantic segmentation. Second, we present a flexible multi-feature fusion network trained on virtual image datasets for robust image retrieval. Third, we propose an end-to-end confidence map prediction network for feature filtering and pose estimation. We demonstrate that large-scale rendered virtual images are beneficial to visual localization. Using virtual images can solve the diversity problem of real images and leverage labeled multi-feature data for deep learning. Experimental results show that our method achieves remarkable performance surpassing state-of-the-art approaches. To foster research on improvement for visual localization using synthetic images, we release our benchmark at https://github.com/YuanXiong/contributions.

CCS Concepts: • **Computing methodologies** → **Camera calibration**; **Neural networks**; **Virtual reality**; • **Information systems** → **Image search**.

Additional Key Words and Phrases: visual localization, virtual reality, image retrieval, rendering

ACM Reference Format:
Yuan Xiong, Jingru Wang, and Zhong Zhou. 2023. VirtualLoc: Large-Scale Visual Localization using Virtual Images. *J. ACM* 37, 4, Article 111 (August 2023), 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Visual localization is a fundamental task in computer vision. Robust and accurate camera pose estimation is a key to many applications in digital twin. These include camera calibration in GPS-denied environments such as the surveillance camera [1] and augmented reality use cases, including mobile navigation[30], photo tourism[44], telepresence[54], and manipulation localization[15, 16].

*Corresponding author.

Authors' addresses: Yuan Xiong, xiongyuanxy@buaa.edu; Jingru Wang, jrwang999@163.com, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, 100191, P.R.China; Zhong Zhou, zz@buaa.edu.cn, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, 100191, P.R.China and Zhongguancun Laboratory, Beijing, P.R.China.

0004-5411/2023/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

It is helpful to map real images to the 3D virtual world with accurate camera poses for an immersive user experience. However, due to the frequent changes in illumination, weather, season and other environment conditions, existing work face great challenges in robustness and accuracy. This is mainly due to the appearance change of features in the query image that are significantly different from those in the dataset. In addition, mainstream visual localization approaches are trained on expensive real-image datasets with insufficient diversity. These datasets lack accurate 3D information and semantic labeling.

In general, visual localization approaches can be divided into three categories according to their use of 3D information. Image-based approaches rely on the understanding of 2D visual cues in the image. Their 3D information is either unused or used only for visualization. These include retrieval-based localization [2, 20, 51] and end-to-end learning-based pose regression [6, 24]. While point-cloud-based approaches [10, 12, 22, 34, 35, 52, 53], on the other hand, extract 3D features as input and directly map them to point clouds. This 3D information can be obtained from additional devices or multi-view stereo reconstruction. Recently, researchers have combined the advantages of 2D and 3D representations to propose a new framework: structure-based localization. These methods [14, 17, 31, 37, 39, 46, 49] match 2D features and 3D coordinates for pose estimation.

Point-cloud-based approaches are difficult to generalize because the query image to be localized may not contain depth information. Structure-based approaches extract 2D features from the input image and map them to existing 3D representations for pose optimization. They have achieved state-of-the-art performance on street view and indoor localization tasks. They usually employ image-based localization in pre-processing for coarse estimation. However, these methods can hardly be generalized to surveillance and UAV images with large elevation differences and significant appearance changes. With the development of oblique photography and Structure-from-Motion (SfM) techniques, the reconstruction of large-scale outdoor scenes becomes accessible and easy to use. As a result, we build our self-collected dataset by rendering large amounts of virtual images of reconstructed large-scale scenes. Besides, our dataset contains pixel-wise depth, relative surface normal, and semantic segmentation, providing supplementary features for different visual tasks.

In summary, the paper makes the following contributions:

- We present the NAVELoc dataset with meshed 3D models, point-wise semantic annotation, calibrated photographs, and rendered virtual images. Convincing results show that the localization performance can be improved when using virtual images for training instead of real images. To foster research on the challenging visual localization task, we release our benchmark at https://github.com/YuanXiong/contributions.
- We propose a multi-feature fusion (MFF) network which is tolerant of season, weather, and illumination changes in the stage of retrieval. The multi-branch fusion can leverage semantic and normal information to provide robust descriptors.
- We design an end-to-end network for confidence map prediction and illustrate how to use it for feature filtering. It inspirits the pose estimation to focus on stable feature points, reducing mismatches caused by viewpoint and appearance changes.

Experimental results prove that the proposed method achieves the best performance, surpassing state-of-the-art visual localization approaches on large-scale datasets.

2 RELATED WORK

Visual localization is similar to place recognition, for they both predict the camera pose of the query image from visual cues. Their differences are mainly reflected in the complexity, accuracy, and data. Tolf et al. [47] introduced commonly used datasets and discussed the performance of state-of-the-art approaches on them.

Image-based visual localization. Image-based visual localization approaches extract 2D features[11, 51] from the input image for visual localization. Recent work employ end-to-end regression[24], classification[19] or retrieval[2, 20, 51] neural networks. Since put forward, end-to-end pose regression methods, such as PoseNet[24] and MapNet, have attracted many researchers because of their ability to predict pose parameters from an input image directly. However, it performs poorly when the images to be localized are randomly scattered around in large-scale scenes. Mainstream retrieval-based localization methods[2, 20, 51] retrieve a database to obtain a group of calibrated images with the highest similarity to the query image. NetVLAD [2] is the most commonly used method because its deep learning-based descriptors are more robust than traditional methods, such as the DenseVLAD[51]. Sattler et al.[43] conduct extensive experiments to prove that current end-to-end regression approaches perform similarly to retrieval-based methods.

Table 1. Comparison with existing urban visual localization datasets. Our dataset with condition changes contains 300k+ images and 6 scenes reconstructed from high resolution photographs. We also provide absolute depth, surface normal and semantic information for virtual images.

Datasets	Capture	3D Model	Images			Localization		Condition Changes			Additional Features	
			Viewport	Train	Query	Pose	Acc.	Weather	Season	Night	Semantic	Norm.
Aachen[41]	hand	point cloud	free	4.3k	922	6DoF	m			✓		
CMU[3]	car	point cloud	sequential	60.9k	56.6k	6DoF	m	✓	✓			
RobotCar[28]	car		sequential	20.8k	11.9k	6DoF	m	✓	✓	✓		
San Francisco[42]	hand	point cloud	free	610k	0.4k	6DoF	m					
Cambridge[24]	hand	point cloud	free	6.3k		6DoF	m					
KITTI 2015[29]	car	point cloud	sequential	200	200	GPS	m				28 classes	
CityScapes[9]	car		sequential	25k		GPS	m				30 classes	
ApolloScape[21]	car	point cloud	sequential	140k		6DoF	cm	✓			28 classes	
NAVELoc(ours)	UAV	mesh	free	300k	10.6k	6DoF	cm	✓	✓		7 classes	✓

Point-cloud-based visual localization. Point-cloud-based approaches extract 3D features included in query data and map them to the reference 3D model in the dataset. The PointNet family [34, 35, 52] and its variants [22, 53] have aroused interest in point cloud analysis. Some [10, 12] include semantic segmentation for robust feature matching. These approaches usually require depth information from additional devices. Some of them work for both RGB and RGBD inputs [5]. Since depth information is hard to obtain and the spatial distribution of query images is often unknown. Additional research is needed before point-cloud-based matching algorithms can be generalized.

Structure-based localization. Early approaches, such as ActiveSearch [40], extract visual vocabulary from the image and directly match them to the SfM point cloud or prestored keypoints. These methods usually require query images to correspond with those in the dataset. Researchers have paid more attention to long-term visual localization in recent years because of its strong robustness in matching images with significant appearance changes. State-of-the-art approaches adopt a hierarchical pipeline similar to hloc[37], recover the coarse camera pose from the image representation, and then optimize it using 3D information. Such an optimization can be finished by a pose solver, such as the most commonly used RANSAC Perspective-n-Point (PnP) solver [55], which relies on the correspondence between 2D pixels in the image and 3D points in the dataset. Researchers have made considerable improvements based on this pipeline. Dusmanu et al.[13] propose the deep learning-based cross-descriptor as an alternative to conventional image representation. S2DNet[17] employs deep neural networks to improve the accuracy of correspondence feature matching. Depth prediction[50] is included to assist feature matching. Barath et al. [4]

replace the RANSAC-PnP solver with learning-based models to improve precision. MeshLoc[31] uses synthesized views for better initialization of the optimization. PixLoc[39] replaces the pose solver with a deep learning-based end-to-end pixel shift estimation network for direct pose change prediction.

Researchers include semantic segmentation for Visual localization in urban scenes to improve robustness [26, 45, 48, 49]. However, most semantic segmentation is trained on autonomous vehicle datasets[9, 21]. FGSN[25] proposes that not all pre-trained semantic segmentation benefits visual localization. Its main contribution is a self-supervised network to generate robust semantic segmentation clusters. SegLoc[33] only uses semantic features for visual localization. Although its accuracy is slightly lower than conventional color-based localization algorithms, it performs well in memory consumption and privacy preservation.

Datasets. Panek et al.[32] show that photographic models with high resolution textures are beneficial to the localization accuracy. However, such data is rare and lacks accurate labeling. As a result, we are motivated to collect our datasets with large-scale 3D models. In Table 1 we compare our dataset with existing visual localization datasets. Mainstream visual localization algorithms rely on urban datasets. Images taken by stereo cameras on street view cars are sequential. As a result, datasets[3, 9, 21, 28, 29] built on them have a strong regularity in the distribution of camera poses. Place recognition datasets with images taken from handheld cameras[24, 41, 42] are not limited to street view scenes but are usually small in size due to the high labor cost in calibration and labeling. Autonomous driving datasets [9, 21, 29] containing localization information can also be used for visual localization tasks. They usually provide semantic segmentation information in their benchmark. However, these datasets either lack condition changes or are not publicly accessible. Besides, their semantic segmentation classification needs to be simplified for visual localization tasks and often leads to incorrect retrieval results.

Regarding precision and diversity, networks trained on these datasets may not be generalized to surveillance and UAV scenes. As a result, we propose our self-collected dataset as a complement for semantic visual localization.

3 VIRTUALLOC: LOCALIZATION USING VIRTUAL IMAGES

The overview of our visual localization framework is shown in Figure 1. Our system consists of two major modules: (1) A coarse retrieval-based localization module is responsible for searching the dataset to obtain high-ranking reference images. (2) A confidence map-based filtering module is used to generate robust feature matching pairs and estimate the camera pose of the query image using the RansacPnP solver. In the retrieval stage, we fuse multiple semantic features to optimize the retrieval result. We generate confidence maps in a completely different way from PixLoc[39] and LearningRansac [4]: we use pixel-wise dense reprojection residuals to train the encoder-decoder network instead of using the feature-based keypoints filtering strategy. Like MeshLoc[31], we use virtual images instead of real images for camera calibration. The major difference between us is that our dataset exhaustively covers the entire space above the 3D scene, while MeshLoc generates images distributed around the locations of the real images. Besides, MeshLoc employs a weighted pose averaging strategy for pose estimation. We never do pose averaging. Instead, we always use the best reference image with minimum reprojection error for re-ranking the retrieval result and iteratively reducing the reprojection error. For query images with significant illumination change, we include illumination augmentation using shaders.

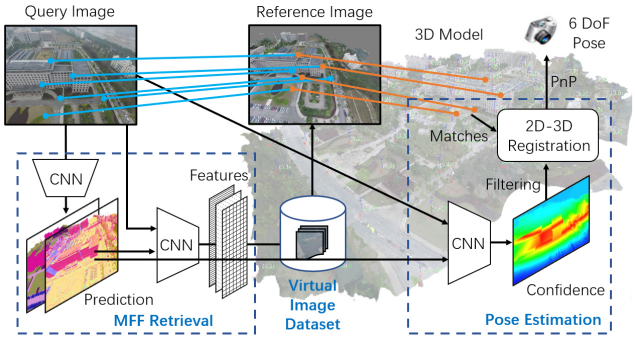
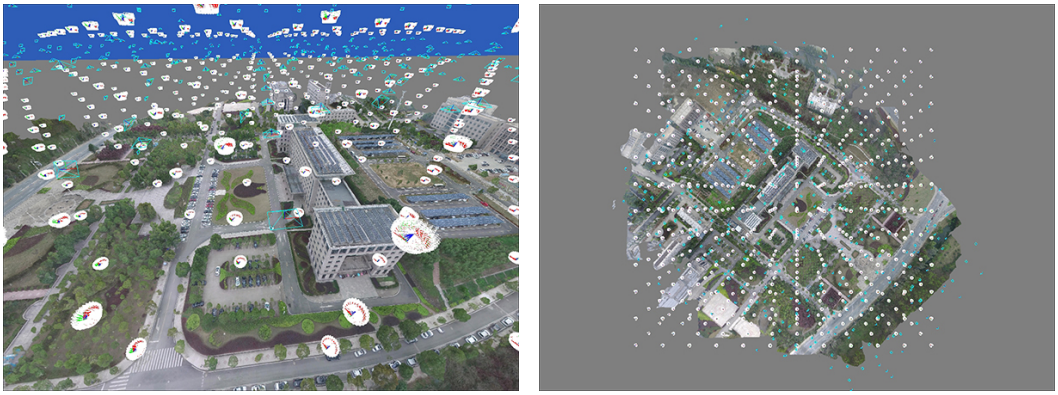


Fig. 1. **Large-Scale Visual Localization using Virtual Images (VirtualLoc)**. VirtualLoc is trained on datasets with multiple features, including texture, semantic segmentation, and relative surface normal. Our goal is to render large amounts of virtual images of urban scenes to calibrate query images, because real images are rare and expensive to capture. Experimental results show that virtual reference images are better than real images because virtual images are widely distributed, and the illumination-based augmentation can enhance feature matching, significantly improves the accuracy of pose estimation.

3.1 NAVELoc dataset

The NAVELoc dataset is labeled on our 3D platform, designed by the Networked Augmented Virtual Environment(NAVE) group. We complete point-wise annotation of large-scale 3D models and render large amounts of virtual images for training visual localization networks. A visualized example of our dataset is shown in Figure 2. In order to cover the scene, we use a dense sampling strategy to distribute virtual images.



(a) Side view (b) Top view

Fig. 2. A visualized example showing one of our scenes. The distribution of cameras is shown from different perspectives. The camera groups of virtual images are highlighted by white wireframes. The cameras of real images are highlighted by cyan wireframes. To cover the $520.41m \times 495.05m$ scene, we set up camera groups in 720 different locations. The distance between them is about 30m horizontally and 20m vertically. Each group contains 120 cameras with different rotation angles. Some cameras with bad viewpoint quality are removed. Finally, 49809 virtual images are rendered for training, and 380 real images are calibrated for testing.

Data acquisition. Our data collection system includes a DJI Phantom 4 RTK (84°FOV, 8.8mm / 24mm focal length(35mm equivalent), 8 - 1/2000s mechanical shutter and 8-1/8000s electronic shutter, 1 inch CMOS with maximum resolution 5472×3648 for photographs and 1080P for video) and a DJI Phantom 3 (94°FOV, 20mm focal length(35mm equivalent), 8 - 1/2000s mechanical shutter and 8-1/8000s electronic shutter, 1/2.3 inch CMOS with maximum resolution 4000×3000 for photographs and 1080P for video) and 3 spare batteries. The maximum flight time of DJI 4 RTK is 30 minutes (15 minutes in winter) and 25 minutes (10 minutes in winter) for DJI 3. Considering the return cost of each task, the actual flight length is 3 to 5 minutes shorter, depending on the weather and fight distance. Considering the height control and security restrictions, we limit the UAV's flight height to 50 to 200 meters. Therefore, the size of our reconstructed 3D scenes is between $0.1km$ to $1km$. We use ContextCapture for 3D reconstruction and texture mapping. For parallel acceleration and rendering optimization, we split our 3D models into planar tiles of $60m \times 60m$.

The NAVELoc dataset contains photographs collected from 6 different cities. Altogether 10594 calibrated photographs are provided as query images. Only some of them are used for reconstruction. Most of them are collected from different seasons and weather with various illumination conditions. We also rendered more than 300k virtual images as training sets. These images are distributed in different scene locations, with different parameters, including the elevation height, yaw-pitch-roll angle and illumination simulation. For query images, we provide their intrinsic parameters (focal length and distortion) and extrinsic parameters (6DoF camera poses). For virtual images in the training set, we employ GPU shaders to directly render additional channels with pixel-wise accurate semantic segmentation, absolute depth, and surface normal.

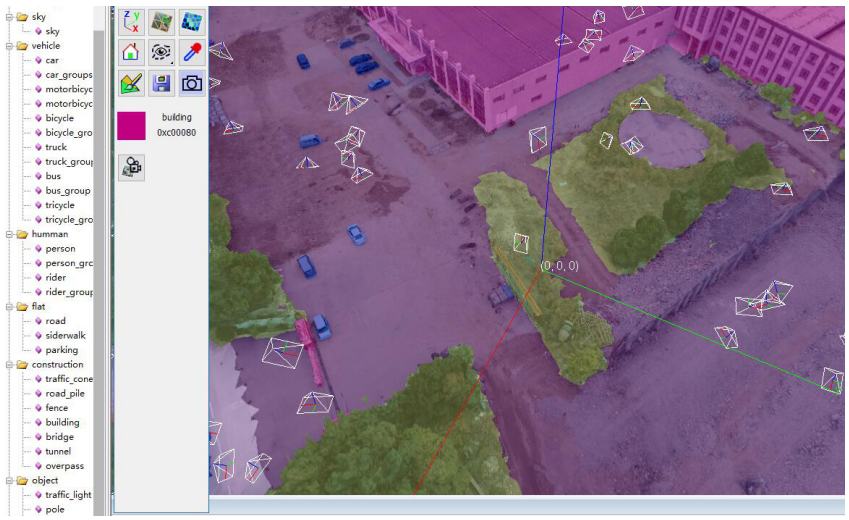


Fig. 3. The user interface of our self-developed 3D labeling and rendering platform. We design practical tools for mesh-wise labeling and massive rendering.

Classification. We follow Apolloscape[21] to create a point-wise 3D model labeling tool and manually label 7 types of semantic segmentation that are helpful to the visual localization problem. In Table 2 we list these classes in detail. Based on the experimental results, we conclude similarly to FGSN [25] that not all pre-trained semantic segmentation classifications are beneficial to visual localization. For example, extracting the semantic segmentation of pedestrians for the autonomous task is important. However, for localization tasks, pedestrians can be categorized as moving objects

and be ignored in feature matching. Another example mentioned by PixLoc [39] is that tiny objects such as poles, trash bins, and traffic signs are rare and often not visible in some cases, which can lead to overfitting of the localization network. Therefore, to improve the performance of the MFF network, it is important that apparently similar images must have similar semantic segmentation results. However, we do not follow FGSN [25] to generate more clusters through self-supervised training. Instead, we simplified our classification definition by selecting those stable scene elements and improve the semantic segmentation accuracy through supervised training.

Table 2. Details of used classes in our dataset. We only extract segmentation classes from ApolloScape[21] that are helpful to the visual localization problem. The reference colors in CityScapes[9] are also given.

Class	Category	Color	ApolloScape Class Id	Reference in CityScapes
building	construction	#c00080	97	#464646
vegetation	nature	#808040	113	#6b8e23
ground	nature	#510051	192	#510051
road	flat	#c080c0	49	#804080
car	vehicle	#00008e	33	#00008e
sky	sky	#4682b4	17	#4682b4
others	other	#000000	0	#000000

Labeling. Our color scheme for semantic segmentation is consistent with ApolloScape [21], so the semantic rendering result is well differentiated and can be directly used by retrieval networks. We develop a single document-based window application as our labeling platform, as shown in Figure 3. The user can load the model tiles and render them in the main window using embedded OpenGL dialog. Unlike point-cloud based annotation, our system considers the occlusion and orientation of meshes, which allows the user to select or deselect objects quickly. For training and labeling convenience, all moving objects are categorized as vehicles and are ignored during localization. Sidewalks and planar surfaces where vehicles are prohibited are classified as "ground". Parking lots and surfaces where vehicles can drive are classified as "road". Grass, bushes, and trees are classified as "vegetation". The "sky" label is reserved for automatic detection and generation.

3.2 Multi-feature fusion for image retrieval

The design purpose of our MFF network is to improve the robustness of retrieval results without retraining the retrieval backbone, as shown in Figure 4. The input query image usually contains only RGB color information. In order to exploit the prior knowledge, including semantic segmentation and surface normal, we have to predict them using trained deep neural networks. In our framework, we employ DeepLabv3+[7] for semantic segmentation prediction. It combines the spatial pyramid pooling module and encoder-decoder structure to predict semantic segmentation with multi-scale contextual constraints and sharp boundaries. However, the pre-trained DeepLabv3+ does not perform well on localization datasets, as shown in Figure 5. It may predict significantly different semantic segmentation results for paired similar images with wrong classification and inaccurate boundaries. As a result, we retrain it on our dataset using the standard pipeline and cross-entropy loss function with our rendered virtual images. Depth in the wild (DIW)[8] is an end-to-end relative depth estimation network. In our approach, it is adapted and retrained on the NAVELoc dataset for relative surface normal prediction with the cosine loss function. Then, we include the backbone

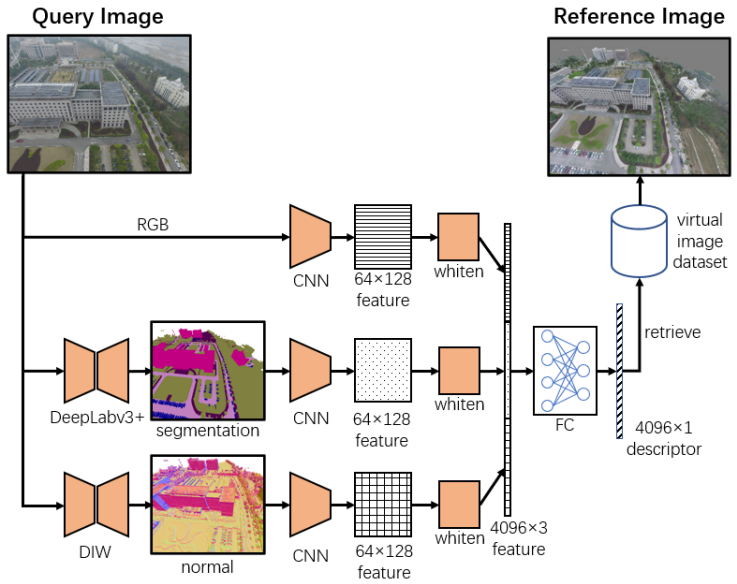


Fig. 4. Our MFF network for image retrieval.

network to compress these input channels (color, semantics, and normal) into 64×128 features. Our method is flexible and generalized, in which the backbone network can be replaced by either traditional visual bag of word approaches or mainstream deep learning-based approaches. As the experimental results Table 3 show, we found that in practice, the best backbone is NetVLAD[2]. In the MFF network, a whitening [23] process is employed to compress descriptors into 4096×1 features. Then we train a fully connected (FC) layer to merge them into a single feature. Instead of merging multiple features in the beginning for NetVLAD retraining, we use the FC layer for late feature fusion. The backbone network, such as the NetVLAD is already trained on the original dataset. Retraining may affect the generality and cause overfitting problems.

To train the MFF network, we generate tuples q, p^q, n_i^q, where q is the query image, p^q is the ground truth positive image randomly chosen from the positive groups p^q, and n_i^q is the negative group chosen from the rest of the dataset. In our dataset, we have the accurate 6DoF pose of each image, so we do not need to pick potential positive and negative images manually. Instead, we directly filter them according to their absolute pose difference to the query image, with the frequently used threshold [2] (25m/15°), which is larger than our dataset sampling interval (20m/10°). We use the triplet loss function with a margin for the ranking loss L, defined as

$$L = \sum_i l(w_i d(q, p^q) + \alpha - d(q, n_i^q)), \tag{1}$$

where $l(x) = max(x, 0)$ is the hinge function, $d(x, y)$ is the L2 distance of input descriptors x and y, $\alpha = 0.1$ is the margin constant, and w_i is the compensation weight. In our implementation, we randomly choose 1 positive image from the positive group and 10 negative images from the negative group in each iteration.

3.3 Confidence-based keypoints filtering

We train an encoder-decoder network to generate a pixel-wise confidence map, which indicates the possibility that a 2D pixel is registered to a correct 3D point. Based on our experience in large-scale

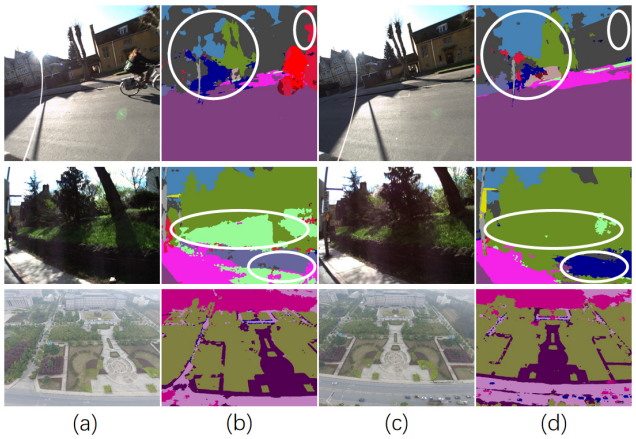


Fig. 5. Comparison of using the pre-trained semantic segmentation model on RobotCar[28] (first row) and CMU Season Ex[3] (second row) with the results of retraining on our dataset (third row). For pairs of similar images (a) and (c) and their semantic segmentation prediction (b) and (d), the CityScapes[9] pre-trained model gives different results (highlighted in white), while the results are more consistent after retraining on our dataset. Please refer to Table 2 for label definitions and color schemes.

3D reconstruction, we define that a good point is a 3D point whose reprojection residuals at different viewpoints are similar to its neighbors. On the contrary, the reprojection residuals of a bad point at different viewpoints differ from its neighbors. Using bad points as feature matching keypoints is more likely to produce large reprojection errors in the camera calibration. To train the network, we generate its ground truth confidence map.

A 2D pixel point $\boldsymbol{x_q}$ in the query image can be represented by a homogeneous vector $[x, y, -f, 1]^T$, where x and y are 2D image coordinates and f is the normalized focal length in the X-right-Y-up-Z-back camera coordinates. It can be converted to another 2D point $\boldsymbol{x_i}$ in the ith reference image viewport using the function F as

$$\boldsymbol{x_i} = F(x_q, i) = P_i M_i (d_i M_q^{-1} \boldsymbol{x_q}), \tag{2}$$

where $M = [R|t]$ denotes the model view matrix with rotation R and translation t that converts a world point in the 3D scene to a 3D point in the camera coordinates, and P is the constant perspective matrix that converts a 3D point in the camera coordinates to the 2D point in the image coordinates. The scale factor d can be obtained from the depth map. With the help of GPU rendering on the NAVELoc dataset, we can accurately obtain these parameters for every calibrated real image and all virtual images. The reprojection residual r of a pixel x between the query image and the ith reference image is defined as

$$r(x, i) = |F(x, i) - x| \tag{3}$$

Then we can calculate the confidence value C_q of a pixel using the average standard deviation of its neighbors using

$$S(x, i) = \sqrt{\frac{1}{n} \sum_{\Delta x} (r(x + \Delta x, i) - \overline{r})^2} \tag{4}$$

$$\overline{r} = \frac{1}{n} \sum_{\Delta x} r(x + \Delta x, i) \tag{5}$$

$$C(x_q) = \frac{1}{m} \sum_{i}^{m} e^{-S(x_q,i)}, \tag{6}$$

where Δx is the offset, n is the number of offsets, and m is the number of referenced images. In our implementation, we search all neighbor pixels around x_q within the range $|\Delta x| < 10$ pixels. The top 20 candidates returned by the retrieval module are used as reference images. The result of Equation 6 can be normalized to generate a ground truth confidence map.

However, the reprojection residual can not be calculated for real query images with RGB channels. Based on the observation of the residual distribution, we found that features on artificial structures and dense vegetation are robust. Moreover, surfaces perpendicular to the current view direction are less likely to be occluded when the viewpoint changes. This shows that in addition to textural features, semantic segmentation, and surface normal information also play an essential role in camera pose estimation. As a result, we design an encoder-decoder network for confidence prediction, as shown in Figure 6. The network accepts multiple features as input, where semantic segmentation and surface normal can be predicted in the stage of retrieval. We use DeepLabV3+ for feature extraction and segmentation and embedding layers for dimensionality reduction. The embedding contains a 1×1 convolutional network(CNN), 2 batch normalization (BN) layers, and 2 ReLU activation layers. To train the network, we use the generated residual confidence map as ground truth labels for supervised learning. We use the standard L1 loss function for backpropagation and Adam optimizer for acceleration. We use DeepLabV3+ as our backbone because it can retain both high-level semantic information and detailed segmentation boundaries. We initially adopted the early fusion strategy to train the network by merging multiple features.

Then, we can use the predicted confidence map to filter out feature points in low confidence areas. The confidence threshold $C < 0.1$ is determined by the result of keypoint matching experiments, as shown in Figure 11. Feature keypoints with low confidence are more likely to be wrongly matched with large reprojection errors. In our implementation, the confidence-based filter is adopted together with the RANSAC PnP solver. Unlike MeshLoc[31], we do not average poses obtained from referenced virtual images. Instead, we choose one with minimum reprojection error and re-rank the retrieval result. We iteratively optimize the pose 5 times to minimize the reprojection error.

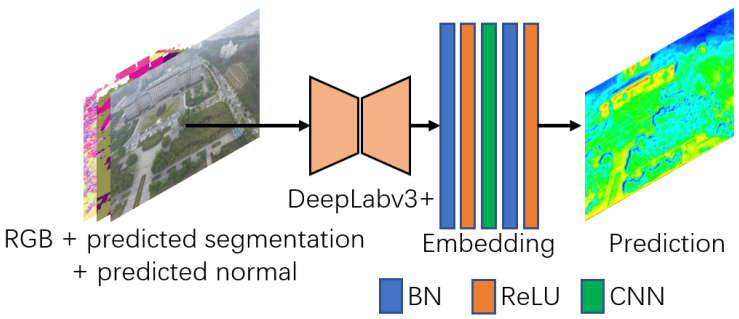
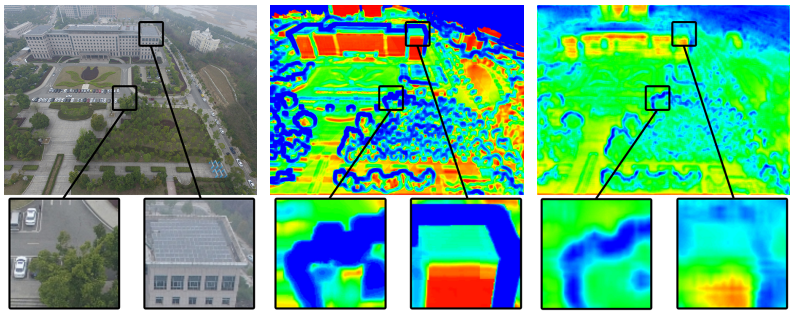


Fig. 6. The design of our confidence prediction network.



(a) Query (b) GT confidence (c) Predicted confidence

Fig. 7. Example of our end-to-end confidence map prediction. (a) query image; (b) ground truth reprojection residual-based confidence map generated by Equation 6; (c) confidence map prediction of our method. Some details are magnified for better visualization.

4 EXPERIMENTS

In this section, we conduct two experiments. First, we compare the proposed MFF network with mainstream retrieval-based localization approaches using the ablation study. Second, we compare our pose estimation with state-of-the-art localization approaches with statistics and visualization.

Dataset. As shown in Table 1, long-term localization datasets do not contain semantic annotations. The localization performance using pre-trained semantic segmentation is poor, as shown in Figure 5. On the other hand, the localization accuracy of semantic segmentation datasets depends on GPS, except for ApolloScape[21]. However, ApolloScape is not publicly accessible yet. The condition change of these datasets is insufficient. Besides, the pose of the vehicle-mounted camera does not cover different pitching angles. Most of them are sequential. Deep learning methods trained on these images may not perform well when dealing with free viewport cases. None of these datasets provide high-quality meshed 3D models and surface normal annotation. As a result, we conduct experiments on our NAVELoc datasets.

Implementation details. Our system runs on a regular Dell workstation computer and is equipped with an NVIDIA GeForce RTX 2080 Ti graphics card, 6 Intel i7-8700 CPU cores @ 3.20GHz, 32GB of RAM, and 2TB of disk. To quickly render the large-scale virtual scene and generate virtual images including texture, depth, normal and semantic segmentation, we employ GLSL shader programs running on the GPU pipeline and directly save the frame buffered images to a file. The average time from rendering to saving a virtual image is about 92.6ms, including additional information. The absolute surface normal is included in vertex data in the stage of triangulation. However, to make it learnable and generalized, we use its relative form by converting them to camera coordinates using the X-right-Y-up-Z-back coordinate system. For the retraining of semantic segmentation using DeepLabv3+[7], we set the batch size to 8 and stopped at 20 epochs. For the retraining of the surface normal network using DIW [8], we stopped after 40 epochs when there was no significant gain. For the feature fusion, we train 1000 epochs with batch size = 128. We stopped training the confidence prediction network after 300 epochs.

4.1 Retrieval-based localization experiments

Baselines. For the retrieval test, we compare NetVLAD [2], DenseVLAD [51], PatchNetVLAD [20], and DIR [18] on our dataset and the improvement included by our MFF module. All baseline

Table 3. Ablation study of our MFF network on the improvement of different approaches. We compare the recall rate of top 1, 5, 10, and 20 ranking results of state-of-the-art retrieval methods. We also compare the features fusion results with/without semantic segmentation and surface normal.

Baseline	Features	Top1↑	Top5↑	Top10↑	Top20↑
PatchNet-VLAD[20]	RGB	67.1%	88.9%	91.9%	93.7%
	semantic	66.4%	88.1%	91.7%	93.3%
	normal	66.5%	85.6%	89.6%	92.2%
	RGB+semantic	74.2%	92.3%	94.4%	95.6%
	RGB+normal	75.8%	92.7%	94.9%	95.6%
	MFF(ours)	**79.1%**	**93.6%**	**95.7%**	**96.2%**
Net-VLAD[2]	RGB	80.1%	95.7%	96.4%	96.7%
	semantic	73.6%	92.3%	94.8%	95.6%
	normal	77.6%	92.8%	94.6%	95.5%
	RGB+semantic	83.3%	95.3%	96.5%	96.7%
	RGB+normal	83.7%	95.8%	**96.6%**	**97.0%**
	MFF(ours)	**85.0%**	**95.9%**	96.5%	96.8%
Dense-VLAD[51]	RGB	43.6%	73.9%	83.4%	89.5%
	semantic	32.9%	64.7%	84.8%	82.1%
	normal	56.0%	79.7%	84.7%	88.6%
	RGB+semantic	48.6%	78.7%	86.2%	91.8%
	RGB+normal	64.5%	88.2%	92.2%	94.7%
	MFF(ours)	**66.3%**	**90.0%**	**93.1%**	**95.3%**
DIR[18]	RGB	74.3%	93.0%	95.8%	96.6%
	semantic	59.2%	85.8%	90.5%	92.6%
	normal	65.0%	88.3%	91.9%	93.9%
	RGB+semantic	78.4%	94.7%	96.0%	96.6%
	RGB+normal	78.9%	94.7%	96.5%	96.9%
	MFF(ours)	**81.4%**	**95.6%**	**96.6%**	**97.2%**

methods are reproduced using their original implementation and pre-trained models. In the input stage, we use retrained DeepLabV3+[7] and DIW[8] to produce semantic and normal features. We also train the feature embedding using the triplet loss function Equation 1.

Metrics. We followed NetVLAD[2] and evaluated the recall rate of each test. A query image is considered successfully localized if at least one of the top N retrieved candidates is good. A reference image is considered a good candidate if the camera pose difference between the query image and the reference image is smaller than the threshold (25m/15°). Since the sampling interval of the dataset is (20m/10°), this ensures that the mutual visual zone between the query image and the correctly retrieved image is large enough to satisfy the requirement of subsequent feature matching and pose optimization algorithms.

Result and discussion. To assess the benefits of our method, we conducted an ablation study for baseline approaches with and without feature fusion. In addition to comparing the performance difference between the baseline method and our method, we also specifically compared the individual channels and their fusion results. The detailed ablation study result is shown in Table 3. From

the comparison of single-feature retrieval results, we see that learning-based methods, including PatchNetVLAD[20], NetVLAD[2], and DIR[18] can better leverage the color information included in RGB channels, and their accuracy is high. However, for the DenseVLAD[51] algorithm, its RootSIFTs do not perform well, and its performance is better when processing normal and semantics. Although PatchNetVLAD[20] was proposed later, we did not find its advantage over NetVLAD[2]. We found that NetVLAD[2] is the best backbone retrieval method especially on datasets with appearance changes. The performance of fusing normal is better than fusing semantics because normal prediction is related to the scene geometry and is more accurate than semantic segmentation. When more features are fused, all methods have significant improvements in the performance of the top 1 recall rate, and fusing both semantic segmentation and surface normal (our MFF) is better than fusing one of them. However, such an improvement in the performance of the top 10 recall rate of NetVLAD is not significant because it has reached its saturation. Experimental results show that our MFF module can enhance mainstream retrieval-based localization baseline methods in terms of recall rates.

Overall, our method can handle season, weather, and illumination changes in the stage of image retrieval, as shown in Figure 8. This is mainly because our method can take advantage of semantic and normal information. With robust retrieval results, we can further improve the accuracy of feature-based visual localization by providing more matching pairs.

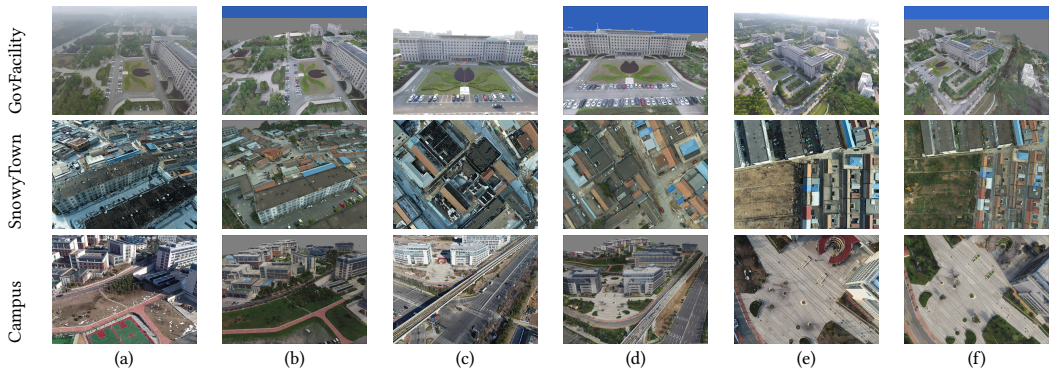


Fig. 8. Visualization of our retrieval results for different scenes with appearance changes. (a), (c) and (e) are query images; (b), (d) and (f) are retrieved virtual images. Images are resized for better visualization.

4.2 Pose estimation experiments

Methods. As mentioned in the related work, structure-based localization approaches achieve the highest performance. As a result, we only compared our method with state-of-the-art structure-based approaches on our dataset, including DenseVLAD [51], hloc [37], D2-net [14], R2D2 [36], and MeshLoc [31]. For hierarchical methods, we used NetVLAD[2] as a retrieve-based baseline method to obtain reference images and coarse estimation of their poses. We also tested our performance with the improvement of our MFF network as an alternative to the NetVLAD retrieval.

Metrics. We follow hloc[37] and evaluate the percentage of successfully localized images with different criteria (0.25m / 0.5m / 5m, 2°/ 5°/ 10°). We decompose the 6DoF camera pose into position and orientation components. For the position components, we measure the Euclidean distance between the position of the query image and the ground truth position. For the orientation, we

convert the rotation matrix to quaternions and evaluate the angular difference between the query image and the ground truth.

Table 4. Experimental results of our pose estimation. For query images, we fuse multiple features using deep neural networks. Instead of matching real images, we render augmented virtual images for robust and accurate visual localization.

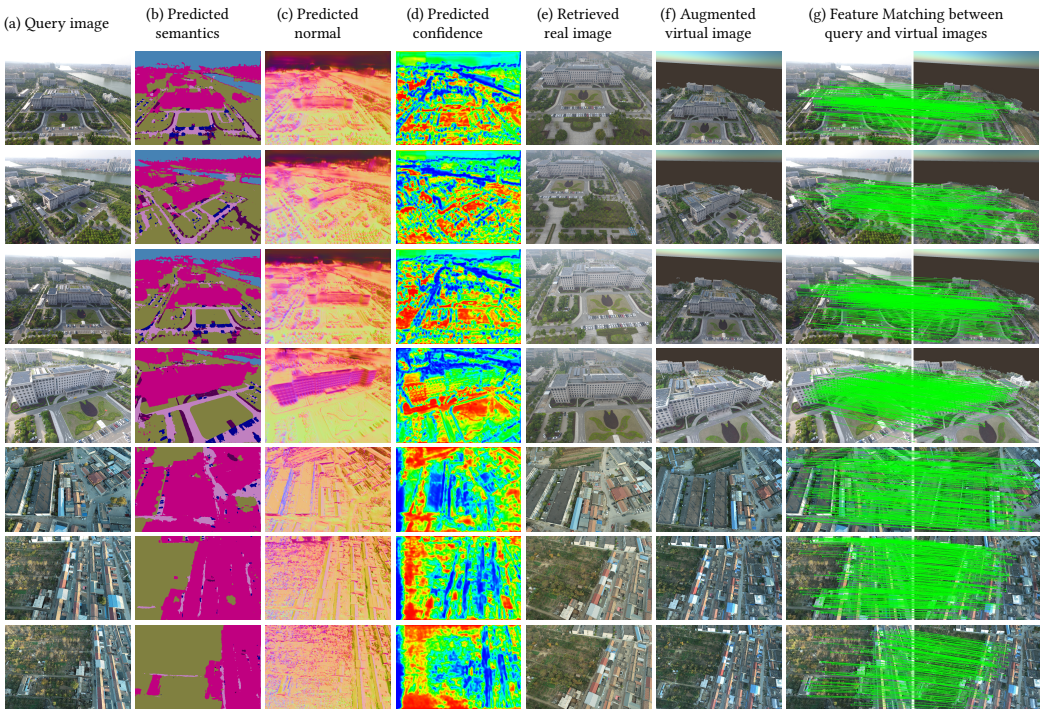


Table 5. Large-scale pose estimation results. Our approach outperforms others on all criteria.

Methods	(0.25m, 2°)	(0.5m, 5°)	(5m, 10°)
hloc [37]	62.1%	85.1%	91.1 %
D2-net [14]	64.3%	81.4%	89.2 %
R2D2 [36]	65.0%	82.2%	87.6 %
MeshLoc [31]	59.5%	86.3%	95.5 %
DenseVLAD [51]	61.7%	80.25%	84.8%
Ours	**67.1%**	**89.8%**	**98.2** %

Results and discussion. The result of the localization is shown in Table 5. Our method outperforms state-of-the-art approaches on all criteria. Notice that the result of our method surpasses the retrieval baseline on the (5m, 10°) criteria. Our method can find a more accurate pose estimation even if the initial retrieval result is not accurate because our method supports re-ranking the retrieval result filtered by the current best estimation, and the PnP solver is robust to tolerate the

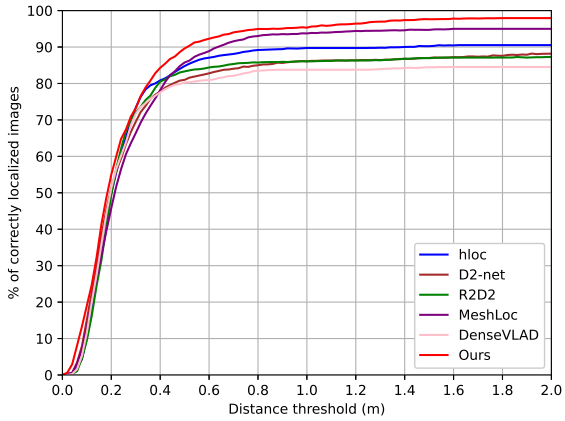


Fig. 9. The cumulative position error. About 90% of our estimates are less than 0.5m in position error.

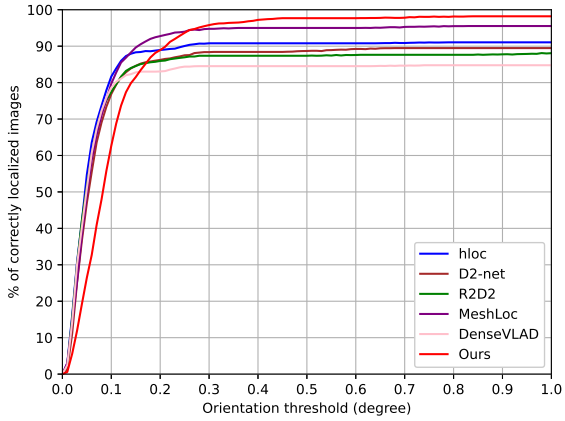


Fig. 10. The cumulative orientation error. About 98% of our estimates are less than 1°in orientation error.

viewpoint differences between the query image and the reference image as long as they have the mutual visual zone. The overall performance of MeshLoc [31] is good, except for the accuracy on the (0.25m, 2°) criteria. This is mainly because the diversity of real images used by other learning-based methods is not enough, and the use of massive virtual images as calibration reference can improve the success rate. However, too many reference virtual images can reduce accuracy when averaging poses. On the other hand, we solve this problem by using the reference image with minimum reprojection error for iterative optimization.

To illustrate the effectiveness of our methods, we visualized our results. As shown in Table 4 (a) and (e), query images and dataset images have different appearances due to the change in illumination and seasonal conditions. Besides, the retrieved real images may have a large difference in viewpoint compared to the query image. However, as shown in Table 4 (f), virtual images are not limited by this. We use illumination-based augmentation to reduce appearance differences, and

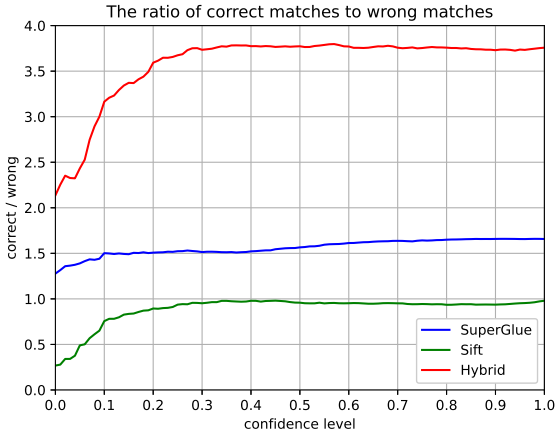


Fig. 11. The ratio of correct matches to wrong matches. The ratio drops sharply when the confidence level is less than 0.1.

cover the virtual space with exhaustive rendering to avoid blind spots. Besides, as shown in Table 4 (b) and (c), our method predicts useful features for MFF. The confidence map in Table 4 (d) shows that the confidence map network successfully identifies stable patterns in the scene (building, open zone, and lane lines, dense woods), ignores faraway objects(sky, distant buildings) and filter out other low confident objects (cars, isolated trees and patterns covered by trees). Table 4 (g) shows the robust and accurate feature matching result filtered by the confidence map and RANSAC.

We used line charts to illustrate the distribution of cumulative position/orientation errors of different methods. As shown in Figure 9 and Figure 10, our method is more accurate than others. About 90% of our estimated results are less than 0.5m in position error, and 98 % of them are less than 1°in orientation error.

In order to determine the threshold of our confidence-based filtering, we design an experiment to find the correlation between the feature matching and the predicted confidence map. We compared three feature matching algorithms: SIFT [27], SuperGlue [38], and our hybrid matching method. Our hybrid method combines SIFT and SuperGlue, which merges keypoints from both algorithms and iteratively optimizes the pose estimation using the fundamental matrix constraint. We collect all keypoint matches from the localization experiments and use the ground truth reprojection error as an evaluation metric. All matches with a reprojection error of less than 1% of the image size (about 10px in our dataset) will be marked as correct, and others will be marked as wrong. As shown in Figure 11, the ratio of correct matches to wrong matches gradually decreases when confidence ≤ 0.3 and drops sharply when confidence ≤ 0.1. This shows that for the query image, matched keypoints with a confidence level less than 0.1 have a larger probability of being wrong. We conclude that filtering out these low-confidence feature matches can reduce the reprojection error of our camera pose estimation.

5 CONCLUSIONS

In this paper, a visual localization dataset is proposed with annotation of semantic segmentation and relative surface normal. An MFF network for improvement of retrieval and a confidence map prediction network for feature filtering are also proposed to prove the usefulness of these features. The major improvement of our method comes from the use of virtual images. We conduct

experiments to prove that virtual images have advantages over real images in terms of dataset size, diversity, and multi-features. The major difference between the proposed approach and other virtual image-based localization methods is that it does not rely on the distribution of real images. We uniformly distributed dense sampling to make virtual images cover as much space as possible. Experimental results show that, due to the use of virtual images with semantics and normal information, our approach surpasses state-of-the-art visual localization approaches.

Limitations. Our method requires high-quality photographs for 3D reconstruction. Therefore, its performance needs to be improved in street view datasets with sparse 3D point clouds. However, its accurate and robust localization is suitable for overlooking scenarios such as surveillance camera registration and drone camera calibration.

Future work. Our future work focuses on increasing the diversity of our dataset, including collecting more data with condition changes and labeling part-level scene semantic segmentation for challenging visual localization tasks.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 62272018.

REFERENCES

[1] Austin Abrams and Robert Pless. 2013. Web-accessible geographic integration and calibration of webcams. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–20.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5297–5307.

[3] Aayush Bansal, Hernán Badino, and Daniel Huber. 2014. Understanding how camera configuration and environmental conditions affect appearance-based localization. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 800–807.

[4] Daniel Barath, Luca Cavalli, and Marc Pollefeys. 2022. Learning to Find Good Models in RANSAC. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 15723–15732. https://doi.org/10.1109/CVPR52688.2022.01529

[5] Eric Brachmann and Carsten Rother. 2021. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE transactions on pattern analysis and machine intelligence* 44, 9 (2021), 5847–5865.

[6] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. 2018. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2616–2625.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.

[8] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-image depth perception in the wild. *Advances in neural information processing systems* 29 (2016).

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

[10] Andrei Cramariuc, Florian Tschopp, Nikhilesh Alatur, Stefan Benz, Tillmann Falck, Marius Brühlmeier, Benjamin Hahn, Juan Nieto, and Roland Siegwart. 2021. SemSegMap–3D segment-based semantic localization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1183–1190.

[11] Pengwen Dai, Siyuan Yao, Zekun Li, Sanyi Zhang, and Xiaochun Cao. 2022. ACE: Anchor-free corner evolution for real-time arbitrarily-oriented object detection. *IEEE Transactions on Image Processing* 31 (2022), 4076–4089.

[12] Renaud Dube, Andrei Cramariuc, Daniel Dugas, Hannes Sommer, Marcin Dymczyk, Juan Nieto, Roland Siegwart, and Cesar Cadena. 2020. SegMap: Segment-based mapping and localization using data-driven descriptors. *The International Journal of Robotics Research* 39, 2-3 (2020), 339–355.

[13] Mihai Dusmanu, Ondrej Miksik, Johannes L. Schönberger, and Marc Pollefeys. 2021. Cross-Descriptor Visual Localization and Mapping. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 6038–6047.

[14] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. 2019. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on*

computer vision and pattern recognition. 8092–8101.

[15] Zan Gao, Shenghao Chen, Yangyang Guo, Weili Guan, Jie Nie, and Anan Liu. 2022. Generic Image Manipulation Localization through the Lens of Multi-scale Spatial Inconsistence. In *Proceedings of the 30th ACM International Conference on Multimedia*. 6146–6154.

[16] Zan Gao, Chao Sun, Zhiyong Cheng, Weili Guan, Anan Liu, and Meng Wang. 2023. TBNet: A two-stream boundary-aware network for generic image manipulation localization. *IEEE Transactions on Knowledge and Data Engineering* 37, 7 (2023), 7541–7556.

[17] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. 2020. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *arXiv preprint arXiv:2004.01673* (2020).

[18] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2017. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision* 124, 2 (2017), 237–254.

[19] Weili Guan, Zhaozheng Chen, Fuli Feng, Weifeng Liu, and Liqiang Nie. 2021. Urban perception: Sensing cities via a deep interactive multi-task learning framework. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 1s (2021), 1–20.

[20] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. 2021. 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*. 14141–14152.

[21] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. 2019. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence* 42, 10 (2019), 2702–2719.

[22] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. 2022. Efficient 3D Point Cloud Feature Learning for Large-Scale Place Recognition. *IEEE Transactions on Image Processing* 31 (2022), 1258–1270.

[23] Hervé Jégou and Ondrej Chum. 2012. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *ECCV-European Conference on Computer Vision*.

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*. 2938–2946.

[25] Måns Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. 2019. Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 31–41.

[26] Konstantinos-Nektarios Lianos, Johannes L Schonberger, Marc Pollefeys, and Torsten Sattler. 2018. Vso: Visual semantic odometry. In *Proceedings of the European conference on computer vision (ECCV)*. 234–250.

[27] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60 (2004), 91–110.

[28] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 2017. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research* 36, 1 (2017), 3–15.

[29] Moritz Menze and Andreas Geiger. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3061–3070.

[30] Andreas Möller, Matthias Kranz, Robert Huitl, Stefan Diewald, and Luis Roalter. 2012. A mobile indoor navigation system interface adapted to vision-based localization. In *Proceedings of the 11th international conference on mobile and ubiquitous multimedia*. 1–10.

[31] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. 2022. MeshLoc: Mesh-Based Visual Localization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*. Springer, 589–609.

[32] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. 2023. Visual Localization using Imperfect 3D Models from the Internet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13175–13186.

[33] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. 2023. SegLoc: Learning Segmentation-Based Representations for Privacy-Preserving Visual Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15380–15391.

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. [n. d.]. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 ([n. d.]), 5105–5114.

[36] Jerome Revaud, Philippe Weinzaepfel, César De Souza, and Martin Humenberger. 2019. R2D2: repeatable and reliable detector and descriptor. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 12414–12424.

[37] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12716–12725.

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

[39] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. 2021. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3247–3257.

[40] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2016. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence* 39, 9 (2016), 1744–1756.

[41] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. 2018. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8601–8610.

[42] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. 2017. Are large-scale 3d models really necessary for accurate visual localization?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1637–1646.

[43] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. 2019. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3302–3312.

[44] Noah Snavely, Steven M Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *ACM siggraph 2006 papers*. 835–846.

[45] Erik Stenborg, Carl Toft, and Lars Hammarstrand. 2018. Long-term visual localization using semantically segmented images. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6484–6490.

[46] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. 2017. City-scale localization for cameras with known vertical direction. *IEEE transactions on pattern analysis and machine intelligence* 39, 7 (2017), 1455–1461.

[47] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. 2022. Long-Term Visual Localization Revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4 (2022), 2074–2088. https://doi.org/10.1109/TPAMI.2020.3032010

[48] Carl Toft, Carl Olsson, and Fredrik Kahl. 2017. Long-term 3d localization and pose from semantic labellings. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 650–659.

[49] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. 2018. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 383–399.

[50] Carl Toft, Daniyar Turmukhambetov, Torsten Sattler, Fredrik Kahl, and Gabriel J Brostow. 2020. Single-image depth prediction makes feature matching easier. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. 473–492.

[51] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 2015. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1808–1817.

[52] Mikaela Angelina Uy and Gim Hee Lee. 2018. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4470–4479.

[53] Tao Ye, Xiangming Yan, Shouan Wang, Yunwang Li, and Fuqiang Zhou. 2022. An Efficient 3-D Point Cloud Place Recognition Approach Based on Feature Point Extraction and Transformer. *IEEE Transactions on Instrumentation and Measurement* 71 (2022), 1–9.

[54] Jacob Young, Tobias Langlotz, Matthew Cook, Steven Mills, and Holger Regenbrecht. 2019. Immersive telepresence and remote collaboration using mobile and wearable devices. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 1908–1918.

[55] H. Zhou, T. Zhang, and J. Jagadeesan. 2019. Re-weighting and 1-Point RANSAC-Based PnP Solution to Handle Outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 12 (2019), 3022–3033.

Received 27 May 2023; revised 21 Jul 2023; accepted 27 Aug 2023