FEW-SHOT DEEP STRUCTURE-BASED CAMERA LOCALIZATION WITH POSE AUGMENTATION

Cheng-Yu Tsai, Shang-Hong Lai

Department of Computer Science
National Tsing Hua University, Taiwan

ABSTRACT

Camera localization predicts the camera pose from a query image. There are two types of deep learning-based camera localization methods: image-based and structure-based. Previous works have shown that data augmentation can improve the performance of image-based methods, but there are no research studies on the structure-based method with data augmentation technique. In this paper, we propose a new pose augmentation procedure that can further improve the performance of the deep structure-based camera localization method, especially under few-shot settings. We investigate different inpainting and rendering strategies and compare their performance with pose augmentation. In addition, we propose a confidence-based sampling scheme that drastically reduces the computation time while maintaining high pose estimation accuracy.

Index Terms— Camera localization, deep learning, pose augmentation

1. INTRODUCTION

Camera localization is to estimate the 6-DoF camera pose, including 3D position and orientation, from an image in a known environment. Traditional methods use feature descriptors [1–3] to establish 2D-3D correspondences between the key points on the 2D image and the 3D model generated by a SfM system [4, 5]. These correspondences can then be used to compute the camera pose of the query image. However, these methods are computationally expensive and suffer from textureless scenes, repetitive patterns, duplicated objects, and highly symmetrical indoor scenes.

Various CNN-based methods have been proposed to take advantage of the strong learning capability of CNN in recent years. They can be divided into two categories: image-based and structure-based. Image-based methods [6, 7] regress poses from images. Structure-based methods [8–12] establish 2D-3D correspondences by using CNN, and then compute the 6-DoF camera pose by solving the Perspective-n-Point (PnP) [8, 13] problem. Structure-based methods typically outperform image-based methods.

To make the most of the training data, previous works [14–16] proposed different methods to augment camera poses. They [14–16] proved that augmented image-pose pairs can improve the performance of camera localization models. However, these augmentation methods have only been applied to image-based models. In this paper, we aim to extend the data augmentation strategy to improve the structure-based camera localization.

Most camera localization models, either image-based or structure-based, need to be trained on specific datasets for each applied scene. In the real world, collecting fine, dense datasets for model training may be hard. In this work, we focus on applying data augmentation to improve structure-based camera localization in few-shot situations.

In this work, we present a pose augmentation method for the structure-based camera localization method. We combine our data augmentation method with a state-of-the-art structure-based model HscNet [11]. We can not only improve the baseline HscNet [11], but also perform better than other state-of-the-art models under few-shot settings.

2. RELATED WORK

Deep learning based camera localization methods can roughly be divided into image-based and structure-based localization approaches. Image-based methods [6, 7] feed the query images into CNN models, and models output regressed camera poses. Structure-based methods [8–12] regress each 2D pixel on the query image into 3D scene coordinates to obtain 2D-3D correspondences. Then the camera pose prediction task becomes a PnP [13] problem and can be solved by the above 2D-3D correspondences. After predicting scene coordinates, [8, 10–12] compute the camera pose from the predicted scene coordinates by the RANSAC-PnP step. The training objective of deep models is the regression of 3D scene coordinates. Due to the precise 3D information during training, structure-based methods perform better than image-based methods. Thus we choose the structure-based approach as our research focus.

HscNet [11] is a state-of-the-art structure-based model for camera localization. Its model architecture is depicted in Figure 1. HscNet [11] clusters all 3D points of training scenes into hierarchical classes with hierarchical k-means clustering and sets clusters as class labels during pre-processing. The model hierarchically classifies labels for each 2D pixel and predicts the closest cluster center to each 2D pixel. Finally, HscNet [11] computes the predicted camera pose by the PnP-RANSAC algorithm.

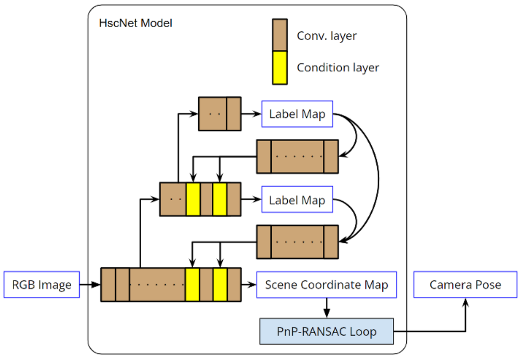

Fig. 1. HscNet [11] network architecture and the pose estimation procedure.

3. PROPOSED METHOD

3.1. Structure-Based Camera Localization Model

We augment camera poses and synthesize images from the new viewpoints. We use the inpainting technique to deal with the blank area caused by the invalid depth and pose changes. In addition to the new RGB images, the rendering pipeline simultaneously generates new depth maps. This allows us to use the extended 3D information to improve structure-based methods. Since our spatially-augmented data do not fit the original label maps provided by HscNet [11], we use Nearest Neighbor search to assign the nearest cluster label to the 3D point of each pixel, and generate the new label maps for the augmented images as the ground truth.

The new camera poses, synthesized RGB images & depth maps, and the new label maps form additional augmented image-pose pairs, which is m times the number of original data. After the pre-processing and data preparation, we train and evaluate HscNet [11] with the augmented data to verify whether the structure-based localization can benefit from the data augmentation method that increases the number of image-pose pairs.

3.2. Data Augmentation

We use the following augmentation pipeline to generate the augmented data. It starts by generating new camera poses. It first applies K-Nearest-Neighbors search to find the k nearest camera poses and dynamically decides the adjustment range for camera poses in three axes and three Euler angle directions according to the maximum distance in k neighbors. After finding the bounds in six dimensions, it randomly adjusts camera poses in the bounds to create additional poses. It augment an image for m times, so augmented images is m times the number of original images. The rendering pipeline projects the pixels of the 2D images onto the 3D point cloud and reprojects them back to 2D images according to new camera poses. We set k to 50 in the K-Nearest-Neighbors Search, but we lower it to fit the smaller and sparser data in the few-shot situation.

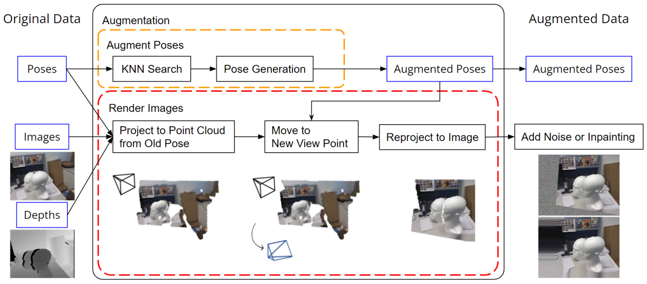

Fig. 2. Flowchart of the proposed data augmentation pipeline.

We observe that different point size settings of the point cloud in the rendering pipeline affect the effect of augmentation. The point size setting affects how large the 3D points are projected onto 2D images. When the point size is larger, the rendered images look coarser and less accurate, but there are fewer invalid pixels because the larger point sizes cover more space. On the other hand, the smaller point size produces more accurate and precise images, but they suffer from more blank areas. Even though smaller point sizes cause the blank area problem, we choose a smaller point size for rendering and use the RGB inpainting technique to cover these blank areas and make the images rendered under smaller point sizes more photo-realistic than the coarser ones. In this work, we apply the inpainting function with the Navier-Stokes based algorithm from the OpenCV Python toolkit to fill in the blanks on the augmented RGB images..

3.3. Confidence-based Sampling

The FPS bottleneck of HscNet [11] is the first part in the PnP-RANSAC algorithm that samples four points to form a valid hypothesis. It spends much time forming a valid hypothesis because it sometimes picks bad points and has to resample. We plan to improve the quality of candidate points so that it does not have to spend too much time resampling.

HscNet [11] uses the one-hot encoding format for the ground truth of two hierarchical label maps. In addition to HscNet [11] using the *"argmax"* function to extract the predicted classes, we also edit the model to output the *"max"* value for each pixel. This raw value indicates how strongly the model considers the pixel the predicted label. We take this raw value as confidence and create a confidence map. We select the top 50% points with higher confidence as the sample range and filter out the other 50% points with lower confidence. The proposed confidence-based sampling scheme samples the points with higher quality for the PnP-RANSAC algorithm and has a higher probability of successfully forming a valid hypothesis, thus making the PnP-RANSAC algorithm more efficient. Because the second classification network performs poorly in some scenes, we only use the raw value as confidence in the first classification network.

4. EXPERIMENTS AND DISCUSSION

4.1. Datasets and Experimental Setup

We evaluate on 7-Scenes and 12-Scenes. 7-Scenes and 12-Scenes are both indoor RGB-D datasets. Under the few-shot condition, we compare our experimental results with our baseline HscNet [11], and Dong et al. [17] on 7-Scenes. Since Dong et al. [17] did not report experimental results on 12-Scenes, we only compare our method with the baseline HscNet [11] on 12-Scenes.

Table 1. Comparison of the experimental results on 7-Scenes with 100% data with different methods. The numbers are median translation (in meters) and rotation errors (in degrees). Here we set *k* to 50 and *m* to 32.

7-Scenes 100%	DSAC++	SANet	DSM	HscNet	Ours
Chess	**0.02**, **0.50**	0.03, 0.88	**0.02**, 0.71	**0.02**, 0.70	**0.02**, 0.59
Fire	**0.02**, 0.90	0.03, 1.08	**0.02**, 0.85	**0.02**, 0.90	**0.02**, **0.84**
Heads	**0.01**, **0.80**	0.02, 1.48	**0.01**, 0.85	**0.01**, 0.90	**0.01**, 0.82
Office	0.03, **0.70**	0.03, 1.00	0.03, 0.84	0.03, 0.80	**0.02**, 0.72
Pumpkin	**0.04**, 1.10	0.05, 1.32	**0.04**, 1.16	**0.04**, **1.00**	**0.04**, 1.03
Kitchen	**0.04**, **1.10**	**0.04**, 1.40	**0.04**, 1.17	**0.04**, 1.20	**0.04**, 1.17
Stairs	0.09, 2.60	0.16, 4.59	0.05, 1.33	**0.03**, 0.80	**0.03**, **0.73**
Average	0.04, 1.10	0.05, 1.68	**0.03**, 0.99	**0.03**, 0.90	**0.03**, **0.84**

Table 2. Experimental results on 7-Scenes dataset under few-shot settings.

7-Scenes	HLoc [18, 19]	DSAC* [9]	Dong et al [17]	HscNet [11]		Ours	
Few-shot	Median err.	Median err.	Median err.	Acc.	Median err.	Acc.	Median err.
Chess (0.5%)	0.04, 1.42	**0.03**, 1.16	0.04, 1.23	77.9	**0.03**, 1.13	77.0	**0.03**, **0.99**
Fire (0.5%)	**0.04**, 1.72	0.05, 1.89	**0.04**, 1.52	56.9	**0.04**, 1.50	62.2	**0.04**, **1.30**
Heads (1%)	0.04, 1.59	0.04, 2.71	**0.02**, **1.56**	63.9	0.04, 2.16	74.6	**0.02**, 1.58
Office (0.5%)	**0.05**, 1.47	0.09, 2.21	**0.05**, 1.47	40.0	0.06, 1.61	49.9	**0.05**, **1.28**
Pumpkin (0.5%)	0.08, 1.70	0.07, 1.68	0.07, 1.75	30.3	0.07, 1.65	33.5	**0.06**, **1.56**
Kitchen (0.5%)	0.07, 1.89	0.07, 2.02	**0.06**, 1.93	27.1	0.07, 2.09	39.4	**0.06**, **1.81**
Stairs (1%)	0.10, 2.21	0.18, 4.80	**0.05**, **1.47**	26.6	0.10, 2.76	39.3	0.06, 1.61
Average	0.06, 1.71	0.08, 2.35	**0.05**, 1.56	46.1	0.06, 1.84	53.7	**0.05**, **1.45**

Table 3. Experimental results on 12-Scenes datasets under few-shot settings.

12-Scenes	Few-shot 1%				Few-shot 0.5%			
	HscNet [11]		Ours		HscNet [11]		Ours	
	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	55.3	0.060, 2.4	**76.4**	**0.028**, **1.2**	39.4	0.259, 16.8	**51.3**	**0.180**, **15.5**

4.2. Experimental Comparison

Table 1 summarizes the results on 7-Scenes with 100% data. All methods are state-of-the-art structure-based methods. HscNet [11] is our baseline that does not use augmentation data. The rightmost column is HscNet [11] combined with our data augmentation method. It shows that the additional augmented image-pose pairs can also improve the performance of the structure-based method (HscNet [11]). It makes HscNet [11] perform better and achieve the best results.

Table 2 demonstrates the experimental results on 7-Scenes under few-shot conditions. We obtain competitive results on median error and accuracy rate. Table 3 shows the experimental comparison on 12-Scenes, under the 1% and 0.5% few-shot conditions. The proposed method can achieve over 50% improvement in median translation and rotation errors and over 30% improvement in accuracy rate. The few-shot conditions on 7-Scenes follow the setting from Dong et al. [17]. The numbers are median translation and rotation errors (m, °), and the percentages of test images accurately predicted (error < 0.05 m, 5°). The results of HLoc [18, 19], DSAC* [9], and Dong et al. [17] are copied from [17].

4.3. RGB Inpainting and Noise

Table 4. shows the comparison of RGB noise and RGB inpainting for filling empty areas on augmented images. Both RGB noise and RGB inpainting can improve the performance of HscNet [11], but RGB inpainting is better than RGB noise.

Table 4. Experimental results on 7-Scenes and 12-Scenes datasets under few-shot settings. Here we set *k* to 4 and *m* to 64 on 7-Scenes, and *k* to 2 and *m* to 32 on 12-Scenes.

Few-Shot	HscNet [11]		w/ aug + Noise		w/ aug + Inpainting	
Average	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
7S	46.1	0.060, 1.8	51.2	0.050, 1.6	**53.7**	**0.047**, **1.5**
12S 1%	55.3	0.060, 2.4	75.0	0.029, **1.2**	**76.4**	**0.028**, **1.2**
12S 0.5%	39.4	0.259, 16.8	50.3	0.182, 15.6	**51.3**	**0.180**, **15.5**

4.4. Rendering Quality

Although the images with smaller point sizes are more accurate and precise, the larger blank regions negatively impact model training. Fortunately, the RGB inpainting technique can fulfill the blanks and improve the benefits of augmented data. Adding RGB inpainting to the blank pixels can improve the training results, especially for images with smaller point size.

Table 5 compares the results of different point size settings and invalid pixel fixing strategies (RGB inpainting/RGB noise). The augmented images with rendering quality ”point size = 2.0” perform best, so we set the point size as 2.0. Table 5. also shows that adding RGB inpainting is better than adding RGB noise with different rendering qualities.

Table 5. Ablation study of different rendering quality (point size settings) and different invalid pixel inpainting strategies on 7-Scenes under few-shot conditions. Here we set *k* as 2 and *m* as 32. The top table is augmented images with RGB inpainting, and the bottom is augmented images with RGB noise.

	Point Size = 3.0		Point Size = 2.0		Point Size = 1.5	
7-Scenes	w/ RGB Inpainting					
	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
Few-shot	48.7	0.06, 1.66	**49.5**	**0.05**, **1.62**	47.9	0.06, 1.67
	w/ RGB Noise					
Average	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
	44.8	0.06, 1.81	44.6	0.07, 1.84		

4.5. Discussion on *k* and *m*

We analyze the influence of different *k* and *m* settings. The parameter *k* for K-Nearest-Neighbors Search affects the distribution of augmented data. The parameter *m* for augmentation multiple affects the density of augmented data. We test different *k* and *m* to find the best parameters for data augmentation. We first adjust the parameter *m* and find that *m* = 64 is the best. And we adjust the parameter *k* with fixed *m* = 64, and find that *k* = 4 is the best, as shown in Table 6. We use this fixed *k*/*m* for the few-shot experiments on 7-Scenes dataset.

Table 6. Ablation study of different *m* and different *k* values on 7-Scenes under few-shot conditions.

7-Scenes	***m* = 48**, *k* = 2		***m* = 64**, *k* = 2		***m* = 80**, *k* = 2	
Few-shot	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	49.6	**0.05**, 1.62	**51.1**	**0.05**, **1.58**	50.6	**0.05**, 1.62
7-Scenes	*m* = 64, ***k* = 2**		*m* = 64, ***k* = 3**		*m* = 64, ***k* = 4**	
Few-shot	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	51.1	**0.05**, 1.58	**53.8**	**0.05**, 1.46	53.7	**0.05**, **1.45**

We compare our results of the previous fixed *k*/*m* and the new dynamic *k*/*m* with Dong et al [17]. Since the dynamic selection of *k* and *m* is not yet mature enough to be automatically selected by some mechanisms, we show the results of dynamic *k*/*m* only in this section. For the few-shot problem settings on 7-Scenes, our results with dynamic *k*/*m* achieve the best. An important future work is how to decide the values of *k* and *m* automatically and dynamically according to the datasets' attributes.

4.6. Confidence-based Sampling

Table 8 shows that after applying the proposed confidence-based sampling, the model's FPS performance is improved by about 55% ˜60% under the few-shot condition. At the same time, the accuracy (recall) and the median translation and rotation errors are about the same. The points with higher confidence bring a higher probability of successfully solving the coarse camera poses during the first part in the PnP-RANSAC algorithm. After reducing the number of failed hypothesis generation cases, the FPS for our method is significantly increased.

Table 7. Experimental results on 7-Scenes under the few-shot settings. The column labeled "Fixed *k*/*m*" means that we use the fixed *k* = 4 and *m* = 64. The column labeled "Dynamic *k*/*m*" means that we use the optimal *k* and *m* for each scene. For comparison with other methods, we use the results with fixed *k*/*m* as our results.

7-Scenes	Dong et al	Ours: Fixed *k*/*m*		Ours: Dynamic *k*/*m*	
Few-shot	Median err.	Acc.	Median err.	Acc.	Median err.
Chess	0.04, 1.23	77.0	**0.03**, 0.99	**78.6**	**0.03**, **0.92**
Fire	**0.04**, 1.52	**62.2**	**0.04**, 1.30	61.2	**0.04**, **1.29**
Heads	**0.02**, **1.56**	**74.6**	**0.02**, **1.58**	**74.6**	**0.02**, 1.58
Office	**0.05**, 1.47	49.9	**0.05**, 1.28	**50.5**	**0.05**, **1.24**
Pumpkin	0.07, 1.75	**33.5**	**0.06**, 1.56	33.2	**0.06**, **1.51**
Kitchen	**0.06**, 1.93	**39.4**	**0.06**, **1.81**	**39.4**	**0.06**, **1.81**
Stairs	**0.05**, 1.47	39.3	0.06, 1.61	**44.0**	0.06, **1.46**
Average	**0.05**, 1.56	53.7	**0.05**, 1.45	**54.5**	**0.05**, **1.40**

Table 8. Comparison of camera localization experiment results with and without using the proposed confidence-based sampling on 7-Scenes and 12-Scenes datasets under few-shot conditions. On 12-Scenes, as we consider the results of Kitchen-2 and Living-2 under the 0.5% condition to be outliers, we exclude them and calculate a new average result (Avg.*) for the 0.5% condition.

7-Scenes Few-shot		Acc.	Median err.	Time(s)	FPS
Avg.	w/ aug	53.7	0.047, 1.45	0.25634	3.90
	w/ aug + cfd.	54.1	0.047, 1.44	0.16568	6.04
12-Scenes Few-shot		Acc.	Median err.	Time(s)	FPS
Avg.(1%)	w/ aug	76.4	0.028, 1.15	0.26413	3.79
	w/ aug + cfd.	75.9	0.029, 1.17	0.16409	6.09
Avg.(0.5%)	w/ aug	51.3	0.180, 15.50	0.42770	2.34
	w/ aug + cfd.	51.8	0.170, 14.43	0.26566	3.76
Avg.*(0.5%)	w/ aug	54.9	0.055, 2.23	0.40210	2.49
	w/ aug + cfd.	55.4	0.053, 2.23	0.24459	4.09

5. CONCLUSION

In this paper, we presented the pose augmentation strategy for the structure-based camera localization method. We combined the proposed data augmentation method with the state-of-the-art structure-based model HscNet [11], and prove that the augmented image-pose pairs can further improve the performance of the structure-based model. Furthermore, our augmentation method can provide reasonable model training results under few-shot settings for the structure-based camera localization model. In addition, we propose a confidence-based sampling scheme for the structure-based camera localization model, which brings about 40% reduction in the inference time. Meanwhile, it maintains high accuracy in the camera localization results.

6. REFERENCES

[1] D.G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157 vol.2.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 06 2008.

[3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[4] Johannes L. Schönberger and Jan-Michael Frahm, “Structure-from-motion revisited,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[5] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys, “Pixelwise view selection for unstructured multi-view stereo,” in *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds., Cham, 2016, pp. 501–518, Springer International Publishing.

[6] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz, “Geometry-aware learning of maps for camera localization,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2616–2625.

[7] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala, “Camera relocalization by computing pairwise relative poses using convolutional neural network,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 920–929.

[8] Eric Brachmann and Carsten Rother, “Learning less is more - 6d camera localization via 3d surface regression,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4654–4662.

[9] Eric Brachmann and Carsten Rother, “Visual camera re-localization from RGB and RGB-D images using DSAC,” *TPAMI*, 2021.

[10] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan, “Sanet: Scene agnostic network for camera localization,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 42–51.

[11] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala, “Hierarchical scene coordinate classification and regression for visual localization,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11980–11989.

[12] Shitao Tang, Chengzhou Tang, Rui Huang, Siyu Zhu, and Ping Tan, “Learning camera localization via dense scene matching,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1831–1841.

[13] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart, “A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,” in *CVPR 2011*, 2011, pp. 2969–2976.

[14] Jian Wu, Liwei Ma, and Xiaolin Hu, “Delving deeper into convolutional neural networks for camera relocalization,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5644–5651.

[15] Tayyab Naseer and Wolfram Burgard, “Deep regression for monocular camera-based 6-dof global localization in outdoor environments,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1525–1530.

[16] Shih Fang-Yu, “Improving the accuracy of deep localization models by spatially-augmented camera poses,” M.S. thesis, National Tsing Hua University, 2020.

[17] S. Dong, S. Wang, Y. Zhuang, J. Kannala, M. Pollefeys, and B. Chen, “Visual localization via few-shot scene region classification,” in *2022 International Conference on 3D Vision (3DV)*, Los Alamitos, CA, USA, sep 2022, pp. 393–402, IEEE Computer Society.

[18] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.