2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Camera Relocalization using Deep Point Cloud Generation and Hand-crafted Feature Refinement

Wang Junyi[1,2] and Qi Yue*[1,2,3]

Abstract— **Visual localization plays an indispensable role in robotics. Both learning and hand-crafted feature based methods for relocalization process keep their effectiveness and weakness. However, current algorithms seldom consider these two kinds of features under one framework. In this paper, focusing on this task, we propose a novel relocalization framework for RGB or RGB-D data source, which is composed of coarse localization process by learning features and pose refinement by hand-crafted features. In particular, coarse stage contains deep point cloud generation and registration. In this stage, instead of regressing camera pose directly, the paper novelly designs a neural network called PGNet to construct sparse point cloud with RGB or RGB-D as inputs. Further more, by means of training set, hand-crafted feature space is established. Based on the obtained camera pose in coarse stage, accurate point-to-point correspondences are set up through searching the space. Then accurate camera pose is obtained by applying RANSAC to correspondences or solving PnP. Finally, experiments on both outdoor and indoor benchmark datasets demonstrate state-of-the-art performance over other existing methods.**

I. Introduction

Visual localization algorithm aims to predict the 6-DoF camera pose containing position and orientation in a known scene from input images, which is widely applied in various areas such as AR/MR, robotics, and autonomous driving [1]–[5]. To deal with this task, current solutions cover three kinds of techniques containing hand-crafted feature based, learning feature based and scene coordinate based approaches.

Hand-crafted feature based approaches [6]–[8] covering a variety of simultaneous localization and mapping (SLAM) systems [9]–[16] solve this task through establishing 2D-3D correspondences. The general process always contains point feature extracting such as SIFT [17] and ORB [18], building correspondences between 3D points and 2D pixels and finally adopting an PnP solver to correspondences. By means of the procedure, it can achieve precise localization performance. However, hand-crafted feature is not sufficiently robust to challenging real-world environments such as changes of illumination, weather, dynamic factors or other conditions. Besides this, in dynamic environments, hand-crafted feature is not stable, which needs to exploit deep learning techniques to detect dynamic objects.

Alternatively, learning feature gains great achievements in detection, segmentation and classification tasks. PoseNet [19] completes the earliest work which adopts deep learning to relocalization process. This method directly regresses position and orientation of camera from images. Through training the dataset which contains images and their corresponding poses, the trained neural network learns correspondences between images and their poses. After PoseNet, subsequent works [20]–[24] focus on network architecture and loss function to improve accuracy. Relocalization through learning features is computationally effective and have no problems of scale drift and accumulative errors in SLAM systems. However, there is still quite a gap in accuracy compared to hand-crafted feature based methods, which limits their actual applications.

In addition to adopting learning features for localization directly, an another process combines deep neural network with RANSAC or PnP pipeline. The core idea locates at generating 3D scene coordinates from corresponding image patches. The method is firstly proposed by using regression forests [25]. Recently, Brachmann et al. [26], [27] presented the differentiable RANSAC pipeline (DSAC) and scaled the method to large scene for camera localization. Further more, KFNet [28] extended the method to the time domain to build correspondences recursively for the pose estimation. At present, this kind of methods achieves the state-of-the-art performance.

Obviously, on the one hand, both hand-crafted and learning features have their superiorities. On the other hand, implementing sectional tasks of whole localization process contributes to more accurate performance. Inspired by both ideas, it is beneficial to combine two kinds of features in a novel localization framework. However, it is difficult to combine learning and hand-crafted features directly because of black box of deep learning. Alternatively, it is feasible to design a framework to exploit both two kinds of features, which aims to achieve precise and stable localization process. Above all, in this paper, we propose a novel localization framework for RGB or RGB-D source data, which is demonstrated in Fig.1. The framework firstly utilizes a deep neural network called PGNet to generate sparse point cloud with color images, depth images and previous camera poses as inputs. Then coarse localization results are obtained by point cloud registration algorithm. These two steps take advantages of learning features, which enhances robustness to environmental changes or dynamic conditions. Further more, to improve accuracy, refine process is designed by establishing hand-crafted feature space and searching feature point correspondence by the obtained coarse pose.

In summary, our main technical contributions are as follows.

[1] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering at Beihang University, Beijing, China
[2] Peng Cheng Laboratory, ShenZhen, China
[3] Qingdao Research Institute of Beihang University, Qingdao, China
Wang Junyi, wangjy0524@163.com
*Corresponding author, Qi Yue, qy@buaa.edu.cn

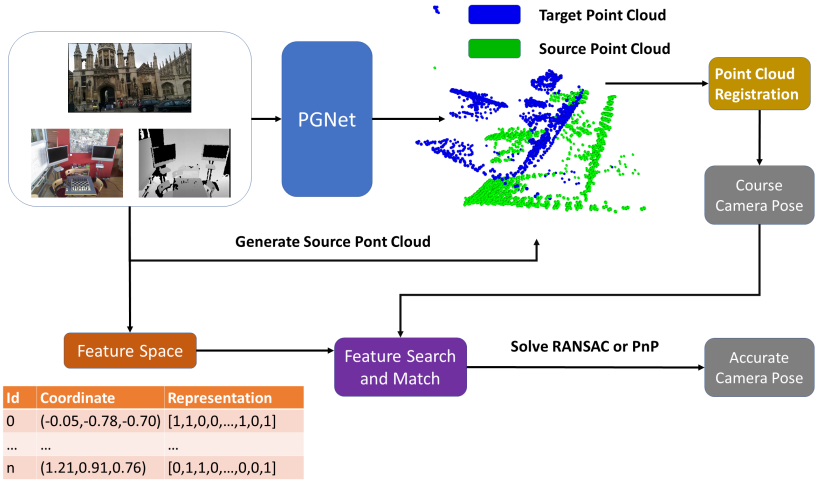

Fig. 1. Outline of localization pipeline. At the start, target point cloud is generated by PGNet with RGB or RGB-D as input. Then point cloud registration is adopted to calibrate the coarse camera pose. Meanwhile, according to training set, hand-crafted feature space is established. Based on obtained pose and feature searching, accurate point-to-point correspondences are built to evaluate final pose.

1. A novel algorithm framework is proposed to exploit both learning and hand-crafted features with RGB or RGB-D source data.

2. By adopting learning features, PGNet is designed to generate target point cloud of input images. Then the coarse camera pose is calibrated by point cloud registration.

3. To promote localization accuracy, the camera refinement stage by hand-crafted features is proposed, which contains space establishing, feature searching and solving PnP or applying RANSAC to the feature corresponding.

4. Experiments on both indoor and outdoor benchmark datasets demonstrate state-of-the-art performance of our algorithm framework.

In following of this paper, related works are summarized in Section 2. Section 3 illustrates detailed localization process, which covers coarse pose estimation by deep features and pose refinement by hand-crafted features. Further more, Section 4 reports the experimental localization results compared with previous state-of-the-art works. At last, Section 5 draws conclusions and future works.

II. Related Work

Hand-crafted feature based localization methods exploit feature extraction, feature matching to calibrate camera pose between images and graph optimization to reduce accumulated pose error [6], [7], [9]–[13]. These methods build 2D-3D correspondences and represent the scene by a 3D model, which estimates camera pose precisely. Various of SLAM systems and Structure From Motion (SFM) algorithms adopt the pipeline and gain some achievements in specific circumstances. However, hand-crafted features are often not robust to changing environmental conditions such as illumination variation or dynamic objects.

Learning feature based processes directly output camera poses with the neural network trained by images and their corresponding poses [19]–[21], [23], [29], [30]. The localization process employs network parameters to represent the scene and store relationships between learning features and camera poses. On the basis of output pose type, learning feature based approaches are classified into absolute and relative pose methods, which share similar process containing feature extracting and camera pose regression. To improve localization precision, different works take attention to the network architecture and loss functions [19]–[21], [23], [29], [30]. Among all learning feature based methods, VLocNet [29] and VLocNet++ [30] achieve the state-of-the-art performance. VLocNet exploited auxiliary learning to promote localization performance, which firstly yielded an accuracy comparable to hand-crafted feature based pipelines. Based on VLocNet, VLocNet++ designed a multitask neural network to learn inter-task relationship between 6-DoF global pose, odometry and semantics.

Coordinate regression based methods mainly regress 3D coordinates from pixel patches, which are firstly proposed by the Shotton et al. [25]. Based on this work, Valentin et al. [31] trained regression forest to predict multi-model distributions of scene coordinates and show how predicted uncertainties are exploited for continuous pose optimization. Algorithm [32] utilized auto-context random forest to achieve localization from RGB images. To achieve end-to-end learning, DSAC [26] presented the differentiable RANSAC pipeline for camera localization. Further more, DSAC++ [27] demonstrated and explained that learning a single component of this pipeline was sufficient. KFNet [28] leveraged Kalman filtering to extend the scene coordinate regression problem to the time domain in order to obtain precise correspondences between 2D pixels and 3D points. Currently, this kind of approaches achieves state-of-the-art localization performance. Compared with these methods, our localization process has some similarities, but mainly differs at following two aspects. Firstly, we construct sparse unordered point cloud instead of regressing scene coordinates, which reduces the uncertainty and runtime. Further more,

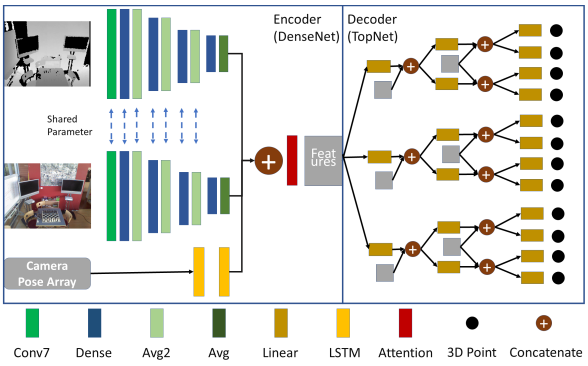

Fig. 2. Architecture of PGNet. PGNet regresses target scene point cloud of the input color image, depth image and previous camera poses. For the layer name at the bottom, Conv7 means 7×7 convolution layer, Dense means Dense Block defined in DenseNet, Avg2 means average pooling layer with stride 2, Avg means average pooling layer which scale the input to 1×1, Linear means linear layer, LSTM means LSTM layer, Attention means attention layer defined in the following.

based on obtained pose by point cloud, hand-crafted features are exploited to calibrate more accurate camera pose.

Point cloud generation by deep learning has drawn scholars' attention in recent years, which mainly focuses on construction and completion of single object. Point set generation network [33] constructed 3D point cloud of target object with an single image, which is the earliest work to generate 3D point clouds. In addition to this, Fan et al. [33] also introduced the Chamfer Distance (CD) between two point clouds, which is also leveraged in later researches and our paper. Subsequent works took attention on improving the reconstruction precision. FoldingNet [34] proposed a folding-based point cloud decoder with MLPs and gained lower reconstruction errors. PCN [35] featured a decoder design with enabling the generation of fine-grained completions. Based on hierarchical rooted tree structure, TopNet [36] presented a novel decoder constrained to generate point clouds, which achieved state-of-the-art performances on point cloud completion. In this paper, we adopt TopNet as point cloud decoder, which outputs scene point cloud by images and camera poses.

III. Localization Framework

The whole framework architecture is revealed as Fig.1, which contains coarse localization stage by learning features and pose refinement stage through hand-crafted features. The coarse process is composed of deep point cloud generation and robust point cloud registration. To exploit learning features, deep point cloud decoder is brought to the localization process instead of regressing camera poses directly. Further more, to gift precision, we establish the hand-crafted feature space, search the space through obtained poses and build point-to-point correspondences. Finally, refined camera poses are calibrated by exploiting RANSAC to the correspondences or solving PnP.

A. Coarse Localization

The core idea of coarse localization process is to calibrate camera pose by exploiting registration algorithm to the source and target point cloud. In detail, the source cloud is obtained from inputs and the target is regressed by the deep network called PGNet. PGNet takes color image, depth image (optional) and previous camera poses as inputs and generates the target sparse point cloud.

Firstly, the constructing process of source and target point cloud is given. In RGB-D case, the source point cloud with camera coordinate is generated by depth image and the target is obtained by converting source point cloud to the world coordinate. In contrast, we construct the sparse point cloud by SFM pipeline with ground truth pose in RGB situation. Then the source point cloud with world coordinate is chosen from sparse point cloud by their view number. As a matter of the course, the target point cloud is calculated by transforming the source point cloud to camera coordinate, which is opposite to the RGB-D case.

As Fig.2 demonstrates, the architecture of PGNet is brief, which includes feature extraction, camera trajectory learning and point cloud construction. The feature extraction part adopts DenseNet [37]. In RGB-D case, the color and depth information are processed by two DenseNet style network with first three shared Dense Blocks. After this, color and depth image are transformed to $d/2-$dimension features, which are then concatenated to $d-$dimension ($d = 1024$ in the experiment). In RGB case, the network with same architecture converts RGB image to $d-$dimension features by exploiting more convolution filters. The camera trajectory learning part aims to gift network performance and avoids limitations of single image in scenes with similar views. In this part, PGNet utilizes two consecutive LSTM layers to transform camera pose array to $d/2-$dimension features. Further more, after obtaining image and pose features, channel attention is used to fuse these features, which are then adopted to point cloud generation part.

The point cloud generation part of PGNet leverages TopNet [36] as point cloud decoder, which transforms fused features to scene point cloud by hierarchical rooted tree structure. For the loss function between scene point clouds, CD finds the nearest neighbor in the other set and sums the squared distances up, which produces high quality results in practice [33]–[36]. The formula is as follows,

$$Loss_{CD} = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{x \in S_2} \min_{y \in S_1} \|x - y\|_2^2,$$

where S_1 and S_2 are two sets of points. Through PGNet and CD loss function, target point cloud is constructed. Further more, with source and generated target point cloud, the paper adopts point cloud registration algorithm to calibrate camera pose. A common approach is to leverage Iterative Closest Point (ICP) algorithm or its variants, but the kind of algorithm requires static scenes and small motion between point cloud pairs, which is not adaptable in our situations. Instead, SUPER 4PCS [38] registers a pair of raw point

clouds in arbitrary initial poses and runs in linear time, which is exploited to obtain coarse camera pose.

B. Pose Refinement

With the learning feature, output points contain certain noises, which are not sufficient to calibrate accurate camera pose. In this subsection, based on the coarse pose, the paper implements pose refinement through establishing hand-crafted feature space, searching the space and calibrating the pose by applying RANSAC to the feature correspondences or solving PnP.

The refinement starts with establishing hand-crafted feature space by training images. In this process, the paper exploits ORB feature [18] because of its robustness and effectiveness. Each slice of information contains the 3D coordinate and descriptor. In RGB-D cases, for each image in training set, ORB features are detected from the RGB source and the corresponding world coordinates are evaluated based on depth information. After all training images are processed, we fuse the information by means of 3D coordinates and ORB features by fixed threshold to reduce the searching time. For the RGB input, the sparse point cloud by SFM pipeline is exploited to construct the feature space.

Based on the built feature space, new correspondences are established through the search process. The RGB-D search firstly extracts ORB features of test image. Then each feature point is converted to world coordinate by coarse camera pose and depth information. Secondly, we search every neighbourhood of world point and find the best ORB match. Further more, based on new matches, RANSAC is exploited to determine the camera pose. With the only RGB input, for each point in known sparse point cloud, the space stores its world coordinate and ORB features. When setting up the correspondence, the world coordinate is converted to the pixel position by obtained camera pose and known focal length. Then according to descriptors, neighbourhood of evaluated pixel position is searched and the best match is selected. Now relations between 3D world point and 2D image pixel are established. Finally, PnP method is leveraged to solve the final camera pose.

Obviously, for RGB and RGB-D, refinement stage adopts different strategies. The reason locates at the depth information. After the feature searching, accurate correspondences are established. With known depth, these correspondences is expressed between 3D points. Then camera pose can be evaluated through 4 pairs of points. Further more, to reduce the effect of outliers, RANSAC algorithm is exploited. When the input only covers RGB information, correspondences are represented between 2D pixels and 3D points. In this situation, PnP algorithm is the best choice.

It is apparent that the refinement process is mainly on account of hand-crafted feature matches. As stated in the introduction, hand-crafted features are not robust to changing environments such as lighting, dynamic factors and so on. Therefore, the refinement may fail. In the actual implementation, to avoid this, when enough points (more than 100 points in the experiment) under the match threshold (40 in the experiment), the refinement process is executed. In other cases, the coarse camera pose is the final result, which is also more accurate than most of approaches.

IV. Experiment

A. Dataset and implementation detail

To validate the performance of our algorithm, we conduct experiments both in indoor and outdoor datasets. The adopted indoor dataset is 7 Scenes and outdoor is Cambridge Landmarks, which recently have been evaluated as benchmarks [21], [23], [27], [28]. Peculiarly, 7 Scenes consists of color and depth information, while Cambridge Landmarks only includes color images, which is convenient for testing both pipelines of our framework.

7 Scenes [25]. The indoor dataset is a collection of tracked RGB-D camera frames in indoor scenes captured from a Kinect RGB-D camera, which contains seven indoor environments named Chess, Fire, Heads, Pumpkin, Office, Red Kitchen and Stairs. Each environment includes 2 - 10 sequences of images. The ground truth camera poses are obtained by KinectFusion [39].

Cambridge Landmarks [19]. The outdoor dataset is composed of several large outdoor environments. In each scene, RGB images are gathered by mobile phone and divided in training and testing sequences with several hundred frames. The ground truth camera poses are calibrated by Visual SFM and sparse point cloud is also provided.

PGNet details. Input color and depth images are resized to 256×256 pixels with intensities from -1 to 1. For 7 Scenes, source point cloud with size 2048 are uniformly sampled from depth image. In Cambridge Landmarks, 2048 points are selected from reconstructed sparse point cloud, thus all training and testing images share same source points and differ at target points transformed by camera pose.

The DenseNet architecture for feature extraction is composed four Dense Blocks. With RGB-D source, color and depth information are passed to two networks with same architecture. In this situation, layer numbers of Dense Block are $2, 3, 6, 4$ respectively and the first three Dense Blocks share the same parameter. Both images are transformed to 512-dimension features, which are then concatenated to 1024 dimensions. When input only contains color image, it is converted to 1024 dimensions with layer number $6, 12, 24, 16$. In camera pose learning part, the paper adopts two LSTM layers with feature size 512 to learn features of 8 previous poses. Finally, the camera pose feature and image feature are connected to 1536 dimensions for point cloud generation. In decode stage, the paper outputs 2048 points with TopNet decoder. In detail, tree level is 6 and the node feature number is set to 8.

Besides above, network is optimised with ADAM optimizer [40] using the default parameters and a learning rate of 4×10^{-4}. The batch size of training is set to 8 on a Tesla P100 GPU. Whole training takes approximately 20k - 80k iterations in 7 Scenes and 80k - 160k iterations in Cambridge Landmarks to achieve loss converges.

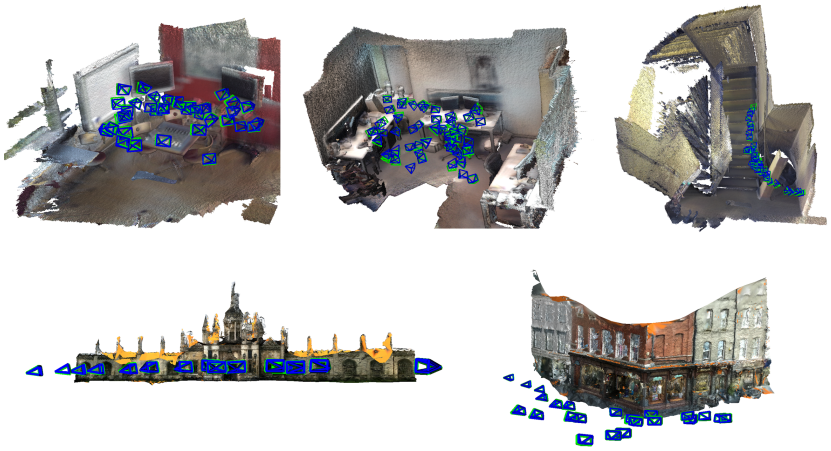

Fig. 3. Visualization of ground truth and prediction camera pose. For each subfigure, green and blue camera pose denote the ground truth and prediction respectively. Three scenes in first row are Chess, Office, Stairs and two in the second row are College, Shop.

Registration and Refinement details. In point cloud registration stage, some parameters in SUPER 4PCS should be determined, which contain overlap, delta and point number. The point number is set to fixed value 400, while delta and overlap is optimized iteratively based on the CD between registered point cloud and generated point cloud.

In refinement stage with RGB-D data, when establishing the feature space and searching the correspondence, 3D space is voxelized to fixed 3D grids, which aims to accelerate the run time. The 3D point in the same grid is regarded as same position. In experiment, the grid size is set to $100 \times 100 \times 100$. For refinement of RGB input, after converting world coordinate to camera view, the near region with size 30×30 is searched by ORB feature.

B. Localization results

Table I lists experimental results of median position and orientation in the indoor 7 Scenes and outdoor Cambridge Landmarks. In detail, contrastive approaches contain PoseNet [21], [23], VLocNet++ [30], Active Search [7], DSAC++ [27] and KFNet [28], which covers hand-feature based, learning based and coordinate regression based methods. Among all previous algorithms, VLocNet++ [30], DSAC++ [27] and KFNet [28] gain highly accurate performance, which are strong competitors.

In indoor 7 Scenes dataset, it is obvious that our approach achieves best mean results in position and orientation, which obtains at least $0.001m, 0.06^\circ$ gift compared with other state-of-the-art methods. Further more, among 7 scenes, our algorithm acquires best position results in 5 scenes and best orientation results in 6 scenes, which are marked bold in table I. In outdoor Cambrige Landmarks, our framework outperforms other approaches in orientation, but is less accurate than DSAC++ [27] and KFNet [28] in position.

For efficiency, the whole run time consists of four parts covering point cloud generation, registration, feature search and pose refinement. In total, it takes about 150ms per image on a Tesla P100 GPU with CUDA implementation of registration, feature search and refinement. Fig.3 illustrates visualization result of camera trajectories for several testing sequences of our method.

	PGNet without pose learning	PGNet	Coarse	Refinement
7 Scenes				
Chess	0.0154	0.0138	0.021,0.85	0.014,0.47
Fire	0.0313	0.0295	0.028,1.42	0.017,0.87
Head	0.0130	0.0125	0.028,1.45	0.014,0.79
Office	0.0204	0.0186	0.043,1.37	0.023,0.84
Pumpkin	0.1032	0.0982	0.045,1.17	0.024,0.87
Kitchen	0.0564	0.0528	0.042,1.65	0.024,0.95
Stairs	0.0644	0.0452	0.048,1.21	0.028,0.93
Mean	0.0434	0.0387	0.036,1.30	0.021,0.82
Cambridge Landmarks				
College	1.33	1.23	0.65,0.59	0.25,0.24
Hospital	1.68	1.52	0.54,0.45	0.14,0.30
Shop	1.71	1.45	0.30,0.58	0.10,0.30
Church	4.36	4.02	0.43,0.74	0.16,0.32
Mean	2.62	2.41	0.55,0.60	0.16,0.29

TABLE II

THE FIRST AND SECOND COLUMN SHOW THE CONSTRUCTION ERROR BY PGNET WITHOUT AND WITH CAMERA POSE LEARNING PART RESPECTIVELY. THE ERROR IS EXPRESSED THROUGH CD. THE THIRD AND FOURTH COLUMN ILLUSTRATE LOCALIZATION RESULT BY COARSE AND REFINE STAGE.

C. Detailed studies

In this subsection, two important parts in localization process are discussed, which contain additional previous camera pose learning part and pose refinement stage. The first and second column of table II demonstrate point cloud distance between predicted result and ground truth. In detail, the first column is obtained by removing camera pose learning part in PGNet, while the second column is evaluated by PGNet. It is evident that the point cloud construction error is reduced by adding pose learning part. Among all scenes in 7 Scenes, the

	PoseNet	Xue et al. [23]	VLocNet++	Active Search	DSAC++	KFNet	Ours
7 Scenes							
Chess	0.13,4.48	0.09,3.25	0.023,1.44	0.04,1.96	0.02,0.5	0.018,0.65	**0.014,0.47**
Fire	0.27,11.3	0.26,10.92	0.018,1.39	0.03,1.53	0.02,0.9	0.023,0.90	**0.017,0.87**
Heads	0.17,13.0	0.17,12.7	0.016,0.99	0.02,1.45	**0.01**,0.8	0.014,0.82	0.014,**0.79**
Office	0.19,5.55	0.18,5.45	0.024,1.14	0.09,3.61	0.03,0.7	0.025,**0.69**	**0.023**,0.84
Pumpkin	0.26,4.75	0.20,3.66	**0.024**,1.45	0.08,3.10	0.04,1.1	0.037,1.02	**0.024,0.87**
Kitchen	0.23,5.35	0.23,4.92	0.025,2.27	0.07,3.37	0.04,1.1	0.038,1.16	**0.024,0.95**
Stairs	0.35,12.4	0.23,11.3	**0.021**,1.08	0.03,2.22	0.09,2.6	0.033,0.94	0.028,**0.93**
mean	0.236,7.87	0.194,7.46	0.022,1.39	0.051,2.46	0.036,1.10	0.027,0.88	**0.021,0.82**
Cambridge Landmarks							
College	0.88,1.04	-	-	0.44,1.01	0.18,0.3	**0.16**,0.27	0.25,**0.24**
Hospital	3.20,3.29	-	-	**0.12**,0.40	0.20,0.3	0.18,**0.28**	0.14,0.30
Shop	0.88,3.78	-	-	0.12,0.40	0.06,**0.3**	**0.05**,0.31	0.10,**0.30**
Church	1.57,3.32	-	-	0.19,0.54	0.13,0.4	**0.12**,0.35	0.16,**0.32**
Mean	1.63,2.86	-	-	0.29,0.63	0.14,0.33	**0.13**,0.30	0.16,**0.29**

TABLE I

MEDIAN LOCALIZATION RESULTS ON INDOOR 7-SCENES AND OUTDOOR CAMBRIDGE LANDMARKS COMPARED WITH OTHER STATE-OF-THE-ART METHODS. UNITS OF POSITION AND ORIENTATION ARE METER(M) AND °. AT EACH DATASET, THE TOP ACCURATE RESULT IS MARKED BOLD.

most gift is reflected in Stairs scene, which has similar views at different position. As explanations for this improvement, on the one hand, the camera movement keeps the regularity and current pose is closely related to previous frames. On the other hand, the generated target point cloud is related the camera pose directly. Further more, when there are many similar views in the scene like Stairs, taking previous camera poses as input also avoids limitation of single image.

The third and fourth column reveal the effect of refinement stage, which gains $0.015m, 0.48°$ promotion in 7 Scenes and $0.39m, 0.31°$ in Cambrige Landmarks. Limited to the uncertainty during deep point cloud generation and registration algorithm, coarse localization process produces more error compared with other methods. Therefore the paper adopts pose refinement by establishing feature space, searching correspondences and solving the final pose.

Our paper adopts hand-crafted features to refine the result by learning features, which achieves state-of-the-art performance finally. Based on this pipeline, an interesting idea is that whether learning features are a better choice than hand-crafted ones for refinement. The explanation is as follows. On the one hand, the correspondence by hand-crafted features is pixel-to-pixel level. With accurate matches, camera pose can be calibrated very precisely. However, the match establishment causes errors with similar hand-crafted feature point, environmental change and so on. So this paper process deep point cloud generation, registration and building feature space to avoid these errors, which obtains accurate results. On the other hand, as far as we are concerned, researches for pose refinement with learning features is limited because of difficulty for refinement design. Graph convolutional network may be a choice. Authors will also make a study of this in our further work.

V. CONCLUSION AND FUTURE WORK

In this paper, inspired by combining hand-crafted and learning features, a novel localization algorithm framework has been proposed for RGB or RGB-D source, which contains coarse pose calibration stage and pose refinement stage. In coarse stage, the novel process composed of deep point cloud generation and registration has been proposed. Instead of outputting camera pose directly by deep neural network, authors have put forward a point cloud construction network called PGNet with color image, depth image and camera poses as inputs. Further more, aiming at improving accuracy, hand-crafted space has been established and point-to-point correspondences has been obtained by searching the space. Experiments on both indoor and outdoor datasets have demonstrated state-of-the-art localization performance compared to other methods.

For the future work, from authors' opinions, two aspects of efforts can be made to promote pose precision. For one aspect, more efficient point cloud decoder can be focused to decrease the construction error. For the other aspect, point cloud registration algorithm by deep neural network is also recommended.

VI. ACKNOWLEDGMENT

This paper is supported by National Natural Science Foundation of China (No. 62072020), National Key Research and Development Program of China (No. 2017YFB1002602), Key-Area Research and Development Progaram of Guangdong Province (No. 2019B010150001), and the Leading Talents in Innovation and Entrepreneurship of Qingdao (19-3-2-21-zhc).

REFERENCES

[1] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image and Vision Computing*, vol. 68, pp. 14–27, 2017.

[2] H. Lim, S. N. Sinha, M. F. Cohen, M. Uyttendaele, and H. J. Kim, "Real-time monocular image-based 6-dof localization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 476–492, 2015.

[3] E. Royer, M. Lhuillier, M. Dhome, and J. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[4] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *2008 12th IEEE International Symposium on Wearable Computers*, 2008, pp. 15–22.

[5] K. Guo, X. Li, and L. Xie, "Simultaneous cooperative relative localization and distributed formation control for multiple uavs," *SCIENCE CHINA Information Sciences*, vol. 63, no. 1, 2020.

[6] L. Liu, H. Li, and Y. Dai, "Efficient global 2d-3d matching for camera localization in a large-scale 3d map," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2372–2381.

[7] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[8] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1455–1461, 2016.

[9] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[10] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular slam for highly dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 393–400.

[11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[13] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[14] K. M. Han and Y. J. Kim, "Robust rgb-d camera tracking using optimal key-frame selection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] T. Caselitz, M. Krawez, J. Sundram, M. V. Loock, and W. Burgard, "Camera tracking in lighting adaptable maps of indoor environments," in *IEEE International Conference of Robotics and Automation (ICRA)*, Paris, France, 2020.

[16] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5231–5237.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[19] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[20] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2616–2625.

[21] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5974–5983.

[22] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637.

[23] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha, "Local supports global: Deep camera relocalization with sequence enhancement," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2841–2850.

[24] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.

[25] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," pp. 2930–2937, 2013.

[26] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[27] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4654–4662.

[28] L. Zhou, Z. Luo, T. Shen, J. Zhang, M. Zhen, Y. Yao, T. Fang, and L. Quan, "Kfnet: Learning temporal camera relocalization using kalman filtering," in *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[29] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," pp. 6939–6946, 2018.

[30] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, 2018.

[31] J. Valentin, M. Niebner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. S. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," pp. 4400–4408, 2015.

[32] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," pp. 3364–3372, 2016.

[33] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[34] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *arXiv preprint arXiv:1712.07262*, 2017.

[35] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[36] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

[37] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[38] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," vol. 33, no. 5, pp. 205–215, 2014.

[39] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking." in *ISMAR*, vol. 11, no. 2011, 2011, pp. 127–136.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.