Efficient Image Retrieval Based Mobile Indoor Localization

Ruoyun He*, Yitong Wang†, Qingyi Tao*, Jianfei Cai* and Lingyu Duan†
* Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore
E-mail: {ryhe, qytao, asjfcai}@ntu.edu.sg
† Institute of Digital Media, School of EE&CS, Peking University, Beijing, China
E-mail: {wangyitong, lingyu}@pku.edu.cn

***Abstract*—Vision based localization has been investigated for many years. The existing Structure from Motion (SfM) technique can reconstruct the 3D models based on the input images. The image retrieval and feature matching allow us to find the correspondence between the query image and the 3D model. According to these, the location can be easily calculated. In mobile scenarios, the limited CPU speed, memory storage and network latency bring in new challenges. The state-of-the-art solution can not be easily adopted due to the complicated calculation and large resource consumption. In this paper, we leverage the techniques developed during the MPEG-7 Compact Descriptors for Visual Search (CDVS) standardization, which aims to provide high performance and low complexity compact descriptors. We show that these techniques are suitable for mobile device and can achieve state-of-the-art retrieval performance in indoor environment. Besides, we propose additional components including blur measurement and result smoothing to improve the performance of the location calculation process. Based on these techniques, a whole system which enables fast vision based localization on mobile device is developed. We present experiments on the real world situation, showing that the system can strike a balance between accuracy and efficiency.**

***Index Terms*—Indoor Localization, Image Retrieval, Compact Descriptors, Feature Matching, Point Cloud**

I. Introduction

The location information of a mobile phone is very important for mobile multimedia applications and location based services (LBS). Smartphones now use different methods to obtain the location information such as using GPS, WIFI and cellular networks. However, they might not work well for the indoor environment or require additional setup. With the great improvements of the structure from motion (SfM) technique which can build 3D models from images and videos effectively [1][2] and the advance of image retrieval and feature matching techniques, the localization of a mobile device can be done via image search, i.e. computing the location of a query image by solving the perspective-n-point (PnP) problem [3].

Recently, the state-of-the-art algorithms can already provide an accurate location based on the 3D models and feature matching [4][5]. But most of these algorithms are quite complicated in terms of computation, which is totally not suitable for mobile devices. Compared with a personal computer, a mobile device has much slower CPU speed and limited memory capacity and storage space. It is unlikely to download the whole database of a large building into the device due to the large amount of data and limited data space of the mobile device. To solve this problem, client-server architecture is commonly used, where the mobile phone sends compressed query information to the server and the server returns the location information to the client [6][7]. In such situation, the network latency and bandwidth consumption need to be considered due to the slow uploading speed and high transmission cost. Because of these problems, most of the existing complicated visual based localization solutions can not be easily adopted in mobile device. There are still no good solutions for the mobile localization problem.

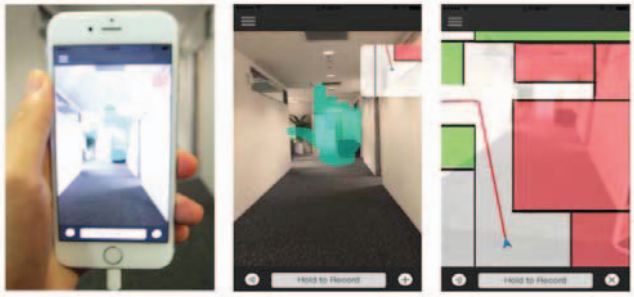


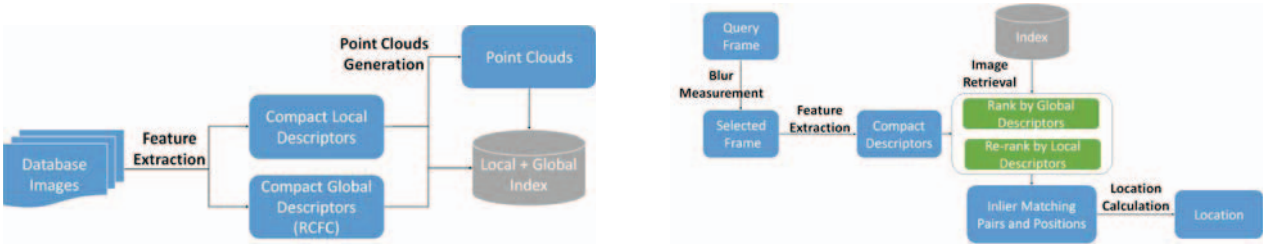
Fig. 1. The exemplar scenario of the proposed localization system.[1]

In this paper, we propose a fast and accurate vision based indoor localization system on mobile device based on recently developed compact descriptor and fast visual search technique, which can efficiently run on current mobile devices (See Fig. 1). In our experiments, we show that the proposed system can achieve state-of-the-art performance in the typical indoor dataset, with light computations and resource consumption.

II. Related work

In the early days, the image based localization problem was treated as an image retrieval problem. Robertson and Cpolla [8] proposed an image localization framework in 2004, which was one of the earliest studies in the field. Later, Zhang and

[1]A demo video of our proposed system is released on https://youtu.be/WyEFsLlrKcA



(a) Off-line Processing (b) Online Processing

Fig. 2. The framework of the proposed system.

Kosecka [9] improved the system by utilizing 2 matching images, the positions of which were known according to the GPS sensor, to triangulate the query image position. Since there was no 3D information in the images in these methods, they cannot give an accurate location result. Recently, with the development of SfM research, large 3D point cloud can be easily constructed based on the images [1][2]. Then, the image localization problem became a 2D-3D point matching problem. Irschara et al. [4] used a vocabulary tree based method for image retrieval together with SIFT local match to get the 2D-3D matching pairs. On the other hand, Li et al. [5] proposed a prioritized feature matching which could directly match the 3D points in the point cloud to the 2D points in the query image. However, all these works were targeted at the outdoor environment using desktop PC processing. In contrast, our considered mobile indoor localization is much more challenging due to the complex indoor environment and limited computing capability of mobile devices.

Recent studies started focusing on the indoor localization problem, which is also crucial to the simultaneous localization and mapping (SLAM) problem in robotics. Van Opdenbosch et al. [6] proposed a binary VLAD for image retrieval to decrease the network usage and achieved the state-of-the-art result. Schroth et al. [7] used a vocabulary tree based method for the data retrieval and pre-downloaded partial vocabularies to solve the network latency and improve the performance. All these studies focused on improving the running time efficiency. Their final localization results were not so accurate. Meanwhile, the MPEG-7 Compact Descriptors for Visual Search (CDVS) [10] standard had been developed to provide high performance and low complexity compact descriptors. During the CDVS standardization, several technologies made remarkable progress. In particular, novel compact descriptors were proposed for efficient image retrieval. Interest points detection and descriptor quantization methods were adopted to provide fast matching. Besides, other techniques such as fast geometry check [12] were incorporated to improve the search accuracy with few extra computation costs. Since CDVS techniques are customized for light computations and low resource consumption, we adopt them in our mobile localization system.

III. Mobile Localization System

Our system includes two parts: the off-line database processing and the online localization. The database images are processed during the off-line phase to build a localization database. In the online localization process, location will be calculated using a query image.

A. Off-line Processing

Fig. 2(a) shows the diagram of the offline sub-system. The input to this sub-system is the database images, and the output includes global descriptors, local descriptors and their corresponding 3D locations. The 3D locations are generated using SfM based on the local descriptors, and they will be used to localize the camera position during the online stage.

Feature Extraction. For our considered mobile scenario, we take advantage of the compact feature extraction technique proposed during the CDVS standardization, which includes the compact local descriptors and compact global descriptors. The local descriptors are generated by the Block based Frequency Domain Laplacian of Gaussian (BFLoG) detector [13] followed by a space transform and scalar quantization [14]. For the global descriptors, a Gaussian component selection based Rate-adaptive Compressed Fisher Code (RCFC) [11] is used, which is also adopted in the CDVS standard. Since BFLoG detection, space transform, scalar quantization and RCFC generation are all performed with low memory footprint and time complexity, they are very suitable to extract compact descriptors for not only database images at the server side but also the query image at the mobile end.

Point Clouds Generation. The goal here is to generate the compact 3D model from database images. We use the PMVS [1][2] algorithm to generate the 3D point cloud. The compact local descriptor is used to provide a fast binarized descriptor matching for image matching and registration. From the 3D point cloud, the relative camera poses for the dataset images can be known as well as the 3D point position of local descriptors. The local descriptor positions will be used in the later query stage for localization.

B. Online Processing

The input to the online localization sub-system is a query frame obtained from the camera in real-time, and the output is the location of the frame or failure to localize. Due to the

limited resources of the mobile phone, client-server architecture is adopted here. The feature extraction of the query image is performed in the client side while the server side is mainly for storing the database and doing image retrieval. The overall diagram is shown in Fig. 2(b). In the following, we introduce individual steps in detail.

Frame Selection by Blur Measurement. Feature extraction is a time consuming process and costs lots of CPU resources. Considering in practice, there are many unuseful frames such as blur images, so we propose a simple gradient based approach to reject the blur frames. It scores an image by calculating the average gradient value of each pixels, i.e.:

$$sign(I) = \begin{cases} 1 & if\, T < t \\ 0 & otherwise \end{cases} \quad (1)$$

where T denotes the average gradient of the frame:

$$T = \frac{\sum_{x \in I} |x(i,j) - x(i+1,j)| + |x(i,j) - x(i,j+1)|}{N} \quad (2)$$

I indicates all image pixels, N indicates the number of pixels in I, $x(i,j)$ denotes the gray scale value of the pixel (i,j) and t is the threshold to control the rejection rate.

Feature Extraction and Image Retrieval. For each query frame, the compact descriptors will be extracted. The size of the descriptor is $\sim$4KB for each frame. Considering the transmission cost is high and the chance of successfully matching with the previous retrieval result is high (see experiment results), the current local compact descriptors will be firstly matched with the last retrieval result. If it fails, the compact descriptors will be uploaded to the server, which will then return Top K similar images. After that, local descriptors will be matched and a fast geometric consistency checking [12] will be used to rerank the top K images. This reranking step consumes little time but improves the performance significantly. We refer readers to [11][12] for more details on global descriptors retrieval and fast geometric consistency checking. After reranking, top N relevant images will be retrieved (In our experience, K is set to 10 and N is set to 5). Finally, the compact local descriptors of all the final result images will be sent back to the client side for the location calculation. Once the mobile device gets the retrieval result, an accurate RANSAC based inlier matching method will be used and the inliers of each reference images will be put together for the subsequent location calculation.

Location Calculation. If there are enough matching inliers, the location of the query image can be calculated. Assuming that the camera intrinsic parameters are known, with the obtained matching pairs, the camera pose can be easily calculated using the interactive optimization method for minimizing reprojection error. Considering that there could have some frames with limited matching pairs or even some wrong pairs, we add a time based smoothing method to stabalize the calculated position P as:

$$P = \frac{\sum_{i=CF-n}^{CF} Pos(i) \times (CT - t(i))}{\sum_{i=CF-n}^{CF} (CT - t(i))} \quad (3)$$

where n indicates the number of nearby frames, CF is the index of current frame, $Pos(i)$ is the calculated position for frame i, CT is the current time and $t(i)$ is the time for frame i. This equation basically considers all the recently calculated locations and uses the past time as their weight.

IV. Experiments

A. Image Retrieval

Since image retrieval is the key to the final location calculation, in the first experiment, we evaluate the image retrieval performance of our system on a publicly available TUM Indoor Dataset [6]. The difference between our experiment and the experiment used in [6] is that the generated virtual images are not used in our work because we find that the final result is totally good without the virtual images. There are 128 query images captured by smartphone with manually annotated position and 6858 database images in this dataset. The retrieval performance is evaluated by $Precision@1$ (P1) and $Precision@3$ (P3). $Precision@N$ denotes the ratio of the relevant results to the all retrieved results up to a rank N. The reference image is considered as a relevant image when it is located with less than 5 meters to the location of the query image. We compare our used RCFC (implemented with 128 GMM centroids and 32 dimensional PCA-SIFT) with the BVLAD proposed in [6], the traditional VLAD [15] and Fisher Vector [16]. The experiment results are shown in Table I.

It can be seen from the table that RCFC outperforms the BVLAD, Fisher Vector and VLAD in $Precision@1$ with equal (resp., lower) descriptor dimension, which means equal or lower memory consumption. Besides, the fast geometric consistency checking based re-rank significantly improves the accuracy with little extra calculation. The retrieval performance and retrieval efficiency demostrate our retrieval system is more reliable compared to other state-of-the-art approaches.

B. System Evaluation

The second experiment focused on resource consumption and positioning accuracy. A 3D model of 500-square-meters office place was constructed using the proposed offline processing method. The client camera is travelling along a 40 meters long track with 2 meters wide recorded at 30fps with a resolution of 1920×1080. The velocity of the camera is stable which is about 0.5m/s and there are some dynamic background clutters like some passing-by person which does not appear in the database. The video is 79.7s seconds long and there are 2392 frames in total. All the frames are converted to 480×270 for processing.

We conduct 5 experiments using the query video and preprocessed model. The tests are run with iPhone6 (Client) and Intel Core i7-3610QM (Server). To evaluate the effects of individual component, we design 5 settings: (1). Baseline without the following components; (2). Baseline (1)+frame selection by blur measurement; (3). Baseline (2)+using features from multiple images to match; (4). Baseline (3)+matching with the last result firstly; (5). Baseline (4)+outlier position discard and final result smoothing.

(a) (b) (c) (d) (e)

Fig. 3. The performance of localization baselines. Red line refers to the ground truth and blue dots refer to the estimated locations.

TABLE I
THE PERFORMANCE OF RETRIEVAL BASELINES. K REFERS TO THE NUMBER OF KMEANS (OR GMM) CENTROIDS AND D REFERS TO THE DIMENSION.

Methods	K	D	P1	P3
VLAD	128	16384	0.55	0.44
FV	128	32768	0.59	0.45
BVLAD [6]	1024	65536	0.62	0.58
BVLAD [6]	128	8192	0.56	0.54
RCFC	128	8192	0.61	0.20
RCFC+Rerank 10	128	8192	0.72	0.57
RCFC+Rerank 50	128	8192	0.79	0.65

Fig. 3 shows the localization results under the 5 different settings. It can be seen that with more components being added, the localization (blue points) becomes more concentrated around the ground truth (red lines) with fewer outliners. Table II provides detailed results on network transmission count, failure count and final result count. The network transmission count indicates the number of times to transmit the data to the server side for image retrieval, which reflects the transmission cost. The failure count measures the number of times that the system fails to localize for consecutive 60 frames (2 seconds), which reflects the reliability of the system for localization. The final result count measures the number of times that obtain the location results (the same as the number of blue points in Fig. 3). We can see from the table that the blur measurement and matching with the last results significantly reduce the transmission cost. Although setting 5 has much less number of position results, its final results are more accurate (see Fig. 3(e)) and it does not reduce the system reliability.

TABLE II
THE POSITIONING RESULTS OF THE SYSTEM.

Test	Network Transmission Count	Failure Count	Final Result Count
1	2392	1	2054
2	2098	1	1829
3	2098	1	1829
4	1191	1	1829
5	1191	1	1486

In addition, we measure the average time consumed for different steps. At the mobile side, the blur measurement takes 2ms for a frame, the feature extraction needs 197ms and the feature matching uses 182ms. At the server side, it takes 37ms for retrieval. The system can run at more than 2fps on the mobile phone.

V. CONCLUSIONS

In this paper, we present a mobile vision-based localization system which enables fast and accurate localization in the mobile phone. Based on the robust 3D model and the efficient image retrieval techniques, our system achieves reliable and accurate positioning performance with low transmission cost.

ACKNOWLEDGEMENT

The research was carried out at the Rapid-Rich Object Search(ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, under its Interactive Digital Media (IDM) Strategic Research Programme.

This work was supported by the Chinese Natural Science Foundation under Contract No. 61271311, and the National Hightech R&D Program of China (863 Program) under Contract No. 2015AA016302.

REFERENCES

[1] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *IJCV*, vol. 80, no. 2, pp. 189–210, 2008.
[2] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *TPAMI*, vol. 32, no. 8, pp. 1362–1376, 2010.
[3] Y. Zheng *et al.*, "Revisiting the pnp problem: A fast, general and optimal solution," in *ICCV*, 2013.
[4] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *CVPR*, 2009.
[5] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *ECCV*, 2010.
[6] D. van Opdenbosch *et al.*, "Camera-based indoor positioning using scalable streaming of compressed binary image signatures," in *ICIP*, 2014.
[7] G. Schroth *et al.*, "Exploiting prior knowledge in mobile visual location recognition," in *ICASSP*, 2012.
[8] D. P. Robertson and R. Cipolla, "An image-based system for urban navigation." in *BMVC*, 2004.
[9] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *3D Data Processing, Visualization, and Transmission*, 2006.
[10] ISO/IEC JTC1/SC29/WG11/N12201, "Call for proposals for compact descriptors for visual search," 2011.
[11] J. Lin *et al.*, "Rate-adaptive compact fisher codes for mobile visual search," *Signal Processing Letters, IEEE*, vol. 21, no. 2, pp. 195–198, 2014.
[12] S. Lepsoy *et al.*, "Statistical modelling of outliers for fast visual search," in *ICME*, 2011.
[13] J. Chen *et al.*, "A low complexity interest point detector," *Signal Processing Letters, IEEE*, vol. 22, no. 2, pp. 172–176, 2015.
[14] S. Paschalakis *et al.*, "Cdvs ce2: Local descriptor compression," in *ISO/IEC JTC1/SC29/WG11/M28179*.
[15] H. Jégou *et al.*, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.
[16] F. Perronnin *et al.*, "Large-scale image retrieval with compressed fisher vectors," in *CVPR*, 2010.