Feature Matching With Labeled Keyframes to Reduce Keyframe Matching Error for SLAM–Based Camera Tracking

Seonghun Park and Junseong Bang

Abstract—Simultaneous localization and mapping (SLAM) is a technology that estimates a camera pose (i.e., its position and rotation) while constructing a 3D spatial map and generating keyframes, where the camera pose is computed by the feature matching between a current camera input image and a keyframe. In the large-scale space which generates a lot of similar keyframes, the SLAM-based camera tracking for a mobile augmented reality (AR) application is easy to fail due to a matching error between a camera input image and a keyframe that is selected among generated keyframes with the highest matching rate of feature points. With this motivation, in this paper, a feature matching approach with labeled keyframes are presented in order to reduce the keyframe matching error, where the labels for keyframes filter out several promising keyframes. Further, the computation load in pose estimation is reduced by the filtering with the labels. This is verified by experiments in indoor and outdoor environments.

Research Keywords— Augmented Reality, Camera Tracking, Labeled Keyframe, Simultaneous Localization and Mapping

1 INTRODUCTION

Augmented reality (AR) is a technology that overlaps virtual objects on a display in a real world environment. Simultaneous localization and mapping (SLAM) for AR is a technology that estimates a camera pose (i.e., its position and rotation), in order to overlap computer-generated virtual objects to the display of the user device (e.g., smartphones, smartpads, etc.). The SLAM constructs a 3D spatial map and generates keyframes. They are used in camera pose estimation by the feature matching between a current camera input image and a keyframe. Especially, SLAM using a camera sensor is called as the visual SLAM (vSLAM).

In placing 3D virtual objects at the appropriate position of a user device, a camera pose needs to be accurately computed. In the large-scale space which generates a lot of similar keyframes, the SLAM-based camera tracking for a mobile augmented reality (AR) application is easy to fail due to a matching error between a camera input image and a keyframe that is selected among generated keyframes with the highest matching rate of feature points. The failure in camera pose estimation disturbs a seamless mobile AR service.

SLAM with an extended Kalman filter is used in robotics [1]. For mobile AR applications, this method may be inappropriate due to heavy computation complexity which comes from the correlation among observations in each image, where the observation means the position of feature points in an image. For the reason, a camera pose in SLAM-based camera tracking for the mobile AR is computed by feature matching between a camera input image and a keyframe which is selected from a 3D pointcloud map that is created in advance [2, 3, 4, 5]. In the SLAM-based camera tracking, ORB (Oriented FAST and Rotation BRIEF) and SIFT (Scale Invariant Feature Transform) can be generally used to determine the degree of matching feature points between the both images, a camera input image and a keyframe. In this paper, the ORB is chosen due to the characteristics of the fast and rotation-invariant under a BSD license.

- *S. Park is with the Computer Software, the Korea University of Science & Technology (UST), Daejeon, South Korea. E-mail: shpark@ust.ac.kr*
- *J. Bang (corresponding author) is with the Creative Contents Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. E-mail: hjbang21pp@etri.re.kr*

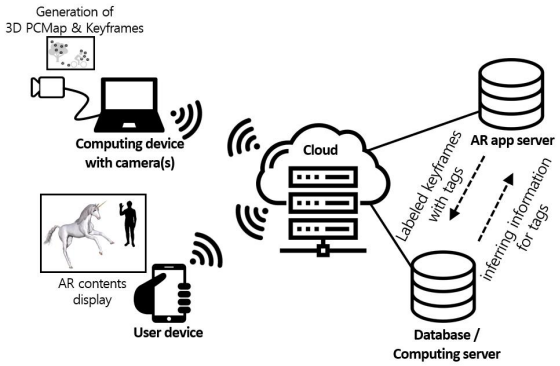


Fig. 1. An example of AR service with a cloud server.

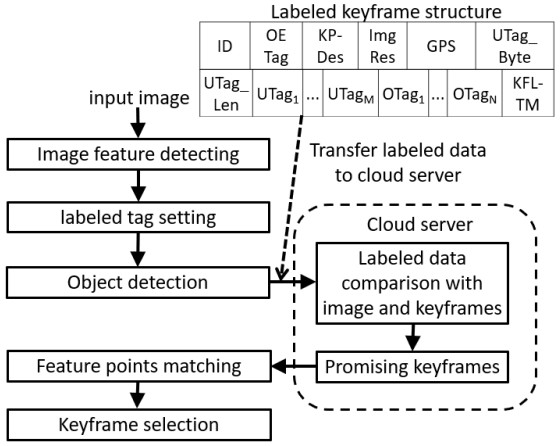


Fig. 2. A process for selecting promising keyframes.

In this paper, a feature matching approach with labeled keyframes are presented in order to reduce the keyframe matching error, where the labels for keyframes filter out several promising keyframes. Further, the computation load in pose estimation is reduced by the filtering with the labels. In Section 2, the feature matching approach with labeled keyframes is described. In Section 3, the performance is verified by experiments in indoor and outdoor environments and the results are discussed. Section 4 concludes this paper.

2 IMAGE FEATURE MATCHING USING LABELED KEYFRAME

A cloud server can be used for an AR services in order to manage resources (e.g., 3D pointcloud maps, graphic images) for AR services and to perform computational tasks (e.g., deep learning-based image recognition) [6]. An example of these kinds of AR services with a cloud server is shown in Fig. 1.

SLAM-based camera tracking builds a 3D pointcloud map (PCMap) with generating keyframes. The PCMap is stored in an AR app server on the cloud and it is downloaded to a user device (e.g., smartphones, smartpads, etc.) as it needs. In the feature matching approach with labeled keyframes, the information (e.g., GPS, recognized objects) to labeled keyframes is managed. Incomplete parts of the information can be fill out by communicating with neighbor servers and inferring with machine learning or deep learning algorithms. This can be performed during idle time of a cloud server, e.g., object recognition in an image [7]. With the information, keyframes are labeled (e.g., GPS, object tags) and classified. A tag for a labeled keyframe can also be added by a user such as a developer or a service provider for the purpose of keyframe classification, e.g., floor or room of the building. To recognize a space for a mobile AR application, a 3D PCMap for the SLAM-based camera tracking is downloaded from the cloud server. The labels for keyframes filter out several promising keyframes. With the promising keyframes, the feature matching error for pose estimation is reduced.

A labeled keyframe is stored with meta data as shown in the labeled keyframe structure of Fig. 2. The KFL-ID is a unique identification number of the corresponding keyframe. The OETag (i.e., Outdoor Environment Tag) with 2 bits checks whether the keyframe is generated an outdoor environment or not. When unspecified, the bits are set to '00'. For the outdoor (or the indoor), the bits are set to '10' (or '01'). The KP-Des means the information to a feature descriptor that is used in generating keyframes. This information is necessary because detected feature points (or other features) in a keyframe are heavily dependent on the feature descriptor. So, in the feature matching, the same feature descriptor has to be used. The feature detection for object classification can be performed in a cloud server [8]. In this paper, ORB is used for the feature detection and description. The ORB feature is capable of extracting features at high speed and detecting feature points in image scale changes due to rotation and motion of the image as the camera rotates [9]. The ImgRes is represent the resolution of a keyframe image. This information is used to adjust the resolution of a keyframe to that of a camera input image, in order to perform a feature matching task in the same condition. The GPS field stores the information of the location (i.e., latitude and longitude) that a keyframe is generated while constructing a 3D pointcloud map. The

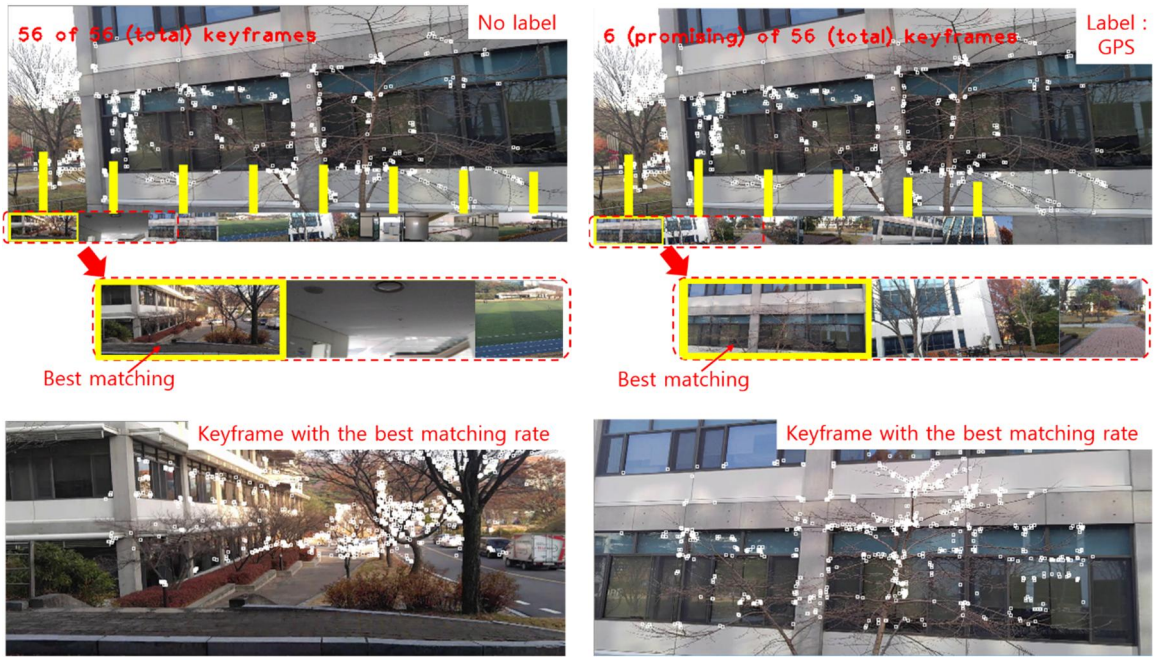


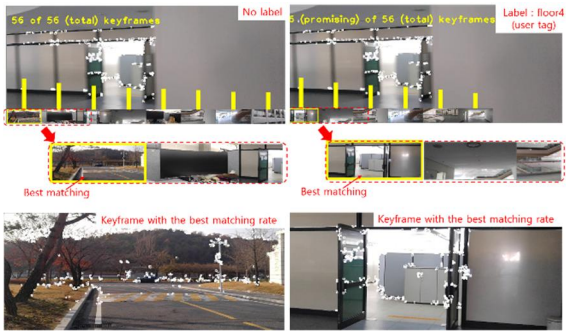
Fig. 3. Keyframe selection (a) without GPS information, and (b) with the GPS.

UTag_Byte and UTag_Len represent the number of user-defined tags to a keyframe and the byte length for the user tags, respectively. Up to M user tags can be used as needed (i. e. $\mathrm{UTag}_1 \cdots \mathrm{UTag}_M$). The OTag field stores object recognition information of a camera input image or keyframe. If several types of objects are detected, N object tags (i. e. $\mathrm{OTag}_1 \cdots \mathrm{OTag}_N$) can be used. The KFL-TM is an indicator indicating the end of a frame structure of the meta data for a labeled keyframe. Fig. 2 also shows a process for improved feature matching by selecting promising keyframes. Labeled tag information (i.e. GPS information, user tags, and object tags etc.) of a camera input image are transmitted to the server from a user device. Then, a keyframe having a matching labeled tag information among the received labeled tag information is selected as a promising keyframe. Feature matching with a camera input image is performed only with the selected promising keyframes.

3 Simulation Results

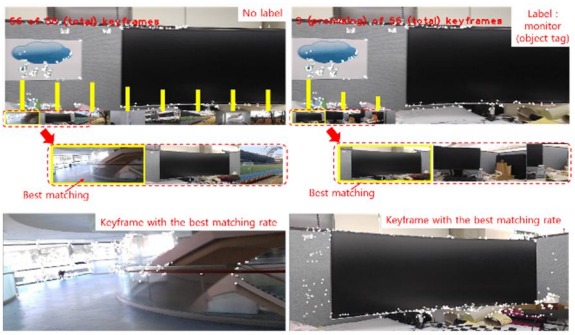
In order to examine this method, camera tracking based on ORB-SLAM is implemented with OpenCV 3.3.0 whose library is used for feature point extraction, and feature point matching. The program is developed with C++ language (Visual Studio 2017) on Windows 10.

The experiment results are shown in Fig. 3, Fig. 4, and Fig. 5. For these experiments, 56 keyframes generated in a large-scale space in indoor and outdoor environments are used. Keyframe selection without/with GPS information is shown in Fig. 3. In an image, feature points are detected by the ORB and marked with white squares. In Fig. 3(a), a mismatched keyframe is selected though feature matching because the keyframe set includes a lot of keyframes with the similar distribution of feature points. This causes the failure of camera tracking. While, in Fig. 3(b), the probability to select an appropriate keyframe for pose estimation is increased because promising keyframes at the reduced number is compared by the filtering with the labels. The both first images of Fig. 3(a) and Fig. 3(b) are the same camera input image. Its feature points are displayed in the image. The keyframes are ordered by the matching rate (i.e., the similar distribution of feature points), which are displayed with yellow bars. The first ones in the bottom sides of the both first images are the keyframe with the best matching, and they are marked with a yellow border. In the both second images which are magnified, the keyframes which have feature matching rates of the highest and the next values are showned. The both third images are the selected keyframes for pose estimation. Compared to the third image of Fig. 3(a), an appropriate keyframe



(a) (b)

Fig. 4. Keyframe selection (a) without a user-defined tag, and (b) with the tag.



(a) (b)

Fig. 5. Keyframe selection (a) without an object tag, and (b with the tag.

in the third image of Fig. 3(b) is selected due to the filtering with the GPS information. With the GPS information, the distance between the camera input image and the keyframe is calculated by using the haversine formula to obtain the shortest distance between two points on the sphere [10].

$$\cos\theta = \sin\phi_A \sin\phi_B + \cos\phi_A \cos\phi_B \Delta L \quad (1)$$

$$D = R\theta \quad (2)$$

Assuming that the distance between the point A and the point B is obtained, ϕ_A is the latitude of A, ϕ_B is the latitude of B and ΔL is the difference of the longitude between the point A and B. θ is calculated by (1), and in (2), the distance D between two points can be determined by multiplying the earth's equatorial standard radius, 6378.14 km. For example, feature matchings are performed with 56 keyframes for feature matching without the GPS (Fig. 3(a)) and 6 keyframe for that with the GPS (Fig. 3(b)), respectively. This means that the accuracy of feature matching and the speed of camera tracking are enhanced. The matching error is caused by computing the matched rate with feature points of a camera input image and a keyframe on the two-dimension. Also, the matching error is easy to be increased by an image with the patterned objects (e.g., windows at the building) and natural objects (e.g., trees).

In the case of no information to the GPS, user-defined and object tags can be used. Keyframe selection without/with a user-defined tag (floor4 of building-7) is shown in Fig. 4. Compared to the third image of Fig. 4(a), only 6 keyframes out of 56 keyframes in the third image of Fig. 4(b) are used for feature matching. Keyframe selection without/with an object tag (e.g., monitor) is shown in Fig. 5. Since the matching error in an image that is difficult to detect feature points or that the feature points are concentrated on certain areas is more increased, an object tag is useful, where the object tag can be generated by the deep learning at a cloud server. Compared to the third image of Fig. 5(a), only three keyframes in the third image of Fig. 5(b) are used for feature matching. The presented feature matching with labeled keyframes are to reduce the keyframe matching error and the computation load.

4 Conclusion

In this paper, the presented feature matching with labeled keyframes helps to SLAM-based camera tracking for seamless mobile AR services in indoor and outdoor environments (especially, for the large-scale space). With this approach, the accuracy of feature matching and the speed of camera tracking is enhanced.

Acknowledgment

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017.

References

[1] N. Ayache and O. Faugeras, “Maintaining Representation of the Environment of a Mobile Robot,” IEEE Trans. on Robotics and Automatation, vol. 5, no. 5, pp. 804-819, May. 1989.

[2] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” Proc. of the Sixth IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR), Nara, Japan, Nov. 2007.

[3] A.J. Davison, I.D. Reid, and N.D. Molton, “MonoSLAM: Real-Time Single Camera SLAM,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052-1067, Jun. 2007.

[4] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. on Robotics, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[5] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-Time Monocular Visual SLAM with Points and Lines," in Int. Conf. in Robotics and Automation (ICRA), pp. 4503–4508, May. 2017.
[6] Z. Huang, W. Li, P. Hui, and C. Peylo, "CloudRidAR: A Cloud-based Architecture for Mobile Augmented Reality," in Proc. of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology Based Systems, pp. 29-34, Jun. 2014.
[7] J. Huang, and S. You, "Detecting Objects in Scene Point Clolud: A Combinational Approach," in Int. Conf. on 3D Vision (3DV), pp. 175-182, Jun. 2013.
[8] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 105–119, Jan. 2010.
[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in IEEE Int. Conf. on Computer Vision (ICCV), volume 13, pp. 2564-2571, Nov. 2011.
[10] R. Bullock, "Great Circle Distances and Bearings Between Two Locations," http://www.dtcenter.org/met/users/docs/write_ups/gc_simple.pdf. 2007.

Seonghun Park He received the B.S. degree in Information Communication Engineering from the University of Baekseok, Cheonan, South Korea, in 2016. Currently, he has studied for the M.S. degree in Computer Software, the Korea University of Science & Technology (UST), Daejeon, South Korea.

Junseong Bang He received the B.S. degree in Computer Science Engineering from Hanyang University, Ansan, South Korea, in 2006; the M.S. and Ph.D. degrees in Information and Communications from Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2009 and 2013, respectively. Since 2013, he has worked at Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. In 2016, he also joined at University of Science and Technology (UST), Daejeon, South Korea. Currently, he is a Senior Researcher at ETRI and an Associate Professor at UST.