2015 IEEE International Symposium on Mixed and Augmented Reality

[POSTER] Abecedary Tracking and Mapping: a Toolkit for Tracking Competitions

Hideaki Uchiyama*
Kyushu University

Takafumi Taketomi†
Nara Institute of Science and Technology

Sei Ikeda‡
Ritsumeikan University

Joao Paulo Silva do Monte Lima§
Universidade Federal Rural de Pernambuco

Abstract

This paper introduces a toolkit with camera calibration, monocular visual Simultaneous Localization and Mapping (vSLAM) and registration with a calibration marker. With the toolkit, users can perform the whole procedure of the ISMAR on-site tracking competition in 2015. Since the source code is designed to be well-structured and highly-readable, users can easily install and modify the toolkit. By providing the toolkit, we encourage beginners to learn tracking techniques and to participate in the competition.

Index Terms: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 Introduction

Camera pose tracking is an essential process for merging real and virtual objects in augmented reality. In order to evaluate the techniques, on-site tracking competitions have been organized in ISMAR since 2008. In 2015, we organize the competition for evaluating visual SLAM techniques. In the competition, the participants first acquire the world coordinate system at a starting area, then move a device along a given path in an unknown environment, and finally do tasks at given coordinates.

In order to encourage beginners to learn tracking techniques and to participate in the competition, we first provide an all-in-one toolkit for performing the whole procedure of the competition. The toolkit includes camera calibration, visual SLAM and registration with a calibration marker. The source code is (i) available online under the modified BSD licenses, (ii) structured to be highly readable with less lines of code, and (iii) dependent on minimum external libraries. Therefore, even undergraduates can easily install the toolkit within a few hours, and implement their own ideas on the toolkit. Since the toolkit is designed to be educational, we name it abecedary tracking and mapping (ATAM). In this paper, we describe the implementation of the techniques used in the toolkit.

2 ATAM

ATAM follows keyframe based SLAM proposed in PTAM [5] and is composed of initialization, tracking, mapping, bundle adjustment (BA) [6] and relocalization as illustrated in Figure 1. The difference is that both tracking and mapping run on the same thread while BA runs on another. We calibrate a camera beforehand [10] and use undistorted images for all the processes. The source code is available online[1] and compiled with C++ Standard Template Library, OpenCV 3.0[2], and a sparse BA library[3].

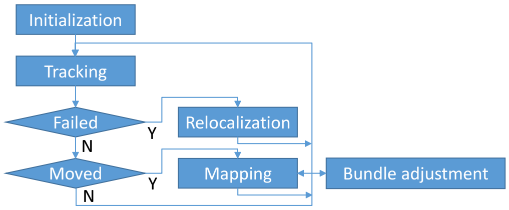

Figure 1: Flow of ATAM. Tracking and mapping run on one thread, while BA runs on another.

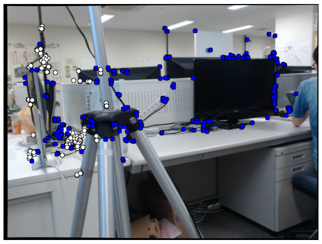

Figure 2: Interactive initialization. After mapping keypoints as while circles, new keypoints are detected for mapping as blue ones.

2.1 Interactive Initialization

As in PTAM, users press a button to start ATAM. FAST keypoints [8] are detected in the first keyframe and tracked in incoming frames with the Lucas-Kanade feature tracker (KLT) [1]. When users press the button again, the second keyframe is captured and the relative pose between the first keyframe and the second one is computed from tracked keypoints [7]. Finally, the 3D positions of the keypoints are computed by triangulation and optimized with BA. After initialization, new keypoints are detected in the second keyframe and tracked with KLT for mapping as illustrated in Figure 2. Note that the quality of the initialization is checked with the mapping criterion in Section 2.3, and the initialization is not finished until satisfying the criterion.

2.2 Tracking

In order to track camera poses, 2D-3D correspondences are acquired in every frame with the following three tracking techniques.

1. Tracking mapped keypoints in consecutive frames.

*e-mail: uchiyama@limu.ait.kyushu-u.ac.jp
†e-mail: takafumi-t@is.naist.jp
‡e-mail: ikeda.sei.jp@ieee.org
§e-mail: jpsml@cin.ufpe.br

[1] https://github.com/CVfAR/ATAM
[2] http://opencv.org/
[3] http://www.uco.es/investiga/grupos/ava/node/39

978-1-4673-7660-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMAR.2015.63

2. Projecting all the mapped keypoints to the current frame.
3. Matching the current frame with the nearest keyframe.

First, mapped keypoints in the previous frame are tracked in the current one with KLT. From the tracked mapped keypoints, a camera pose is computed by solving the Perspective-n-point problem. Second, all the keypoints in the map are projected onto the current frame and matched if FAST keypoints are detected near the projected ones. Third, keypoints in the current frame are matched with those in the nearest keyframe with ORB descriptors [9] as well as relocalization in Section 2.5.

2.3 Mapping

Keypoint correspondences between two keyframes are triangulated and inserted into the map. A current frame is stored as a keyframe if the normalized baseline B between the current frame c and the nearest keyframe nk is wider than a certain threshold. B is computed as

$$B = \frac{\|(-\boldsymbol{R}_c^{-1}\boldsymbol{T}_c) - (-\boldsymbol{R}_{nk}^{-1}\boldsymbol{T}_{nk})\|/2}{\|\boldsymbol{P}_m - ((-\boldsymbol{R}_c^{-1}\boldsymbol{T}_c) + (-\boldsymbol{R}_{nk}^{-1}\boldsymbol{T}_{nk}))/2\|} \tag{1}$$

where $[\boldsymbol{R}_c|\boldsymbol{T}_c]$ is the pose of the current frame, $[\boldsymbol{R}_{nk}|\boldsymbol{T}_{nk}]$ is that of the nearest keyframe and $\boldsymbol{P}_m$ is the 3D median coordinate of tracked mapped keypoints. The correspondences are obtained by tracking keypoints from the previous keyframe to the current frame with KLT. Since KLT runs in real time, one thread is sufficient for both tracking and mapping. By contrast, epipolar search for computing the correspondences in PTAM is highly computational.

2.4 Bundle Adjustment

Both keyframe poses and points in the map are optimized with BA [6]. With M keyframes, the cost function E is designed as

$$E = \sum_{i}^{M} \sum_{j \in \boldsymbol{F}_i} \left\| \boldsymbol{p}_{ij} - proj\left(\boldsymbol{T}_i, \boldsymbol{R}_i, \boldsymbol{P}_j\right)) \right\| \tag{2}$$

where $\boldsymbol{F}_i$ represents a set of points observed in keyframe i, $\boldsymbol{p}_{ij}$ represents an observed image coordinate of point j in keyframe i, $\boldsymbol{T}_i$ and $\boldsymbol{R}_i$ represent translation and rotation of keyframe i and $\boldsymbol{P}_j$ is 3D coordinate of point j. $proj()$ is a function for projecting the point $\boldsymbol{P}_j$ to the keyframe i using $\boldsymbol{T}_i$ and $\boldsymbol{R}_i$. In the current implementation, local BA with a few keyframes is incorporated and runs every time a new keyframe is stored. Global BA with all the keyframes can also be implemented.

2.5 Interactive Relocalization

We provide an interface for indicating a desirable camera position for relocalization if tracking has failed similar to [3]. The keyframe for relocalization is automatically selected based on the camera position where tracking has failed or manually selected on the interface. In the interface, edges detected in the selected keyframe are overlaid onto the camera image as illustrated in Figure 3. Relocalization is based on matching the current frame with the selected keyframe with ORB descriptors.

3 Registration

In ATAM, the local coordinate system is equivalent to the camera coordinate system of the first keyframe and can be registered with a different coordinate system. The registration is generally necessary for acquiring the scale of real environment in monocular SLAM [4]. In the competition, the world coordinate system will be defined on a calibration marker.

The transformation $[\boldsymbol{R}|\boldsymbol{T}]$ between camera pose in the world coordinate system $[\boldsymbol{R_w}|\boldsymbol{T_w}]$ and that in local one $[\boldsymbol{R_l}|\boldsymbol{T_l}]$ is described as

$$\begin{bmatrix} \boldsymbol{R_w} & \boldsymbol{T_w} \\ \boldsymbol{0} & 1 \end{bmatrix} = \begin{bmatrix} s\boldsymbol{R_l} & s\boldsymbol{T_l} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{3}$$

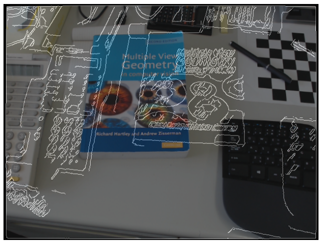

Figure 3: Interactive relocalizaion. Edges detected in a keyframe image for relocalization are overlaid onto a camera image.

where s is a scale factor from the local coordinate system to the world one. The scale factor is computed from both world and local camera poses at view i and view j as

$$s = \frac{\|\boldsymbol{T}_{wi} - \boldsymbol{R}_{li}\boldsymbol{R}_{lj}^{-1}\boldsymbol{T}_{wj}\|}{\|\boldsymbol{T}_{li} - \boldsymbol{R}_{li}\boldsymbol{R}_{lj}^{-1}\boldsymbol{T}_{lj}\|} \tag{4}$$

where subscripts w and l represent the world coordinate system and the local one respectively. We can compute the average scale factor from multiple sets of the world and local camera poses. After computing the factor, $[\boldsymbol{R}|\boldsymbol{T}]$ is computed by Equation 3. With multiple sets of the camera poses, the average translation is calculated and the average rotation is calculated by extracting orthogonal elements from summed rotation matrix by SVD [2].

4 Conclusion

This paper introduce a complete toolkit for performing the on-site tracking competition of ISMAR. Though the toolkit is initially designed for tracking competitions, its usage is obviously not limited to the competitions. Users can develop any applications such as 3D interfaces and robot motion control with the toolkit. In lectures and exercises on computer vision, the toolkit can be utilized for teaching SLAM techniques. We expect beginners to use the toolkit, to develop their own ideas and to participate in the competition.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 26880016 and 50610664.

References

[1] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm, 2000.

[2] W. D. Curtis, A. L. Janin, and K. Zikan. A note on averaging rotations. In *Proc. VRAIS*, pages 377–385, 1993.

[3] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox. Interactive 3D modeling of indoor environments with a consumer depth camera. In *Proc. UbiComp*, pages 75–84, 2011.

[4] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

[5] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. ISMAR*, pages 1–10, 2007.

[6] M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[7] D. Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26(6):756–770, 2004.

[8] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. ECCV*, pages 430–443. 2006.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571, 2011.

[10] Z. Zhang. A flexible new technique for camera calibration. *TPAMI*, 22(11):1330–1334, 2000.