2016 International Conference on Image, Vision and Computing

Register Based on Efficient Scene Learning and Keypoint Matching for Augmented Reality System

Zhen-Wen Gui
China Electronics Technology Group Corporation NO. 7 Research Institute
Guangzhou, China
e-mail: whut_gzw@163.com

Abstract—**Register is steadily gaining in importance due to the drive from various computer vision applications, such as Augmented Reality (AR), mobile computing, and human-machine interface. Efficient keypoint-based approachs are widely used in scene register. This paper focuses on designing a robust and flexible registration method for wide-area augmented reality applications using scene recognition and natural features tracking techniques. A linear structured SVM classifier is used to perform scene learning online, which allows us to quickly adapt our model to a given environment. And the hybrid tracking strategy is implemented by combining both wide and narrow baseline techniques. The experiments have been conducted to demonstrate the validity of our methods.**

Keywords-augmented reality; natural features; wide-area registration; scene recognition

I. Introduction

Augmented Reality (AR) is to add virtual objects to real environments, allowing computer-generated 3-D graphics or 2-D information to be overlaid on the real world in such a manner as to enhance people's understanding of the real word [1], [2]. Registration between real and synthetic worlds is one of the major technological issues in order to create AR systems. As the user moves his/her head or viewpoint, the virtual objects must be properly aligned with the objects in the real world, or the coexistence of the virtual world and the real world will be compromised.

Keypoint-based method for scene register has become a corner stone of modern computer vision, enabling great advances in areas such as AR and Simultaneous Localization and Mapping (SLAM) [3].These register approaches model an object as a set of keypoints, which are matched independently in an input image. Robust estimation procedures based on RANSAC [4]-[6] are then used to determine geometrically consistent sets of matches which can be used to infer the presence and transformation of the scene. There has been a great deal of progress in making these approaches suitable for real-time applications.

The applications for the approach of keypoint-based register include location recognition [7]-[10], autonomous robot navigation [11]-[13] and augmented reality [14]-[16]. Broadly speaking, there are two prevalent approaches that have been used to achieve registration for AR. The first addresses the method of Simultaneous Localization and Mapping (SLAM), where the camera is localized within an unknown scene. In the second method use the knowledge of a prior map or 3D scene model. Several recent methods fall in the second method [7], [8], [14], [17], and this renewed interest has been sparked by progress in structure from motion (Sfm) [18], [19], which makes it possible to easily reconstruct large scenes in great detail.

In this paper, we propose a new approach for continuously register within wide area, which have already been reconstructed using Sfm. Our algorithm runs over long periods with low fluctuations in the frame-rate. At its core lies a fast keypoint tracker. Keypoints (Binary robust invariant scalable keypoints) [20] from one frame are tracked in the consecutive frame by optical flow tracker, and the lost points are recovered by matching to candidate keypoints within a local search neighborhood [21] in the next frame when they are lost.

The rest of this paper is organized as follows: Section II is the related work and our contributions. Section III is the registration method. Section IV discusses natural features detecting and matching problem. Section V gives the online mapping and camera tracking method. Section VI deals with the problems of the nature features online learning and tracking. Section VII shows some experimental results. Section VIII discusses is a conclusion.

II. Related Work and Our Contributions

Simultaneous Localization and Mapping (SLAM) and online Structure from Motion (SFM) are two kinds of prevalent techniques that have been used to achieve wide-area registration for AR [22]. Visual SLAM systems have recently been used for realtime AR [23], by utilizing Parallel Threads for Tracking and Mapping (PTAM) [15], using multiple local maps [24] and performing fast relocalization using random ferns [25]. However, these approaches are susceptible to the problem of drift and error accumulation in the pose estimate, and existing solutions for fast relocalization do not scale to larger scenes. Sfm is used to estimate 3D coordinates for the landmarks in recent work [7], [17], [26]. The approaches scale well and some variants use the GPU [7]. However, these methods address the single-image registration problem, and are not fast enough for real-time registration from video.

Recently, a continuous registration method was proposed for scenes reconstructed using Sfm [14] which uses keyframe recognition repeatedly on video frames to indirectly recover 2D-3D matches. The traditional powerful SIFT algorithm

978-1-5090-3755-1/16/$31.00 ©2016 IEEE

has large computation which is the bottleneck in the repeatedly matching method, which runs at 6 fps on a single thread and at 20 fps using parallel threads on four cores. The binary descriptors [21], [27] can be used to amortize the cost of feature computation over time, which is performed on low-power device. Instead of keyframe-based matching, we use 2D-to-3D matchi32ng interleaved with tracking. This allows us to exploit spatiotemporal coherence and lowers the per-frame latency.

In our research, we proposed to deal with learning problem for scenes by using SVM classifiers. Each scene is represented by a predefined numbers of 3D points and local binary features detected from the ten keyframes. These local features will be used to train the multi-index table built in advance for online scene recognition use.

III. REGISTERATION METHOD

Most of natural scene have many depth features, such as outdoor buildings, automobiles; room desks, bookcases, etc. Therefore, we need to build 3D model and select a small number of key frames for pre-unknown scenes. When the 3D model and key frame information is known, we can effectively achieve the outdoor scene tracking. In this paper, an effective scene learning and tracking algorithm is proposed for registration based on the construction of 3d model. The flowchart of which as shown in Fig. 1, the specific steps are as follows:

Offline stages:

Step 1: The 10 keyframes of nature scene are captured from different perspectives, the feature points of keyframes are extracted and matched;

Step 2: The 3d models of the real scene are reconstructed;

Step 3: The point 2D-3D mapping table of the 3d models is established;

Step 4: The descriptors of the reference image for all the scenes are generated;

Step 5: The 3D point weights are trained for all scenes;

Step 6: The multi-index list for all image descriptors is trained.

Online stage:

Step 7: Capture frame of the current scene, extract feature points and calculate the feature descriptors;

Step 8: Match features between the current frame and all keyframes by searching multi-index list;

Step 9: Identify the current scene according to the matching features;

Step 10: If the identification is successful, the current scene is present in the reconstructed scenes, turn to step (12);

Step 11: If the identification is failure to and count number of failure, when the number of identification failure is more than three times, return to step (1).

Step 12: Perform online scene learning to establish correspondence between the 3d coordinates of the scene to and the 2d coordinates of the image;

Step 13: Solve the rotation and translation matrices R, T for the current image with respect to the real scene using PNP algorithm;

Step 14: Assign the registration matrix to the virtual camera to execute fusion display for virtual and real scene;

Step 15: Use optical flow tracking algorithm to track subsequent image frames; When the number of tracked points are less than the threshold is performed to recover the lost features;

Step 16: If tracking failure to subsequent frame, go back to step (7); otherwise return to (12).

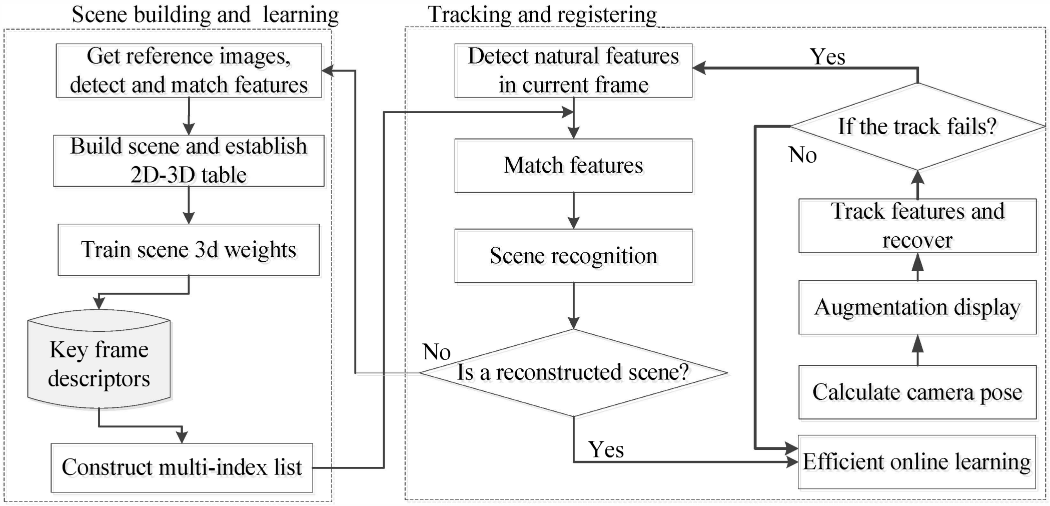

Scene building and learning
Tracking and registering
Get reference images, detect and match features
Build scene and establish 2D-3D table
Train scene 3d weights
Key frame descriptors
Construct multi-index list
Detect natural features in current frame
Match features
Scene recognition
Is a reconstructed scene?
No
Yes
If the track fails?
Yes
No
Track features and recover
Augmentation display
Calculate camera pose
Efficient online learning

Figure 1. Flowchart of the proposed algorithm

IV. NATURAL FEATURES DETECTING AND MATCHING

In 2011, the BRISK (Binary Robust Invariant Scalable Keypoints) [25] are found to be eminently suitable for low-power device because of a lower computational complexity and memory efficiency. Nearest Neighbor (NN) search on binary codes has been widely applied in image recognition, local features matching and parameter estimation. There has been growing interest in mapping image data onto compact binary codes for fast near neighbor search in scene recognition. Because binary codes comparisons require just a small number of machine instructions; millions of binary codes can be compared to a query in less than a second. But the most compelling reason for binary codes is their use as direct indices (addresses) into a hash table, yielding a dramatic increase in search speed compared to an exhaustive linear scan.

The value of index is used for the cluster center and radius R is used as the hamming threshold of the sub-vectors so as to all the sub-vectors of image descriptors are assigned to the corresponding linked list as shown in Fig. 2. Detailed steps are:

1) Calculate hamming distances between the sub-vectors and the cluster center.

2) When the hamming distance is less than or equal to the threshold R, descriptor ID and image ID of sub-vector are inserted into the inverted list according to the serial number of sub-vectors.

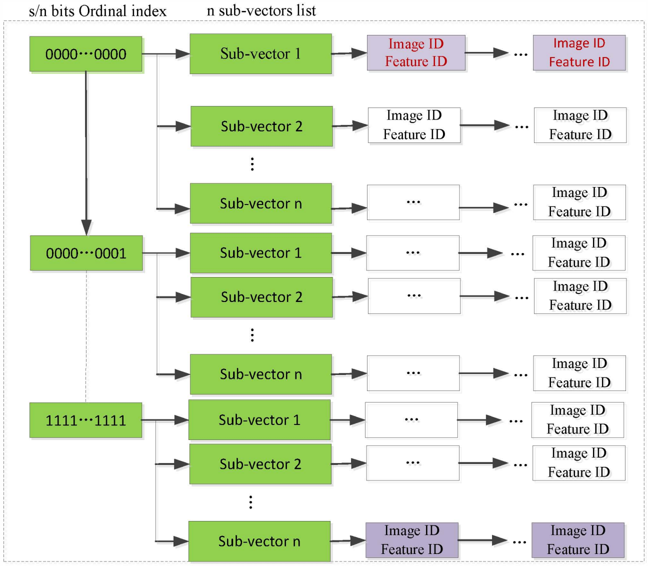


Figure 2. Ordinal multi-index list

V. Scene Building and Training

The scene reconstruction and reference image descriptor detection is completed in the offline stage which is foundation of the online registration process, this stage includes:

Step 1) Scene reconstruction and feature descriptor generation. The process of scene reconstruction involves selecting keyframes of scenes, calculating the two-dimensional position for the feature point of the keyframes and the corresponding three-dimensional position, finally computing initial camera pose.

Step 2) 2D-3D mapping table is established for the keyframe so as to find quickly the corresponding 3D position of image feature points.

Step 3) Training initial weight for 3D points.

Step 4) Training multi-index list of BRISK feature points for all key frames. When online tracking started, the scene is first need to be recognized by searching the multi-index list, and then to perform online learning and tracking.

A. Scene Building

Before scene reconstruction, internal parameters of the camera need to be calibrated. Zhang [28] have proposed a calibration method which is widely applied in teaching and research region because it relatively easy to implement and has fewer restrictions by the environment.

In this paper, the SFM algorithm is used to reconstruct three-dimensional structure of the scene as in [29] and capture 10 key-frames at different view point for every scene, as shown in Fig. 3.

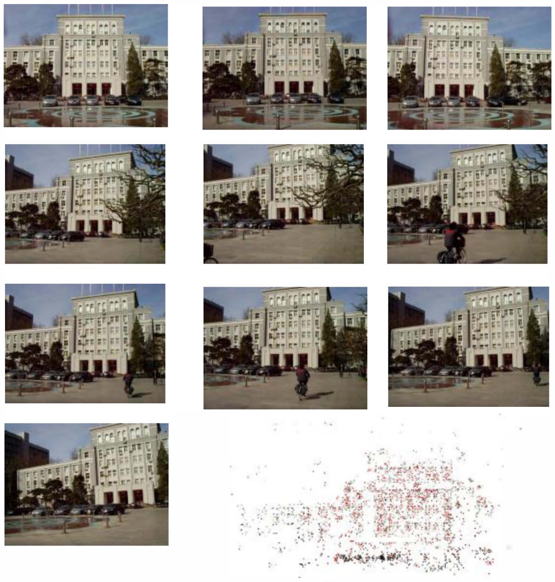


Figure 3. 3D scene reconstruction

B. 2D-3D Mapping Table Construction

Before feature tracking, the 2D-3D mapping table needs to be constructed in order that the 3D points of scene are fast found by the corresponding 2D features of key-frame on the same scene. And each 3D point may be corresponding to multiple 2D features on key-frames, so the mapping table only stores key-frame ID, 3D points and the corresponding 2D pointer in order to reduce memory in practical applications, such as Fig. 4.

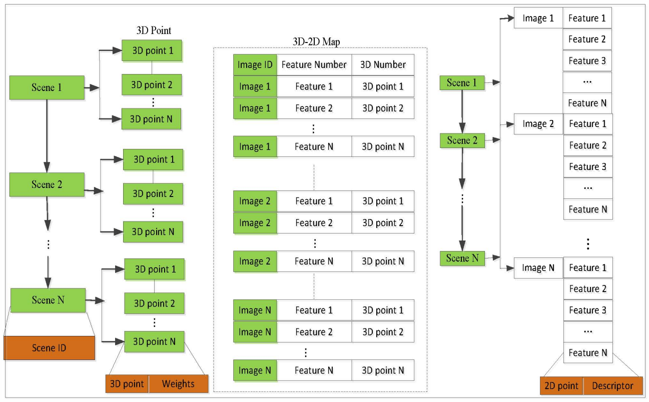


Figure 4. 2D-3D map table

C. Training 3D Point Weights

In order to calculate the correct transformation between scene model and the current image, the correspondence of 2D feature and matching 3D points need to be ensured precision. The larger correct matching pairs, the more accurate the transformation between scene model and the current image. The common methods are used to choose the optimal transformation matrix P by calculating the highest score or counting max number of matching points to match up the transformation matrix as in (1). If these methods are directly used which are unfeasible for smartphones because all related transformation matrix is calculated with expensive computational cost. In this paper, the 3D points are set an initial weights to use for calculating the highest score of transformation matrix and the cycle times are set like PROSAC approach when the scoring values do not increase as the termination condition which avoid computing scores for all transformation matrix. The detail implementation as equation (2), (3), (4), (5), $I=\{x_1,x_2\cdots,x_k\}$ is provided for the 2D point coordinate of the image, $D=\{d_1,d_2\cdots,d_k\}$ is the corresponding descriptors, $M=\{X_1,X_2\cdots,X_k\}$ is the 3D points, $C=\{(X_j,x_k,s_{jk})\mid X_j\in M,x_k\in I,s_{jk}\in R\}$ is the matching set for 2D points and 3D points, s_{jk} is their matching score, R is the set of scores.

$$P=\arg\max_{P'\in \mathrm{T}} S(M,I,P') \tag{1}$$

where M represents a scene model, I represents the scene image, P' is a transformation matrix between scene model and the image, T represents a collection of all the transformation matrices, S represents the score function, P represents the transformation matrix of the maximum value.

$$S(C,P)=\sum_{(X_j,x_k)\in C} E(\|x_k-P(X_j)\|_2<\tau) \tag{2}$$

$$S_w(C,P)=\sum_{(X_j,x_k)\in C} E(\|x_k-P(X_j)\|_2<\tau)=<w,L(C,P)> \tag{3}$$

$$L_j(C,P)=\begin{cases} d_k & \exists(X_j,x_k)\in C:\|x_k-P(X_j)\|_2<\tau \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$L(C,P)=[L_1(C,P),L_2(C,P)\cdots,L_J(C,P)]^T \ \mathrm{L}_j(1\le j\le J) \tag{5}$$

where $w=[w_1,w_2\cdots,w_J]^T$ represent a collection of weight for the 3D point and P is chosen as best projection matrix when his score is maximum.

VI. Online Learning and POSE Estimating

The ultimate goal of the online stage is to achieve registration for live scenes. On this stage, we firstly need to load scene structure, mapping table, key-frame descriptors into memory that have been completed on offline stage. If these have already in memory, we would not have loaded. The main processes are described in the following sections.

A. Scene Learning

In the tracking process, the algorithm of online scene learning is performed to update w weights to adapt to the changing environment. The weight w is subsequently updated to adapt to environmental changes. The weight updating algorithm [30] is applied to renew the w for each 3D point of the scene by computing a transform matrix between the current image and the 3D scene, With the transform matrix computed, we choose two transformation matrices with the highest score and the second highest, and use these scores to update the weights w, given as follows.

$$\begin{aligned} w_j^{t+1} \leftarrow (1-\eta_t\lambda)w_j^t + E(\max_{P\neq P_t}\{\Delta(P_t,P)-\delta S_w^i(P)\}>0)\eta_t\alpha_j^t \\ + E(u_j\in C_t^*)E(\max_{k'\neq k}\{1-<w_j,d_k-d_{k'}\}>0)\eta_t\nu\beta_j^t \end{aligned} \tag{6}$$

$$\alpha_j^t=L_j(C_t,P_t)-L_j(C_t,\hat{P}) \tag{7}$$

$$\beta_j^t=d_k-d_{\hat{k}} \tag{8}$$

$$P=\arg\max_{P\neq P_i}\{\Delta(P_t,P)-\delta S_w^i(P)\} \tag{9}$$

$$\hat{k}=\arg\max_{k'\neq k}\{1-<w_j,d_k-d_{k'}>\} \tag{10}$$

where j is 3D point number and t is the current frame number. w_j^t is the weight of the feature j for the current frame and w_j^{t+1} is the weight for the next frame. P_t represents the transformation matrix with the highest score for the current frame. $\eta_t=1/\lambda t$ is the step size and λ is balance factor which is used to weigh training accuracy and weight vector regularization.

B. Pose Estimating

With the obtained mapping relationship for the current image features m_t and the 3D points M_t of corresponding scene , the internal parameters K of the camera K is known, we compute a more robust and stable camera pose $T=[R\mid t]$ by using PNP [31] algorithm, as follows:

$$m_t=\lambda K[R\mid t]M_t \tag{11}$$

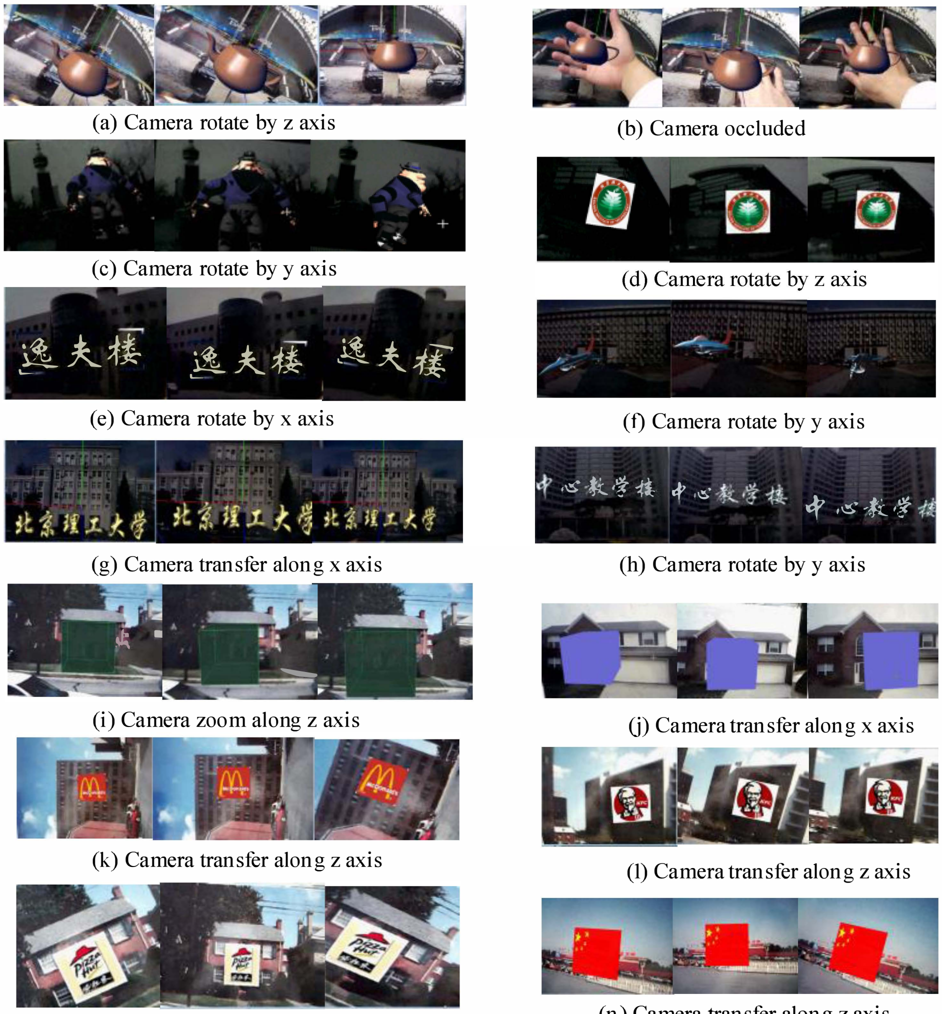
VII. Experiments and Results

The proposed algorithm is implemented on smartphone with1.7GHz processor and 2G RAM. The software is written in JAVA and NDK C++ on Android 4.1 operating system with MicroSD (32G).

Experimental dataset include five scenes in UKBENCH dataset and eight outdoor scenes shot on campuses altogether 13 scenes, and the resolution of video frames are set to $320*240$. BRISK [20] algorithms are used to detect feature points and generate descriptors respectively; and then the descriptors are trained and stored to construct multi-index searching engine.

A. *Registration Result*

The first experiment is taken to demonstrate the robustness of the proposed algorithm in different perspectives, different distances and different lighting conditions. As can be seen from Fig. 5(a)-(f), under different natural environment, even in the dark part of the scene illumination or camera rotate along different axes, the proposed algorithm can be accurately complete the registration in real time. Left part of Fig. 5 shows registration result of the campus scene, the right part shows registration result on scene of UKBENCH library. Fig. 5(a)-(g) gives the results with changes of the volumes, viewing angles, and illumination, respectively. The proposed algorithm is able to successfully identify all the sub-scenes and switch between them automatically to complete exact registration.



(a) Camera rotate by z axis

(b) Camera occluded

(c) Camera rotate by y axis

(d) Camera rotate by z axis

(e) Camera rotate by x axis

(f) Camera rotate by y axis

(g) Camera transfer along x axis

(h) Camera rotate by y axis

(i) Camera zoom along z axis

(j) Camera transfer along x axis

(k) Camera transfer along z axis

(l) Camera transfer along z axis

(m) Camera transfer along z axis and zoom

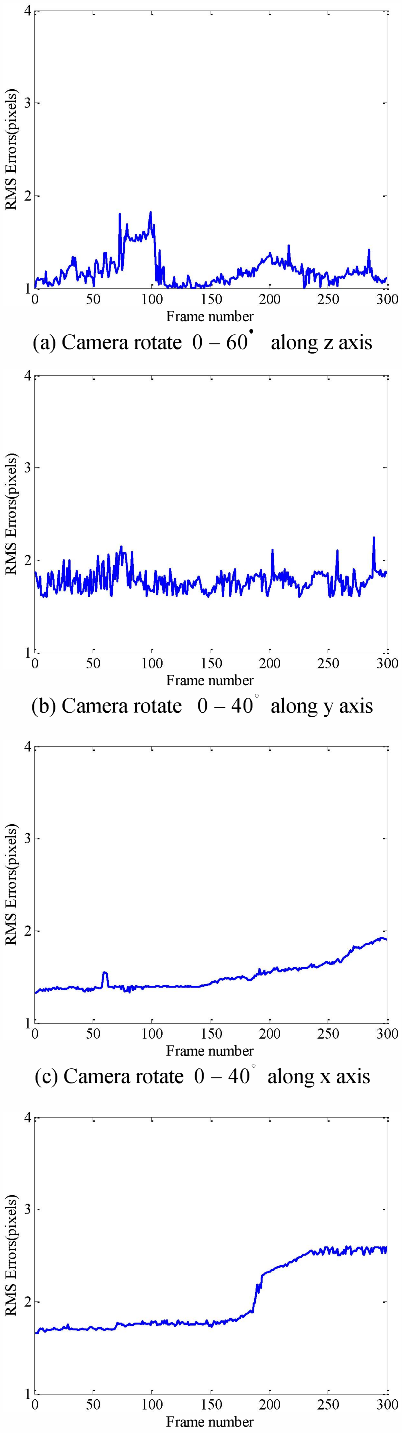
(n) Camera transfer along z axis

Figure 5. Tracking and register result

B. *Registration Accuracy*

We continuously track 300 frames and use RMS errors to test the accuracy of registration. We are interested in the RMS errors under the circumstance of changes in rotations and zooming ratio. The movement patterns of camera are simulated by rotating along X, Y, Z axis or moving from far to near the scene along the Z-axis to test the registration accuracy.

Fig. 6(a-c) gives the RMS errors of the proposed method when camera rotates along Z-axis from 0 to 60. The purpose is to simulate the case when users make large changes in view angles. Fig. 6(a) shows RMS errors are less than one pixel when the camera when the camera rotates 0-60 degrees along Z-axis. Fig. 6(b) shows RMS errors are close to 2.5 pixels when the camera 0-40 degrees along Y-axis. Fig. 6(c) shows RMS errors are close to 2.8 pixels when the camera rotates 0-40 along the X-axis. Fig. 6(d) shows RMS errors are close to 2 pixels when moving users move close to or far from the scene. All the above errors are below 3 pixels which demonstrate the accuracy of the proposed method.



(a) Camera rotate $0-60^{\circ}$ along z axis

(b) Camera rotate $0-40^{\circ}$ along y axis

(c) Camera rotate $0-40^{\circ}$ along x axis

(d) Camera move along the z axis

Figure 6. RMS error for the proposed algorithm

VIII. Conclusion

Register for augmented reality on smartphone is a challenging task. In this paper, several new approaches are proposed to improve the stability of the Register performance: a linear structured SVM classifier is used to perform scene learning online, which allows us to quickly adapt our model to a given environment; and the hybrid tracking strategy is implemented by combining both wide and narrow baseline techniques. The proposed approaches are very efficient and demonstrate excellent performance for large scene.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments. This research is supported by the Collaborative Innovation and Platform Environment Construction Major Project of Guangdong Province under Grant No. 2014B090901024.

References

[1] Duan, L.Y.; Guan, T.; Yang, B. Registration Combining Wide and Narrow Baseline Feature Tracking Techniques for Markerless AR Systems. Sensors, 2009, 9, 10097-10116.

[2] David, Y.; Efron, U. The Image Transceiver Device: Studies of Improved Physical Design. Sensors, 2008, 8, 4350-4364.

[3] Sam Hare, Amir Saffari, Philip H S. Torr. Efficient Online Structured Output Learning for Keypoint-Based Object Tracking. Proceedings of the 2012 Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Los Alamitos, CA, United States, Jun. 16-21, 2012, pp. 1894-1901.

[4] Chum O, Matas J. Matching with PROSAC-Progressive sample consensus. Proceedings of the 2005 Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, United States, Jun. 20-25, 2005, pp. 220-226.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, June 1981.

[6] P. H. S. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Computer Vision and Image Understanding, 78(1):138–156, Apr. 2000.

[7] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure from- motion point clouds to fast location recognition. Proceedings of the 2009 Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2599–2606, 2009.

[8] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. Proceedings of the European Conference on Computer Vision, 2010.

[9] D. Robertson and R. Cipolla. An image-based system for urban navigation. In BMVC, pp. 819–828, 2004.

[10] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. Proceedings of the 2007 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–7, 2007.

[11] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. PIXHAWK: A system for autonomous flight using onboard computer vision. Proceedings of the ICRA, pp. 2992–2997, 2011.

[12] E. Royer, M. Lhuillier, D. Michel, and J.-M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. IJCV, 74:237–260, Sep. 2007.

[13] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. Proceedings of the ICRA, 2011.

[14] Z. Dong, G. F. Zhang, J. Y. Jia, and H. J. Bao. Keyframe-based Realtime Camera Tracking. Proceedings of the 2009 International Conference on Computer Vision, 2009.

[15] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. Proceedings of the ISMAR, November 2007.

[16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. Proceedings of the TVCG, 16:355–368, 2010.

[17] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. Proceedings of the 2011 International Conference on Computer Vision (ICCV), 2011

[18] Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and I.-S. Kweon. Pushing the envelope of modern methods for bundle adjustment. Proceedings of the 2010 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1474–1481, 2010.

[19] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. In IJCV, 2008.

[20] Leutenegger S, Chli M, Siegwart R Y. BRISK: Binary robust invariant scalable keypoints[A].Proceedings of the 13th IEEE International Conference on Computer Vision. Piscataway: Institute of Electrical and Electronics Engineers Inc, 2011. 2548-2555.

[21] D. Ta, W. chao Chen, N. Gelfand, and K. Pulli. SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors. Proceedings of the 2009 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[22] T. Guan, C. Wang. Registration Based on Scene Recognition and Natural Features Tracking Techniques for Wide-area Augmented Reality Systems. IEEE Transactions on Multimedia, 2009, 11(8): 1393-1406.

[23] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. PAMI, 26(6):1052–1067, 2007.

[24] R. O. Castle, G. Klein, and D. W. Murray. Wide-area augmented reality using camera tracking and mapping in multiple regions. Computer Vision and Image Understanding, 115(6):854–867, 2011.

[25] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. Proceedings of 11th IEEE International Conference on Computer Vision (ICCV). Piscataway, NJ, USA: IEEE, 2007: pp. 2160-2167.

[26] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In ECCV, 2010.

[27] Alahi A, Ortiz R, Vandergheynst P. Freak: Fast retina keypoint. Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Washington:IEEE, 2012. 510-517.

[28] Zhang Z Y. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330-1334.

[29] Delabarre, B.; Marchand, E. Camera localization using mutual information-based multiplane tracking. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), pp. 1620-5, 2013.

[30] Sam Hare, Amir Saffari, Philip H S. Torr. Efficient Online Structured Output Learning for Keypoint-Based Object Tracking. Proceedings of the 2012 Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Los Alamitos, CA, United states, Jun. 16-21, 2012, pp. 1894-1901.

[31] Lu C, Hager G, Mjolsness E. Fast and Globally Convergent Pose Estimation from Video Images. Transaction on Pattern Analysis and Machine Intelligence (S0162-8828), 2002, 22(6): 610-622.