An Improved Augmented Reality Registration Method Based on Visual SLAM

Qing Hong Gao[1], Tao Ruan Wan[3], Wen Tang[2], Long Chen[2], and Kai Bing Zhang[1]

[1] College of Electronic and Information, Xian Polytechnic University, Xian, China
[2] Faculty of Science and Technology, Bournemouth University,Bournemouth, UK
[3] Faculty of Informatics, University of Bradford, Bradford, UK

Abstract. Markerless Augmented Reality registration using standard Homography matrix is instable and has low registration accuracy. In this paper, we present a new method to improve the augmented reality registration method based on the Visual Simultaneous Localization and Mapping (VSLAM). We improved the method implemented in ORB-SLAM in order to increase stability and accuracy of AR registration. VSLAM algorithm generate 3D scene maps in dynamic camera tracking process. Hence, for AR based on VSLAM utilizes the 3D map of the scene reconstruction to compute the location for virtual object augmentation. In this paper, a Maximum Consistency with Minimum Distance and Robust Z-score (MCMD_Z) algorithm is used to perform the planar detection of 3D maps, then the Singular Value Decomposition (SVD) and Lie group are used to calculate the rotation matrix that helps to solve the problem of virtual object orientation. Finally, the method integrates camera poses on the virtual object registration. We show experimental results to demonstrate the robustness and registration accuracy of the method for augmented reality applications.

Keywords: Augumented reality, SLAM algorithm, Virtual regristration and fusion, Point cloud

1 Introduction

Augmented Reality (AR) is the technology of mixing real scenes with virtual scenes, an emerging field of huge application potentials. The technology makes the use of computer-generated virtual information within the real world to enhance the human perception of the world. As defined by Azuma, it is an integration of virtual world and the real world with real-time interactions via three-dimensional registrations [2]. The recent rapid development of software as well as hardware technologies in virtual reality and computer vision, AR has a wider range of applications in medicine, military, entertainment and others [15] [4]. Virtual registration, however, remains a challenge in AR research. Initially Simultaneous Localization and Mapping (SLAM), as a probability algorithm, has been mainly used for positioning robots in unknown environments [14] [3]. More recently, researchers have started to utilize the accuracy and real-time performance of SLAM for virtual registration in AR. Davison et al. [5] [6] have used a monocular camera to achieve fast 3D modeling and positioning of cameras in unknown environments, which has presented many practical uses of the algorithm. Klein [9] applied a SLAM algorithm in the creation of three-dimensional point clouds, as well as Reitmayr [12] demonstrated the use of SLAM and sensor fusion techniques in an accurate virtual reality registration with markerless tracking.

The method of computing homography matrix in AR systems for the three-dimensional registration c̃iteUematsu2005 [7] [11]is simple and efficient. This algorithm requires the detection of four point coordinates of a plane in order to determine the translation and rotation of the camera relative to the world coordinate system. In spite of its simplicity and efficiency, since it is based on the 2D plane registration, the four points of detection algorithm is prone to the error of misplacement of the virtual object registration, resulting in virtual objects being unstable with distracting visual effects (e.g. flashing visual artifacts). Previous approaches [9] [12] have attempted to make the use of the three-dimensional map information generated by SLAM for this process. In this paper, we present a method of improvement to the registration and tracking process of virtual objects by using map information generated by VSLAM [12] technology. The three-dimensional information of a scene generated by VSLAM cannot be used directly, due to the interference points and the large error of point clouds. Therefore, a robust Maximum Consistency with Minimum Distance and Robust Zscore (MCMD_Z) [1] algorithm have been used to detect the 2D plane more accurately. Our improved MCMD_Z method computes plane point matrix by using the plane normal vector fund by Singular Value Decomposition (SVD). A method of lie group is then used to convert the normal vector into the rotation matrix to register the virtual object using the plane information. We use the precise positioning function of the VSLAM to change the camera poses to the rendering coordinate system under the camera perspective for the three-dimensional registration of the virtual object.

The main contribution of this paper is to develop a method that can effectively produce stable and high registration accuracy for virtual reality fusion.

2 AR System Overview

The AR system consists of two software modules: VSLAM module and registration module as shown in Fig. 1 for an overview of the system. Tracking in the VSLAM module is to locate the camera position through processing each image frame, and decide when to insert a new keyframe. Firstly, the feature matching is initialized with the previous frame and Bundle Adjustment BA [16] is used to optimize the camera poses. While the 3D map is initialized and the map is successfully created by the VSLAM module, the registration module is called. Once the point cloud of the scene is generated, the computation for the plane detection is started, and the center and the normal vector of the plane are calculated. The center of the plane is used to determine the exact position of the

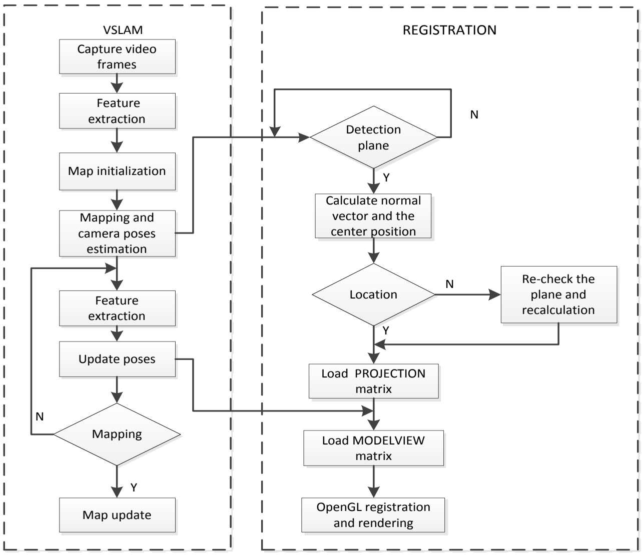


Fig. 1. System overview

virtual object and the normal vector is used to determine the orientation of the virtual object. Camera poses obtained by VSLAM are then transformed to the modelview matrix of OpenGL, which will be generated by the transformation of three-dimensional virtual objects to the center of the plane to achieve the virtual augmentation.

2.1 Tracking and 3D Map building

Our system is based on a visual simultaneous mapping and tracking approach by extracting and matching the Oriented Features From Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRIEF) (ORB) [13]feature points and compute two models: a homography matrix that is used to compute a planar scene and a fundamental matrix that is used to compute a non-planar scene. Each time two matrices are calculated, and a score ($M = H$ for the homography matrix, $M = F$ for the fundamental matrix) is also calculated as shown in equation 1. The score is used to determine which model is more suitable for the camera posture.

$$S_M = \sum_i \left(\rho_M\left(d^2_{cr,M}\left(x^i_c, x^i_r\right) + \rho_M\left(d^2_{rc,M}\left(x^i_c, x^i_r\right)\right)\right)\right) \tag{1}$$

$$\rho_M\left(d^2\right) = \begin{cases} \mathit{\Gamma} - d^2 \; if & d^2 < T_M \\ 0 \quad\;\; if & d^2 \geq T_M \end{cases}$$

where d_rc and d_cr is the measure of symmetric transfer errors [8],T_m is the outlier rejection threshold based on the χ^2, $\mathit{\Gamma}$ is equal to,T_m,x_c is the features of the current frame, x_r is the features of the reference frame. The BA is used to optimize camera poses, which gets a more accurate camera position as in following equation.

$$\{R, r\} = \arg\min_{R,t} \sum_{i \in \chi} \rho \left(\left\| x^i - \pi \left(RX^i + t \right) \right\|_{\Sigma}^2 \right) \tag{2}$$

where $R \in \mathcal{SO}^3$ is the rotation matrix, $t \in \mathbb{R}^3$ is the translation vector, $\S^i \in \mathbb{R}^3$ is a three-dimensional point in space,$x^i \in \mathbb{R}^2$ is the key point, ρ is the Huber cost function,Sigma item is the covariance matrix associated to the key point, π is the projection function.

After obtaining the accurate position estimation of the camera, the three-dimensional map point cloud is obtained by triangulating the key frame through the camera poses, and finally the local BA is used to optimize the map. A detailed description of the approach is given in [10].

3 Plane Detection and Calculation of the Normal Vector

The map created in ξ2.1 is composed of a sparse point cloud. Because of the error of the point cloud data with large number of abnormal values, MCMD_Z is used for plane detection. A MCMD_Z algorithm is used to fit the data according to a search model. The idea of this algorithm is to use Principal Component Analysis (PCA) for a reliable selection of the registration plane, using Robust Z-score to remove invalid points at once. This method not only effectively avoids the threshold setting, but also runs fast. The MCMD_Z algorithm detects the plane as follows:

Algorithm 1 The MCMD_Z algorithm

```
loop:
Randomly select any 3 points in the original point cloud, and calculate the corresponding initial plane and its normal vector;
Ranking the points according to the value of $d_i$ that are the distance of the map point cloud to the initial plane;
Select the threshold $t = 2\sigma$. $\sigma$ is the standard deviation of the point cloud to the initial plane model distance. When the $d_i > t$, the point is removed as an exception point, and vice versa as valid data retention, the number of valid data, $P_n um$;
if The set of points reach the minimum Eigen values then
    Stop the loop.
Calculate Robust Z-score uses the Equation 3. Zscore greater than 2 is considered an outlier and will be removed.
```

$$Rz_i = \frac{\left| od_i - \operatorname*{median}_j (od_i) \right|}{a \cdot \operatorname*{median}_i \left| od_i - \operatorname*{median}_j (od_i) \right|} \tag{3}$$

The detection of the plane determines the plane location, while providing a super-position of the location for a virtual object. Although the location of the virtual object is determined, virtual objects will not appear parallel to the plane but at a certain angle to the plane. In order to solve this problem, we need to calculate the normal vector of the plane and the rotation matrix.

The SVD of the matrix in the plane interior point is obtained, and the right singular vector corresponding to the minimum eigenvalue is the normal vector of the plane. Because there are two normal vectors in the plane, it is important that the normal vector direction is pointing outward. Specifically, the vector of the camera to the plane is found by the camera's posture. Through the vector and the relationship be-tween the plane vectors, we can then determine the direction of the normal vector. The rotation matrix is obtained from the known normal vector by Lie group using the following equation:

$$R_{3x3} = exp\left(\hat{w}\right) = I + sin\left(\|w\|\right) \cdot \frac{\hat{w}}{\|w\|} + \left(1 - cos\left(\|w\|\right)\right) \cdot \frac{\hat{w}\hat{w}}{\|w\|^2} \tag{4}$$

$$w = \frac{n_y \times n_p}{\|n_y \times n_p\|} \cdot arctan \frac{\|n_y \times n_p\|}{n_y \times n_p}$$

Where n_y is $y - axis$ unit vector, n_p is normal vector of the plane, w is a column vector, $\hat{w}$ is the anti-symmetric matrix of the vector w. Finally, the transformation matrix of OpenGL is composed of a translation vector and a rotation matrix. The rotation matrix is obtained and the translation vector is found to be the center of the plane.

3.1 Virtual Registration

The virtual object is finally registered in the real world, which must go through the transformation of the coordinate systems (from the world coordinate system to the camera coordinate system to the crop coordinate system, and to the screen coordinate system). The transformation sequences can be described by Equation 4 from left to right: the world coordinate system is transformed into the camera coordinate system by a rotation matrix $R_(33)$ and a translation matrix $T_(3x1)$. Those matrices are made up by the camera's position and the detected plane information. Then the camera coordinate system is then transformed into the screen coordinate system (u, v) by the focal length (f_x, f_y) and the principal point (d_x, d_y). These parameters are obtained by the camera calibration [25]. Finally, the virtual object is registered in the screen to the real world.

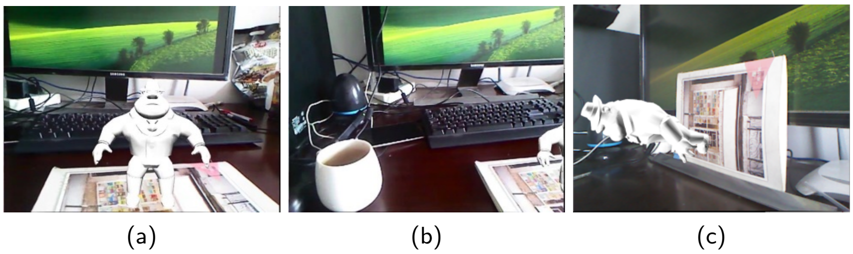


Fig. 2. AR tracking and registration (left to right (a)-(c))

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & d_x & 0 \\ 0 & f_y & d_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5}$$

4 Experiment and Evaluation

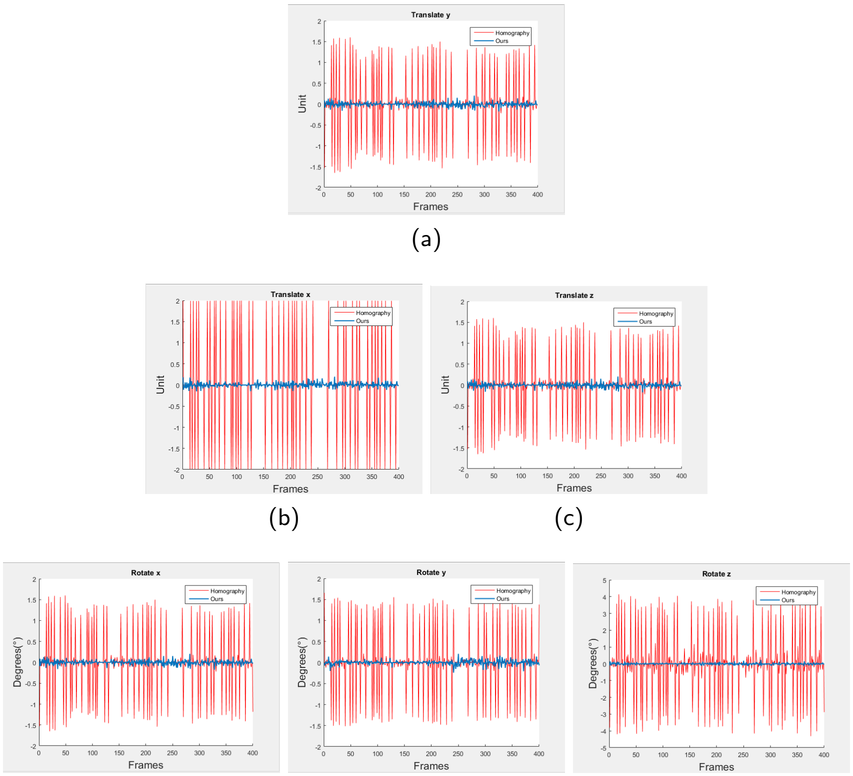
Our experiment is run under Ubuntu 14.04 system, CPU clocked at 2.3GHz, 8GB memory and graphics card for the NVIDIA GeForce GTX 960MB. The camera reso-lution is 640 by 480 pixels at 30 Hz. The experimental scene is indoors and the length of the image collection is 1857 frames. Fig 5 (a) - (b) show the indoor scene under the AR tracking and registration. We can see that the tracking and registration effect. Fig 2 (c) shows the correct virtual object orientation.

4.1 Plane Detection Analysis

Our method based on the CMCD_Z, which achieves better results than Random Sample Consensus(RANSAC). In contrast to these two algorithms, we use the Gaussian distribution to produce 1000 point with outlier percentages (10and20) using the same input parameters used previously. The inliers have means (15.0, 15.0, 10.0) and variances (10.0, 2.0, 0.5). The outliners have means (15.0, 15.0, 10.0) and variances (10.0, 2.0, 0.5). The program ran 1000 times. We compared Correct Identification Rate (CIR) and Swamping Rate (SR) [20]. The RANSAC sets iterations 50 times.

Table 1. Correct Identification Rate (CIR) ,Swamping Rate (SR) and Time

Methods	Outlier percentage				Time(s)
	10		20		
	CIR	SR	CIR	SR	
RANSAC	100	24.3	100	8	0.307849
MCMD_Z	100	6	100	0.5	0.116869



(a)

(b) (c)

(d) (e) (f)

Fig. 3. Registration error

4.2 Registration Error Analysis

A comparison method is used with fixed camera positions to evaluate the robustness of the method. The three-dimensional registration of the virtual object is carried out by using the described method and the standard homography matrix method. Six components of the three-dimensional registration results are analyzed. The difference between the transformation matrix of the current frame and the corresponding com-ponent of the transformation matrix of the previous frame is used as the basis for comparison. The results are shown in Fig.3, where Translate x, Translate y and Trans-late z are the errors of the translation components, respectively, and Rotate x, Rotate y, Rotate z are relative to the x, y, z axis of the rotation component error and where is obtained by subtracting the previous frame from the current frame. The result of the rotation component is obtained by dividing the respective components with the dot product of the corresponding coordinate axis, and the translation component is the result obtained by the normalization process.

In Fig 3, the red curves in the figures are the results of using only the homography matrix. The blue curves are the results of the new registration method used in this paper. As can be seen from the Fig 3, the use of the homography matrix method to register the virtual objects has produced large fluctuations of registration errors that are equivalent to virtual object registration instability. However, the new method tested on each rotation component has been kept the error in a small range below 0.5 degrees. The errors with Translate x, Translate y and Translate z are also small similar to the result of the rotation components.

Through the experimental results, it can be seen that the new method produces stable virtual registration and solve the flickering phenomenon in the virtual reality registration, hence, improves the stability of the AR system.

5 Conclusions and Future Work

This paper presents a stable and realistic tracking method based on three-dimensional map information generated by VSLAM method to track the registration of virtual objects to ensure the stability and real-time performance of registration. Our proposed method is faster and is able to achieve more accurate registration results. The experimental results show that the proposed method can effectively suppress the virtual object jitter, have a higher tracking accuracy with good tracking performance. The current three-dimensional map used in this paper is a sparse point cloud, which can only access limited space configuration information.

While this work has served to propose and prototype with experiments to show the effectiveness of the proposed approach, future work will consider the use of dense point cloud based on our proposed method.

Acknowledgement

This work is supported by Shanxi Province Science and Technology Department-Projects (2016JZ026 2016KW-043) and (2016GY-047)

References

1. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Computer Graphics and Applications 21(6), 34–47 (Nov 2001)
2. Azuma, R.T.: A survey of augmented reality. Presence: Teleoperators and Virtual Environments 6(4), 355–385 (aug 1997)
3. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping (SLAM): part ii. IEEE Robotics Automation Magazine 13(3), 108–117 (Sep 2006)
4. Bimber, O., R.R.: Spatial Augmented Reality Merging Real and Virtual Worlds. A K Peters Ltd Isbn 306 (2005)
5. Davison, A.J., Mayol, W.W., Murray, D.W.: Real-time localization and mapping with wearable active vision. In: Proc. Second IEEE and ACM Int. Symp. Mixed and Augmented Reality. pp. 18–27 (Oct 2003)

6. Davison, A.J., Mayol, W.W., Murray, D.W.: Real-time workspace localisation and mapping for wearable robot. In: Proc. Second IEEE and ACM Int. Symp. Mixed and Augmented Reality. pp. 315–316 (Oct 2003)
7. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 590–596 vol. 2 (Jun 2005)
8. Hartley, R., Z.A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
9. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Proc. 6th IEEE and ACM Int. Symp. Mixed and Augmented Reality. pp. 225–234 (Nov 2007)
10. Mur-Artal, R., Montiel, J.M.M., Tards, J.D.: Orb-SLAM: A versatile and accurate monocular SLAM system. IEEE Transactions on Robotics 31(5), 1147–1163 (Oct 2015)
11. Prince, S.J.D., Xu, K., Cheok, A.D.: Augmented reality camera tracking with homographies. IEEE Computer Graphics and Applications 22(6), 39–45 (Nov 2002)
12. Reitmayr, G., Eade, E., Drummond, T.W.: Semi-automatic annotations in unknown environments. In: Proc. 6th IEEE and ACM Int. Symp. Mixed and Augmented Reality. pp. 67–70 (Nov 2007)
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Proc. Int. Conf. Computer Vision. pp. 2564–2571 (Nov 2011)
14. Strasdat, H., Montiel, J., Davison, A.J.: Visual SLAM: Why filter? Image and Vision Computing 30(2), 65–77 (feb 2012)
15. Szalavri, Z., Gervautz, M.: The personal interaction panel a two-handed interface for augmented reality. Computer Graphics Forum 16(3), C335C346 (sep 1997), http://onlinelibrary.wiley.com/doi/10.1111/1467-8659.00137/abstract
16. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment — a modern synthesis. In: Vision Algorithms: Theory and Practice, pp. 298–372. Springer Nature (2000)