Increasing Camera Pose Estimation Accuracy Using Multiple Markers

Jong-Hyun Yoon, Jong-Seung Park, and Chungkyue Kim

Department of Computer Science & Engineering, University of Incheon,
177 Dohwa-dong, Nam-gu, Incheon, 402-749, Republic of Korea
{jhyoon, jong, ckkim}@incheon.ac.kr

Abstract. If the geometry of a marker is known and camera parameters are available, it is possible to recover a camera pose. The transformation between a camera and a marker is defined relative to the local coordinate system of the marker. This paper proposes a real-time camera tracking method using multiple markers while the camera is allowed to move freely in a 3D space. We utilize multiple markers to improve the accuracy of the pose estimation. We also present a coordinate registration algorithm to obtain a global optimal camera pose from local transformations of multiple markers. For the registration, a reference marker is automatically chosen among multiple markers and the global camera pose is computed using all local transforms weighted by marker detection confidence rates. Experimental results show that the proposed method provides more accurate camera poses than those from other methods.

1 Introduction

The methods of camera motion estimation are classified into a marker-based approach and a feature-based approach. The feature-based methods do not use specific markers and have an advantage that they do not need the preparation of markers. In other side, there are some shortcomings that the camera pose estimation is frequently unstable and inaccurate since it heavily depends on the accuracy of the feature extraction and tracking result. The feature-based methods require a relatively large number of feature points, hence they are slower than the marker-based methods. Our approach falls into the marker-based approach for the reason that real-time interactive augmented reality applications require a fast and stable camera tracking.

Among the various kinds of markers, the most commonly used types are planar markers. Different planar patterns are attached to the markers to distinguish the multiple markers. Once the camera pose is recovered, the virtual objects can be easily synthesized to the captured image frame.

Many previous studies have focused on reducing the recovered camera pose errors[1][2]. Ababsa[3] proposed a method based on tracking calibrated 2D fiducials in a known 3D environment. When a camera is calibrated, the distortion factor is available and it can be used to increase the camera pose accuracy. Efficient calibration methods are readily available[4]. Camera motion recovery without a camera calibration has also been presented[5]. Besides the consideration

Z. Pan et al. (Eds.): ICAT 2006, LNCS 4282, pp. 239–248, 2006.

of the camera distortion factor, camera pose errors are affected by environment variables such as occlusion and lighting. Skrypnyk introduced a method to reduce such errors[6]. As well as the user's viewpoint, the positions of light sources can be considered for a seamless registration. Kanbara used a marker and a mirror ball to estimate the relationship between the real and virtual worlds and the positions of light sources[7].

In this paper, we propose a fast and reliable camera pose estimation method using multiple markers. Our approach is different from previous works in the sense that our method is adaptive to the number of detected markers. If there is at least one accurate detected marker in a frame, a reliable camera pose estimation is guaranteed even when there also exist erroneous markers in the frame. In the next section, we describe the related works about fast and stable camera tracking using markers. Section 3 presents our proposed method based on multiple markers. Section 4 describes experimental results of our method for real video frames. Finally, we conclude with an overall evaluation and summary in Section 5.

2 Marker-Based Camera Tracking

For the realization of practical augmented reality systems, marker-based camera pose estimation methods have been actively investigated. Using a single marker, the transformation between the marker and the camera can be calculated and the camera motion can also be estimated by the marker transform. However, the single marker tracking has several limitations. It cannot be applied to the camera motion recovery if the marker is not visible, which is commonly occurred when a camera is moved freely in a wide space. A solution for such a problem is using multiple markers to cover a wide area.

The purpose of our research is to develop an accurate algorithm to track the camera motion using multiple markers whose positions are known in the real environment[8]. Uematsu proposed a multi-marker registration method in a projective space[9]. In the algorithm, a projective matrix is computed for each plane to relate it to the real world. The projection matrices are computed from the coordinate systems that are defined by the planes without geometrical relationship to each other. Then, all the projection matrices are merged into a single transformation matrix via the projective 3D space. Zauner proposed a mixed reality application using multiple markers and applied it to a mobile device[10]. The method assumed that the geometry between markers are known in advance. To detect multiple markers, the algorithm uses a simple reference marker that has the highest confidence value among detected markers. To improve the accuracy and stability, the average filter and the linear regression filter methods are used. The average filter reduces the errors of estimated values for input image sequences. The linear regression filter considers the history of the tracking results. Via the linear regression filter, the estimated marker transformation becomes more stable.

2.1 Camera Tracking Using a Single Marker

ARToolkit and ARTag are two most representative methods that detect a planar marker and calculate the relation between the marker and the camera. ARToolkit is the representative marker tracking system and it is the widely used method for a fiducial marker system in AR. There are some steps to calculate the relationship between the marker and the camera. First, the input image is converted to a binary image by the grey scale thresholding. Then, four outline segments of marker's square are extracted from the binary image.

The square region for each detected marker is found by the binary image analysis. For each marker, the pattern inside the square is captured and compared with pre-defined pattern templates to identify the type of the detected marker. If the captured pattern is matched to one of the predefined templates, the corresponding pattern number is assigned to the detected marker. After the marker detection, the 3D position and the orientation of the marker is computed[11]. The rotation matrix is calculated using the line segments and the translation matrix is computed using the four corner points of the marker that are the intersections of the line segments. The transformation matrix $\mathbf{T}_m^c$ from the marker coordinate system to the camera coordinate system consists of the 3×3 rotation matrix $\mathbf{R}$ and the 3×1 translation vector $\mathbf{t}$. The estimation error of $\mathbf{T}_m^c$ can be reduced by the iterative optimization process minimizing the differences between the detected corner points and the projections of the marker vertices.

The ARTag is an extension of the ARToolKit. Its marker is a combination of special planar patterns[12]. ARTag finds four outlines of the rectangular marker using an edge detection method. The marker detection in ARTag is more robust in illumination than that in ARToolKit since ARTag uses edge detection instead of binary thresholding. Each detected marker is identified using its corners and inner grid cells. An ARTag marker is a 10×10 grid rectangle and the inner pattern is a 6×6 grid rectangle of black-and-white cells. The inner pattern is sampled to determine whether each grid cell corresponds to zero or one. Then, we could obtain the marker identification number.

2.2 Camera Tracking Using Multiple Markers

This section explains a typical multi-marker registration method based on the works by Uematsu[9] and Zauner[10]. The registration method by Uematsu estimates extrinsic parameters of the camera and aligns virtual objects according to the parameters[9]. The method uses the relationship between the projective space and the reference image. A projection matrix $\mathbf{P}_i$ that relates the ith marker coordinate system to the input image is computed by the planar homography for each marker. The subscript i is the marker identification number. The transformation matrix $\mathbf{T}_i^{WP}$ is a 4×4 symmetric matrix and it relates the ith marker's coordinate system with the projective space. A point in the world space $\mathbf{X}_W$ and the corresponding point in the projective space $\mathbf{X}_P$ are related by $\mathbf{X}_P \simeq \mathbf{T}_i^{WP}\mathbf{X}_W$. Using the projection matrix $\mathbf{P}_i$ and the transformation matrix $\mathbf{T}_i^{WP}$, we could compute another transformation $\mathbf{T}_i^{PI}$ that relates the projective space to each image by $\mathbf{T}_i^{PI} = \mathbf{P}_i \left(\mathbf{T}_i^{WP}\right)^{-1}$. After obtaining all the transformation matrices, all $\mathbf{T}_i^{PI}$ are merged into the transformation matrix $\mathbf{T}^{PI}$. The precalculated transform matrices relate between the projective space and the input images. So, it is possible to merge $\mathbf{T}_i^{WP}$. After the merging process, virtual objects are aligned and posed by $\mathbf{T}^{PI}$.

Zauner proposed another multi-marker registration technique for mixed reality in mobile devices[10]. The method consists of two steps: finding the reference marker and applying the average filter and the linear regression filter. First, a marker is chosen as the reference marker among the detected multiple markers. The center of the reference marker is used as the reference point. Next, the position and the orientation of the camera relative to the marker are computed. The average filter is applied to the positions and the orientations of all the detected markers in order to get the average position vector and the average orientation quaternion. Since the result using only the average filter is unstable, the history of the previous positions and orientations of the camera is considered. The linear regression provides two straight lines representing the correlation between the past time and the tracked placement. The linear regression minimizes the average error of the camera pose estimation.

3 Increasing Camera Pose Accuracy Using Multiple Markers

The transformation between the camera and the marker could be recovered from the marker detection. The transformation consists of the rotation and the translation of the camera, which is represented by:

$$[X_c, Y_c, Z_c, 1_c]^T = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} [X_m, Y_m, Z_m, 1_m]^T \ . \tag{1}$$

The equation (1) represents the camera transformation based on the local coordinate system of the marker. The camera position corresponds to the translation vector $\mathbf{t} = [t_x \ t_y \ t_z]^T$ and camera orientation corresponds to the rotation matrix $\mathbf{R}$. We propose an accurate camera pose estimation method that estimates $\mathbf{R}$ and $\mathbf{t}$ from multi-markers. Our method is fast and reliable in most cases. The overall steps of our method are described as following pseudo code.

Step 1 (Initialization):
1. Detect all visible markers, m_i $(i = 1, \ldots, N)$, where N is the number of detected markers. The four corner points of the ith marker are denoted by $\mathbf{c}_{i,j}$ $(j = 1, \ldots, 4)$.
2. Calculate the transformation matrix $\mathbf{T}_i$ for the ith detected marker.

Step 2 (Choosing a reference marker):
1. Compute the marker center $\overline{\mathbf{c}}_i$ using corner points by $\overline{\mathbf{c}}_i = 0.25 \sum_{j=1}^{4} \mathbf{c}_{i,j}$.
2. Calculate the distance d_i between $\overline{c}_i$ and the image center $\overline{\mathbf{o}}$ by $d_i = \parallel \overline{\mathbf{c}}_i - \overline{\mathbf{o}} \parallel$.

3. Select the reference marker m_k by finding the marker of the minimum $\arg\min_{1 \le i \le N} d_i$.

Step 3 (Setting the reference marker):

1. Align the local coordinate system L_i to the reference coordinate system L_k by applying $\mathbf{T}_{i,k}$ where k is the reference marker index and $\mathbf{T}_{i,k}$ is the predefined transformation matrix from the multi-marker setting in the real world.
2. Change the camera transformation matrix in regard to the local coordinate system $\mathbf{T}_i$ to the transformation in regard to the reference coordinate system $\mathbf{T}'_i$ by $\mathbf{T}'_i = \mathbf{T}_i(\mathbf{T}_{i,k})^{-1}$.

Step 4 (Computing error rates):

1. Calculate error rates ρ_i of all detected markers by $\rho_i = w_i / \sum_{i=1}^{N} w_i$ where w_i is the weight of the ith detected marker.

Step 5 (Computing camera pose):

1. Compute the weighted sum of all estimated transformations using ρ_i by $\mathbf{T}'' = \sum_{i=1}^{N}(\rho_i \mathbf{T}'_i)$.
2. Calculate the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ from $\mathbf{T}''$.

3.1 Calculating the Camera Transformation

From each input image, we first find out all visible markers and calculate the transformations for the detected markers. Accurate corner extraction of the marker directly affects the reliability of the recovered marker transformations. Next, we select the reference marker among detected markers. We choose the closest marker to the image center as the reference marker. In the previous step, we found out all visible square areas and the corner points of detected markers. Using the corner points, we calculate the coordinates of the marker center in the image coordinate system. Then, we calculate the distance between the marker center and the image center. We choose the marker that has the minimum distance as the reference marker. The reference marker provides the best confident camera transformation among the markers since the reference marker has the least projective distortion.

Fig. 1(a) shows the relationship of the local coordinate systems of the markers and the reference coordinate system. We have to calculate the camera pose in terms of the reference coordinate system. Each of the detected markers has its local coordinate system. We need to register the local coordinate systems of the detected markers into the coordinate system of the reference marker. When the local coordinate system L_i is aligned to the reference coordinate system L_k, the aligned local transformation $\mathbf{T}'_i$ of $\mathbf{T}_i$ should be equal to the reference transformation $\mathbf{T}_k$:

$$\mathbf{T}'_i = \mathbf{T}_i(\mathbf{T}_{i,k})^{-1} = \mathbf{T}_i \mathbf{T}_i^{-1} \mathbf{T}_k = \mathbf{T}_k \ . \tag{2}$$

Due to the estimation error, the transformation $\mathbf{T}_i$ in Eq. (2) is not accurate and the equality does not be satisfied.

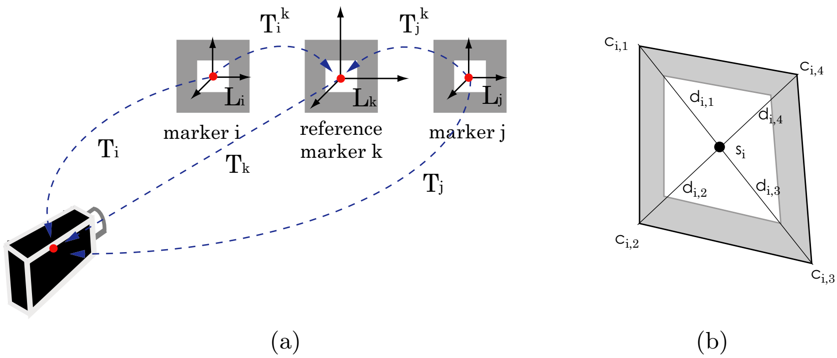

Fig. 1. (a) multiple markers and their geometric relationships, (b) the error rate of each detected marker

We suppose the geometric relations of the multiple markers in real world are known. The transformations $\mathbf{T}_{i,k}$ can be computed from the known geometric relation between the ith marker and the kth marker. We denote $\mathbf{p}_i$ as the ith marker center points. Then, the transformation matrix $\mathbf{T}_{i,k}$ is as follows:

$$\mathbf{T}_{i,k} = \begin{bmatrix} \mathbf{I} & \mathbf{p}_k - \mathbf{p}_i \\ \mathbf{0} & 1 \end{bmatrix} . \tag{3}$$

$\mathbf{T}_i$ is aligned as $\mathbf{T}_i^{'} = \mathbf{T}_i(\mathbf{T}_{i,k})^{-1}$. After the registration, each local coordinate system corresponding to the detected marker is aligned to the reference marker orientation by $\mathbf{T}_i' = \mathbf{T}_i(\mathbf{T}_{i,k})^{-1}$ $(i = 1, \ldots, N)$.

3.2 Camera Motion Reconstruction from the Marker Transformation

The accuracy of the camera pose can be improved using the aligned transformations. For each detected marker, we define the error rate to indicate the confidence rate of the marker. The error rate is related with the camera viewing direction to the target marker. We define the error rate for a marker using the four corner points and the number of pixels inside the marker rectangle. Let $c_{i,1}$, $c_{i,2}$, $c_{i,3}$ and $c_{i,4}$ be the marker corner points as shown in Fig. 1(b). Two diagonal lines intersect at a point s_i. For each marker, we compute v_i using the following equation:

$$v_i = \frac{1}{4}\sum_{n=1}^{4}(d_{i,n} - \bar{d}_i)^2 \text{ where } \bar{d}_i = \frac{1}{4}\sum_{n=1}^{4} d_i^n \text{ and } d_{i,k} = \| s_i - c_{i,k} \| . \tag{4}$$

The projective distortion should be considered for the error rate of the marker. The variance of $d_{i,k}$, the distance from the corner $c_{i,k}$ to the intersection s_i as shown in Fig. 1(b), is proportional to the projective distortion. In the square case,

Fig. 2. Experimental setup for the linear camera movement

Fig. 3. Experimental setup for the circular camera movement

all $d_{i,k}$ should have the same distance from the intersection s_i. We calculate the variation v_i of the lengths $d_{i,k}$ $(k = 1, \ldots, 4)$. If v_i goes to zero, the camera orientation approaches to the normal direction of the marker. The image area of the marker should also be considered for the error rate. The area a_i for the ith marker is measured as the number of pixels inside the detected marker.

Using v_i and a_i, we compute the error rate ρ_i as follows:

$$\rho_i = \frac{w_i}{\sum_{i=1}^{N} w_i}, \text{ where } w_i = \frac{a_i{}'}{v_i'},\ a_i{}' = \frac{p_i}{\sum_{i=1}^{N} p_i} \text{ and } v_i' = \frac{v_i}{\sum_{i=1}^{N} v_i}\ . \tag{5}$$

The weight of the ith marker w_i which is used to compute the error rate ρ is calculated by the pixel area a_i and the degree of variation v_i. The error rate ρ is between zero and one and, if ρ is close to one, the camera orientation closes to the normal direction of the marker. The error rate ρ is used in order to improve the reliability of the camera motion. Multi-markers give more useful information than single marker. The transformations from multi-markers are integrated as follows into the improved camera pose $\mathbf{T}''$:

$$\mathbf{T}'' = \sum_{i=1}^{N} (\rho_i \mathbf{T}_i') \text{ where } \sum_{i=1}^{N} \rho_i = 1\ . \tag{6}$$

All the camera motions corresponding to the detected markers are integrated in the proportion of the error rates. The estimated camera pose $\mathbf{T}''$ is more reliable than the pose from either the single marker methods or the common multi-marker methods.

4 Experimental Results

We have implemented and tested the proposed camera pose tracking method on a number of real data sets. To measure the accuracy of the proposed method, we estimated the camera pose for each image acquired from a moving camera. Two types of camera movements, a linear movement and a circular movement, were tested and their camera trajectories were recovered. Fig. 2 and Fig. 3 show the images captured while the camera is moving along the two predefined trajectories. In Fig. 2, the camera was moved in a parallel direction with respect to the marker direction. The camera position is 70cm far from the markers and

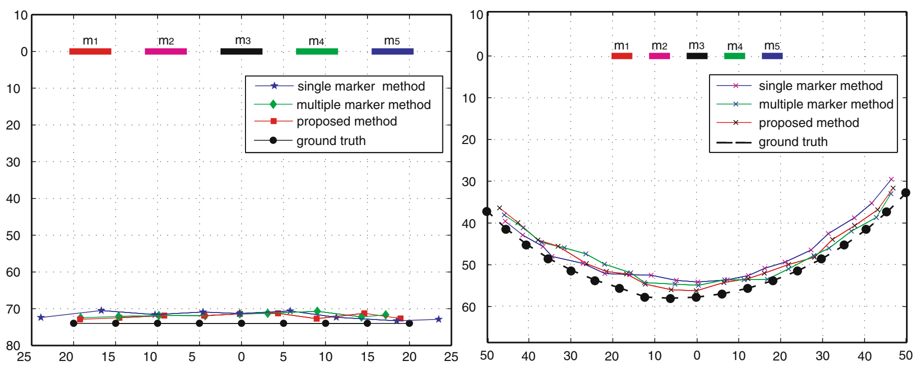

Fig. 4. Comparison of the recovered trajectories

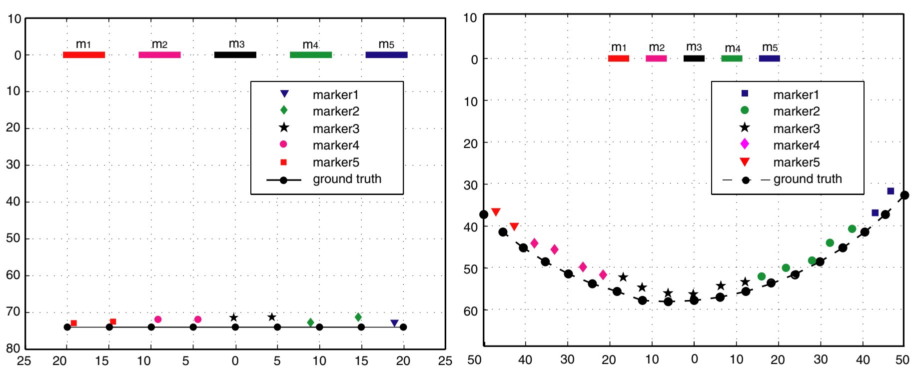

Fig. 5. The selected reference markers

the camera orientation is parallel to the marker normal. In Fig. 3, the camera is moved along a circular path around the markers. Each camera angle for two consecutive images is differed by 5 degrees. The radius of the circular path is 50cm. For the recovery, we use both the ARToolKit markers and the ARTag markers. All the markers have the same size and each marker is a regular square 5cm on a side. Horizontally aligned five markers are used for the multiple marker estimation.

The ground truths of the two camera movement trajectories were measured in advance for the comparison of the recovery accuracy. Fig. 4 shows the recovered trajectories using several different methods together with the ground truth trajectory. The trajectory recovered from our method is compared with the classical single and multiple marker based methods. The left figure is the recovered trajectories for 9 captured frames from the linear camera movement. The right figure is the recovered trajectories for 19 captured frames from the circular camera movement. The comparison of trajectories shows that our method provides the best accurate trajectory toward the ground truth trajectory.

At each frame, a reference marker is selected among the detected markers to align makers into the reference coordinate system. Fig. 5 shows the selected reference markers at the camera positions of the recovered trajectory. The left figure is for the linear camera movement and the right figure is for the circular camera movement.

We measured the errors and compared them with the ground truth for both camera movements. For the linear movement, the single marker method has 2cm error on average, the multiple marker method has 1cm error on average, and the proposed method has 0.5cm error on average. For the circular movement, the single marker method has 3.3cm error on average, the multiple marker method has 2.7cm error on average, and the proposed method has 0.2cm error on average. In all cases, the proposed method shows the best results.

5 Conclusion

This paper has presented a novel method of camera pose estimation using known multiple markers. Our approach is different from previous works in the sense that the proposed method is adaptive to the number of detected markers. If there is at least one accurate detected marker in a frame, reliable camera pose estimation is guaranteed even when there are also erroneous markers in the frame.

We detect all markers shown in each image frame and choose a reference marker among the detected markers. Then, we calculate the transformation from the local coordinate system of each marker to the reference coordinate system of the reference marker. A global optimal camera pose is estimated from the weighted sum of the transformations of multiple markers. Experimental results verified that our method provides more accurate camera poses than those from classical single marker-based methods or multiple marker-based methods.

Our future research includes enhancing the robustness of the camera motion estimation method to provide reliable camera poses even when the relative positions between markers are unknown.

Acknowledgements. This work was supported in part by grant No. RTI05-03-01 from the Regional Technology Innovation Program of the Ministry of Commerce, Industry and Energy(MOCIE) and in part by the Brain Korea 21 Project in 2006.

References

1. Abdullah, J., Martinez, K.: Camera self-calibration for the artoolkit. In: Proceedings of First International Augmented Reality Toolkit Workshop. (2002) 84–88
2. Malbezin, P., Piekarski, W., Thomas, B.: Measuring artoolkit accuracy in long distance tracking experiments. In: Proceedings of First International Augmented Reality Toolkit Workshop. (2002)
3. eddine Ababsa, F., Mallem, M.: Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. In: VRCAI '04, New York, NY, USA, ACM Press (2004) 431–435
4. Bianchi, G., Wengert, C., Harders, M., Cattin, P., Szekely, G.: Camera-marker alignment framework and comparison with hand-eye calibration for augmented reality applications. In: ISMAR '05, Washington, DC, USA, IEEE Computer Society (2005) 188–189
5. Micilotta, A.S., Ong, E.J., Bowden, R.: Real-time upper body 3d pose estimation from a single uncalibrated camera. In: Eurographics. (2005) 41–44
6. Skrypnyk, I., Lowe, D.G.: Scene modelling, recognition and tracking with invariant image features. In: ISMAR 2004. (2004) 110–119
7. Kanbara, M., Yokoya, N.: Real-time estimation of light source environment for photorealistic augmented reality. In: ICPR. Volume 02., Los Alamitos, CA, USA, IEEE Computer Society (2004) 911–914
8. Ledermann, F., Reitmayr, G., Schmalstieg, D.: Dynamically shared optical tracking. In: ART'02, IEEE Computer Society (2002)
9. Uematsu, Y., Saito, H.: Ar registration by merging multiple planar markers at arbitrary positions and poses via projective space. In: ICAT2005. (2005) 48–55
10. Zauner, J., Haller, M.: Authoring of mixed reality applications including multi-marker calibration for mobile devices. In: EGVE 2004, Grenoble, France (2004) 87–90
11. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: IWAR '99, Washington, DC, USA, IEEE Computer Society (1999) 85
12. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: CVPR '05, Washington, DC, USA, IEEE Computer Society (2005) 590–596