Registration Based on Online Estimation of Trifocal Tensors Using Point and Line Correspondences

Tao Guan, Lijun Li, and Cheng Wang

Digital Engineering and Simulation Centre,
Huazhong University of Science and Technology,
No.1037 Luoyu Road,430074 Wuhan, China
qd_gt@126.com, lilijun@hust.edu.cn, wangcheng@hust.edu.cn

Abstract. This paper illustrates a novel registration method based on robust estimation of trifocal tensors using point and line correspondences synthetically. The proposed method distinguishes itself in following ways: Firstly, besides feature points, line segments are also used to calculate the needed trifocal tensors. The registration process can still be achieved even under poorly textured scenes. Secondly, to estimate trifocal tensors precisely, we use PROSAC instead of RANSAC algorithm to remove outliers. While improving the accuracy of our system, PROSAC also reduces the computation complexity to a large degree.

Keywords: Augmented Reality, Registration, Trifocal Tensors, Fundamental Matrix, PROSAC.

1 Introduction

Registration is one of the most pivotal problems currently limiting AR applications. It means that the virtual scenes generated by computers must be aligned with the real world seamlessly. Recently, many promising natural feature based registration approaches have been reported. However, most of them use local, viewpoint invariant regions [1],[2],[3]. These regions are extracted independently from reference images, characterized by a viewpoint invariant descriptor, and finally matched with online frames. Those methods are robust to large viewpoint and scale changes. However, the number of produced matches depends on the amount of visible distinctive texture contained in scenes. Images of poorly textured scenes provide only a few matches, and sometimes none at all. Despite lacking texture, these scenes often contain line segments which can be used as additional features. Line segments convey an important amount of geometrical information about the constitution of the scene, whereas regions capture the details of its local appearance. Using both types of features allows to exploiting their complementary information, making registration more reliable and generally applicable.

R. Shumaker (Ed.): Virtual Reality, HCII 2007, LNCS 4563, pp. 42–51, 2007.

In this paper, we propose a novel registration method based on online estimation of trifocal tensors using point and line correspondences, which distinguishes itself in following ways:

Firstly, we relax the restriction that the four specified points used to establish world coordinate system must form an approximate square. The only limitation of our approach is that these four coplanar points should not be collinear.

Secondly, besides feature points, line segments are also used to calculate the needed trifocal tensors. Using both types of features, the registration process can still be achieved even under poorly textured scenes.

Thirdly, benefiting from the tensor of previous frame and the normalized cross-correlation (NNC) operation, we propose a method to match features between current and reference images directly. By this method, not only do we fulfill the task of establishing points and lines correspondences respectively, but also constitute a NNC based criterion to evaluate the quality of points and lines matches in a uniform framework. In deed, this is an important precondition of the PROSAC algorithm we used to estimate the needed tensor.

Finally, to estimate trifocal tensors precisely, we use PROSAC to remove outliers. The matches with higher quality (normalized cross-correlation score) are tested prior to the others, by which the algorithm can arrive at termination criterion and stop sampling earlier. While improving the accuracy, the proposed method can also reduce sample times to a large degree.

2 Overview of Proposed Approach

Our method can be divided into two stages, namely, offline initialization and online registration. In initialization stage, a high quality set of point and line correspondences are obtained for two spatially separated reference images of the scenes in which we want to augment. The world coordinate system is established based on projective reconstruction technique and the four coplanar points specified by user in the two reference images respectively. In online stage, feature matches of the reference images are tracked in current frame benefiting from the tensor of previous frame. With the feature triplets, the trifocal tensor is calculated using PROSAC based algebraic minimization algorithm. Using this tensor, the four specified coplanar points are transferred into the living image, and the homographies between the world plane and the moving frame is recovered via the correspondence of these four points. Then the registration matrix is recovered using above homographies and the virtual objects is rendered on the real scenes using the graphics pipeline techniques, e.g., OpenGL.

3 Establishing World Coordinate System

Before establishing the world coordinate system, we should first recover the epipolar geometry between the two reference images. For a point $\mathbf{x}_i$ in the first reference image, its correspondence in the second image, $\mathbf{x}_i^{'}$, must lie on the epipolar line in the second image, which is known as the epipolar constraint. Algebraically, in order for $\mathbf{x}_i$ and $\mathbf{x}_i^{'}$ to be matched, the following equation must be satisfied [4]:

$$\mathbf{x}_i^{'} \mathrm{F} \mathbf{x}_i = 0 \quad i = 1, \dots, n \tag{1}$$

Where F, known as the fundamental matrix, is a 3×3 matrix of rank 2, defined up to a scale factor, which is also called the essential matrix in the case of two calibrated images. Let F be the fundamental matrix between two reference images. It can be factored as a product of an antisymmetric matrix $[e^{'}]_{\times}$ and a matrix T, i.e., $\mathrm{F} = [e^{'}]_{\times} \mathrm{T}$. In fact, *e'* is the epipole in the second image. Then, two projective camera matrices can be represented as follows:

$$\mathrm{P} = [\mathrm{I} \mid 0], \mathrm{P'} = [\mathrm{T} \mid e^{'}] \tag{2}$$

Given a pair of matched points in two reference images: $(\mathbf{x}_i, \mathbf{x}_i^{'})$, let $\mathbf{X}_i$ be the corresponding 3D point of real world. The following two equations can be obtained:

$$\mathbf{x}_i = \lambda \mathrm{P} \mathbf{X}_i \tag{3}$$

$$\mathbf{x}_i^{'} = \lambda^{'} \mathrm{P}^{'} \mathbf{X}_i \tag{4}$$

Where λ and $\lambda^{'}$ are two arbitrary scalars. Let p_i and $\mathrm{p}_i^{'}$ be the vectors corresponding to the i-th row of P and $\mathrm{P}^{'}$ respectively. The above two scalars can be computed as follows:

$$\lambda = 1 / \mathrm{p}_3^T \mathbf{X}_i \tag{5}$$

$$\lambda^{'} = 1 / \mathrm{p}_3^{'T} \mathbf{X}_i \tag{6}$$

With equations (3) to (6), we can reconstruct $\mathbf{X}_i$ from its image matches $(\mathbf{x}_i, \mathbf{x}_i^{'})$ using the linear least square technique.

The next step is to specify four coplanar points $\mathbf{x}_i = (x_i, y_i, 1)^{\mathrm{T}}$, $(i = 1, \dots, 4)$ in each of the two reference images, respectively, to establish the world coordinate system. Previous work [4][5] has the restriction that these four points should form an approximate square, and the origin of the world coordinate system is set to the center of the square defined by the four known points, the X-axes and Y-axes are the direction of two different parallel sides respectively. However, this method still needs some special square planar structures to help to specify the planar points accurately in the control images.

From the study of [3], we know that the registration matrix can be computed from the homographies between current frame and the world plane, and to get this homographies, we need only four coplanar but non-collinear points of the world plane and their projections on the current image. Motivated by this property, we propose a new method to define world coordinate system without the needs of special square structures, and the only limitation of our approach is that the four specified coplanar points should not be collinear. Let $\mathbf{X}_i = (X_i, Y_i, Z_i, 1)^{\mathrm{T}}, (i = 1, \dots, 4)$ be the projective 3D coordinates of the four specified points, the origin of the world coordinate system will be the $\mathbf{X}_1$, the X-axes will be the direction of the vector $\overrightarrow{\mathbf{X}_1 \mathbf{X}_2}$, the Z-axes will be the vertical direction of the plane defined by above four 3D points, and the Y-axes will be the cross product of Z-axes and X-axes. To improve accuracy, when one point has been specified in a reference image, its epipolar line in another image is drawn to limit the searching area of the correspondence in this image, because the correspondence is limited on this epipolar line according to the property of epipolar geometry.

4 Feature Tracking

We propose a normalized cross-correlation based method to match the features between current and reference images directly. We assume that the tensor of previous frame has been calculated accurately. With this tensor, the corresponding feature points detected from two reference images are transferred onto the live image, the correspondence is identified by searching in a small area surrounding the transferred point for a point that correlate well with one of the two reference matched points.

To match a line segment, following steps must be carried out. Firstly, the midpoint of the shorter one of the corresponding lines in reference images is extracted, and the correspondence of this midpoint on the longer segment in another image is also detected, in fact, this correspondence is the intersection of the longer segment and the epipolar line of the shorter segment's midpoint in this image. Secondly, with the previous tensor, the midpoint of the shorter segment and its correspondence is transferred onto the current image, and an appropriate searching window around the transferred point is fixed to limit the searching area of the correspondence. Finally, the operation similar to feature points is inflicted on the points belonging to the detected lines and within this searching window. The correspondence is the line with the point that has the maximal normalized cross-correlation score with one of the two extracted points in reference images.

In above steps, not only do we fulfill the task of establishing points and lines correspondences respectively, but also constitute a NNC based criterion to evaluate the quality of points and lines matches in a uniform framework. In deed, this is a very important precondition of the PROSAC algorithm we used to estimate trifocal tensor in section 5.

5 Estimating Trifocal Tensor Using PROSAC

We now turn to the problem of calculating trifocal tensor using point correspondences obtained from previous section. The trifocal tensor plays a similar role in three views to that played by the fundamental matrix in two. It encapsulates all the projective geometric relations between three views that are independent of scene structure [6]. For a triplet of images, the image of a 3D point $\mathbf{X}$ is $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{x}''$ in the first, second and third images respectively, where $\mathbf{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ is homogeneous three vectors. If the three camera matrices are in canonical form, where $\mathrm{P} = [\mathrm{I} \mid 0]$, $\mathrm{P}' = [a^i_j]$, $\mathrm{P}'' = [b^i_j]$, and the a^i_j and b^i_j denote the *ij*-th entry of the matrix P' and P'' respectively, index *i* being the row index and *j* being the column index. Then the trifocal tensor can be computed by:

$$\mathrm{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \quad , j,k = 1,\ldots,3 \, , i = 1,\ldots,3 \tag{7}$$

Where the trifocal tensor $\mathrm{T} = [\mathrm{T}_1 , \mathrm{T}_2 , \mathrm{T}_3]^{\mathrm{T}}$ is a $3 \times 3 \times 3$ homogeneous tensor. Using the tensor a point can be transferred to a third image from correspondences in the first and second:

$$x_l^{\prime\prime} = x_i^{\prime} \sum_{k=1}^{3} x_k \, \mathrm{T}_k^{jl} - x_j^{\prime} \sum_{k=1}^{3} x_k \, \mathrm{T}_k^{il} \, , i,j = 1,\ldots,3 \, , l = 1,\ldots,3 \tag{8}$$

Similarly, a line can be transferred as

$$l_i = \sum_{j=1}^{3} \sum_{k=1}^{3} l_j^{\prime} l_k^{\prime\prime} \, \mathrm{T}_k^{ij} \tag{9}$$

The trifocal tensor has 27 elements, 26 equations are needed to calculate a tensor up to scale. Each triplet of point matches can give four independent linear equations for the entries of the tensor, and each triplet of line matches can provide two independent linear equations, accordingly, 7 points or 13 lines or something in the middle, if one uses both line and point matches are needed to compute the trifocal tensor linearly. The above method called linear algorithm or normalized linear algorithm when input data is pretreated. The drawback of these algorithms is that they do not take into account the internal constraints of the trifocal tensor and cannot necessarily yield a geometrically valid tensor. Algebraic minimization algorithm may be a good choice to obtain a valid tensor. However, due to using all available correspondences, this method is prone to being affected by the presence of mismatches (outliers). To overcome the disturbance of the mismatches, the 6-point RANSAC method has been brought forward. This method has the capability of generating a precise tensor even in the presence of a significant number of outliers. The shortcoming of RANSAC is that its computational complexity increases dramatically with the number of correspondences and proportion of mismatches. Moreover 6-point RANSAC approach does not take into consider the line matches, which is also an important clue to compute the need tensor. In our research, we propose a novel method to calculate trifocal tensor using point and line matches synthetically, which take full advantage of PROSAC and algebraic minimization algorithm.

5.1 Sampling with PROSAC

Unlike RANSAC, which treats all matches coequally and extracts random samples uniformly from the full set, PROSAC [7] samples are semi-randomly drawn from a subset of the data with the highest quality, and the size of the hypothesis generation set is gradually increased. The size of the set of potential matches has very small influence on its speed, since the solution is typically found early, when samples are taken from a smaller set. The improvement in efficiency relies on the presupposition that potential correspondences with high similarity are more likely to be inliers. In fact, PROSAC is designed to draw the same samples as RANSAC, but only in a different order. The matches that more likely to be inliers are tested prior to the others, thereby, the algorithm can arrive at termination criterion and stop sampling earlier.

In our method, the set of *K* potential matches is denoted as U_K. The data points in U_K can be either point or line correspondence and are sorted in descending order with respect to the normalized cross-correlation score *s*.

$$u_i, u_j \in U_K : \; i < j \Rightarrow s(u_i) \geq s(u_j) \tag{10}$$

A set of *k* data points with the highest score is represented as U_k. Then, the initial subset contains the least top-ranked matches that can give 26 need equations. If all of the valid samples from the current subset $U_n = (u_1, u_2, ..., u_n)$ have been tested, then the next subset is $U_{n+1} = (u_1, u_2, ..., u_n, u_{n+1})$, and the following samples consist of u_{n+1} and the data points drawn from U_n at random. In above process, all the samples should contain the least matches (7 points or 13 lines or something in the middle) that can generate 26 equations needed in normalized linear algorithm.

5.2 Algebraic Minimization Algorithm

The standard algebraic minimization algorithm takes following steps [6]:

1. From the set of point and line correspondences construct the set of equations of the form. $\mathrm{At} = 0$, where A come from the equation (8) or (9), t is the vector of entries of tensor T_i^{jk}.
2. Solve these equations using normalized linear algorithm to find an initial estimate of the trifocal tensor.
3. Find the two epipoles $e^{'}(a_4)$ and $e^{''}(b_4)$ from the initial tensor as the common perpendicular to the left null-vectors of the three T_i.
4. According to Equation (7), construct the 27×18 matrix E such that $\mathrm{t} = \mathrm{Ea}$, where a is the vector representing entries of a_i^j and b_i^k.
5. Compute the tensor by minimizing the algebraic error $\|\mathrm{AEa}\|$ subject to $\|\mathrm{Ea}\| = 1$.
6. Find an optimal solution by iteration over the two epipoles using an iterative method like Levenberg-Marquardt algorithm.

To get a fast non-iterative algorithm, we omit the last iteration step in our algorithm. It has been proved that the negative influence of this predigestion is very slightly and can be ignored.

5.3 Estimating Method

The following steps give the outline of our PROSAC based algebraic minimization algorithm.

1. Construct 26 equations from the sample given by PROSAC algorithm.
2. Solve these equations to get a candidate tensor using the method described in section 5.2.
3. Reproject all of the potential matches on to the current frame using above tensor and equation (8) or (9).
4. If the number of inliers is less than the predefined threshold T (varying with different environment), then generate a new sample using PROSAC and repeat the above steps.
5. If the number of inliers is greater than T, then re-estimate the tensor using these inliers and terminate.

The criterion to judge outliers is described as follows:

For points, we take a correspondence as outlier, if the distance between the reprojection and the detected point in current frame is greater than 3 pixels.

For line segments, if the orientation difference between the reprojection and the detected segment is larger than 5° or the distance between the detected segment and the reprojection of the selected points of the lines on the references is greater than 3 pixels, we consider it as outlier.

6 Experimental Results

The proposed method has been implemented in C++ on a fast Graphics Workstation. On average, our system can run at 10fps with 320×240 pixel images. Some experiments have been carried out to demonstrate the validity of the proposed approach.

6.1 Tracking Accuracy

The first experiment is carried out to test the accuracy of our method. The reprojection errors between the original specified points and their reprojections are compared. Fig.1 shows the reprojection errors of our method. The average reprojection error of the first 500 frames is 3.2 pixels, which validates the accuracy of proposed method. Some images are also exhibited in Fig. 2.

6.2 Sample Times

Using the video sequence obtained from this experiment, we also compare the sample times between our modified RANSAC and standard RANSAC algorithm. The average and maximum sample times of the standard RANSAC is 36.7 and 194 respectively, which are 9.4 and 12.1 times of the values of our method. This experiment proves that our modified RANSAC is more stabile and can reduce sample times significantly, which makes our method suitable for online implementation.

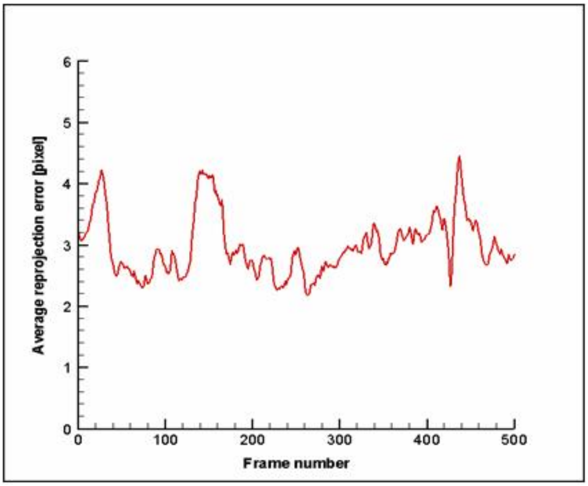
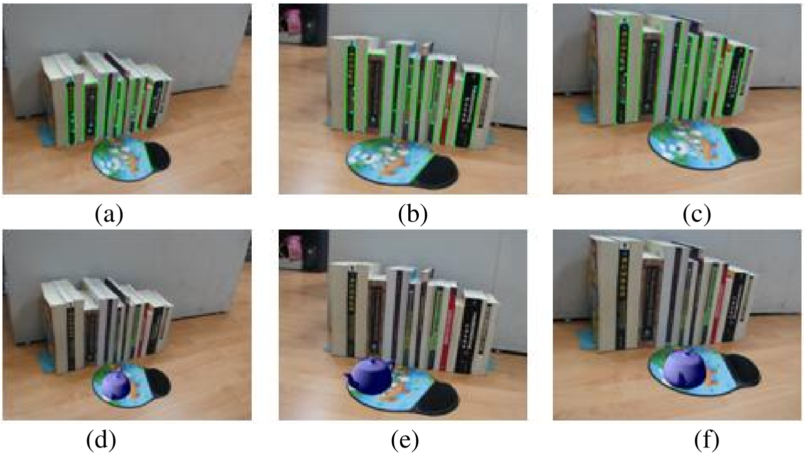


Fig. 1. Reprojection errors of the first 500 frames


(a) (b) (c)

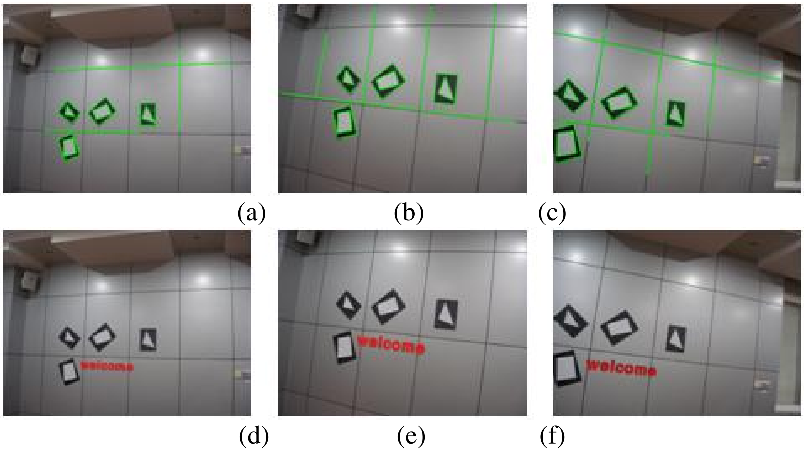
(d) (e) (f)

Fig. 2. Examples in first experiment. (a)-(c) are the 45th ,86th and 257th frames of the input video sequence with the inliers used to calculate the needed tensor, respectively. (d)-(f) are the corresponding registration images.

6.3 Registration with Line Segments

We also take an experiment to validate the usability of our method under the poorly textured scenes. In this experiment, 37 line correspondences (only lines of length 10 pixels or more are considered) are extracted from the two reference views. In online stage, only these line segments are tracked to compute the needed tensor. We have successfully augmented a 3D virtual word on the wall over 450 consecutive frames.

Fig.3 shows some images of the augmented sequence. This experiment proves that our method is effective even under the less textured scenes.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Registration with Line Segments. (a)-(c) are the 23th, 56th and 233th frames of the input sequence with the inliers, respectively. (d)-(f) are the corresponding registration images.

7 Conclusion

In this paper, we presented a registration method for augmented reality systems based on robust estimation of trifocal tensor using point and line correspondences simultaneously. With both types of features, the robustness of the system is improved to a large degree. As shown in our experiments, the proposed method can still work even under poorly textured scenes. To calculate trifocal tensor, we put forward a PROSAC based algebraic minimization algorithm. While improves the accuracy, this method also reduces the computation complexity to a large degree.

References

1. Simon, G., Fitzgibbon, A., Zisserman, A.: Markerless tracking using planar structures in the scene. In: Proc. of the ISAR, pp. 120–128 (2000)
2. Pang, Y., Yuan, M.L., Nee, A.Y.C., Ong, S.K.: A Markerless Registration Method for Augmented Reality based on Affine Properties. In: Proc. of the AUIC, pp. 25–32 (2006)
3. Simon, G., Berger, M.: Reconstructing while registering: a novel approach for markerless augmented reality. In: Proc. of the ISMAR, pp. 285–294 (2002)
4. Yuan, M.L., Ong, S.K., Nee, A.Y.C.: Registration Based on Projective Reconstruction Technique for Augmented Reality Systems. IEEE Trans. on Visualization and Computer Graphics 11(3), 254–264 (2005)

5. Yuan, M.L., Ong, S.K., Nee, A.Y.C.: Registration using natural features for augmented reality systems. IEEE Trans. on Visualization and Computer Graphics 12(4), 569–580 (2006)
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
7. Chum, O., Matas, J.: Matching with PROSAC - progressive sample consensus. In: Proc. of CVPR. 1(1), pp. 305–313 (2005)