

An Improved Real-time Natural Feature Tracking Algorithm for AR Application

Chen Jing, Wang Yongtian, Li Yu, Hu Wenze, Zang Xiaojun

Department of Optical Electronic Engineering, Beijing Institute of Technology, 100081

E-mail: { chen74jing29, wyt, l_y_o, zangxiaojun }@bit.edu.cn; hez2000@gmail.com

Abstract

A real-time camera registration algorithm using nature features for augmented reality applications is presented. The system uses a single camera for visual tracking of the nature features extracted from the real scene. A limited number of calibrated key-frames and a rough 3D model of the part of the real environment are required. Accurate camera registration can be achieved by matching the input image and the key-frame, whose viewpoint is as close as possible to each other. The problem of wide baseline correspondence is solved by rendering an intermediate image. Extended Kalman filter is applied for jitter correction. The performance of the algorithm is tested using real image sequences. Experimental results demonstrate that our registration algorithm is accurate and robust, and it can handle significant aspect changes.

1. Introduction

The crucial technique in an augmented reality (AR) system is to perfectly overlay a virtual computer generated 3D graphic model on the scenery. In order to achieve this we have to get the relationship between the real world and the camera's viewpoint. Although commercial products are available for offline camera registration, robust real-time tracking is still an open issue. Up to now many of the real-time tracking algorithm described in the literatures lack robustness, tend to drift and error accumulation which make them unsuitable for AR application [1-3]. In order to overcome these problems, we developed an efficient online natural feature tracking algorithm for AR applications, which can handle long-term drift, aspect changes and camera displacement.

Our system requires a set of keyframes calibrated in offline stage and a rough 3D model of the part of the real scene. The keyframes and the 3D model can be easily created by using commercial software Boujou

from 2D3 cooperation [4]. Registration can be realized by matching current live image and the keyframe, whose viewpoint is as close as possible to the current one. Wide baseline correspondence problem has been solved by re-rendering intermediate image. Registration results are smoothed by Kalman filter for jitter correction.

In the remainder of the paper, we first discuss related work. Section 3 describes some important details about our natural feature based tracking algorithm. Section 4 explicates the corresponding experimental results and section 5 the conclusion and future work.

2. Related works

To overlay virtual objects on the real world at the right position and orientation, the speed and accuracy of registration are of most importance. Offline structure and motion technology appears to offer significant possibilities for accurate registration with accuracies of around 0.2 pixels and negligible jitter. However these techniques take advantage of bundle adjustment techniques, which is time-consuming and not likely to be suitable for real-time AR implementation. Up to now, many methods perform the task of the real-time application but prone to be less reliable or impractical. The most popular approach applied in AR application is to use ARToolKit software for accurate registration [5], which requires man-made markers. This method is greatly confined in outdoor AR systems. Because markers are very sensitive to outdoor lightening, and when some of the markers are randomly hidden from the view, this method becomes feeble.

Model-based approaches are commonly used for dealing with unknown environment [6-8], which can achieve real-time tracking by re-projecting features of the given 3D model into the 2D image. Position and orientation can be found by least-squares minimization

of an objective function. However the optimization may lead to error local minimum, when target object features become to close to each other. Another approaches for unknown environment pose tracking is to use concatenating transformations, which tend to suffer from error accumulation in long image sequences. Recently, Nister has reported on first success with real-time pose estimation in completely unknown scenes[9]. However, long-term stability is poor (the algorithms tend to drift), and drift is limited by insertion of a "firewall", which leads to a system restart when the drift becomes too large[10].

Using some of the absolute information is a way to eliminate the drift problem. [11] use two reference frames to track the whole sequence without to solve aspect change problem. In [12] a set of reference frames are utilized and FFT transform is used to complete 2D image registration. Time consuming is the mainly drawbacks in this approach. This paper describes a method which has no such limitations.

3. Natural Feature-based Registration Algorithm

In this section we will explain our natural feature based algorithm which need a small number of calibrated keyframes and a rough 3D model. This *priori* knowledge makes our algorithm robustness and accuracy. First we describe how to set up the offline keyframes, and then how to choose the most appropriate keyframe with the current image and how to calculate the camera pose by matching the chosen keyframe.

3.1. Building Offline Keyframes

During offline stage, a set of keyframes which represent the scene from different viewpoints has to be created and calibrated. In order to build these keyframes we make use of commercial Boujou software to retrieve the 3D structure of the scene and the corresponding camera projection matrix P . Because the commercial Boujou software relies on batch computation to obtain camera pose and the 3D model of the object, the results can be expected to be accurate and reliable enough. By back-projecting the interest points detected on the object surface, a 3D/2D correspondence (m_k, M_k) includes a 3D point M , which is always represented in homogenous form $(X, Y, Z, 1)^T$, and a 2D pixel point m , which is also represented in homogenous form $(x, y, 1)^T$, is set up. (m_k, M_k) is related by the 3×4 projective matrix P :

$$m_k = \lambda PM_k, P = K[R|T]$$

where R is a 3×3 rotation matrix, T is the translation vector of the camera and λ is the homogenous scale factor which is dependent of PM . When R, T is known for each keyframe, we can obtain following information for each keyframe: the two sets of m_k and M_k of 2D and 3D points; camera projection matrix P which can decompose into the internal matrix K , rotation matrix R and translation matrix T ; the surface normal \bar{n} for every aspect of the object model. Figure 1a shows one of the frames of the tracked object; 1b the 3D model of the tracked box and the corresponding camera trajectory set up by the Boujou software.



Figure 1a. A image of the coffee box and the interest points detected on its surface

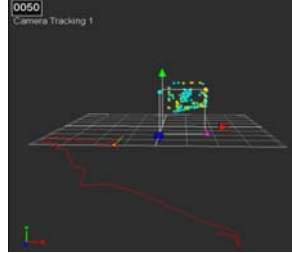


Figure 1b. the camera trajectory and the 3D model of the coffee box

3.2 Keyframe Choice

When the camera moves around the scene, the system have to choice which keyframe's viewpoint is as close as possible to the current captured image. This step is a critical task on which the quality of our algorithm depends. Since the camera pose for the current image is unknown, we choose the keyframe by calculating the normalized cross-correlation of the current live image $C(x, y)$ with the keyframe image $R(x, y)$. At the location (i, j) , the computing equation is given in a computationally efficient form:

$$C * R(i, j) = \frac{2 * \sum_{m,n} C(m-i, n-j) * R(m, n)}{r_1 * \sum_{m,n} C(m-i, n-j)^2 + r_2 * \sum_{m,n} R(m, n)^2}$$

Where $r_1 = \frac{\sum_{m,n} C(m-i, n-j)}{\sum_{m,n} R(m, n)}$ and $r_2 = \frac{1}{r_1}$. The keyframe

with the best score is chosen to evaluate the transformation between it and the current live image.

3.3 Intermediate Image

When we match an input image against the chosen keyframe, the viewpoint of these two images is not as close as possible to that of the consecutive ones. In order to perform wide baseline matching in real time, we use matching strategy described in [13]. First we synthesize an intermediate image by skewing the 15×15 pixel patches around each interest point from the chosen keyframe viewpoint to the previous image viewpoint (Harris corner detector is used to extract interest points of the images). Based on the assumption that the patches around the interest point is approximate to the object surface, through homography H the pixels m_k around interest point m_{k0} in the chosen keyframe to the pixels m_{re} in the intermediate image can be written as: $m_{re} = Hm_k$. Homography H can be induced by the equation[14]:

$$H = K_r (RR_r^T - (T - RR_r^T T_r) R_r \bar{n} / (d - T_r^T R_r \bar{n})) K^{-1}$$

The $K_r [R_r | T_r]$ and $K [R | T]$ are the project matrix of the chosen keyframe and the previous image respectively. \bar{n} is the normal vector of the plane which interest point lies on, and d the distance of the interest point to the origin. Figure 2a shows the current image and 2b the intermediate image which is re-rendered by the interest points' neighborhood from the chosen keyframe by using previous image viewpoint.

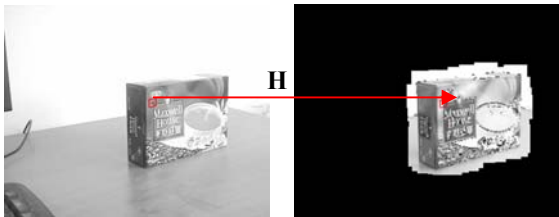


Figure 2. a) The current live frame 80; b) the intermediate image re-rendered from chosen keyframe by the previous image's (frame 79) viewpoint

3.4 2D/2D Correspondences

To match the 2D points between current image m_i and the intermediate frame m_{re} , we select

initial correspondences in the current image for each one in the re-rendered frame by maximizing the cross-correlation in a 7×7 window. All these point to point correspondences form a set of initial matches. In order to refine the correspondences between the current image and the intermediate image, RANSAC algorithm is employed to obtain the accurate interesting point correspondence, which can discard outliers and retain a set of points free from error accumulation. When applying RANSAC algorithm to obtain the 2D/2D correspondences between two images. Here the point p in the intermediate image and its correspondence q in the input image are represented as $\tilde{p}(x, y, 1)^T$, $\tilde{q}(x, y, 1)$ respectively. The process to match interest point by RANSAC approach can be written as follows:

(1) Eight matches are randomly chosen from the initial set of point matches and the fundamental matrix F is calculated by epipolar constraint $\tilde{p}^T F \tilde{q} = 0$.

(2) With F , the point p in the input image is mapped to the epipolar line in the intermediate image, and vice versa. Thus we can find good matches between these two images. For a point match (p, q) , epipolar line of point p is defined as $l_p = F\tilde{p}$. If the match is perfect, the point q in the re-rendered frame should lie on the epipolar line l_p . The distance d_q of point q to the

epipolar l_p is calculated by: $d_q = \frac{|\tilde{q}^T F \tilde{p}|}{\sqrt{(F\tilde{p})_1^2 + (F\tilde{p})_2^2}}$,

Where $F\tilde{p}_i$ is the i -th component of vector $F\tilde{p}$.

(3) The distance d_p of point p to the epipolar line l_q is also calculated similarly. If $\max(d_p, d_q)$ is below the threshold, the two points are considered to be matched.

(4) Re-estimate F from all correspondences classified as inliers by minimizing cost function.

(5) Further interest point correspondences could be found using the optimized F and these newly found correspondences can be determined as inliers directly. Steps (4) and (5) can be iterated until the number of correspondences is stable.

As a result, some of the current image points m_i are matched to the intermediate image points m_{re} :

$$m_i^j \leftrightarrow m_{re}^i$$

Since the relationship between m_{re} and M_k is known. The 3D points M_i^j can be associated to the 2D points m_i^j , giving in this way the 3D coordinates of the 3D points:

$$M_i^j = M_k^i$$

3.5 2D/3D Correspondences

After building the 2D/3D correspondences (m_t, M_t) in the current image, and the camera's intrinsic matrix K has already been known and fixed, the rotation matrix R and translation vector T can be easily recovered from equations by P-N-P algorithm[15]:

$$\tilde{m}_t = \lambda K[R|T]\tilde{M}_t$$

In order to improve the tolerance to data errors and the accuracy of estimation result, we apply Tukey M-estimator to calculate the camera pose. For M-estimation techniques it minimizes the sum of the function of the residuals:

$$\min_P \sum_i^n \rho(r_i)$$

where ρ is a continuous, symmetric function, and $r_i = \|m_t^i - \lambda PM_t^i\|$ is the re-projection error. For Tukey M-estimator $\rho(x)$ is described as:

$$\begin{aligned} \text{if } |x| \leq c, \rho(x) &= \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{x}{c}\right)^2\right)^3 \right] \\ \text{if } |x| > c, \rho(x) &= \frac{c^2}{6} \end{aligned}$$

By minimizing the residual sum, the camera rotation and translation matrix R and T for the current frame can be estimated accurately. In our algorithm, M-estimator is initialized with the camera pose for the previous frame.

4. Extended Kalman Filter

Actually only using keyframe based approach to calculate camera pose will result in jitter because the successive camera pose would be recovered independently for each frame. In our algorithm we avoid jitter problem by using extended kalman filter to smooth the estimation results. For the camera dynamic motion model we assume constant angular velocity and constant translational acceleration and represent motion by a state vector $\bar{x}_m = [q, p, \omega, v, a]^T$. Unit quaternion $q = (q_w, q_1, q_2, q_3)$ is used to represent the orientation of the camera. ω is the corresponding angular velocities, p the translations of the object along the x, y and z axes, v and a are their corresponding velocities and acceleration. ΔT denotes the duration over the sample period. The state

transition equation for the model is: $\hat{x}_{t+1}^- = Ax_t + \eta_t$, A is the transformation matrix, η_t the noise matrix and

$$\bar{x}_{t+1}^- = \begin{bmatrix} (q)_t \otimes \begin{bmatrix} \cos(0.5\omega\Delta T) \\ \sin(0.5\omega\Delta T) \frac{\omega}{\|\omega\|} \end{bmatrix} \\ \omega_t \\ p_t + v_t\Delta T + 0.5a_t\Delta T^2 \\ v_t + a\Delta T \\ a_t \end{bmatrix}$$

The measurement equation can be expressed by:

$$\bar{x}_{t+1} = \bar{x}_{t+1}^- + k(z_m - h_m(\hat{x}_{t+1}^-))$$

z_m is a column vector representing the real measurements from the image sequence for n selected interest points in the object. h_m is the output projection

function $h_m = h \left[\frac{\partial u_i}{\partial x} \frac{\partial v_i}{\partial x} \right]^T$. In our system, a fixed

number of interest points (e.g. 30 in our experiment) extracted by the tracker are passed to the filter for pose estimation.

5. Experiment Results

To evaluate the accuracy of our algorithm, we performed the tests depicted by real image sequences. A camera is used to capture a 720×576 sequence of the real scene and the camera has been calibrated with a chessboard using Zhang's method[16]. Commercial Boujou software is used to obtain the rough 3D model of the object and to calibrate a small number of keyframes. Because Boujou use batch techniques which can yield accurate estimation results, we also apply it as the ground truth data. The algorithm is initialized by manually putting the target object in a position which is close to one of the keyframes.

Harris corner detector is used to extract interesting points on the image sequences. Table 1 shows the cross-correlation result of some of the inputting frames and the keyframes. The keyframe with the best score is used to render the intermediate image. The intermediate image was re-rendered by skewing the 15×15 pixel patches around each interest point from the chosen keyframe viewpoint to the previous image viewpoint. Figure 3 shows some of the intermediate image set up by this approach.

Table 1. The cross-correlation score

Keyframe Current frame	Keyframe 1	Keyframe 2	Keyframe 3
Frame 20	0.7678	0.4972	0.5107
Frame 80	0.8292	0.4807	0.4815
Frame 150	0.8095	0.5184	0.5261



Figure 3. Intermediate image re-rendered from the keyframe

Finally, the register information is used to merge computer generated model to real scene at the right position. Figure 4 shows some frames from the sequence where the tracked coffee box augmented with the wireframe. Figure 5. shows another augmented scene, a 3D virtual girl standing on the table near the coffee box.



Figure 4. Coffee box augmented with wireframe



Figure 5. Coffee box augmented with 3D virtual girl

In order to test the performance of our algorithm, we also compared the estimated parameters with the ground truth data. Figure 6a shows the estimated coordinate of the camera center(translation along x axes) and 6b the rotation angle(rotation around x axes). The black square depicts the ground truth data obtain from the 2D3 software, the solid plot demonstrates the result by our natural feature tracking algorithm. From the plots we can see that the estimated parameters smoothed by kalman filter keeps closer to the ground truth data without exhibiting jitter

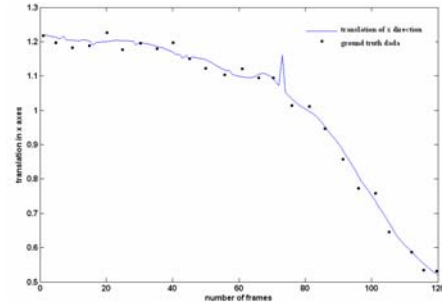


Figure 6a. Plots showing the translation in x direction

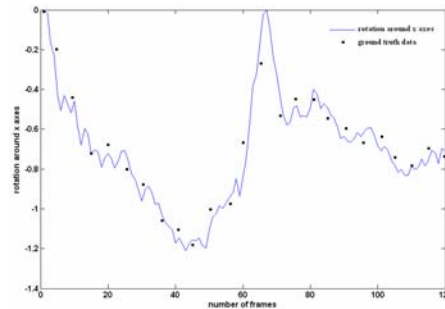


Figure 6b. Plots showing the yaw angle

6. Conclusion

In this paper we present a nature feature based real-time tracking algorithm for augmented reality systems. Our system relies on the passive vision techniques to obtain the camera pose online. 3D model of the part of the scene and a set of calibrated keyframes are used for tracking purpose. We select the most appropriate keyframe at each time step for image registration. Wide baseline correspondence problem was solved by rendering intermediate image. Extended kalman filter was utilized for jitter correction. Experimental results show the accuracy and the robustness of our algorithm.

Acknowledgments. This project is supported by the National Basic Research Program of China (National 973 Project, Grant No. 2002CB312104) and by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, Grant No. (2006)331.

References

- [1] G.Simon, A. Fitzgibbon, A.Zisserman, "Markerless Tracking Using Planar Structures in the Scene", In Proc. of International Symposium on Augmented Reality, 2000.

- [2] U.Neumann, S.You, Y.Cho, J. Lee, and J. Park., “Natural Feature Tracking for Augmented Reality”, *IEEE Transactions on Multimedia*, Vol. 1, no.1, 1999, pp. 53-64.
- [3] M.Uenohara, and T.Kanade, “Vision Based Object Registration for Real Time Image Overlay”, *Computers in Biology and Medicine*, 1996.
- [4] <http://www.2d3.com>
- [5] H.Kato, and M.Billinghurst, “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System”, In *IEEE and ACM International Workshop on Augmented Reality*, 1999.
- [6] B.Reinhold, P.Jun, and S.Venkataraman, “Model-Based Visual Tracking for Outdoor Augmented Reality Applications”, *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2002.
- [7] T. Drummond, and R.Cipolla, “Real-time Tracking of Multiple Articulated Structures in Multiple Views”, In *ECCV (2)*, 2000, pp.20–36.
- [8] E. Marchand, P. Bouthemy, F. Chaumette, V.Moreau, “Robust Real-time Visual Tracking Using a 2D-3D Model-Based Approach”, In *International Conference on Computer Vision*, Corfu, Greece, 1999, pp.262–268.
- [9] D.Nister, “An Efficient Solution to the Five-point Relative Pose Problem”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.26, no.6, 2004, pp. 756-770.
- [10] D.Nister, O.Naroditsky, J.Bergen, “Visual Odometry”, *Proc. CVPR*, 2004, pp.652-659.
- [11] K.W.Chia, A.D.Cheok, and S.J.D. Prince, “Online 6 DOF Augmented Reality Registration from Natural Features” In *International Symposium on Mixed and Augmented Reality*, 2002.
- [12] D. Stricker, P. Daehne, F. Seibert, I. Christou, L.Almeida, R.Carlucci, and N. Ioannidis, “Design and Development Issues for Archeoguide: An Augmented Reality Based Cultural Heritage on-site Guide”, In *International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging*, Mykonos, Greece, 2001.
- [13] V. Luca, L.Vincent, F.Pascal, “Fusing Online and Offline Information for Stable 3D Tracking in Real-Time”, In *Proceeding CVPR*, 2003, pp.241-248.
- [14] R.Hartley, and A.Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000
- [15] C. Lu, G.Hager, E. Mjolsness, “Fast and Globally Convergent Pose Estimation from Video Images”, *Transaction on Pattern Analysis and Machine Intelligence*, Vol. 22, no.6, 2002, pp.610–622.
- [16] Z.Zhang, “Flexible Camera Calibration by Viewing a Plane from Unknown Orientations”, *Proceedings of 7th ICCV*, 1999, pp.666–673.