

Robust Camera Pose Estimation Using 2D Fiducials Tracking for Real-Time Augmented Reality Systems

Fakhr-eddine Ababsa*

Laboratoire Systèmes Complexes. CNRS FRE 2494
40, Rue du Pelvoux, 91020 Evry ,France

Malik Mallem†

Laboratoire Systèmes Complexes. CNRS FRE 2494
40, Rue du Pelvoux, 91020 Evry ,France

Abstract

Augmented reality (AR) deals with the problem of dynamically and accurately align virtual objects with the real world. Among the used methods, vision-based techniques have advantages for AR applications, their registration can be very accurate, and there is no delay between the motion of real and virtual scenes. However, the downfall of these approaches is their high computational cost and lack of robustness. To address these shortcomings we propose a robust camera pose estimation method based on tracking calibrated fiducials in a known 3D environment, the camera location is dynamically computed by the Orthogonal Iteration Algorithm. Experimental results show the robustness and the effectiveness of our approach in the context of real-time AR tracking.

Keywords: Augmented reality, fiducials tracking, camera pose estimation, computer vision.

1 Introduction

AR systems attempt to enhance an operator's view of the real environment by adding virtual objects, such as text, 2D images, or 3D models, to the display in a realistic manner. It is clear that the sensation of realism felt by the operator in an augmented reality environment is directly related to the stability and the accuracy of the registration between the virtual and real world objects, if the virtual objects shift or jitter, the effectiveness of the augmentation is lost.

Several AR systems have been developed these last years, they can be subdivided into two categories: Vision-based AR systems (indirect vision) and see-through AR systems (direct vision). Vision-based techniques have more advantages for AR applications. First, the same video camera used to capture real scenes also serves as a tracking device. Second, the pose calculation is most accurate in the image plane, thereby minimizing the perceived image alignment error. Additionally, processing delays in the video and graphics subsystems can be matched, thereby eliminating dynamic alignment errors [Neumann and Cho, 1996]. Recently, several vision based methods of estimating position information from known landmarks in the real world scene have been proposed. Bajura and Neumann used LEDs as landmarks and demonstrated vision-based registration for AR systems [Bajura and Neumann, 1995]. Uenohara and Kanade used template matching for object registration [Uenohara and Kanade, 1995]. State et al. proposed a hybrid method of combining landmark tracking and magnetic tracking (they used color markers as landmarks) [State et al. 1996].

In this paper we propose a robust camera pose estimation method based on tracking calibrated 2D fiducials in a known 3D environment. To efficiently compute the camera pose associated with the current image, we combine results of the fiducials tracking method with the Orthogonal Iteration (OI) Algorithm [Lu et al. 2000]. Indeed, the OI algorithm usually converges in five to ten iterations from very general geometrical configurations. In addition, it outperforms the Levenberg-Marquardt method, one of the most reliable optimization methods currently in use, in terms of both accuracy against noise and robustness against outliers. Knowing the camera poses for each image frame, we can integrate virtual objects into a video segment.

The remainder of this paper is organized as follows. Section 2 is devoted to the system overview. Section 3 describes in details the 2D fiducials tracking algorithm. Section 4 introduces the Orthogonal Iteration Algorithm and its adaptation to compute the camera pose. Experimental results are then presented in section 5, which show the stability, the robustness to scale, orientation, and the computational performance of our approach. Finally, section 6 provides conclusions.

2 System Overview

Our vision-based AR system is composed of four main components (figure1):

- 2D fiducials detection: detect 2D markers in each new video image.
- 2D-3D correspondence: identification of the detected fiducials allows to match 2D image features with their calibrated 3D features.
- Camera pose estimation: estimating camera pose based on 2D-3D correspondence.
- Virtual world registration: the final output of the system is an accurate estimate of camera pose that specifies a virtual camera used to project the virtual world into the current video image.

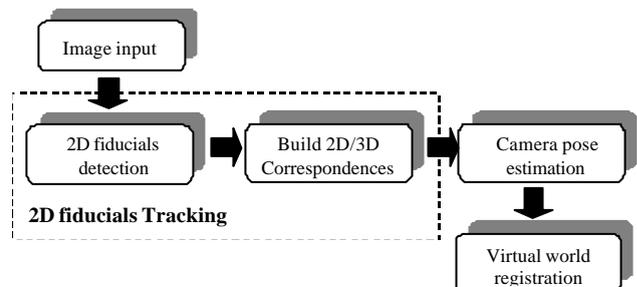


Figure 1. Vision-based AR system architecture

*e-mail:ababsa@lsc.univ-evry.fr

†e-mail:malleml@lsc.univ-evry.fr

3 Fiducials Tracking Algorithm

In our approach we have considered a square-shaped fiducial (figure 2.a) with a fixed, black band exterior surrounding a unique image interior. The outer black band allows for location of a candidate fiducial in a captured image and the interior image allows for identification of the candidate from a set of expected images. The four corners of the located fiducial allow for the unambiguous determination of the position and orientation of the fiducial relative to a calibrated camera. Furthermore, in order to estimate location of a moving camera in the world coordinate system, Fiducials are placed in the fixed, physical environment, in this case, the cupboard and the wall (figure 2.b).

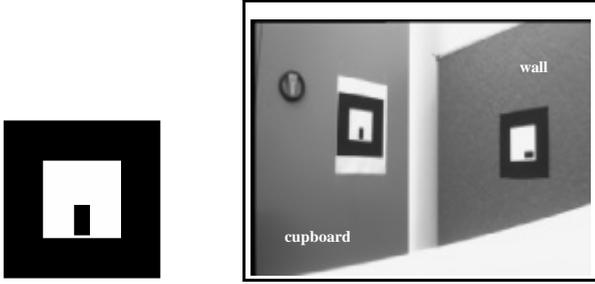


Figure 2. (a) Fiducial, (b) 3D environment with two calibrated fiducials

Our 2D fiducials tracker must uniquely identify any valid patterns within the video frame. Using a method similar to [Kato and Billingham, 1999], the recognition algorithm proceeds as follows:

Image binarization: the program uses an adaptive threshold to binarize the video image (figure 3b). Binary images contain only the important information, and can be processed very rapidly.

Connected regions Analysis: the system looks up connected regions of black pixels (figure 3-c) and only select the quadrilateral ones. These regions become candidates for the square marker. For each candidate found, the system segregates the contour chains (figure 3-d) into the four sides of the proposed marker, and fits a straight line to each using principal components analysis (PCA). Finally, the coordinates of the four corners are found by intersecting these lines (figure 3-e) and are stored for the next processes.

Fiducials recognition: for each selected region, the system takes the four corners points and maps the enclosed area to a standard 100x100 template shape. The normalized templates are then compared to the stored ones at all four orientations. A variety of methods are possible for comparing images, we have used the correlation coefficient method because it is luminance invariant. So, the mean and standard deviations for the normalized template I and stored pattern P are first computed:

$$\mu_I = \frac{1}{xy} \sum_x \sum_y I(x,y) \quad (1)$$

$$\mu_P = \frac{1}{xy} \sum_x \sum_y P(x,y) \quad (2)$$

$$\sigma_I = \sqrt{\sum_x \sum_y (I(x,y) - \mu_I)^2} \quad (3)$$

$$\sigma_P = \sqrt{\sum_x \sum_y (P(x,y) - \mu_P)^2} \quad (4)$$

Then, the correlation coefficient is computed as:

$$\rho = \frac{\sum_x \sum_y [I(x,y) - \mu_I][P(x,y) - \mu_P]}{\sigma_I \sigma_P} \quad (5)$$

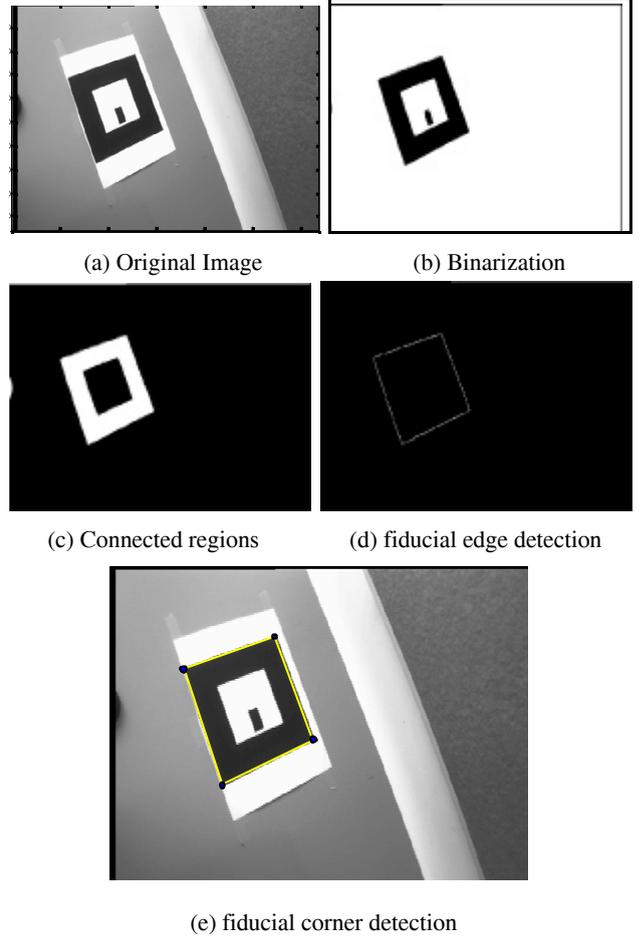


Figure 3. Fiducial extraction process

Finally, a correlation matrix is created, relating each found marker to each stored template. It allows to allocate the markers to templates by finding the greatest correlation coefficient.

4 Camera Pose Estimation

The recognized marker region is used for estimating the current camera position and orientation relative to the world coordinate system. From the coordinates of four corners of the marker region on the projective image plane, a matrix representing the translation and rotation of the camera in the real world coordinate system can be calculated. Several algorithms have been developed last years. Examples are the Hung-Yeh-Harwood pose estimation algorithm [Hung et al. 1985] and the Rekimoto 3-D position reconstruction algorithm [Rekimoto and Ayatsuka, 2000]. In this work we adapted the algorithm

proposed by Lu et al. [Lu et al. 2000], namely the Orthogonal Iteration Algorithm, to perform the camera pose estimation.

4.1. Camera Model and Coordinates

The configuration of our system includes only a moving CCD video camera. There are three principal coordinate systems, as illustrated in Figure 4: the world coordinate system W , the camera-centered coordinate system C , and the 2D image coordinate system U .

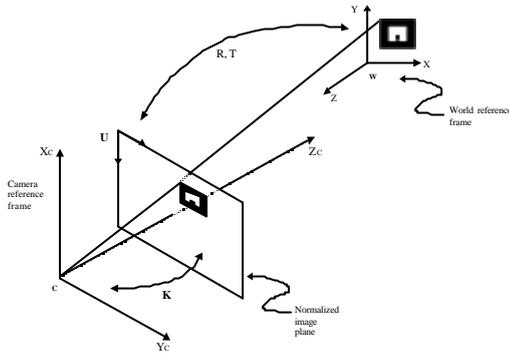


Figure 4. Camera model and the related coordinates systems

A pinhole camera models the imaging process. The origin of C is at the projection center of camera. The transformation from W to C is:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = [R \quad | \quad T] \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (6)$$

where the rotation matrix R and the translation vector T characterize the orientation and the position of the camera with respect to the world coordinate frame. Under perspective projection, the transformation from W to U is:

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = [K][R \quad | \quad T] \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (7)$$

where the matrix K :

$$K = \begin{bmatrix} \alpha_x f & 0 & u_0 \\ 0 & \alpha_y f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

contains the intrinsic parameters of the camera, f is the focal length of camera, α_x , α_y are the horizontal and vertical pixel sizes on the imaging plane, and (u_0, v_0) is the projection of camera center (principal point) on the image plane.

4.2. Camera Calibration

Internal, as well as, external camera parameters are determined by an automated (i.e. with no user interaction) camera calibration

procedure. A highly precise camera calibration is required for a good initialisation of the camera pose tracker. For that purpose, we have used our fiducias tracking algorithm to generate enough 2D-3D matched points. The calibration parameters are then computed by an iterative least-squares estimation [Faugeras, 1993].

The intrinsic parameters K remain constant during the camera tracking mode. The external parameters describe the transformation (rotation and translation) from world to camera coordinates and undergo dynamic changes during a session (e.g. camera motion). Once the camera calibration is finished, the system passes in tracking mode, and uses the obtained external camera parameters for the first initialisation of the camera pose. The current camera pose is then computed using the OI algorithm described below.

4.3. Orthogonal Iteration Algorithm

The OI algorithm allows to dynamically determine the external camera parameters using 2D-3D correspondences established by the 2D fiducias tracking algorithm from the current video image.

The main idea of this algorithm is first in defining pose estimation using an appropriate object space error function, in this case object-space collinearity error vector, and then in showing that this function can be rewritten in a way which admits an iteration based on the solution to the 3D-3D pose estimation or absolute orientation problem [Arun et al. 1987].

Otherwise, the OI algorithm converge to an optimum for any set of observed points and any starting point $R^{(0)}$. However, in order to reduce the average number of iteration taken by OI to converge, we initialize it near the optimum for each new acquired image. So, at time t (corresponding to the current image), we initialize the rotation matrix by the matrix R found at time $t-1$ (corresponding to the previous image).

5 Results and Discussion

In our experiments we recorded an image sequence from a moving camera pointing at the wall and the cupboard (Figure 2.b). One fiducial can be seen, at least, in this area. The frame rate is 25 frames/s and there are 1000 frames in the over 40 second long sequence. We tracked the 2D fiducials on every frame. When the system identifies a detected fiducial, the corresponding overlay information is retrieved from the database (in this case 3D two wire frame models: a cube and a pyramid). Using the estimated camera pose, these virtual objects can correctly be superimposed on the video image.

Figure 5 shows four frames of the video sequence showing virtual objects rendering. For each frame, the camera pose was estimated using two 2D detected fiducials. From figures (5-a), (5-b), (5-c) and (5-d) we can see that virtual objects are well superimposed on the real world. Our current implementation exhibits an average reprojection error between 0.7 and 1.2 pixels.

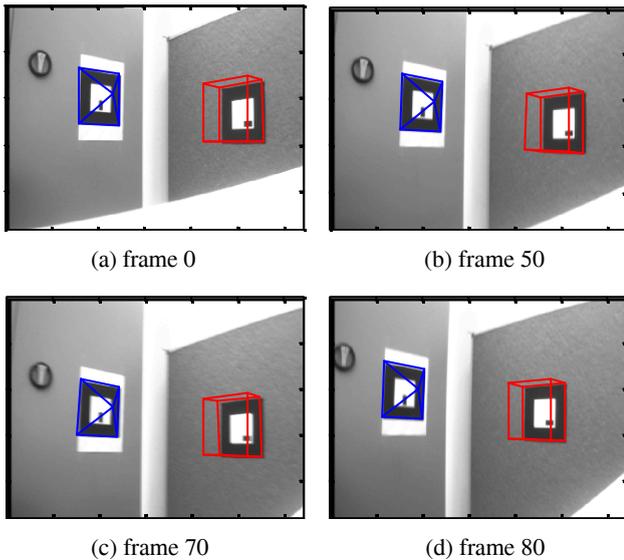


Figure 5. Camera tracking results

Figure 6, illustrates the robustness of our approach to:

- Effects of scales, the major advantage in using corners for tracking is that corners are invariant to scale. Figure (6-a) shows that our 2D fiducials tracker can detect and identify markers in spite of the large range of distances from the camera.
- Poor detection: figure (6-b) illustrates the ability of our system to well estimate the camera pose when only one fiducial is detected.
- Effects of orientations, due to perspective distortion, a square on the original pattern does not necessarily remain square when viewed at a sharp angle and projected into image space. Figure (6-c) illustrates the efficiency of our system in such situations.

Otherwise, real-time performance of our system has been achieved by carefully evaluating each processing step. We have implemented our system on an Intel Pentium 3 500 MHz PC equipped with a Matrox 2 acquisition card and an iS2 IS-800 CCD camera. The average processing time per frame when viewing two fiducials is as follows:

Fiducials identification : 29 ms

Camera pose estimation : 4 ms

Augmentation time : 2 ms

As can be seen, processing times are very acceptable for real time implementation.

6 Conclusion

In this paper we described a robust solution for vision based augmented reality tracking that identifies and tracks, in real-time, known 2D fiducials made up of corners, in order to estimate the camera pose. The major advantages of tracking corners are their detection robustness at a large range of distances, and their reliability under severe orientations. Additionally, we have adapted the orthogonal iteration algorithm to our problem and have demonstrate its efficiency in such applications.

An overview of the developed system was described, and experiments demonstrated the feasibility and reliability of the system under various situations.

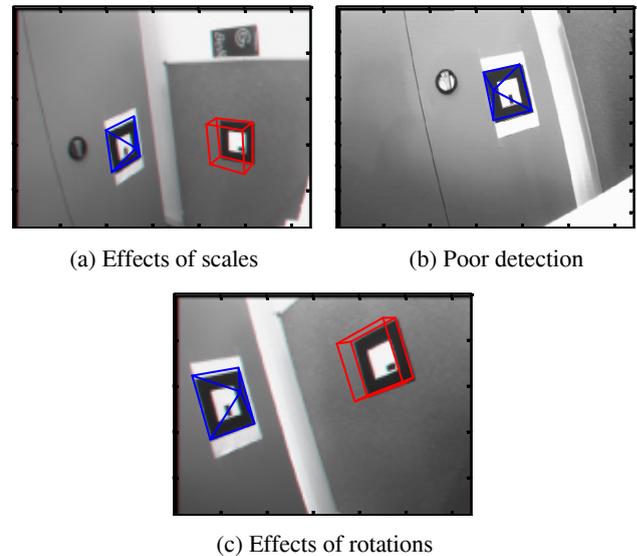


Figure 6. The system robustness

References

- Neumann, U., AND Cho, Y. 1996. A self-tracking Augmented Reality Systems". In *Proceedings of ACM Virtual Reality Software and Technology*. 109-115.
- Bajura, M., AND Neumann, U. 1995. Dynamic registration correction in augmented reality systems. In *Virtual Reality Annual International Symposium (VRAIS'95)*. 189-196.
- Uenohara, M., AND Kanade, T. 1995. Real-time vision based object registration for image overlay. *Journal of the Computer in Biology and Medicine*. 249-260.
- State, A, Hirota, G., Chen, D. T., Garrett, W. F., AND Livingston, M. A.. 1996. Superior augmented registration by integrating landmark tracking and magnetic tracking. In *SIGGRAPH'96 Proceedings*.
- Lu, C. P, Hager, G. D., AND Mjolsness, E. 2000. Fast and globally convergent pose estimation from video images. In *IEEE trans. Pattern Analysis and Machine Intelligence*, Vol. 22 no. 6, 610-622.
- Kato, H., AND Billinghurst, M. 1999. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In *Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*. 85 -94.
- Hung, Y., Yeh, P., AND Harwood, D. 1985. Passive Ranging to Known Planar Point Sets. In *Proceeding of IEEE International Conference on Robotics and Automation*, Vol. 1.,80-85.
- Rekimoto, J., AND Ayatsuka, Y. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. *Designing Augmented Reality Environments*. In *DARE (2000)*.
- Faugeras, O. 1993. Three-dimensional computer vision: ageometric viewpoint. MIT Press.
- Arun, K.S., Huang, T.S., AND Blostein, S.D. 1987. Least-Squares Fitting of Two 3D Point Sets. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, 698-700.