A Robust Circular Fiducial Detection Technique and Real-Time 3D Camera Tracking

Fakhreddine Ababsa
IBISC Laboratory– CNRS FRE 3190. University of Evry Val d'Essonne, France
Email: ababsa@iup.univ-evry.fr

Malik Mallem
IBISC Laboratory – CNRS FRE 3190. University of Evry Val d'Essonne, France
Email: mallem@iup.univ-evry.fr

***Abstract*— In this paper a new marker-based approach is presented for 3D camera pose tracking in indoor Augmented Reality (AR). We propose to combine a circular fiducials detection technique with a particle filter to incrementally compute the camera 3D pose parameters. In order to deal with partial occlusions, we have implemented an efficient method for fitting ellipse to scattered data. So even incomplete data will always return an ellipse corresponding to the visible part of the fiducial image. The other advantage of our approach comparing to the related camera pose estimation works is its capacity to naturally discard outliers which occur because of image noises. Results from real data in an augmented reality setup are presented, demonstrating the efficiency and robustness of the proposed method.**

***Index Terms*—Augmented Reality, Camera tracking, Circular fiducial, Particle filter**

I. Introduction

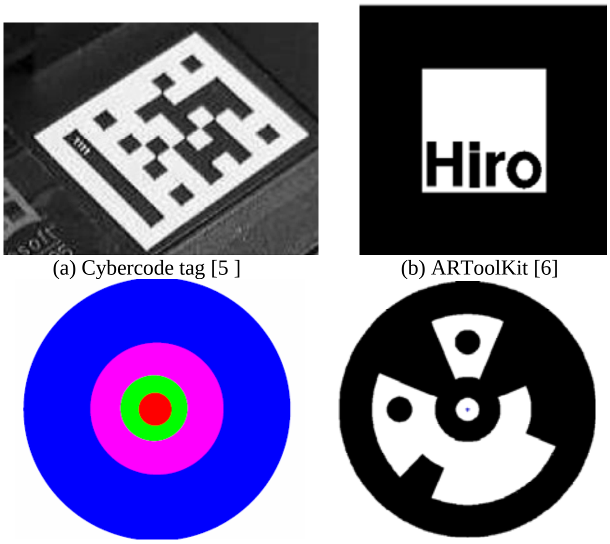
Augmented Reality (AR) systems superimpose virtual information (text, 3D graphics, etc.) onto view of the real world. Tracking computation which refers to the problem of estimating over the time, the position and orientation of the camera viewpoint, is crucial in order to maintain correct registration of real and virtual worlds. Camera tracking remains one of the most important key requirement for AR systems. Fiducial-based tracking is generally used to construct reliable AR systems. This approach supposes that the position and the orientation of fiducials placed through-out the workspace are *a priori* known. To recognize the extracted fiducial, a code pattern is placed inside it for a template matching algorithm . In AR project two kinds of coded fiducials are generally used: square and circular fiducials. Square shape gives four points per fiducials to compute the camera pose. Circular fiducial is invariant to viewing direction and angle (its center corresponds to a centroid), and can be determined with sub-pixel accuracy. Fiducial detection still a reliable and accurate technique. However, a significant challenge remains : fiducial detection requires that fiducial is completely visible to extract the code inside it allowing thus its recognition.

In this paper, we propose a novel approach to deal with fiducials partial occlusion. Indeed, we don't use any code inside the fiducials, they simply consist of a black dot placed arbitrary in the scene. The detection fiducial algorithm uses an efficient method for fitting ellipse to scattered data. So, even incomplete data will always return an ellipse corresponding to the visible part of the fiducial image. Furthermore, we propose to combine the fiducial detector with a particle filter in order to track the 3D camera pose. The filter measurements are then based on inlier/outlier counts of correspondence matches for a set of 3D circular targets who's positions in the scene are known. We use SIS with resampling [1] at each iteration to improve stability. The algorithm is robust to outliers, simple to implement and computation times are linear in the number of particles and scene features. This paper is an extension of our previous work [2] on circular fiducials tracking. The proposed algorithms are more described and discussed and new experiments results are illustrated.

The remainder of this paper is organized as follows. In section 2 we present the related work on artificial fiducials and vision-based racking in AR application. Section 3 presents our circular fiducial detector. Section 3 is devoted both to the camera pose problem formulation and the particle filter-based tracker implementation. Finally, section 4 provides conclusions and pointers for future work

II. Related Work

Last years, artificial fiducials have been widely used in both computer vision and augmented reality applications. Several circular fiducials-based methods have been proposed for calibrating a camera. Kim et al [3] demonstrated that camera calibration is possible using two views of a concentric circle with known size. Their algorithm is based on the quite simple geometric characteristics and uses non-linear minimization approach to estimate the calibration parameters. Unlike the previous method, Abad et al [4] proposed an algorithm that does not require any a priori information about the camera parameters to calibrate it. They also employed two concentric circles to recover the full pose of the camera. Furthermore, several pose estimation methods using artificial fiducials have been proposed for AR systems. Rekimoto and Ayatsuka [5] developed a square fiducial system called Cybercode that is a visual tagging system based on a 2D barcode technology. Figure (1.a) shows an example of CyberCode tags, the, information is encoded in a two-dimensional pattern, and can be optically recognized from image data. Cybercode tags are used to determine the 3D position of the tagged object as well as its ID number.



(a) Cybercode tag [5] (b) ARToolKit [6]

(c) Multi-ring color [8] (d) InterSense [9]

Figure 1. Artificial fiducials used in AR

Kato and Billinghurst [6] designed their popular ARToolkit library AR applications. ARTolkit markers are square-shaped fiducials (see fig. 1.b) 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. However, this approach is still limited though, because for any large database, template matching becomes a computationally expensive procedure with high rate of false alarms. Recently, Fiala [7] has developed the ARTag system which uses arrays of the square markers added to objects or the environment allowing a computer vision algorithm to compute the camera pose in real time. Furthermore, Cho and Neumann [8] developed multi-ring color fiducial systems (see fig 1.c) for scalable fiducial tracking AR systems. Colored areas are detected by expanding candidate pixels compared against reference colors. A centroid for the feature is computed by weighting the pixel's by their distance from the reference color. The value of this centroid will give one 2D point for each target. Since the camera is calibrated and the positions of the markers are known, at least three fiducials are needed to estimate the camera's pose. Recently, Naimark and Foxlin [9] designed a new 2D barcode circular fiducial that can generate thousands of different codes and can be used for wide area tracking. Every fiducial has an outer black ring, two data rings, and an inner black ring (see fig. 1.d). In order to extract fiducials from the scene and to read their barcodes, the authors applied a modified form of homomorphic image processing, which is designed to eliminate the effect of non-uniform lighting in images.

Besides, last years have also seen the emergence of vision based algorithms for real-time camera tracking. The basic idea of these methods is to find correspondences between 2D image features and their 3D coordinates in a definite world frame. The camera pose is then obtained by projecting the 3D coordinates of the feature into the image and minimizing the distance to their corresponding 2D feature. Numerical nonlinear optimization techniques like the Newton-Raphson or Levenberg-Marquardt algorithm are used for the minimization [10][11][12]. Other approaches use Kalman Filtering Framework to update the camera pose by auto-calibrating point or line features in the environment. Hence, Jiang and Neumann [13] developed an Extended Kalman Filter (EKF) tracking method that integrates pre-calibrated fiducials and natural line features for camera pose estimation. Yoon et al. [14] presented a model-based object tracking to compute the 3D camera pose. Their algorithm uses an EKF to provide an incremental pose-update scheme in a prediction-verification framework, and is robust to partial accuracy. Kyrki and Kragic [15] used an EKF to integrate model-based cues with automatically generated model-free cues. They extended their works and developed a method for automatic initialization of pose tracking based on robust feature matching and object recognition suitable for textured objects [16]. Furthermore, Particle Filters could provide improved robustness over the Kalman approach [17]. PFs also tend to be more flexible, particularly with respect to the observation model, and in general they are simpler to implement. Nevertheless, despite their strengths there are few papers on particle filter-based 3D pose estimation for Augmented Reality systems. Recently, Marimon et al. [18] [19] developed a tracking system that fuses a marker-based cue (MC) and a feature point-based cue (FPC). In their framework, measurements from both cues are fed into a particle filter in order to track the camera position and orientation.. Klein and Murray [20] developed a complex self-occluding three-dimensional structures based on a particle filter. Their tracker has robustness advantages over previous systems, particularly when exposed to rapid, unpredictable accelerations. However, a disadvantage of the proposed system is increased jitter in stationary scenes.

III. Circular Fiducial Detection

In our approach we have considered a circular-shaped fiducial with a fixed, white band exterior surrounding a unique black circular dot (see fig. 2). The center of the located fiducial allows 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 at fixed location through-out the workspace.

Figure 2. Our circular fiducial

To extract observed fiducials from the current image, we have developed a robust and fast fiducial detection algorithm based on an efficient method for fitting ellipses to scattered data [21]. Our fiducial extraction algorithm proceeds in several steps :

A. *Image binarization*

the program uses an adaptive threshold to binarize the original video image. Binary images contain only the important information, and can be processed very rapidly.

B. *Connected regions extraction*

the system looks up connected regions of black pixels whose the number of pixel is lower than a given threshold. These regions become candidates for the circular markers. For each candidate found, the system fits an ellipse using the Direct Least Square Fitting algorithm []. Finally, the image coordinates of the centres of the fitted ellipses are stored and used for the camera pose tracking

C. *Direct Ellipse-Specific Fitting Algorithm [21]*

Ellipse fitting approaches are generally based on least-square algorithms, where the mean idea is to find the parameters that minimize the distance between the data points and the ellipse. A general conic can be represented by an implicit second order polynomial:

$$\begin{aligned} F(A,X) &= A.X \\ &= a.x^2 + b.x.y + c.y^2 + d.x + e.y + f = 0 \end{aligned} \tag{1}$$

where $A = [a \quad b \quad c \quad d \quad e \quad f]^T$ and $X = [x^2 \quad x.y \quad y^2 \quad x \quad y \quad 1]^T$. *F(A,X)* corresponds to the algebraic distance of the point *(x,y)* to the conic *F(A,X)=0*. So, this fitting problem is equivalent to the classical minimization problem of the squared algebraic distances:

$$D(A) = \sum_{i=1}^{N} F(X_i)^2 \tag{2}$$

between the N points X_i and the ellipse. Generally the parameter vector A is constrained in some way in order to avoid the trivial solution $A = \mathbf{0}_6$. In this research work, we have implemented the solution proposed by Fitzgibon et al. [21]. The authors suggested to use the constraint $4.a.c - b^2 = 1$ to force the conic to be an ellipse. Their method incorporates this ellipticity constraint into the normalization factor and combines several advantages: It is ellipse-specific, so that even bad data will always return an ellipse. It can be solved naturally by a generalized eigensystem. It is extremely robust, efficient, and easy to implement.

IV. CAMERA POSE TRACKING

In this research work, the particle filter is used to estimate the 3D camera pose parameters over the time. The camera state is represented by position and rotation of the camera with respect to a world coordinate system. Rotations can be represented by several different mathematical entities (matrices, axe and angle, Euler angles, quaternions). However, quaternions have proven very useful in representing rotations due to several advantages above the other representations: more compact, less susceptible to round-off errors, avoid discontinuous jumps. A quaternion representation of rotation R is written as a normalized four dimensional vector $q = [q_0 \quad q_x \quad q_y \quad q_z]$ where $(q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1)$. Thus, the camera state is given by:

$$X = [q_0 \quad q_x \quad q_y \quad q_z \quad t_x \quad t_y \quad t_z] \tag{3}$$

where $T = [t_x \quad t_y \quad t_z]^T$ is the camera position (the translation vector).

We denote the camera state at time k by the vector X_k. Each particle X_k^n corresponds to a potential pose of the camera. The most probable particle will have important weights. These give the approximation to the posterior density. Basically, the key components of the Particle Filter are the state dynamics and the observations used.

A. *State dynamics*

Particle filter requires a probabilistic model for the state evolution between time steps, i.e. $p(X_x | X_{k-1})$. Since we have no prior knowledge of camera movement, we use a simple random walk based on a uniform density about the previous camera state [22]:

$$p(X_k | X_{k-1}) = U(X_{k-1} - v, X_{k-1} + v) \tag{4}$$

where $v = [v_1 \quad v_2]^T$ represents the uncertainty about the incremental camera movement (v_1 for rotation and v_2 for translation).

The camera undergoing a random walk moves a certain random distance Δd and deviating from its previous direction by some random quantity $\Delta\theta$. The proposed Uniform Random Walk model has a probability density distributed according to $v_i \cdot (2 \cdot Rand - 1)$ i=1,2, with *Rand* uniformly distributed between 0 and 1. The parameters v_i are found empirically.

B. Observation model

We consider a pin-hole camera model and we assume that the intrinsic camera parameters are known. Let $\{F_i\}$ a set of scene circular Fiducials. In our approach we represent each fiducial F_i by its center $\mathbf{p}_i = (x_i, y_i, z_i)^t$ defined in the world reference frame (see figure 3). The center $\mathbf{p_i}$ defined in world coordinates frame can be expressed in the camera frame as well (see figure 3):

$$\mathbf{q}_i = R\mathbf{p}_i + T \tag{5}$$

where $R = (\mathbf{r}_1^t, \mathbf{r}_2^t, \mathbf{r}_3^t)^t$ and $T = (t_x, t_y, t_z)^t$ are a rotation matrix and a translation vector, respectively.

R and *T* describe the rigid body transformation from the world coordinate system to the camera coordinate system and are precisely the parameters associated with the camera pose problem.

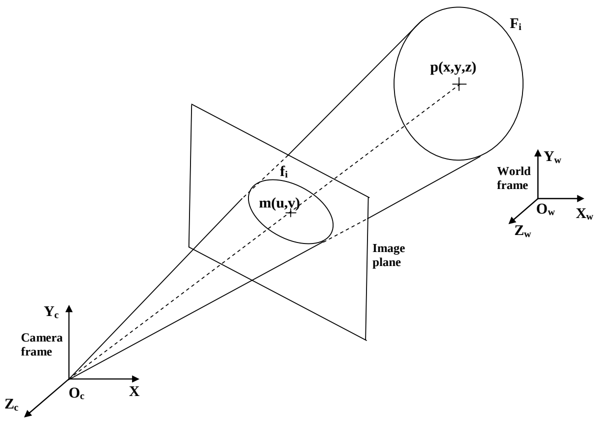


Figure 3. Perspective projection of the circular fiducials

Let $\mathbf{c_i}$ the image projection of the circular target F_i. The image point $\mathbf{m}_i = (u_i, v_i, 1)^t$ corresponds to the projection of center $\mathbf{p}_i$ on the normalized image plane. Using the camera pinhole model, the relationship between $\mathbf{m}_i$ and $\mathbf{p}_i$ is given by:

$$s \cdot \mathbf{m}_i = M_c \cdot [R \quad T] \cdot \mathbf{p}_i \tag{6}$$

where s is a scale factor and M_c contains the intrinsic parameters of the camera. Equation 6 describes the geometric process for image formation using an ideal pinhole camera model. The process is completely determined by choosing a perspective projection center and an image plane. The projection of a scene point is then obtained as the intersection of a line passing through this point and the center of projection with the image plane. Mathematically, this process is realized by two transformations: a rigid transformation (from 3D world coordinates frame to 3D camera coordinate frame using the camera pose parameters *R* and *T*) followed by a perspective transformation (from the camera coordinates frame to the image plane using the camera intrinsic parameters Mc).

Furthermore, let y_k be the observation at frame *k* and $y_{1:k}$ the set of observations from frame 1 to frame *k*. In our case, observations correspond to the image centers $\mathbf{m}_i$ of the detected fiducials. We assume also that we have a set of circular fiducials distributed through the environment. The positions of all fiducials are known in the same reference frame. Let $Z = \{p_1, p_2, \ldots, p_M\}$ be the fiducials center defined in the world coordinate frame.

We define, with the camera motion state X_k, a projection function $C(\mathbf{p}_i, X_k)$, which gives the projection of the 3D fiducial center $\mathbf{p}_i$ on the image plane :

$$C(\mathbf{p}_i, X_k) = M_c \cdot (R_k \mathbf{p}_i + T_k) \tag{7}$$

Equation 7 corresponds to the evolution in time of the projection equation 6.

C. The camera pose update

The solution of the particle filter is to obtain successive approximations to the posterior density $p(X_k | y_{1:k}, Z)$. This is generally provided in the form of weighted particles $\{(X_k^1, w_k^1), \ldots, (X_k^n, w_k^n)\}$, where X_k^n is a state space sample and the weights w_k^n are proportional to $p(y_k | X_k^n)$, such as:

$$\sum_{n=1}^{S} w_k^n = 1 \tag{8}$$

where *S* is the particles number.

The likelihood function $p(y_k | X_k, Z)$ describes the relationship between the state and the observation at a certain time step t. It influences both the re-sampling behaviour of the particle filter and the tracking performance. Lichtenauer et al [23] have shown that, for robust tracking, the true observation probability is not always the optimal likelihood function because the limited amount of particles used in practice must be concentrated as much as possible on the most important areas instead of approximating the complete posterior distribution. Hence, one straightforward but nonetheless good way for defining the likelihood function is to use an exponential function [24] [25]. In our case, the density function explicitly models uncertainty in the sensing process, and can be defined as a comparison between the observations y_k and the projections $C(\mathbf{z}_i, X_k)$ of the features set Z into the reference frame.

The likelihood $p(y_k | X_k, Z)$ is based on the closeness of the projected point $C(\mathbf{p}_i, X_k)$ to the observed image point $\mathbf{m}_i$. We use a function related to the number of the reference 3D fiducials centers whose projections into the image plane are within a given threshold of extracted image centers, i.e.

$$p(y_k|X_k,Z) = \exp\left\{\sum_{i=1}^{I}\sum_{j=1}^{M} d_p(\mathbf{m}_i, \mathbf{p}_j, X_k)\right\} \quad (9)$$

where I is the number of the 2D extracted points from the current image and M is the number of the 3D model points (a priori known). We used an exponential function to compute the particle likelihood because it allows to generate high weights for the good particles and small weights for the bad ones. Thus, only good particles will have more importance/contribution when computing the current camera pose.

$d_p(\mathbf{m}_i, \mathbf{p}_j, X_k)$ indicates whether the 3D point $\mathbf{p}_j$ is an inlier or outlier with respect to the observation $\mathbf{m}_i$ and the state X_k, i.e.

$$d_p(\mathbf{m}_i, \mathbf{p}_j, X_k) = \begin{cases} 1 & \text{if } \left\| \mathbf{m}_i - C(\mathbf{p}_j, X_k) \right\|^2 < \varepsilon_p \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where ε_p is a threshold which defines the tolerance in the point matching.

Comparing to classical model-based pose estimation techniques, our approach don't need to implement any 3D-2D matching module to discard outliers. This is done naturally by the particle filter equations as explained above. This property improves the robustness of our algorithm as we will see in section 5. Furthermore, using circular fiducials will ensures the extracted of image points even if they are occluded.

Finally, the weights w_k^n, for both point and line features, are given by:

$$w_k^n = \frac{p(y_k|X_k^n, Z)}{\sum_n^S p(y_k|X_k^n, Z)} \quad (11)$$

The output of the particle filter is given by:

$$\hat{X}_k = \sum_{n=1}^{S} w_k^n \cdot X_k^n \quad (12)$$

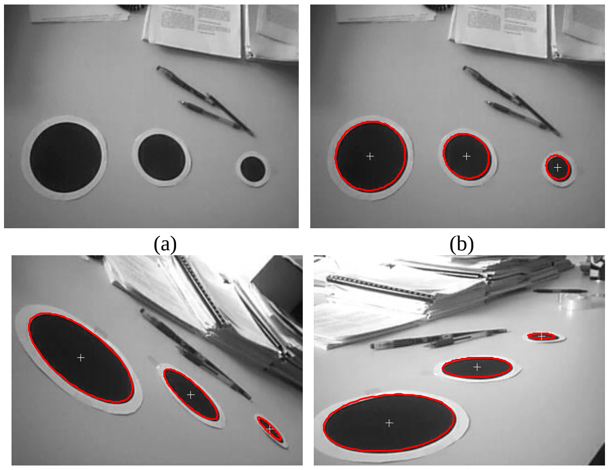
Finally, to avoid the degeneracy of the particle filter method, a resampling stage may be used to eliminate samples with low importance weights and multiply samples with high importance weights. In our work, we have implemented the selection scheme proposed by Gordon et al. [26].

V. Experimental Results

In order to study the robustness of our algorithm, we have done two experiments. The first experiment is intended to show the efficiency of our circular fiducial detection algorithm, whereas the second one focuses on the performance of our 3D camera tracker when used in real conditions of work. We have used multiple circular fiducials printed out on a single sheet of paper. Our fiducials are composed of the simplest possible primitives: a black dot against a white background. We have generated several test sequences to evaluate the performances of our tracking framework (processing time, occlusion,…). The sequences are recorded from a moving camera pointing towards the sheet paper. The frame rate is 25 frames/s (25 Hz) and there are 1000 frames in the over 40 second long sequence. The resolution of the video images is 320×240 pixels.

A. Fiduciasl detection

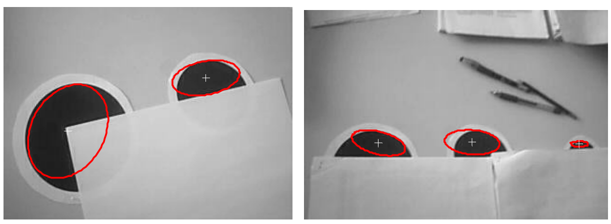
For this test we have used three circular fiducials of different sizes to simulate the camera scale changing (see figure 4-a). The obtained results for different camera view points and distances are shown in figures 4-(b-c-d). They demonstrates the robustness of our approach to the scale and orientation changing. Indeed, in figure (4-b) we can see that all the three fiducials are well extracted in spite of their different sizes. In addition, due to perspective distortion, a circular on the original pattern are generally transformed on ellipse when viewed at a sharp angle and projected into image plane. Figures (4-c) and (4-d) illustrate the efficiency of our fiducial detector in such situations.



(a) (b) (c) (d)

Figure 4. Fiducials detection

Furthermore, we have tested the robustness of our detection approach to the partial occlusion of the fiducials. Figures (5-a) and (5-b) show that the visible parts of the three fiducials are successfully detected although the fiducials are occluded more than 50%.



(a) (b)

Figure 5. Robustness to partial occlusion

Indeed, our fiducials detection algorithm is ellipse-specific, so that even incomplete data, due to noise and occlusion conditions, will always return an elliptical "blob", and thus an estimation of the fiducial position is always available.

Otherwise, our fiducials detection algorithm has been implemented on a desktop PC with four 3GHz CPU. The computational time more depends on the number of extracted fiducials. The current computation times for extracting only one fiducial is about 3 ms, which demonstrates the real time performance of our fiducial detection algorithm.

B. The 3D camera tracker

We have placed six circular fiducials at known locations in the workspace (see figure 6). During this experiment, the camera is hand-held and undergoes arbitrary 6DoF motions, viewing the calibrated fiducials from different positions and orientations. The camera pose is continually computed by the particle filter tracker.

Figure 6. RMSE analysis for PF and EKF Instrumented workspace

Furthermore, Frame 1 is used to calibrate the camera and also to initialize the pose estimation algorithm.

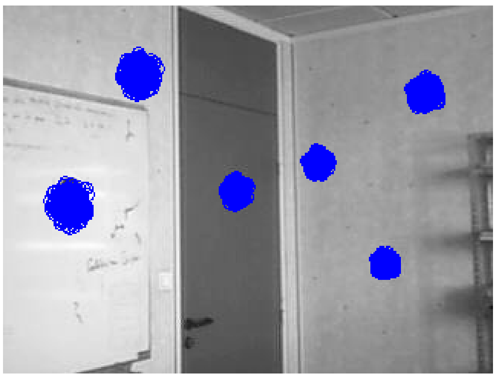


Figure 7. Particle filtering initialization

So, at $k=1$ the state vector X_1 is initialized with the camera pose parameters given by the calibration procedure. Figure 7, illustrates the initialization of the particle filtering. We have tuned the particle filter parameters v_1 and v_2 in such a way that the particles cloud well surround the targets. This will ensure a good initialization for the fiducilas tracking. In addition, we optimized the number of particles used in the estimation step in order to minimize the computational time while assuring high performances. A good compromise found is $N = 200$.

Furthermore, we used the image registration error (in pixels) in order to estimate the camera pose accuracy. It corresponds to the mean distance between the detected features (detected fiducial centres) in the image (inliers) and the re-projected 3D reference points. The root mean square error (RMSE) RMSE is given by:

$$RMSE = \frac{1}{M}\sum_{i=1}^{M}\left\|m_i - \hat{m}_i\right\|^2 \qquad (13)$$

where m_i are the projected image points, and $\hat{m}_i$ the observed ones.

Figure 8 shows the evolution of the registration error during the images sequence. The mean error is about 2.56 pixels with a standard deviation of 0.87 pixels. This demonstrates the good localization accuracy of our camera tracking algorithm.

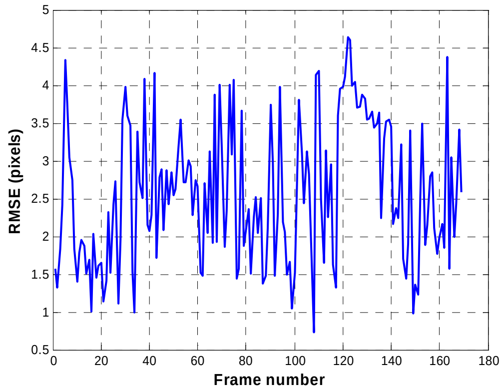


Figure 8. The registration error

When the camera pose is estimated we re-project on the current frame the 3D wireframe model of the workspace. We can see in figure 9 that the 3D model is well superimposed on the real world for different frames of the images sequence. This demonstrates the tracking success within the time.

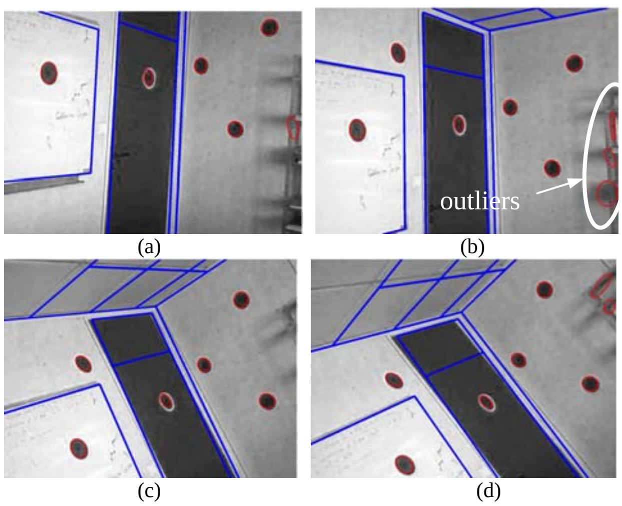


Figure 9. Camera pose estimation results

In addition, we noted that despite the detection of several outliers corresponding to erroneous circular fiducials extraction (see figure 9-b), the camera pose still well estimated. Indeed, when using particle filter, we don't need to implement a 3D-2D matching algorithm to discard outliers as it is the case in the classical pose estimation techniques. In our approach, this is done naturally by the equation of the filter: the erroneous matching generate particles with low weights and thus are naturally discarded, whereas the good matching generate particles with high weights which are using to compute the current camera pose. Figure 10 shows the number of elliptic blobs being detected during the sequence.

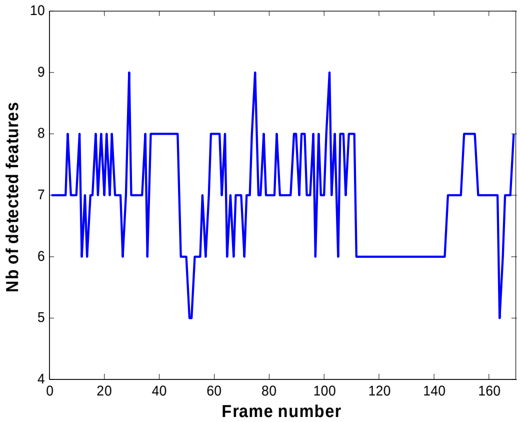


Figure 10. Detected elliptic features

Figure 11 shows the recovered egomotion for the hand – held camera with respect to the workspace coordinate system. Our approach handles the truly 6-DOF camera motion.

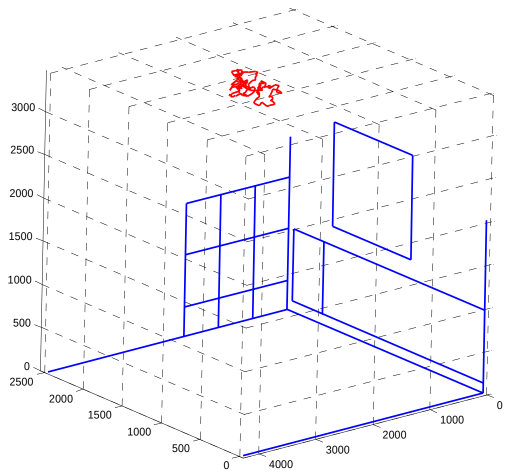


Figure 11. Recovered camera egomotion

Finally, figure 12 shows the whole computational time spend by our algorithm to estimate the camera pose. All computations were performed in Matlab. The average time is about 39 ms per frame. 46 % of this processing time is spent in circular fiducials extraction. The remaining time is spent in particle filter tracking and pose estimation. However, this time depends on several parameters in particular the number of particles used in the tracking process.

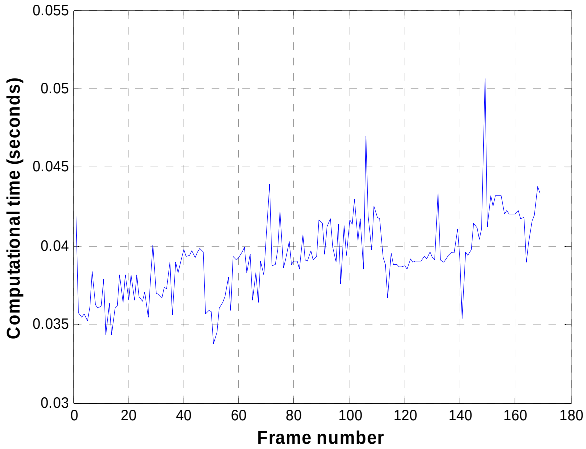


Figure 12. Computational times

VI. Conclusion

In this paper, we have presented a camera pose estimation method which combine circular fiducials detection and particle filtering. We have used a least square fitting method which is specific to ellipses to detect fiducials. This method is speed, accurate and extremely robust to noise and occlusion conditions. Furthermore, the simplicity of the particle filter implementation and the flexibility in defining the observation model are the major advantages for the camera pose tracking. The experimental results show that our algorithm can track the camera pose successfully and accurately under various conditions such as sever occlusion. Future research efforts will be made to increase the robustness of our method by using other kinds of features and by incorporating uncertainty information about the extracted features.

References

[1] N.J. Gordon, "A hybrid bootstrap filter for target tracking in clutter". *IEEE Trans on Aerospace and Electronic Systems*, vol. 33, pp. 353-358, 1997.

[2] F. Ababsa and M. Mallem, "Robust Circular Fiducials Tracking And Camera Pose Estimation Using Particle Filtering". *Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2007)*, pp. 1159-1164, October 2007.

[3] J. S. Kim, H. W. Kim and I. S. Kweon, "A Camera Calibration Method using Concentric Circles for Vision Applications". *5th Asian Conf. on Computer Vision (ACCV 2002)*, pp. 22-25.

[4] F. Abad, E. Camahort and R. Vivó, "Camera Calibration Using Two Concentric Circles". *Proc. of Inter. Conf. on Image Analysis and Recognition (ICIAR 2004)*, pp. 688-696, 2004.

[5] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags". *Proc. of Inter. Conf. on Designing augmented reality environments, (Dare 2000)*, pp. 1–10.

[6] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality

conferencing system". *Proc. of ACM/IEEE Int. Workshop on Augmented Reality (IWAR 1999)*, pp. 85–92.
[7] M. Fiala, "Artag, a fiducial marker system using digital techniques". *Proc. of IEEE Int. Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. Vol. 2, pp. 590–596.
[8] Y. Cho and U. Neumann, "Multi-Ring Color Fiducial Systems for Scalable Fiducial Tracking Augmented Reality". *Proc. of the Virtual Reality Annual International Symposium (VRAIS 1998)*, pp. 212.
[9] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker". *Proc. of ACM/IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR 2002)*, pp. 27–36.
[10] D.G. Lowe, "Fitting Parameterized Three-Dimensional Models to Images". *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 441-450, 1991.
[11] R.M. Haralick, "Pose Estimation From Corresponding Point Data". *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19:6, pp. 1426-1446, 1989.
[12] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code". *Int. J. of Computer Vision*, vol. 15:1-2, pp. 123–141, 1995.
[13] B. Jiang, and N. Neumann, "Extendible tracking by line auto-calibration". *Proc. of IEEE/ACM Int. Symp. on Augmented Reality (ISMAR 2001)*, pp. 97-103.
[14] Y. Yoon, A. Kosaka, J. B. Park and A. C. Kak, "A New Approach to the Use of Edge Extremities for Model-based Object Tracking". *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA 2005)*. pp. 1883-1889.
[15] V. Kyrki and D. Kragic, "Integration of Model-based and Model-free Cues for Visual Object Tracking in 3D". *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA 2005)*. pp. 1554-1560.
[16] D. Kragic and V. Kyrki, "Initialization and system modeling in 3-d pose tracking". *Proc. of the Int. Conf. on Pattern Recognition (ICPR 2006)*, pp. 643-646.
[17] F. Ababsa, M. Mallem and D. Roussel, "Comparison Between Particle Filter Approach and Kalman Filter-Based Technique For Head Tracking in Augmented Reality Systems". *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA2004)*, pp. 1021-1026.
[18] D. Marimon, Y. Maret, Y. Abdeljaoued and T. Ebrahimi, "Particle filter-based camera tracker fusing marker- and feature point-based cues". *Proc. of the IS&T/SPIE Electronic Imaging Conference on Visual Communications and Image Processing*, 2007.
[19] D. Marimon and T. Ebrahimi, "Combination of video-based camera trackers using a dynamically adapted particle filter". *Proc. of Int. Conf. on Computer Vision Theory and Applications (VISAPP'07)*, pp. 363-370.
[20] G. Klein and D. Murray, "Full-3D Edge Tracking with a Particle Filter". *Proc. of the British Machine Vision Conference (BMVC 2006)*, pp. 1119-1128.
[21] A. Fitzgibbon, M. Pilu, and R. B. Fisher, 'Direct Least Square Fitting of Ellipse'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21: 5, 1999, pp. 476-480.
[22] M. Pupilli, and A. Calway, Real-Time Camera Tracking Using a Particle Filter". *Proc. of the British Machine Vision Conference (BMVC 2005)*, pp. 519-528.
[23] J. Lichtenauer, M. Reinders and E. Hendriks, "Influence of the observation likelihood function on particle filtering performance in tracking applications". *Proc. of IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG 2004)*, pp. 767-772.
[24] L. Ronnie, M. Johansson and R. Suzic, "Particle filter-based information acquisition for robust plan recognition". *Proc. of Int. Conf. on Information Fusion*, July 2005.
[25] J. Czyz, "Object Detection in Video via Particle Filters", *Proc. of the Int. Conf. on Pattern Recognition (ICPR'06)*, pp. 820-823,.
[26] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "A novel approach to nonlinear/non Gaussian Bayesian state estimation". *IEE Proc. on Radar and Signal Processing*, 1993, vol. 140, pp. 107-113.

Fakhreddine Ababsa received the PhD degree in Robotics from the University of Evry Val d'Essonne in 2002. He joined the IBISC-Complex System Laboratory in December 1999. Since 2004, he has been assistant Professor in Electrical and Computer Sciences at the University of Evry Val d'Essonne. His research interests are augmented reality, pattern recognition, motion tracking, sensor fusion and image processing.

Malik Mallem received the PhD degree in Robotics and Computer Sciences from Paris XII University in 1990. His research deals with augmented reality (AR) applied to robotics and telerobotics at the IBISC–Complex System Laboratory. Since 1999, he has been Professor at Evry University and Head of the AR team which is involved in several projects in cooperation with French industries and research laboratories (http://lsc.univ-evry.fr/techno/EVRA/index.html).