Hybrid Camera Pose Estimation Combining Square Fiducials Localization Technique and Orthogonal Iteration Algorithm

Jean-Yves Didier, Fakhr-Eddine Ababsa, Malik Mallem

International Journal of Image and Graphics

Hybrid camera pose estimation combining square fiducials localisation technique and orthogonal iteration algorithm

Jean-Yves Didier
didier@iup.univ-evry.fr

Fakhr-eddine Ababsa
ababsa@iup.univ-evry.fr

Malik Mallem
mallem@iup.univ-evry.fr
Laboratoire IBISC, Université d'Évry Val d'Essonne, 40 rue du Pelvoux
CE 1455 Courcouronnes, 91020 Evry Cedex, FRANCE

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Camera pose estimation from video images is a fundamental problem in machine vision and Augmented Reality (AR) systems. Most developed solutions are either linear for both n points and n lines, or iterative depending on nonlinear optimization of some geometric constraints. In this paper, we first survey several existing methods and compare their performances in an AR context. Then, we present a new linear algorithm which is based on square fiducials localisation technique to give a closed-form solution to the pose estimation problem, free of any initialization. We propose also an hybrid technique which combines an iterative method, in fact the orthogonal iteration (OI) algorithm, with our own closed form solution. An evaluation of the methods has shown that this hybrid pose estimation technique is accurate and robust. Numerical experiments from real data are given comparing the performances of our hybrid method with several iterative techniques, and demonstrating the efficiency of our approach.

Keywords: Augmented reality, pose estimation, fiducial localization, orthogonal iteration

1. Introduction

Camera pose estimation is the problem of determining the position and orientation of an internally calibrated camera from known 3D reference points and their images. It is essential to the so-called registration problem in an Augmented Reality context. Indeed, the objects in the real and virtual world must be properly aligned with respect to each other, which requires knowing the camera's pose. Accurate estimation of the 3D pose data will absolutely affect the accuracy and visual performance of virtual objects in the AR space.

In this paper we propose efficient real-time solutions to the camera pose estimation problem that can handle large camera displacements. While commercial products, are already available for offline camera pose estimation, efficient online registration remains an open issue because it must be fast and reliable. Many of the real-time algorithms described in the literature still lack robustness, tend to drift, and are prone to jitter that makes them unsuitable for applications such as Augmented Reality[3].

To efficiently solve the camera pose estimation problem, we have developed two approaches. The first one is an analytical solution from 4 points, it is based on square fiducials localization technique and returns a unique solution, free of any initialization. The second developed solution is an hybrid approach that combines the orthogonal iteration (OI) algorithm [19] with our analytical pose estimation technique. It is a real time extension of some of our previous works using OI for camera pose recovery using fiducials [1]. As we will see, the performances of the OI algorithm are heavily affected by the initialization process, so it is very important that the initialization is close to the optimum. The original OI algorithm uses a week-perspective approximation to initialize the rotation matrix, so we propose to use our analytical solution to initialize the OI algorithm. This combination results in a system that does not suffer from any of the above difficulties and can deal with real-time aspect. Experimental results demonstrate that our hybrid algorithm is extremely efficient and converges in few iterations. It outperforms the original OI algorithm in terms of convergence and accuracy.

The remainder of this paper is organized as follows. First, we will study related works in camera pose estimation algorithms as well as augmented reality systems using fiducial detection. Then, section 3 is devoted to the formulation of the camera pose estimation problem and to the original OI algorithm review. Section 4 describes the developed square fiducials localization technique whereas section 5 will introduce an hybrid method composed of OI initialized using the algorithm detailed in the previous section. Experimental results are then presented in section 6, detailed performance analysis are given to compare our method to existing methods. Finally, section 7 provides conclusions and suggests some directions of our future works.

2. Related work

The camera pose computation is based on the extraction of geometric primitives which allow to match the 2D points (extracted from images) and the 3D points (known on the object). To deal with this problem, many approaches have been developed these last years, they can be subdivided into two main categories: 1) analytical methods based on a low number of points and/or lines, and 2) optimization methods based on minimizing an error criteria. Finally, we will survey some augmented reality localization tools based on fiducials tracking.

2.1. *Analytical methods*

The analytical methods use a low number of points and have a finite set of solutions. Their complexity is generally low, and when implemented their computing time is short. The pose estimated by these methods is generally accurate. However, the pose computation is dependent on the extraction process of the image points to be matched with points of the 3D object. According to the quality of the acquired image and the processing carried out, these methods can produce less accurate results. Many analytical methods were proposed these twenty last years. Using 3 points, the problem generically has four possible solutions. Haralick et al.[12] review many old and new variants of the basic 3-points method and carefully examine their numerical stabilities due to different orders of substitution and elimination. Lastly, in 1992, Dementhon and Davis also proposed[5] a three points based pose estimation technique. They have shown that by using two perspective approximations: paraperspective and orthoperspective, 2D lookup tables can be built and can be used to reduce the number of runtime floating-point operations needed to compute pose estimates. If a unique solution is required, additional information must be given, a fourth point generally suffices. But there are certain degenerate cases for which no unique solution is possible. All these critical configurations for which multiple distinct or coinciding (unstable) solutions occur are known. In 1981, Fischler and Bolles propose several methods to resolve the pose estimation problem. In their paper on the RANSAC method[9] (RAndom SAmple Consensus), they have shown that, knowing the coordinates of a number of 3D points and their corresponding image points, it is possible to compute the pose of the camera using a geometric closed-form technique. They also described results on the conditions under which multiple solutions exist for various numbers of correspondences between image and target, particularly for the Perspective-4-Points (P4P) problem. They extended their solution to 4 points by taking subsets and using consistency checks to eliminate the multiplicity for most point configurations. Horaud et al.[13] developed a closed form solution on 4 points which avoids this reduction to a 3 points solution. These closed form methods can be applied to more points by taking subsets and finding common solutions to several polynomial systems, but the results are susceptible to noise and the solutions ignore much of the redundancy in the data. Hung et al.[15] have proposed in 1985 a method for fiducial pose estimation using 4 non-aligned and coplanar points. Quan and Lan[22] propose a family of linear methods that yield a unique solution to 4- and 5-points pose determination for generic reference points. They also extended their 5-points method to handle more than five points. Their method does not degenerate for coplanar configurations and even outperform the special linear algorithm for coplanar configurations in practice. Furthermore, methods for pose estimation using line segments instead of points as image features have also been developed. Dhome et al.[7] developed algebraic solutions for 3-lines algorithms. Only three edges of the object are used to estimate the pose. A polynomial of degree 8 is then resolved and the obtained solutions sorted

according to the validity of the edge's configuration. Liu, et al.[18] combined points and line segments into the same pose estimation procedure.

2.2. *Optimization methods*

As the analytical methods are quite dependent on the quality of the acquired image. The solution thus is to carry out the pose computation on a larger set of points. However, For more than four points, closed form solutions do not exist. So, it is necessary to formulate pose estimation as a nonlinear least-squares problem of a polynomial equation in the image observables and the pose parameters and to solve it by nonlinear optimization algorithms, most typically, the Gauss-Newton method. Iterative solutions are a subset of optimization methods and are based on minimizing the error in some nonlinear geometric constraints. The pose is computed first once, then an iterative algorithm progressively refines the estimate. These methods allow to obtain a high accuracy of the camera localization within its environment. However, each iteration carried out involves an additional cost in computing execution time of the pose estimation. So, it is necessary to take into account this constraint in order to optimize the iteration number and the operations carried out during each iteration. Kumar and Hanson[17] have developed an iterative algorithm based on constraints on image lines using an update step adapted from Horn's solution[14] of the relative orientation problem. Dementhon and Davis[6] initialize their iterative scheme (named POSIT) by relaxing the camera model to scaled orthographic. It uses at least four non coplanar points. Haralick et al.[11] introduced also an iterative pose estimation algorithm which simultaneously computes both object pose and the depths of the observed points. Lu et al.[19] reformulate the pose estimation problem as that of minimizing an object-space collinearity error. They combine a constraint on the world points, effectively incorporating depth, with an optimal update step in the iteration. These iterative approaches typically suffer from slow convergence for bad initialization, convergence to local minima and the requirement of a large number of points for stability.

Table 2.2 summarizes the properties of the various pose estimation techniques surveyed above, it gives some practical advice allowing to choose the appropriate approach in a given situation.

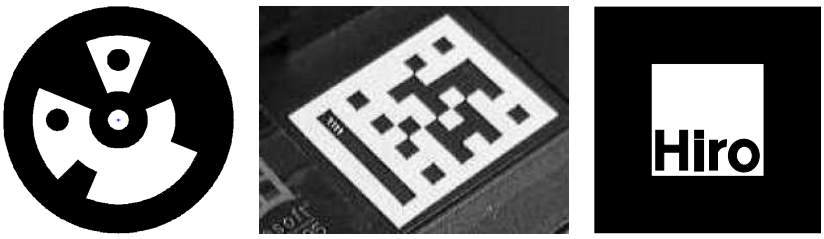
2.3. *The use of fiducials in Augmented Reality*

At the same time, in augmented reality projects, coded fiducials techniques were applied to determine feature points and estimate camera pose in real-time in a video sequence. Amongst those projects some are using circular coded fiducials like Cho et al. [4] or later the Intersense system [20]. Such fiducial shape implies that there are several of them stuck on an object to recover the camera pose. That's why the most current systems in augmented reality are using square fiducials. This shape gives 4 points per fiducials, which is enough to compute the camera pose relative to the fiducial even if there is only one of them. The number of

Table 1. Summary of camera pose estimation techniques.

Techniques	Rely on	Failure mode	Initialization	Accuracy
Haralick et al.[12] DeMenthon and Davis[5]	Linear - 3 points	Noisy data + occlusion	No	Can jitter
Horaud et al.[13] Quan and Lan [22]	Linear - 4 and 5 points	Noisy data + occlusion	No	Can jitter
Fischler and Bolles[9]	Linear - 4 points and more	Robust to noise and occlusion	No	Accurate
Dhome et al.[7]	Linear - 3 edges	Noisy data + occlusion	No	Can jitter
Kumar and Hanson[17]	Iterative - 3 lines	Robust to noise and occlusion	Yes	Highly accurate
Dementhon and Davis[6]	Iterative - 4 non coplanar points	Fast motion + noisy data	No	Accurate
Haralick et al.[11] Lu et al.[19]	Iterative - 3 points	Fast motion + robust to noise	Yes	Highly accurate

systems using such fiducials have increased in the past ten years as well as interest is growing in using them. We can cite the Matrix system [23] which became later the Cybercode system [24]. More and more augmented reality projects are using the ARToolkit library [16] [10], which is working with the same kind of principle and which is available freely to public. To these systems, we must add some that were developed by several laboratories as we can conclude from reading Zhang [25] comparative study of 4 different systems. Recently some work has been performed to increase the robustness of fiducial recognition and code extraction [21] [8]. Some of these fiducials are shown in figure 1.



(a) InterSense (b) Cybercode (c) ARToolkit

Fig. 1. Some examples of fiducials frequently used in augmented reality systems

This illustrates the potential interest of fiducials in augmented reality applications. We will now study in detail the pose estimation problem related to the extraction of feature points from ARToolkit-like markers. First, we will study a

generic iterative algorithm which the OI. We will then expose an analytic pose estimation method based on particular geometrical constraints of the square fiducials and then study a variation combining the two previous methods.

3. Formulation of the pose estimation problem

Given a calibrated camera and correspondences between 3D reference points and 2D points found in the image, the goal in pose determination is to find the rotation and translation matrices which map the world coordinate system to the camera coordinate system. In this paper we assume a perspective projection model. Let $\mathbf{p}_i = (x_i, y_i, z_i)^t, i = 1, ..., n, n \geq 3$ a set of 3D non-collinear reference points defined in an object-centered reference frame, the corresponding camera-space coordinates $\mathbf{q}_i = (x_i^{'}, y_i^{'}, z_i^{'})$ are given by :

$$\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t} \tag{1}$$

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

Let the image point $\mathbf{v}_i = (u_i, v_i, 1)^t$ be the projection of $\mathbf{p}_i$ on the normalized plane. Using the camera pinhole model, the relationship between $\mathbf{v}_i$ and $\mathbf{p}_i$ is given by :

$$u_i = \frac{\mathbf{r}_1^t \mathbf{p}_i + t_x}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \qquad v_i = \frac{\mathbf{r}_2^t \mathbf{p}_i + t_y}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \tag{2}$$

or

$$\mathbf{v}_i = \frac{1}{\mathbf{r}_3^t \mathbf{p}_i + t_z}(R\mathbf{p}_i + \mathbf{t}) \tag{3}$$

which is known as the *collinearity equation*. This equation is used in a different manner according to the desired approach to be implemented.

3.1. *For linear solutions*

Each pair of correspondences $\mathbf{p}_i \rightarrow \mathbf{v}_i$ and $\mathbf{p}_j \rightarrow \mathbf{v}_j$ gives a constraint on the unknown camera-point distances $d_i = \|\mathbf{p}_i - \mathbf{c}\|$ (cf. Fig. 1) :

$$d_{ij}{}^2 = d_i{}^2 + d_j{}^2 - 2d_i d_j cos\theta_{ij} \tag{4}$$

where $d_{ij} = \left\|\mathbf{p}_i - \mathbf{p}_j\right\|$ is the known inter-point distance between the i-th and j-th reference points and θ_{ij} is the 3D viewing angle subtended at the camera center by the i-th and j-th points. The cosines of this angle is directly computed from the image points and the calibration matrix of the internal parameters of the camera[22]. Using the points geometric constraints for 3 or 4 points allows to obtain a polynomial system for the unknown distances d_i. Linear algebra is often used to resolve this system in closed form[2,22]. The recovered camera-point distances d_i are used to estimate the coordinates of the 3D reference points in a camera-centered 3D frame. To find the camera pose, the rigid 3D motion that best aligns these points with their known world-frame coordinates is then estimated.

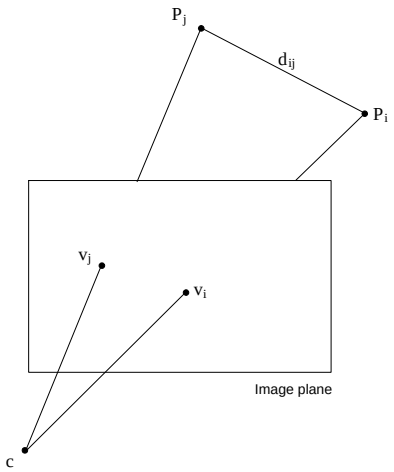


Fig. 2. Geometry of camera pose from points correspondences

3.2. *For solutions using optimization techniques*

The pose estimation problem is to develop an algorithm for finding the rigid transform $(R, \mathbf{t})$ that minimizes some form of accumulation of the errors, generally the summation of squared errors of the collinearity equations (see Fig. 2). It is formulated as the problem of optimizing the following objective function :

$$\sum_{i=1}^{n} \left[\left(\hat{u}_i - \frac{\mathbf{r}_1^t \mathbf{p}_i + t_x}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \right)^2 + \left(\hat{v}_i - \frac{\mathbf{r}_2^t \mathbf{p}_i + t_y}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \right)^2 \right] \tag{5}$$

Given observed image points $\hat{\mathbf{v}}_i(\hat{u}_i, \hat{v}_i, 1)^t$. The minimization is over image-space collinearity. To resolve this problem, two commonly optimization algorithms are used: the Gauss-Newton method and the Levenberg-Marquardt method.

4. Analytical square fiducials pose estimation

The aim is to recover the camera pose relative to a square fiducial. We will use the geometry constraints of this marker to compute its localization.

In this section, we will assume that the camera is calibrated (its intrinsic parameters are known). At the same time, we assume that the fiducial is already detected and the matching of 2D/3D points is already done.

First we will introduce some notations, then detail our algorithm and describe how we will be able to use it with orthogonal iteration.

4.1. *Notations*

We will use three different frame coordinates. The first one is the world space coordinates. They're computed relative to the fiducial. The center of the fiducial is the

origin of world space coordinates. The second one is the camera space coordinates. The last one is the image space coordinates. The coordinates of a 3D point M are written (X, Y, Z) in world coordinates, (x, y, z) in camera space and (u, v) are the coordinates of the corresponding point in the image plane.

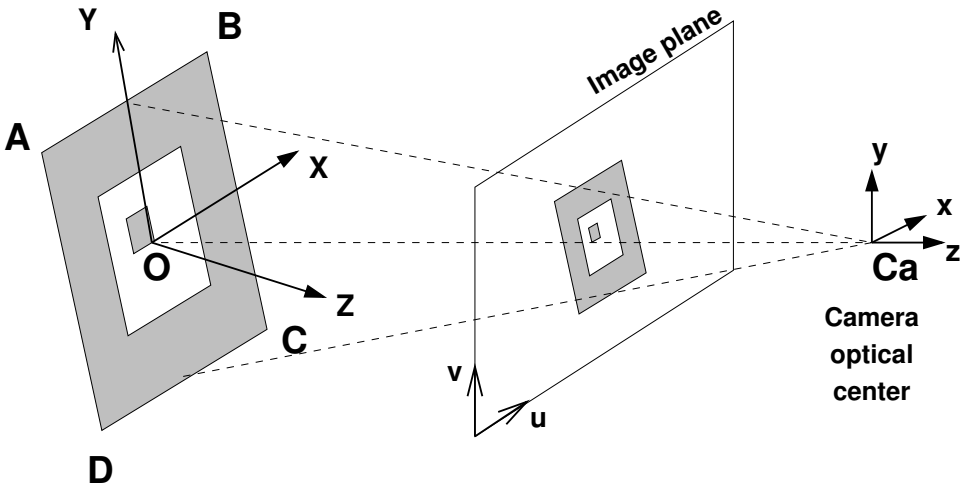


Fig. 3. Notations

The four corners of the fiducials are called A, B, C and D as shown in Fig. 3.

4.1.1. *Pin-hole camera model*

The pin-hole camera model is a widely spread model for camera calibration. In our case, we will assume that the optical distortion are already corrected. In its simplest formulation, the pin-hole model, when we're writing the relations in camera space are linking an image point m with image coordinates (u, v) to a 3D point M of coordinates (X, Y, Z) in world space by the following relations :

$$X = \frac{Z}{fk_u}(u - u_0) \qquad Y = \frac{Z}{fk_v}(v - v_0) \tag{6}$$

In our method, we will first apply an arbitrary depth estimation of our points A, B, C and D in camera space using geometrical constraints of our fiducial. Then we will compute the real depth using metric constraints.

4.1.2. *Applying geometrical constraints*

Since our fiducial has a square geometry, we can write the following property :

$$\overrightarrow{AB} = \overrightarrow{DC} \Leftrightarrow \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \\ Z_B - Z_A \end{bmatrix} = \begin{bmatrix} X_C - X_D \\ Y_C - Y_D \\ Z_C - Z_D \end{bmatrix}$$

We can multiply the two first lines of the equation by f (the camera focal length parameter) and apply 6 :

$$\begin{bmatrix} \frac{Z_B}{k_u}(u_B - u_0) - \frac{Z_A}{k_u}(u_A - u_0) \\ \frac{Z_B}{k_v}(v_B - v_0) - \frac{Z_A}{k_v}(v_A - v_0) \\ Z_B - Z_A \end{bmatrix} = \begin{bmatrix} \frac{Z_C}{k_u}(u_C - u_0) - \frac{Z_D}{k_u}(u_D - u_0) \\ \frac{Z_C}{k_v}(v_C - v_0) - \frac{Z_D}{k_v}(v_D - v_0) \\ Z_C - Z_D \end{bmatrix} \tag{7}$$

If we look at the first line, we notice we can simplify it and rewrite it like this :

$$\begin{aligned} u_B Z_B - u_A Z_A - u_0(Z_B - Z_A) = \\ u_C Z_C - u_D Z_D - u_0(Z_C - Z_D) \end{aligned} \tag{8}$$

Since $Z_B - Z_A = Z_C - Z_D$, we can simplify again this line. We can do the same for the second line and then we obtain :

$$\begin{bmatrix} u_B Z_B - u_A Z_A \\ v_B Z_B - v_A Z_A \\ Z_B - Z_A \end{bmatrix} = \begin{bmatrix} u_C Z_C - u_D Z_D \\ v_C Z_C - vD Z_D \\ Z_C - Z_D \end{bmatrix} \tag{9}$$

4.1.3. *Arbitrary relative depth estimation*

To differentiate the arbitrary depth to the real ones, we will call them Z'_A, Z'_B, Z'_C and Z'_D. In the first move, we will arbitrarily set Z'_A to 1, so that we will be able to compute Z'_B, Z'_C et Z'_D, we will then have to solve the following system :

$$\begin{pmatrix} u_B & -u_C & u_D \\ v_B & -v_C & v_D \\ -1 & 1 & -1 \end{pmatrix} \begin{bmatrix} Z'_B \\ Z'_C \\ Z'_D \end{bmatrix} = \begin{bmatrix} v_A \\ u_A \\ -1 \end{bmatrix} \tag{10}$$

We will then obtain :

$$\begin{aligned} Z'_B &= \frac{1}{\delta}[u_A(v_C - v_D) + v_A(u_D - u_C) - (u_C v_D - u_D v_C)] \\ Z'_C &= \frac{1}{\delta}[u_A(v_B - v_D) + v_A(u_D - u_B) + (u_D v_B - u_B v_D)] \\ Z'_D &= \frac{1}{\delta}[u_A(v_B - v_C) + v_A(u_C - u_B) - (u_B v_C - u_C v_B)] \\ \delta &= (u_C v_D - v_C u_D) + (u_D v_B - u_B v_D) + (u_B v_C - u_C v_B) \end{aligned}$$

4.1.4. *Real depth estimation*

The arbitrary depths we obtained are related to the real depth by a scale factor, therefore we note r_1 as the ratio $Z_A/Z_C = Z'_A/Z'_C$ and r_2 the ratio $Z_B/Z_D = Z'_B/Z'_D$. Since we can recover the arbitrary depth, we know the values of the ratios.

We will now express some equations related to r_1. The same principle could be applied to r_2. Using equations 6 we can write :

$$\begin{array}{ll} X_A = \frac{Z_A}{\alpha_u}(u_A - u_0) & Y_A = \frac{Z_A}{\alpha_v}(v_A - v_0) \\ X_C = \frac{Z_C}{\alpha_u}(u_C - u_0) & Y_C = \frac{Z_C}{\alpha_v}(v_C - v_0) \end{array} \tag{11}$$

Where $\alpha_u = fk_u$ and $\alpha_v = fk_v$. We know the value of $\|\overrightarrow{AC}\|$ since the metric constraints of our fiducial is known. Using this constraint and the value of r_1, we can express the value of Z_A :

$$Z_A = \frac{-\|\overrightarrow{AC}\|}{\sqrt{(r_1 - 1)^2 + (f(r_1))^2 + (g(r_1))^2}} \tag{12}$$

with

$$f(r_1) = \frac{r_1(u_C - u_0) - (u_A - u_0)}{\alpha_u} \tag{13}$$

$$g(r_1) = \frac{r_1(v_C - v_0) - (v_A - v_0)}{\alpha_v} \tag{14}$$

Once Z_A is computed, we can obtain Z_C using r_1. We can perform the same process to obtain Z_B and Z_D values. By using 11, we can determine the missing coordinates of points A, B, C and D.

4.1.5. *Pose estimation*

When the real depth are known, we can determine the translation and the rotation associated to our fiducial in camera space. The fiducial origin is placed in its center which coordinates are computed using those of A, B, C and D. The fiducial's center coordinates are also the translation between the camera and the fiducial.

We can then compute a rotation matrix (we will note it $R_{3\times3}$) its lines will be noted r_{1*}, r_{2*}, r_{3*} and will be equal to the following :

$$r_{1*} = \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \quad r_{2*} = \frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} \quad r_{3*} = r_{1*} \wedge r_{2*} \tag{15}$$

The camera pose is obtained by inverting the found transformation. Once the pose is known, we can combine this algorithm with OI by feeding the initialization step using the newly computed pose. We will now see how it benefits to the OI algorithm.

5. Hybrid Solution for pose estimation

In this section we propose an hybrid approach for camera pose estimation that combines the orthogonal iteration (OI) algorithm[19] with our analytical pose estimation technique described above. We propose to use the (OI) for its accuracy, global convergence and rapidity. Another strong point is it is taking into account the structure of the rotation parameter which is not achieved when using standard optimization method such as Levenberg-Marquart or Gauss-Newton.

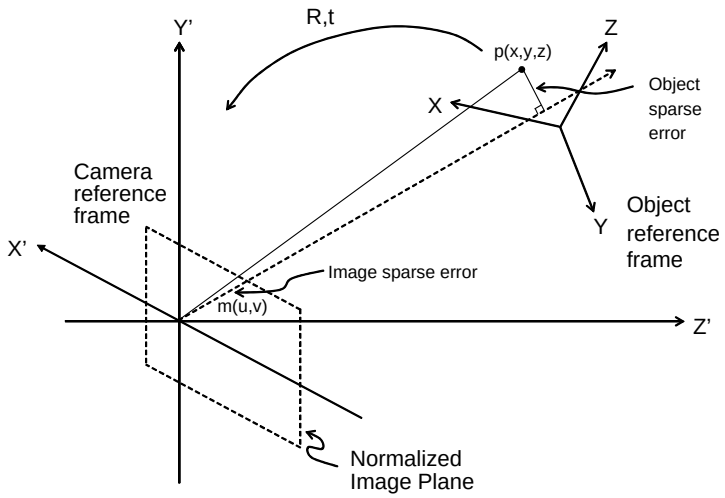


Fig. 4. The reference frames

5.1. *The OI algorithm overview*

The starting point for the algorithm is to state the pose estimation problem using the following object-space collinearity error vector (cf. Fig. 4) :

$$\mathbf{e}_i = \left(I - \hat{V}_i\right)(R\mathbf{p}_i + \mathbf{t}) \tag{16}$$

where $\hat{V}_i$ is the observed line-of-sight projection matrix defined as :

$$\hat{V}_i = \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t \hat{\mathbf{v}}_i} \tag{17}$$

The algorithm minimize the sum of the squared error

$$E(R,t) = \sum_{i=1}^{n} \|e_i\|^2 = \sum_{i=1}^{n} \left\| \left(I - \hat{V}_i\right)(R\mathbf{p}_i + \mathbf{t}) \right\|^2 \tag{18}$$

over R and $\mathbf{t}$. Given a fixed rotation R, the optimal value of $\mathbf{t}$ can be computed in closed form as :

$$\mathbf{t}(R) = \frac{1}{n}\left(I - \frac{1}{n}\sum_j \hat{V}_j\right)^{-1} \sum_j \left(\hat{V}_j - I\right) R\mathbf{p}_j \tag{19}$$

Thus, equation (18) can be written :

$$E(R) = \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t}(R) - \mathbf{q}_i(R)\|^2 \tag{20}$$

The matrix R can be computed iteratively as follows : First, assume that the kth estimate of R is $R^{(k)}$, $\mathbf{t}^{(k)} = \mathbf{t}\left(R^{(k)}\right)$, and $\mathbf{q}_i^{(k)} = R^{(k)}\mathbf{p}_i + \mathbf{t}^{(k)}$. The next estimate,

$R^{(k+1)}$, is determined by solving the following classical absolute orientation problem [14] :

$$R^{(k+1)} = \arg\min_R \sum_{i=1}^{n} \left\| R\mathbf{p}_i + \mathbf{t} - \hat{V}_i \mathbf{q}_i^{(k)} \right\|^2 \tag{21}$$

The next estimate of translation is then computed using (19).

$$\mathbf{t}^{(k+1)} = \mathbf{t}\left(R^{(k+1)}\right) \tag{22}$$

The process is repeated. A solution R^* to the pose estimation problem using the (OI) algorithm is defined to be a fixed point to (21), that is, R^* satisfies :

$$R^* = \arg\min_R \sum_{i=1}^{n} \left\| R\mathbf{p}_i + \mathbf{t} - \hat{V}_i \left(R^*\mathbf{p}_i + \mathbf{t}(R^*)\right) \right\|^2 \tag{23}$$

5.2. *Initialization of the OI algorithm*

The OI algorithm is initiated using a weak perspective approximation. Weak perspective assumes that the object points lie in a plane parallel to the image plane passing through the origin of the object frame [13]. In this case, the image points $\mathbf{v}_i$ are treated as the first hypothesized scene points. This leads to an absolute orientation problem between the set of 3D reference points $\mathbf{p}_i$ and the set of image points $\mathbf{v}_i$ considered as coplanar 3D points. This initial absolute orientation problem allows to compute the initial rotation matrix $R^{(0)}$ and thus to start the (OI) algorithm.

The performances of the OI algorithm are heavily affected by the initialization process, so it is very important that the initialization is close to the optimum. We propose to use our square fiducials localization method to compute a good initial guess better than given by the weak perspective approximation. This will considerably improve the pose solution accuracy and computing time as demonstrated in section 6.

6. Experimental results

In this section, we will compare the results given by OI method, our pose estimation method for square fiducials, the hybrid method using OI (we will call it later Customized OI or C-OI) and initialized using our analytical method and a ground truth : the least mean squares (LMS) algorithm for pose estimation. The three first methods are using the four corners of a detected fiducial (Fig. 5) whereas the LMS needs more points. To use this last one, we had to detect internal corners of the fiducial.

We will compare the results given by the three methods according several criteria :

- Computation time,

Image size		736×571	
Projection parameters		**Distortion parameters**	
Scale factors		Radial distortion coefficients	
α_u	706.1	k_1	-0.2279
α_v	731.1	k_2	0.1479
Optical center projection		Tangential distortion coefficients	
u_0	388.0	p_1	-0.0007985
v_0	269.6	p_2	0.0006245

Table 2. Intrinsic parameters of the Sony XC-555P used in experiments.

- Reconstruction error, i.e. the error in pixels between reprojection on the image of the fiducial model,
- Generalization error, i.e. the error in pixels when we reproject other objects on the images according to the pose computed using only one fiducial,
- Distance estimation error, i.e. the metric error we measure between the estimated pose and the actual one.

Each study is performed by computing the pose of only one fiducial, placing us in the worst case scenario of fiducials detection.

The test-bench we're using for experiments is composed of one computer PIII 1.1 GHz, a camera Sony XC-555P, a Matrox Meteor II framegrabber. The whole is orchestrated by a linux operating system. The camera is calibrated using Zhang method [26]. Intrinsic parameters are displayed in table 2.

6.1. *First series of results*

For the first series of experiments, we hold the camera by hand and we're freely moving around a set of two square fiducials of 6cm side-length. One is needed to compute the camera pose and the reconstruction error. During the pose estimation, the computation time is also determined. The projection of the second fiducial on the image is performed to determine the generalization error. It leads to the figures given in table 3. The best values for each criterion of comparison is highlighted in green whereas the worst are the red ones. The angular error given is extrapolated from the results given by generalization error study.

If we have a first look on these values, one can notice that our direct pose estimation algorithm is the fastest as expected since it is a specific method developed for square fiducials. At the same time, the LMS is quite slow. This is partly due to the additional corners detection. The reconstruction errors are roughly the same for each algorithm and looks like in figure 5 whereas it drastically changes for generalization error. The customized-OI method seems to have the best performances on generalization error contrary to the OI algorithm with its standard initialization which has one of the worst performances. This is mainly due to the fact that OI

Algorithm	Direct	OI	C-OI	LMS
Computation time				
Mean (μs)	20	153	111	13129
Standard deviation	3.10	12.7	17.6	3664
Reconstruction error				
Mean (*pixel*)	0.48	0.50	0.42	2.98
Standard deviation	0.38	0.44	0.30	1.83
Generalization error				
Mean (*pixel*)	9.52	16.6	8.90	14.1
Standard deviation	8.85	26.6	8.03	16.24
Angular error				
Mean angular error (degrees)	1.63	2.82	1.52	2.76
Number of computed poses				
Number of computed poses	6541	6362	6571	4410

Table 3. Raw results on the different experiments performed.

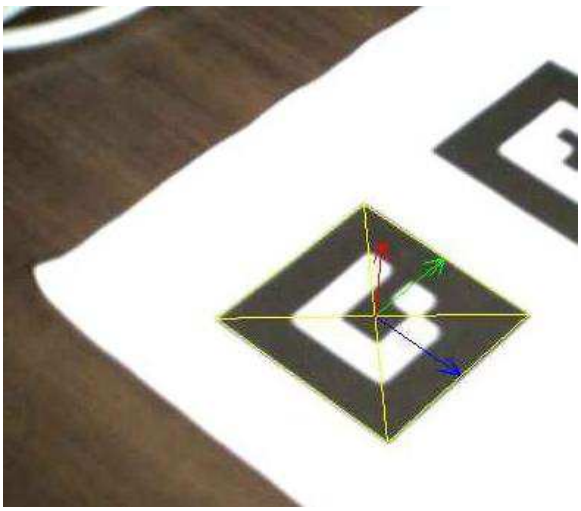


Fig. 5. Registration sample on a coded fiducial

is not designed for solving the camera pose problem using coplanar points as it is the case in our application. It is a degenerate case where OI can converge to two different solutions. If the initialization of the pose estimation is not close enough to the real solution then OI will be able to converge to the wrong solution, hence having poor results for generalization error.

If we plot the generalization error according to distance separating two fiducials,

we will have the curve shown in figure 6. It shows that, for some series, OI has performed a wrong pose estimation, resulting in a jittering curve. Mean performances of our analytical algorithm and customized-OI are basically not very different unless we separate the two fiducials by a distance of seven times the square side size of each fiducial. This is mainly due to the fact that our direct algorithm doesn't return a real rotation matrix but an approximation of it, whereas OI is intended to optimize such a matrix.

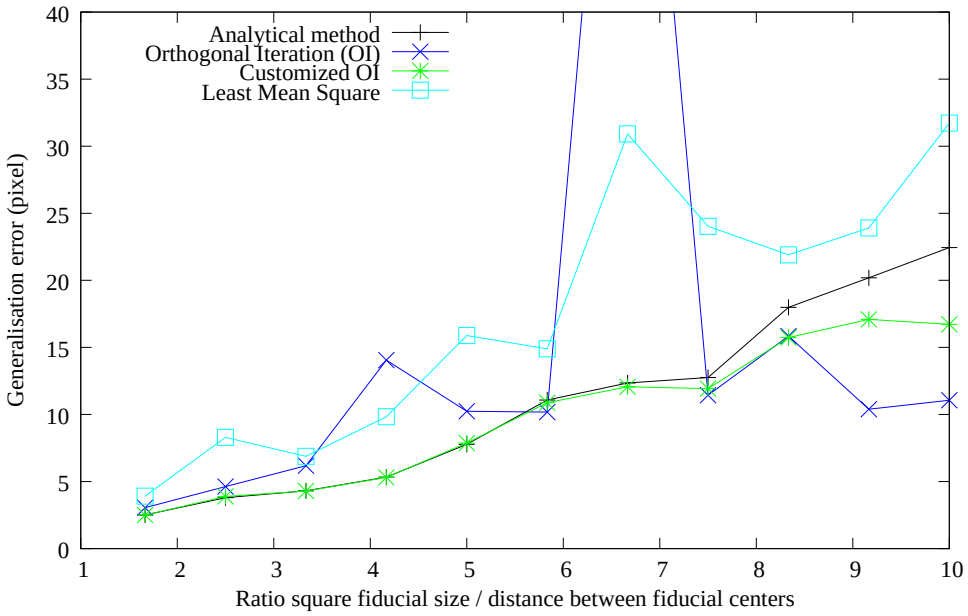


Fig. 6. Generalization error according to distance separating fiducials

6.2. *Distance estimation error*

For this experiment, we changed the square fiducial size (20cm side-length). The camera was mounted on a XY-axes robot realizing the setup from Fig. 7. The robot coordinate frame and the camera frame were calibrated. The robot positions taken from a given trajectory (Fig. 8 are then giving the actual camera pose we can compare with our pose estimation algorithms. Fig. 9 is a scatter plot of the pose estimations we obtained using the different methods. One can notice that the Direct and OI methods are having an error of distance estimation around two percents whereas the LMS method has a rapidly increasing distance error estimation.

Since the scatter plot is not easily readable, we separated data into ten classes (Fig. 9(c)) to elaborate a simplified graph (Fig. 9(b)). The middle of each bar is the mean value for the distance evaluation in a given class and the two extremities are

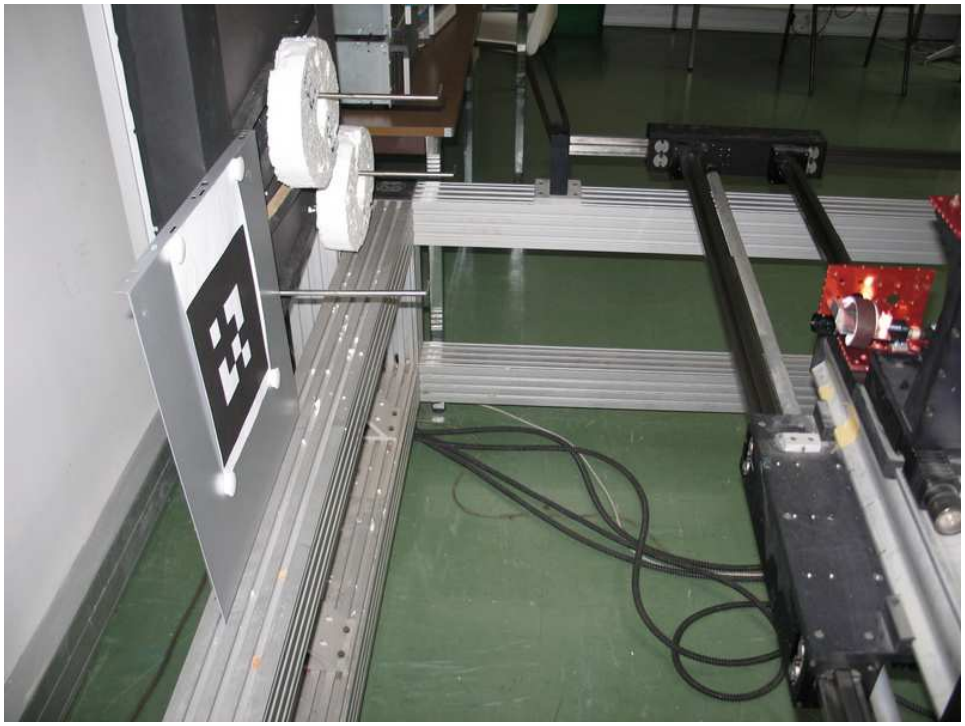


Fig. 7. Robot test-bench for distance estimation error

showing the standard deviation around the computed mean. It particularly shows that the pose estimations coming from our analytical method is more sensible to noise than customized-OI or OI itself.

6.3. *Effects of OI customization*

By studying the provided results, the customized OI has several strong points genuine OI and our direct analytical method do not have. First of all, C-OI is more stable than the two other algorithms, providing a more accurate pose estimation with an increase of the accuracy for objects far from the vertices of the fiducials used to estimate the camera pose.

The second one is that the C-OI is compensating some of the side effects of OI, that is to say the initialization problem as well as the computation time. The provided initialization is close enough to the pose estimation we want to reach so that the OI converge about 30 percents faster to the right solution.

However, even if the C-OI has some interesting features, the computation time cost to increase accuracy is five times bigger than the basic analytical method.

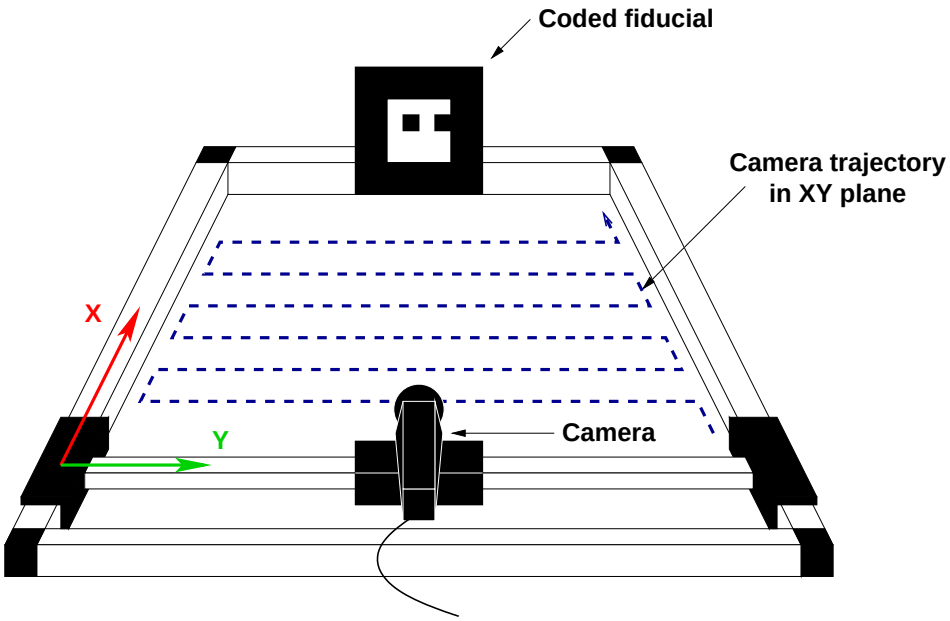


Fig. 8. Camera trajectory in the XY robot plane

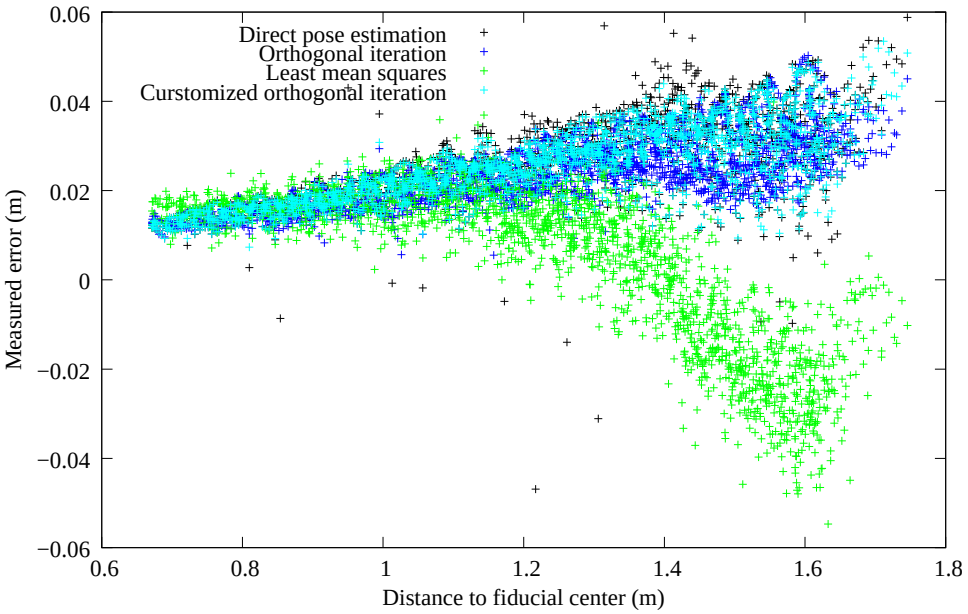
7. Conclusion

In this paper, we proposed the use of a square fiducials pose estimation to compute a good initial guess for the OI method. According to evaluations performed on 4 different algorithms, this customized-OI is presenting the following strong points :

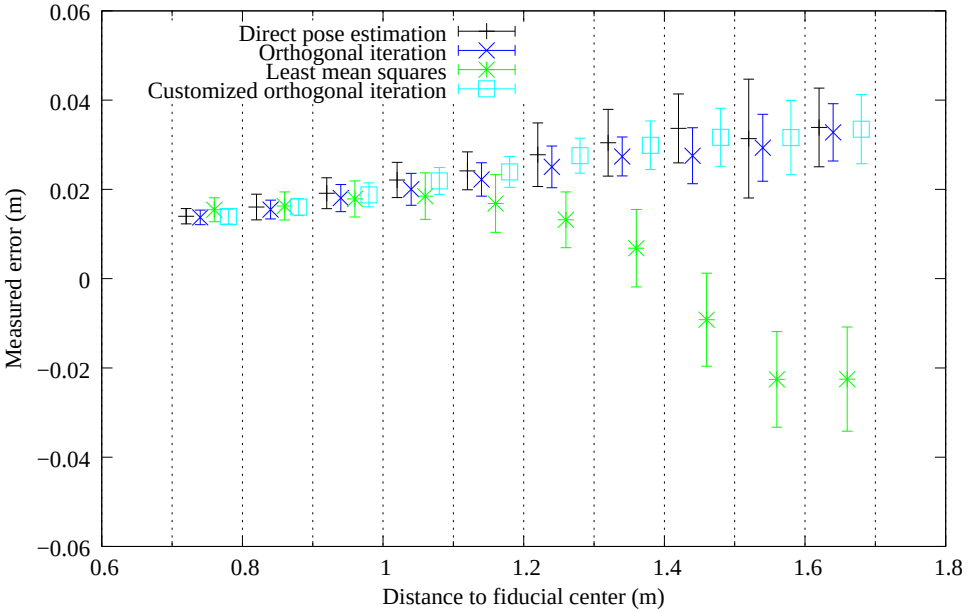
- the distance evaluation accuracy has less than two percents of relative error,
- the rotation estimation is accurate and optimized using the OI algorithm,
- its computation speed is quite a good compromise between speed and accuracy, the pose being computed in about 100 μs,
- this method is more stable than all compared method.

This study will help us to establish guidelines in using fiducials in augmented reality projects according to the requirements of applications in terms of tracking, robustness and accuracy in overlaying virtual images on real images as well as accuracy of localization. Some of them would answer to questions such as how many markers should we place, where and what size should they have ?

Further area of investigation will be towards the improvement of the robustness of our square fiducial localization algorithm, especially in case of occlusions of a part of the fiducial. We will also try to combine this method with markerless algorithms to bypass the drawbacks of both family of methods.



(a) Scatter plot of the measured distance error for each computed pose



(b) Means and standard deviations for 10 classes subdividing the point set

Classes	[0.7,0.8[	[0.8,0.9[	[0.9,1.0[	[1.0,1.1[	[1.1,1.2[	[1.2,1.3[
Population	123	143	157	184	206	229
Classes	[1.3,1.4[	[1.4,1.5[	[1.5,1.6[	[1.6,1.7[	Outliers	Total
Population	235	248	258	133	41	1957

(c) Classes and their population.

Fig. 9. Distance estimation error, 1957 pose estimations

References

1. F. Ababsa and M. Mallem. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. In *Proceedings of ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI2004)*, pages 431–435, 16-18 December 2004.
2. A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003.
3. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov/Dec 2001.
4. Y. Cho and U. Neumann. Multi-ring color fiducial systems for scalable fiducial tracking augmented reality. In *VRAIS '98: Proceedings of the Virtual Reality Annual International Symposium*, page 212, Washington, DC, USA, 1998. IEEE Computer Society.
5. D. F. DeMenthon and L. S. Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105, Novembre 1992.
6. D. F. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.
7. M. Dhome, M. Richetin, and J.-T. Lapreste. Determination of the attitude of 3d objects from a single perspective view. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(12):1265–1278, 1989.
8. M. Fiala. Artag, a fiducial marker system using digital techniques. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 590–596, Washington, DC, USA, 2005. IEEE Computer Society.
9. M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, Juin 1981.
10. K. H, B. M., P. I., I. K., and T. K. Virtual object manipulation on a table-top ar environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR 2000)*, pages 111–119, Munich, Germany, Oct. 2000.
11. R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446, 1989.
12. R. M. Haralick, C. Lee, K. Ottenberg, and M. Nlle. Analysis and solutions of the three point perspective pose estimation problem. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 592–598, Maui, Hawaii, 1991.
13. R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, March 1997.
14. B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonomal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, April 1988.
15. Y. Hung, P. Yeh, and D. Harwood. Passive ranging to known planar point sets. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 80–85, St.Louis, Missouri, 1985.
16. H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–92, Washington, DC, USA, 1999. IEEE Computer Society.

17. R. Kumar and A. R. Hanson. Robust methods for estimaging pose and a sensitivity analysis. *Computer Vision and Image Understanding*, 60(11):313–342, 1994.
18. Y. Liu, T. S. Huang, and O. D. Faugeras. Determination of camera location from 2d to 3d line and point. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.
19. C. P. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.
20. L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, pages 27–36, Washington, DC, USA, 2002. IEEE Computer Society.
21. C. Owen, X. Fan, and P. Middlin. What is the best fiducial? In *Augmented Reality Toolkit, The First IEEE International Workshop*. IEEE, 2002.
22. L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(7):774–780, July 1999.
23. J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *APCHI '98: Proceedings of the Third Asian Pacific Computer and Human Interaction*, pages 63–68, Washington, DC, USA, 1998. IEEE Computer Society.
24. J. Rekimoto and Y. Ayatsuka. Cybercode: designing augmented reality environments with visual tags. In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, pages 1–10. ACM Press, 2000.
25. X. Zhang, S. Fronz, and N. Navab. Visual marker detection and decoding in ar systems: A comparative study. In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 97, Washington, DC, USA, 2002. IEEE Computer Society.
26. Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision*, volume 1, page 666, Corfu, Greece, September, 20-25 1999.

Photo and Biography

Jean-Yves Didier recieved his M.S. degree in Virtual Reality and Computer Science from University of Evry Val d'Essonne, France, and his Ph.D. degree in robotics from the same university in 2002 and 2005 respectively.

Since 2005, he is a temporary teacher in a computer engineering school named Institut d'Informatique d'Entreprise (France), and a researcher at the IBISC Laboratory. His research works are focused on software architecture for rapid prototyping of augmented reality applications.

Fakhr-eddine Ababsa received a Ph.D. degree in Robotics from the University of Evry Val d'Essonne (France) in 2002. Since 2004, he is assistant professor in electrical and computer sciences at the University of Evry Val d'Essonne and researcher at the IBISC Laboratory (ex. Complex System Laboratory).

His current research are focused on robust estimation, motion tracking, human-machine interaction and sensor fusion with applications to scientific problems, in particular the development of real-time augmented reality systems.

Malik Mallem recieved a Ph.D. degree in robotics and computer science from Paris XII University.

Since, he has been working on Augmented Reality applied to robotics and telerobotics at Complex System Laboratory - CNRS FRE 2494, Evry, France which became IBISC (CNRS FRE 2873) in January 2006. Since 1999, he is professor at the University of Evry Val d'Essonne.