Pose Estimation for Planar Structures

Gilles Simon and Marie-Odile Berger
Loria - University of Nancy - Inria Lorraine
email:{gsimon, berger}@loria.fr
Loria, Campus Scientifique, BP 239
54506 Vandœuvre-les-Nancy cedex , FRANCE

Abstract

Tracking, or camera pose determination, is the main technical challenge in creating augmented reality. This paper describes an efficient and reliable method for real time camera tracking. This method is based on the knowledge of a piecewise planar structure in the scene and does require neither markers nor sensors to recover the viewpoint. Our system yields results comparable in accuracy with full structure-and-motion methods but with better reliability. Results are presented, demonstrating tracking both for indoors and outdoor urban scenes. Videos of augmented scenes are available at the URL http://www.loria.fr/~gsimon/Cga

1 Introduction

Augmented reality (AR) systems allow users to interact with real and computer generated objects by displaying 3D virtual objects registered in a user's natural environment. Applications of this powerful concept range from entertainment to military applications and include interior design, architectural design, computer-aided repair and learning systems, medicine systems or battlefield augmented reality systems. The interested reader may consult [16] which describes a large variety of current and near future applications of AR as well as [1] for an overview of recent advances in augmented reality.

All these interactive applications require that the augmented scene is continually updated as the user moves about the real scene. Hence, while there are several problems in building AR systems, one of the most basic challenge to overcome is the registration problem: the objects in the real and the virtual world must be properly aligned with respect to each other or the illusion that the two worlds coexist will be compromised. It is therefore essential to determine accurately the location and the optical properties of the camera.

In this paper, we address the registration problem for interactive AR applications. Such applications require real-time registration process. Though the registration problem has received a lot of attention in the computer vision community, the problem of real time registration is still far from a solved problem. Ideally, an AR system should work in all environments without the need to prepare the scene ahead of time and the user should walk anywhere he pleases. In the past, several AR systems have achieved accurate and fast tracking and registration, putting dots over objects and tracking the dots with a camera [9, 10]. Registration can also be achieved by identifying features in the scene for which real world coordinates can be carefully measured. However, such methods restrict the flexibility of the system. Hence, there is a need to investigate registration methods which work in unprepared environments and which reduce the need to know the geometry of the objects in the scene.

In the present paper we propose an efficient solution to real time camera tracking for scenes which contain planar structures. The type of scenes which can be considered with our method is large: this is commonly true of indoor environments where the ceiling or the ground plane are often visible. This is also often true for urban outdoor scenes because the façades of the buildings, the roads and the squares are often visible and can be used for registration. We show that our system is reliable and can be used for real time applications. Results are presented demonstrating real time camera tracking on indoor and outdoor scenes.

2 Background

The viewpoint parameters can be obtained using different kinds of sensors: mechanic or magnetic sensors, GPS, compass... However, the use of sensors has proven to be constraining in practice (restricted user displacements, perturbations from the environment...). By contrast, viewpoint computation using artificial vision does not require any instrument except the acquiring camera. Moreover, the augmentation re-

sults are generally more accurate than results obtained by using sensors (except for abrupt motions), as they are directly computed from features extracted from the images to be augmented.

In particular, movematching algorithms appear to offer significant possibilities for general, accurate registration [4, 7]. Such systems simultaneously estimate camera motion and 3D structure of the imaged scene, by tracking key-points along the sequence. These systems permit accurate registration and negligible jitter but are extremely time consuming. Besides the heavy calculation, these methods require a batch bundle adjustment in order to achieve a highly accurate registration. Furthermore, the world-coordinate system is arbitrarily chosen, generally aligned with the first camera. As a result, the insertion of virtual objects requires further registration of object- to world-coordinate systems. This additional step is manual and cumbersome unless some feature correspondences are made between the recovered 3D structure and the augmenting models. These drawbacks preclude the sequential implementation required in interactive applications.

Model-based techniques rely on the identification in the images of features from the object model. Hence, a direct correspondence between the 3D object-coordinate system and each image is set up. Examples include point features [10, 14], edges [5] or curves [12]. Artificial markers can help generate such features in the images [14, 10]. Other works extract them from the natural structure of the scene [5, 11, 15, 12]. Pose estimation techniques can then be used to estimate the camera position on each single image. This capability of treating each image independently makes such methods more appropriate for real time implementations. Another consequence of model-based tracking is the absence of drift.

However, it is commonly true that few features are available for registration. Moreover, noise in the image measurements hampers their accurate detection and consequently corrupts the estimated pose. As a result, the tracking suffers from high-frequency jitter. More importantly, such methods require significant manual intervention to construct the model.

Our approach combines both kinds of methods and lies between the model based approach and movematching techniques because we use 2D metric information rather than 3D one while using a planar model of the scene. In this paper, we consider piecewise planar environments, like indoor scenes showing textured walls, or outdoor scenes including planar structures such as buildings and/or flat ground. The planes are tracked in consecutive images using key-points. This work is an extension of [13] where the case of a single plane visible in the scene was addressed. It appears to be a good alternative for real-time AR because: 1. no marker is required ; 2. the matching of the key-points is constrained by the planar structure hypothesis and therefore is robust ; 3. no jitter sways out the augmenting virtual object ; 4. it performs in real-time, owing to the existence of a closed-form solution to the n-planes registration problem.

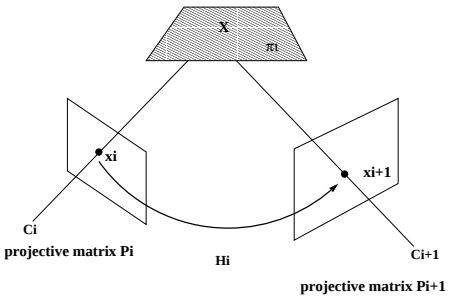

Figure 1: The homography induced by a plane.

The paper is organized as follows: section 3 deals with the 1-plane registration problem, section 4 extends it to the case where several planes are visible. An overview of the system is given in section 5, and experimental results are shown in section 6. We conclude by discussing some of our current research efforts.

3 Single-plane registration

A single-plane temporal registration system was described in [13]. In this section, we remind how to compute a 3×4 projection matrix P^i in image i, from P^{i-1} and a planar homography H^i between these images (fig. 1).

We consider the pinhole camera model, which associates a point $\mathbf{x}^i$ in image i to a point $\mathbf{X}$ in the scene:

$$\mathbf{x}^i = \mathsf{P}^i\mathbf{X} = \mathsf{K}\left[\ \mathsf{R}^i|\mathbf{t}^i\ \right]\mathbf{X}$$

The matrix K represents the internal calibration parameters of the camera which are supposed to be known. $\left[\ \mathsf{R}^i|\mathbf{t}^i\ \right]$ is the viewpoint matrix to be estimated.

Let us now restrict $\mathbf{X}$ to lie on a plane Π and suppose that we know the associated planar homography H^i between image i and image $i-1$ (e.g. thanks to at least 4 point correspondences between these images - see section 5). The following relation is valid for all points on plane Π:

$$\mathbf{x}^i \simeq \mathsf{H}^i\mathbf{x}^{i-1} \tag{1}$$

Let M be a transformation matrix between $\mathcal{R}_0$, a coordinate system where the equation of Π is $z = 0$, and the world-coordinate system. For all points $\mathbf{X}$ on plane Π, we have:

$$\mathbf{x}^i = \mathsf{P}^i\mathbf{X} = \mathsf{P}^i\mathsf{M}\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \left\langle \mathsf{P}^i\mathsf{M} \right\rangle \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix} \tag{2}$$

where $\langle \mathsf{A} \rangle$ denotes the matrix A deprived of its third column. $\langle \mathsf{P}^i\mathsf{M} \rangle$ is invertible unless plane Π goes through the origin of the camera.

As a result, combining equations (1) and (2) yields:

$$\langle \mathsf{P}^i\mathsf{M} \rangle \simeq \mathsf{H}^i \langle \mathsf{P}^{i-1}\mathsf{M} \rangle \quad (3)$$

Depriving $\mathsf{P}^i\mathsf{M}$ of its third column does not prevent from recovering the full viewpoint parameters. Indeed, knowing $\langle \mathsf{P}^i\mathsf{M} \rangle$ from equation (3), as well as the internal parameters K leads to:

$$\mathsf{K}^{-1} \langle \mathsf{P}^i\mathsf{M} \rangle \simeq \left[\begin{array}{ccc} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{array} \right]$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal vectors. The third column for the rotation matrix of the viewpoint is merely given by $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. In practice, the orthonormality conditions are never perfectly met, and renormalization must be applied ($\mathbf{r}_2 = \mathbf{r}_2/||\mathbf{r}_2||, \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2/||\mathbf{r}_1 \times \mathbf{r}_2||, \mathbf{r}_1 = \mathbf{r}_2 \times \mathbf{r}_3$).

Thus, if we know the camera pose in the first frame, we can compute the camera position for every frame of the sequence, using only the frame-to-frame homographies between consecutive images. As these homographies are directly computed from the images, we then have a sequential method to automatically compute the camera pose along the sequence.

4 Multi-plane registration

Our experiments proved that the accuracy of the single-plane registration method is not allays sufficient to obtain a good visual impression of the augmented scene (see section 6). That is the reason why we investigate the use of several visible planes for registration. Hence, we suppose that several planes are visible in the scene and that their equation is known in any frame. It must be noticed that indoor scenes as well as outdoor urban scenes often contain a lot of perpendicular planes. Then finding the equations of these planes is very easy and our n-plane method can be used for a large variety of sequences.

In this section, we suppose that $n > 1$ homographies H^i_p are known for n planes Π_p. From equation (3), we get for each plane:

$$\langle \mathsf{P}^i\mathsf{M}_p \rangle \simeq \mathsf{H}^i_p \langle \mathsf{P}^{i-1}\mathsf{M}_p \rangle . \quad (4)$$

The problem is to compute the camera viewpoint from this set of equations given the intrinsic camera parameters. As is, this problem is non linear and we have to resort to numerical iterative process to solve (4). Unfortunately, real time computation cannot be achieved through iterative minimization. So, we investigate here two other methods (LIN1, LIN2) which lead to a linear estimation of the viewpoint.

We first show how to estimate iteratively the viewpoint parameters. Then we present the first linear solution for the computation of P^i. Finally, we propose a linear estimator of the viewpoint, using an approximation of the rotation matrix.

4.1 Iterative computation of the viewpoint (ITER)

For each correspondence $\mathbf{x}_j \leftrightarrow \mathbf{x'}_j$ between image $i-1$ and image i, we have

$$\mathbf{x'}_j \simeq \mathsf{H}^i_p \mathbf{x}_j, \quad (5)$$

where p is the subscript of the plane containing the related 3D point $\mathbf{X}_j$. A residual r_j can hence be computed for each pair of points:

$$r_j = dist \left(\mathsf{H}^i_p \mathbf{x}_j, \mathbf{x'}_j \right),$$

where $dist$ is the Euclidean distance between two 2D points expressed in homogeneous coordinates. From (4), we get

$$\mathsf{H}^i_p \simeq \langle \mathsf{P}^i\mathsf{M}_p \rangle \langle \mathsf{P}^{i-1}\mathsf{M}_p \rangle^{-1}. \quad (6)$$

Consequently, the residuals expressively depend on the 6 parameters of the viewpoint in image i (translation $[t_x t_y t_z]^\top$ and Euler angles α, β, γ). These parameters can be recovered through a least squares minimization (e.g. Powell's optimization method) of the criterion:

$$\min_{t_x, t_y, t_z, \alpha, \beta, \gamma} \sum_j r_j^2$$

with P^{i-1} providing the initial parameters.

This method has proven to be stable and accurate in our experiments. However, the iterative process is relatively slow (typically 0.5 sec per frame in the conditions of the tests presented in section 6.1), and does not permit real time on common computers.

4.2 Linear computation of the projection (LIN1)

This section takes its inspiration from the reasoning used in [7] for the linear computation of a homography. From equations (5) and (6), and considering that $\langle \mathsf{P}^i\mathsf{M}_p \rangle = \mathsf{P}^i \langle \mathsf{M}_p \rangle$, we obtain for each correspondence $x_j \leftrightarrow x'_j$:

$$\mathbf{x'}_j \simeq \mathsf{P}^i \langle \mathsf{M}_p \rangle \langle \mathsf{P}^{i-1}\mathsf{M}_p \rangle^{-1} \mathbf{x}_j,$$

that is

$$\mathbf{x'}_j \simeq \left(\begin{array}{c} \mathbf{p}^{1\top}\mathsf{Q}_p\mathbf{x}_j \\ \mathbf{p}^{2\top}\mathsf{Q}_p\mathbf{x}_j \\ \mathbf{p}^{3\top}\mathsf{Q}_p\mathbf{x}_j \end{array} \right), \quad (7)$$

where $\mathbf{p}^{k\top}$ is the k^{th} row of matrix P^i, and $\mathsf{Q}_p = \langle \mathsf{M}_p \rangle \langle \mathsf{P}^{i-1}\mathsf{M}_p \rangle^{-1}$. Writing equations (7) in terms of cross products gives

$$\left(\begin{array}{c} y'_j\mathbf{p}^{3\top}\mathsf{Q}_p\mathbf{x}_j - w'_j\mathbf{p}^{2\top}\mathsf{Q}_p\mathbf{x}_j \\ w'_j\mathbf{p}^{1\top}\mathsf{Q}_p\mathbf{x}_j - x'_j\mathbf{p}^{3\top}\mathsf{Q}_p\mathbf{x}_j \\ x'_j\mathbf{p}^{2\top}\mathsf{Q}_p\mathbf{x}_j - y'_j\mathbf{p}^{1\top}\mathsf{Q}_p\mathbf{x}_j \end{array} \right) = \mathbf{0},$$

where $\mathbf{x'}_j = \begin{bmatrix} x'_j & y'_j & w'_j \end{bmatrix}^\top$. Finally, as we have $\mathbf{p}^{k\top}\mathsf{Q}_p\mathbf{x}_j = \mathbf{x}_j^\top\mathsf{Q}_p^\top\mathbf{p}^k$, we get the linear system of equations:

$$\begin{bmatrix} \mathbf{0}^\top & -w'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top & y'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top \\ w'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top & \mathbf{0}^\top & -x'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top \\ -y'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top & x'_j\mathbf{x}_j^\top\mathsf{Q}_p^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0}. \quad (8)$$

Although there are three equations, only two of them are linearly independent (we omit the third equation). Therefore, each correspondence that belongs to one plane Π_p gives two equations in the entries of P^i. Finally, we obtain a linear system of equations having the form $\mathsf{A}\mathbf{p} = \mathbf{0}$, where A is a $2m \times 12$ matrix, m being the number of point correspondences. This system may be solved very quickly using a Singular Value Decomposition (SVD).

4.3 Linear approximation of the viewpoint (LIN2)

The above method linearly computes the 11 entries of P^i. However, no profit is taken from the a priori knowledge of the matrix K and that R is a rotation matrix. Therefore, this method is in practice less accurate than the iterative method (see section 6.1).

To overcome this problem, we suppose that the camera rotation between two images is small. Thus we can perform a first order approximation of this rotation $\Delta\mathsf{R}^i(\Delta\alpha, \Delta\beta, \Delta\gamma)$, where $\Delta\mathsf{R}^i = \mathsf{R}^i(\mathsf{R}^{i-1})^t$. We obtain a linear expression of the entries of P^i in the coefficients $t_x, t_y, t_z, \Delta\alpha, \Delta\beta, \Delta\gamma$:

$$\mathsf{P}^i \approx \mathsf{K}\left[\begin{bmatrix} 1 & -\Delta\alpha & \Delta\beta \\ \Delta\alpha & 1 & -\Delta\gamma \\ -\Delta\beta & \Delta\gamma & 1 \end{bmatrix}\mathsf{R}^{i-1} \quad \begin{matrix} t_x \\ t_y \\ t_z \end{matrix}\right] \quad (9)$$

Combining equations (8) and (9) provides a linear system of equations in the viewpoint parameters.

5 Overview of the system

This section gives an overview of our registration method. The scene is modelled with a set of 3D planar polygons $\mathcal{L}_p (1 \leq p \leq n)$. In the following, the main steps of this algorithm are described with further details (Fig. 2).

5.1 Computing P^0

Different methods can be used to recover the projective matrix in the first frame: markers may be added to the scene to make the projection recovery easier. One may also take advantage of natural points in the scene with known 3D coordinates. In this case, the user has to point out these points in the first image.

Particular scene structures can also be used to recover P^0. For example, [13] shows how to compute P^0 from the observation of a rectangle in the scene provided that the principal point is the center of the image and that the aspect ratio is known. We used this technique to process the indoor sequence presented in section 6.2: a poster was placed on the ground and was used to recover P^0.

Obviously, more accurate internal parameters can be recovered if a classical calibration target is used. However, placing a calibration target is not always possible and alternative calibration methods based on particular scene structure are of great interest in such cases.

Initialization stage:

1. Give the equation of the observed planes used for registration ,
2. Compute the projective matrix for the first frame P^0,

Computation of the projective matrix P^i for $i > 0$:

1. Compute the set of matched key-points between images $i-1$ and i for each observed plane.
2. Compute the homographies induced by the observed plane between $i-1$ and i.
3. Compute P_i from P_{i-1} and the computed homographies

Figure 2: Overview of the multi-planar tracking method

5.2 Extracting and matching key-points

The Harris detector [6] is used to detect key-points and the matching process is performed in a classical way. The normalized cross correlation score [17] is computed between a key-point in image i and all key-points lying in its neighbourhood in image $i-1$. We only retain the set of candidate matches presenting with the maximum score, provided that this score is greater than a predefined threshold (typically, $s = 0.8$).

5.3 Robust homography estimation

Let $\mathcal{C}_p^i$ be the set of correspondences $x_j \leftrightarrow x'_j$ such that x_j belongs to the projection of the planar polygon $\mathcal{L}_p$ in image $i-1$ using P^{i-1}. Robust estimation of the homography H_p^i can be achieved using the RANSAC paradigm [3]: randomly samples of four pairs are selected in $\mathcal{C}_p^i$ and the corresponding homography is computed. The homography is tested against all the correspondences: the set of inliers is the set of pairs $x_j \leftrightarrow x'_j$ for which the distance between $\mathbf{x}'_j$ and

$\mathsf{H}_p^i \mathbf{x}_j$ is below a predefined threshold $r = 1.25$. Finally, the homography with the largest consensus set is chosen.

In the following, the set of inliers computed for each planar polygon $\mathcal{L}_p$ is denoted $\mathcal{I}_p^i$.

5.4 Computing P^i

The projective matrix P^i is computed using the full set of inliers $\bigcup_p \mathcal{I}_p^i$ stemming from the visible planar structures. The computation is performed using one of the methods described in section 4 if several polygons are visible in the scene. Otherwise, if only one polygon is visible, the method described in section 3 is used.

Once P^i computation is achieved, the homographies are updated using equation (6). Then, new inliers are detected and used to update P^i. This loop is continued until the number of inliers does not change. This refinement stage generally converges in 2 or 3 iterations.

6 Experimental results

We first conducted experiments on a calibration target. The estimates of the viewpoint for the three methods were compared with estimates given by a classic calibration method. These experiments (section 6.1) clearly show that the method LIN2 is a good compromise between computation rates and accuracy of the viewpoint. Then our methods has been tested on various indoor and outdoor scenes. The indoor sequence (section 6.2) is very difficult because the scene is poorly textured. Despite this, we get very good results and the visual impression of the augmented scene is nice. Finally, we proved the efficiency of our method for outdoor urban scenes. Using the ground and the facade of some buildings in our campus allows us to recover good viewpoints estimates.

6.1 A target sequence

This sequence contains 98 images. The camera is moving around the calibration target, roughly pointing the intersection of the three planes (see Fig. 3). The initial matrix P^0 is obtained using the method proposed in [2], from 3D/2D correspondences of points belonging to the target. The three planes of the target are registered using the different methods.

Comparison of the three methods

Fig. 4 shows the temporal evolution of one viewpoint parameter (x translation), for each of the proposed methods. The other viewpoint parameters show similar curves and errors. Actual values (represented by crosses in Fig. 4), are computed every ten images, using classical calibration from points on the target. ITER gives rise to the best results, LIN1 to the worst.

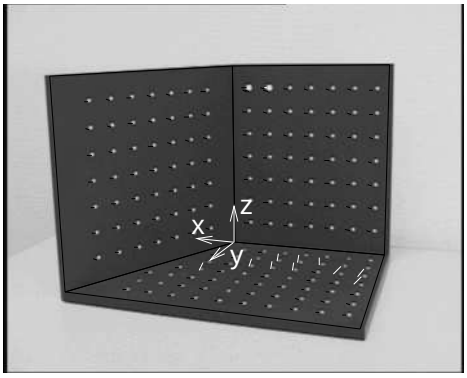

Figure 3: One image of the target sequence.

The graph obtained for LIN2 is relatively close to the actual graph, but it slowly diverges from it: this expresses an accumulation error due to the successive approximations of the rotation matrix. However, the pose estimation error obtained at the end of the sequence is only 6.3 cm for a distance target - camera equal to 127 cm. This error is almost not perceptible with regard to the projection of the target.

Influence of the number of planes

The above results were obtained by registering three planes of the target. Figures 5 shows results obtained by using ITER (the most accurate method) with one or two planes. The titles of the curves indicate which planes were used: X represents the vertical left plane of the target (see Fig. 3), whose equation is $\mathrm{X} - 0.577\mathrm{Y} = 0$, Y the vertical right plane ($\mathrm{Y} = 0$) and Z the horizontal plane ($\mathrm{Z} = 0$). It is interesting to notice that the most accurate results were obtained when the horizontal plane was involved: curves XZ, YZ and Z are very close to the expected results, whereas the other curves diverge. This is due to the fact that the depth information is better represented by the horizontal plane, particularly in the second half of the sequence. Fig. 5 shows that the results obtained from a single plane are much more irregular than the results obtained from two or three planes, which illustrates our contribution with regard to [13].

Computation rates

The computation rates are detailed in table 1 for each step of the algorithm. They were obtained with LIN2 method on a Pentium III 900 Mhz. The images were 360×288 pixels wide. The whole algorithm performs in 62.70 ms per frame on average, which leads to an approximate processing rate of 16 frames per seconds. The iterative method (ITER) is more time consuming (approx. 2 frames per second).

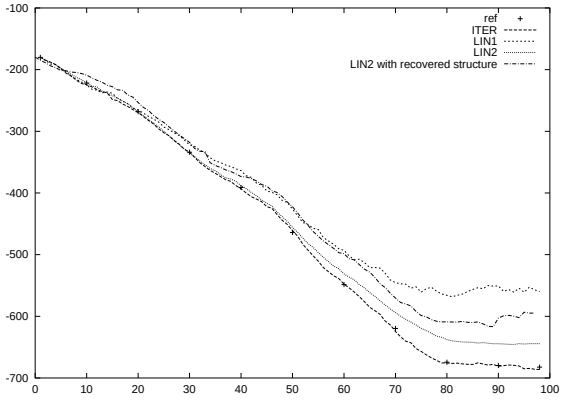

Figure 4: Temporal evolution of t_x with three planes, for ITER, LIN1, LIN2 and LIN2 with the recovered structure. The crosses represent the actual values of t_x, obtained by performing a classical calibration from points on the target.

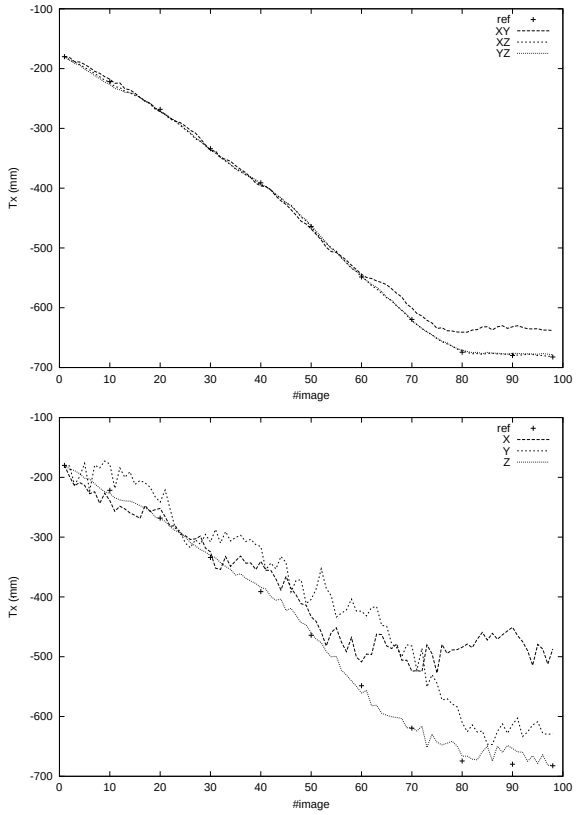

Figure 5: Temporal evolution of t_x with two planes (a) or one plane (b) are used.

6.2 An indoor sequence

Results were also obtained on a 200-frames indoor sequence, shot in the basement of our laboratory. These experiments were conducted within a project of AR for e-commerce. One of the goal is to allow the user to visualize the products in its future environment. Here we attempt to put a new sofa in a room.

MIC	30,33 ms
Matching	18,62 ms
RANSAC	12,13 ms
LIN2	1,62 ms
Total	62,70 ms

Table 1: Computation rates obtained on a Pentium III 900 Mhz (mean over 98 images).

The sequence contains three parts (see Fig. 6): backward movement of the camera with respect to the corner of the target, (frames 0 to 100), panoramic from left to right, making the left wall disappear (frames 100 to 150) and panoramic from right to left (frames 150 to 200). This sequence is particularly difficult to treat because the scene is very poorly textured (hardly a few stains on the ground and walls). Moreover, the camera motion is relatively fast in the second half of the sequence (up to 20 pixels of disparity between two images), and some images are blurred.

The projection matrix in first image was obtained using an ordinary poster laid on the ground (see Fig. 6). The aspect ratio was fixed to 1, and the principal point was assumed to lie at the center of the image. As the angle between the two visible walls was a right angle, no measure had to be taken to recover the equations of the three planes.

Despite the difficulties mentioned above and the approximative knowledge of the internal parameters of the camera, the system succeeded in registering the two or three planes that were visible in the sequence. Fig. 6 shows the matching result and the projection of a cube after registration using LIN2, in four images of the sequence. A final composition is shown in Fig. 7.

6.3 An outdoor sequence

The effectiveness of our approach is also demonstrated on an 600-frames outdoor scene. The campus of our University was shot by a pedestrian walking with an handheld camera. Here our aim is to incrust annotations in order to help the visitor to find some laboraties of our university: the mathematic reseach center and the biological research center (BIO RC and MATH RC). We also add a Maya statue to bring an exotical impression to our campus Fig. 9.

In this application the ground plane and two façades were used to compute the viewpoint. They are drawn in blue on Fig. 8 (note that the cube in blue is not used for registration). Fig. 8 also exhibits the points which are tracked in each planar structure to recover the homography: the points drawn in red are considered as outliers by the algorithm and are discarded from the homography comptation.

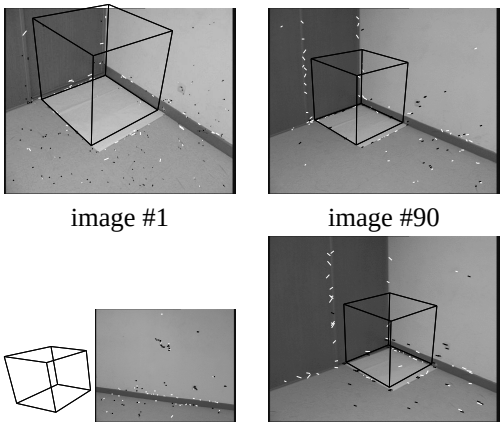

Figure 6: Key-points matching and projection of a cube after registration in four images of the indoor sequence (black segments are inliers, white segments are outliers).

TTo recover the orientation of the third plane with respect to the two others, we use a modelling tool inspired from the work of [8]: as the planes is perdicular to the ground plane, pointing out one segment on the plane allows us to recover its planar equation.

Some snapshots of the augmented scene are shown in Fig. 9. The visual impression is good and the annotations seems to be part of the scene. However, if we look carefully at the full video available at out web site, we notice that the Maya statue seems to slide slightly along the ground at some time instant.

7 Discussion

A real time markerless registration system for augmented reality was presented. It provides accurate and reliable results for scenes which include planar structures. Among the three method we proposed for solving the n-planes registration problem, one of them (LIN2) has proven to be a good compromise between computation rates and accuracy of the composition. Our implementation yields results comparable in accuracy with full structure-and-motion methods but with better reliability. In this article, we proved that this method is applicable to a wide range of environments both for complex indoor and outdoor urban scenes. The accuracy obtained on the viewpoint is sufficient for numerous applications: it especially allows annotations to be displayed, providing the relevant, critical information for a users's context.

We now plan to further investigate how this framework can be improved to be able to handle long sequences. Indeed, the method may progressively diverge because of successive approximations. This problem may be shaped by considering homography with more distant images, or by performing a bundle adjustment on a small number of images (the last five images for example, in the spirit of [4]). Of course, a hybrid system could also increase robustness and avoid drift by taking advantage of a partial 3D knowledge on the scene.

Finally, the possibility of recovering the multi-planar structure of the scene is currently under study. This will be of particular interest when the structure of the observed scene is not obvious, especially when the observed planes are not perpendicular.

References

[1] R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, pages 34–47, December 2001.

[2] O. D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL (USA)*, pages 15–20, 1986.

[3] M. A. Fischler and R. C. 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, 1981.

[4] A.W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Images Sequences. In *Proceedings of 5th European Conference on Computer Vision, University of Freiburg (Germany)*, pages 311–326, June 1998.

[5] D. Gennery. Visual Tracking of Known Three Dimensional Objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[6] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of 4th Alvey Conference*, Cambridge, August 1988.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[8] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating Architectural Models from Images. In *EUROGRAPHICS'99, Milano, Italy*, 1999.

[9] J. P. Mellor. Realtime Camera Calibration for Enhanced Reality Visualization. In *Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine'95 (CVRMed'95)*, pages 471–475, April 1995.

[10] U. Neumann and Y. Cho. A selftracking augmented reality system. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 109–115, 1996.

[11] S. Ravela, B. Draper, J. Lim, and R. Weiss. Tracking Object Motion Across Aspect Changes for Augmented Reality. In *ARPA Image Undertanding Worshop, Palm Spring (USA)*, August 1996.

[12] G. Simon and M.-O. Berger. A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. In *Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pages 261–266, January 1998.

[13] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. International Symposium on Augmented Reality*, pages 137–146, October 2000.

[14] A. State, G. Hirota, D. Chen, W. garett, and M. Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In *Computer Graphics (Proceedings Siggraph New Orleans)*, pages 429–438, 1996.

[15] M. Uenohara and T. Kanade. Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medecine*, 1996.

[16] J. Vallino. *Interactive Augmented Reality*. PhD Thesis, University of Rochester, December 1998.

[17] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, 78:87–119, October 1995.

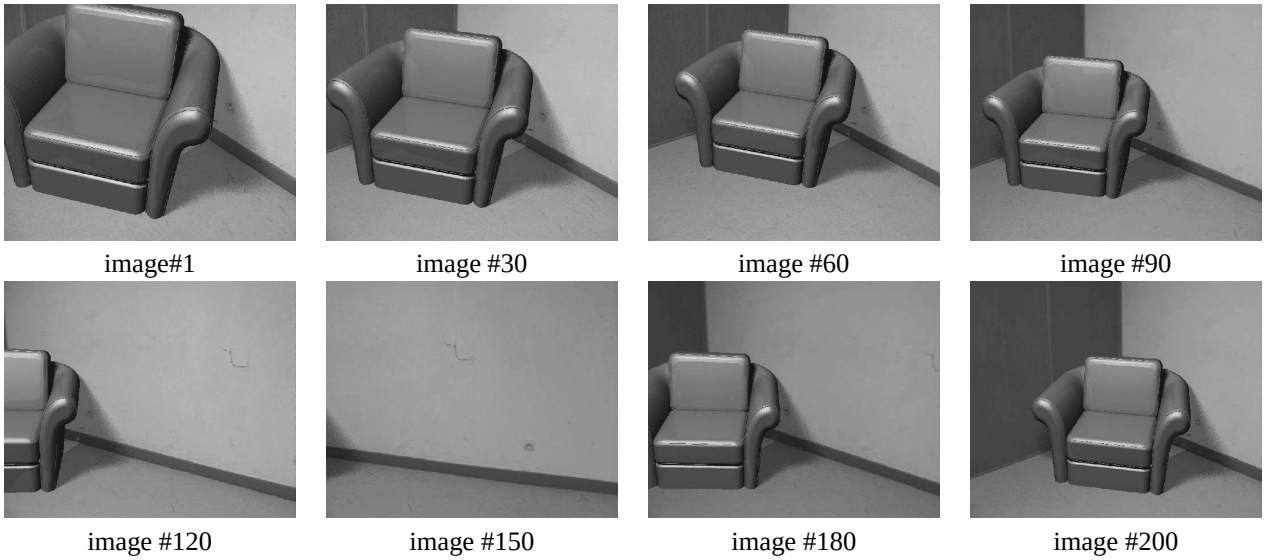

Figure 7: Augmented sequence: a sofa has been added.

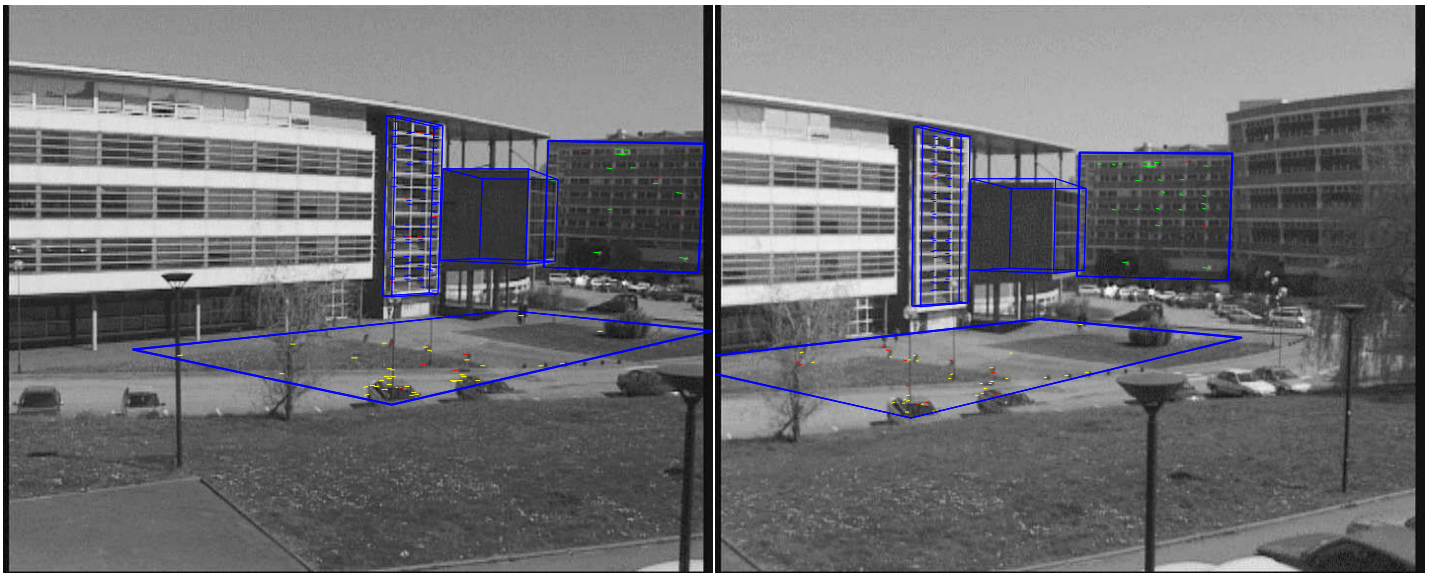

Figure 8: Some intermediary steps towards the augmented campus. The planes used for registration are shown as well as the points used for homography computation.

Figure 9: Some views of the augmented campus. Annotations have been added on the mathematical research center and the biological research center as well as a maya statue.