# Using Geometric Properties for Automatic Object Positioning

Boubakeur Boufama, Roger Mohr, Luce Morin

# Using geometric properties for automatic object positioning

Boubakeur Boufama[1], Roger Mohr[2] & Luce Morin[3]

[1]Math and C.S. Dep.
University of P.E.I.
550 University Av
Charlottetown, PE Canada C1A 4P3
bboufama@upei.ca

[2]Lifia–Inria
46, Av Félix Viallet
38000 Grenoble
France
mohr@imag.fr

[3]Irisa
Campus de Beaulieu
35042 Rennes Cédex
France
lmorin@irisa.fr

**Abstract**

This paper presents an application of some recent results in computer vision, in particular, the use of geometric properties. The problem we examine here is the accurate positioning of an object with respect to another, say a tool. Such applications could be used in complex and hazardous environments like nuclear plants.

Because high accuracy in positioning is our goal here, we suppose that the objects have planar faces on which targets can be put. Without loss of generality, we have used only 2 objects in our experiments. Throughout this paper, we have called them the *reference object* and the *unknown object* respectively. The positions of the targets of the *reference object* with their associated projective invariants are computed in an off-line stage. Hence, given at least 2 images of the two objects, we can automatically identify the *reference object* points in the images and reconstruct the points of the *unknown object* relative to the *reference object*. The experiments show that a precision of 0.1mm in relative positioning can be reached for an object observed at a distance of 2m.

# 1 Introduction

This paper is primarily an application paper which illustrates how recent developments to geometry for vision can be applied to solve delicate industrial problems. The problem we examine here is the accurate positioning of an object with respect to another in a complex and hazardous environment, for instance a nuclear plant.

Three main subjects are treated in this paper: firstly, the use of projective invariants for point identification in a way pioneered by Morin [14] and Meer [12]. For a more detailed description, but restricted to the linear case, see the work of Maybank[11]. In this paper, 2D projective invariants for point identification are considered; the next section describes how these invariants are used. For a more general study on projective invariants and their use in computer vision see [15] and [18] for invariant-based recognition.

Second, the image points (the target centers in our case) have to be located with a high accuracy. We use here an approach originally proposed by Gruen [8] and redesigned by Brand [4]. This work will not be described here.

Finally we need a reconstruction from point matches using uncalibrated cameras. Following the work on projective reconstruction [6] [10], we implemented a method for going from the projective reconstruction to the Euclidean world using a priori available Euclidean information [3]. In our case the positions of some points in their reference frame provide this Euclidean information.

# 2 Point identification

The point identification procedure is used in our process to identify the projections of the 3D *reference object*'s points in the images. As both the calibration of the camera and the position of the *reference object* are unknown, identification is done using projective invariants. The invariants characterizing the *reference object*'s points are computed, in an off-line step, using an image only containing the *reference object*. The *reference object*'s points can then be identified in an image containing several other objects, by comparing the invariants values.

An accurate 2D positioning of image points is required to reliably compute projective invariants. This is achieved by fixing targets on the objects. All image points are defined by the targets centers position which can be extracted with very good accuracy.

## 2.1 Projective invariants

In our algorithm, we have used projective invariants from sets of coplanar points only. In fact it can be shown that projective invariants can be recovered from non-coplanar points [7][9][17]. However a minimum of 6 points is required, whereas only 5 points are needed in the coplanar case. Furthermore, invariants of non-coplanar points can not be extracted from a single image. Using several images would require first matching the points, whereas we want to use projective invariants to

identify points prior to any matching procedure.

The fundamental projective invariant is the *cross-ratio*. The cross-ratio of four collinear points $A$, $B$, $C$, $D$ is defined by :

$$[A, B, C, D] = \frac{\overline{CA}}{\overline{CB}} \times \frac{\overline{DB}}{\overline{DA}} \tag{1}$$

where $\overline{AB}$ is the algebraic measure of $AB$.

The cross-ratio of a pencil of four lines $l_1$, $l_2$, $l_3$, $l_4$ intersecting on point $O$ is defined as the cross-ratio $[P_1, P_2, P_3, P_4]$ of the points of intersection of the $l_i$ with any line $l$ not containing $O$. The obtained value is independent of the choice of $l$.

Five points $O$, $P_1$, $P_2$, $P_3$, $P_4$ define a pencil of four lines with vertex $O$. The associated cross-ratio can be computed from the points homogeneous coordinates according to the Möbius formula:

$$[O; P_1, P_2, P_3, P_4] = \frac{|OP_1P_3|\,|OP_2P_4|}{|OP_1P_4|\,|OP_2P_3|} \quad where \quad |P_iP_jP_k| = \begin{vmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{vmatrix} \tag{2}$$

and $(x_i, y_i, z_i)$ are the homogeneous coordinates of $P_i$.

## 2.2 Evaluating similarity between cross-ratios

In order to compare projective invariants, one needs a measure of distance. An Euclidean distance is not adapted to compare projective quantities. A similarity measure can be derived from the probability density function of the cross ratio [1]. We prefer to use the Mahalanobis distance defined from a first order linearization of the cross ratio. This approximation is valid as long as we stay close to the point where the linearization is made and far from the singularities of the cross ratio function. The first condition is guaranteed by the very good accuracy of the image points measurements (1/20 of a pixel). In order to avoid singularities when computing the cross ratio from 5 coplanar points, we have to exclude points which are aligned, and for which a separate type of projective invariants can be computed.

We also assume independent Gaussian noise with standard deviation $\sigma$ on each of the $x$ and $y$ coordinate measurements of all the targets. Therefore the distance between the reference cross ratio $c$ and the computed value $c'$ from points

$\{(x_i, y_i)\}_{i=1..5}$, is :

$$\frac{(c - c')^2}{\sigma^2 \sqrt{\sum_{i=1}^{5} \left( \frac{\partial c'}{\partial x_i}^2 + \frac{\partial c'}{\partial y_i}^2 \right)}} \tag{3}$$

## 2.3 Index computation

Considering target points on the *reference object*, we want to find projective invariants which enable us to identify these points among $n$ points in an image. The minimum number of coplanar points with projective invariants is 5, and 5 coplanar points have only 2 functionally independent projective invariants. In order to consider the $C_n^5$ nonordered sets instead of the $A_n^5$ ordered sets (both numbers are of order $n^5$ however), the invariants should be computable regardless of the order of the points in the set. Two such independent values can be computed using symmetrical functions such as symmetrical polynomials or min and max functions. However, experiments have shown that these symmetrical invariants are less discriminative than simple cross ratios, and also that a redundant characterization is more efficient in practice than a characterization with the minimum number of invariants[14].

There is an implicit perspective invariant which is not used at this point : the preservation of the convexity and orientation for observed points in the scene. This can be very useful in the following way.

Five points may lie in one of the following configurations as displayed in Figure 1. These configurations are preserved under perspective projection, and for each of them a particular set of cross ratios can be computed, which is independent of the order of the points in the set. For instance, considering the third case of 5 points on the convex hull, 5 cross ratios are computed taking each point as the origin of a pencil of lines, with the lines in the pencil ordered using the standard plane orientation. The starting point is chosen so that the largest value is the first.

Using this characterization, two sets of nonordered points can be compared: the computation of the invariant values in both sets gives a measure of distance between the two sets, as well as a point-to-point correspondence. In order to improve the time of computation, invariants for the *reference object*'s points are computed off-line and used as indices in a table containing the 5 points sets.

Experiments conducted for this setup proved that the index table was really discriminant, leading to an extremely low confusion rate, from about 1/4000 to 1/2000 for different sets of simulated data depending essentially on the minimum distance between the considered points. The use of convexity improved these rates

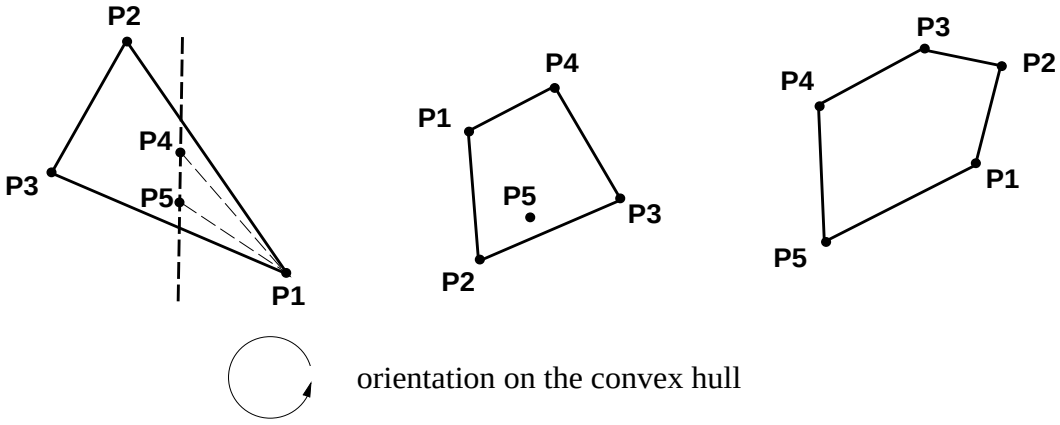



Figure 1: Different convex hull for five points and plane orientation. Note how each case can lead to the consideration of a particular pencils

by a factor of 10, the use of a redundant set of data by a factor of 5. Part of these results were confirmed by separate experiments [12].

It can be noted that such a characterization has been generalized recently by Carlsson [5] and called "combinatorial structure" of a point set.

# 3 The steps of the positioning process

We have used 2 objects in our experiments: one of them is considered a *reference object* and the other one an *unknown object*. We do not suppose that the *reference object* is known, except that it has planar faces on which targets can be put. Note that the use of targets here is only for the purpose of high accuracy, since the target centers can be located in the images with subpixel precision. Our system (the whole process) can still work without any change if, for example, corners are used instead of target centers.

The system proposed in this paper consists of several steps listed below; all these steps are completely automatic.

## 3.1 Reference object's points reconstruction

The first step is the positioning or reconstruction of the *reference object*'s points relative to some reference frame. The goal here is not to obtain an absolute

reconstruction but rather a relative one, so any known set of 3D points could be used as a reference frame. This off-line step could be called "the *reference object* recognition"; it enables us to use the *reference object* as a known 3D reference frame in the future.

For this purpose, the *reference object* is placed together with a highly precise calibration pattern (see Figure 2) and several images (at least 2) are taken from different viewpoints. Note that we have no knowledge of the camera's positions or intrinsic parameters, the camera (or cameras if many) is assumed unknown. However, the precise calibration pattern has point coordinates known in a Euclidean frame.

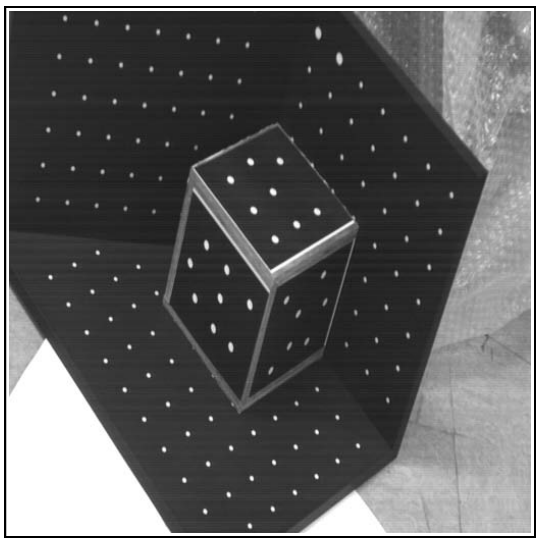


Figure 2: One image of our *reference object* (unknown at this stage) placed together with a highly precise calibration pattern. Using at least 2 images, the *reference object*'s points will be reconstructed.

Once all target centers are extracted [8] from the images, the targets on the *reference object* are identified and separated from the calibration pattern's targets. This is done automatically since the calibration pattern is known. At this stage we need to match the *reference object*'s points (target centers) in the different images. This is also done automatically by using epipolar geometry between images. The epipolar geometry between a pair of images is computed using the point correspondences of the calibration pattern.

Some problems may arise at this stage, such as a correspondence conflict, i.e., for a point in one image there are 2 or more points in the other image lying

on the corresponding epipolar line. This problem was solved simply by ignoring the match. This choice was adopted for two main reasons: firstly this case is statistically rare and secondly our implementation deals with missing matches. This process is very fast, in particular we have used a new faster method for epipolar geometry computation [2].

At this stage, we have the image points of the *reference object* matched in at least 2 images. Using these matches, the *reference object*'s points are reconstructed (see Paragraph 3.3 for the reconstruction method we have used).

## 3.2 Using the reference object to position an unknown object

We have shown in the above paragraph how to compute the relative positions of the *reference object*'s points. The same method is applied here; suppose we are given at least two images of a scene containing the *reference object* together with one *unknown object* (see Figure 3). The target centers are first extracted from each image, then the *reference object*'s points are identified among all points using projective invariants already computed in an off-line stage according to Section 2. Using 2D invariants, the *reference object*'s points are then matched in the different images and used to compute the epipolar geometry. The remaining points, points of the *unknown object(s)*, are then matched using the epipolar geometry.

Finally, the *unknown object* is reconstructed relative to the *reference object*; the reconstruction method is described below.

Note that for our particular case where the *unknown object* has planar faces, it is possible to use 2D invariants for point-matching in different images. However, as we have stated before, the *unknown object(s)* is not constrained to have planar faces; i.e., our system can reconstruct any unknown set of points. Hence, we have chosen not to use 2D invariants for matching the points of the *unknown object* to keep the method valid in the general case.

## 3.3 Positioning from multiple views

This reconstruction is done using matched points in at least two images. The algorithm was originally described in [13]. It computes the projective 3D structure of points from a sequence of images, and uses the a priori knowledge of the $3D$ scene, for example the 3D Euclidean coordinates of at least 5 points, to transform the projective reconstruction to a Euclidean one. However, our algorithm can be used so that the Euclidean reconstruction is obtained directly when the Euclidean positions of at least 5 points are supplied.

Consider $v \geq 2$ images of a scene composed of $p$ points, thus providing $p \times v$ image points. Let $\{\mathbf{P}_i, i = 1, \ldots, p\}$ denote the unknown $3D$ coordinates of the points projected in each image. Each point $\mathbf{P}_i$ is represented by a column vector of its homogeneous coordinates $(x_i, y_i, z_i, t_i)^T$, or its non-homogeneous coordinates $(X_i, Y_i, Z_i)^T = (\frac{x_i}{t_i}, \frac{y_i}{t_i}, \frac{z_i}{t_i})^T$. In image $j$, the point $\mathbf{P}_i$ is projected to the point $\mathbf{p}_{ij}$, represented by a column vector of its three homogeneous coordinates $(u_{ij}, v_{ij}, w_{ij})^T$, or its image's non-homogeneous coordinates $(U_{ij}, V_{ij})^T$. Let $\mathbf{M}_j$ denote the $3 \times 4$ projection matrix of the $j$-th image.

For the non-homogeneous coordinates of the image points we have:

$$\begin{cases} U_{ij} = \frac{m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i} \\ \\ V_{ij} = \frac{m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i} \end{cases} \tag{4}$$

These equations express the collinearity of the space points and their corresponding projection points. Since we have $p$ points and $v$ images, we have $2 \times p \times v$ equations. The number of unknowns is $11 \times v$ for the projection matrices which are defined up to a scaling factor, plus $3 \times p$ for the space points. Therefore if $v$ and $p$ are large enough, we have a redundant set of equations.

The solution of system (4) can only be defined up to a collineation. As a matter of fact, if $\mathbf{M}_j$ and $\mathbf{P}_i$ are a solution for some $i, j$, so are $\mathbf{M}_j \mathbf{W}^{-1}$ and $\mathbf{W}\mathbf{P}_i$, where $\mathbf{W}$ is a collineation of the 3-D space, i.e., a $4 \times 4$ invertible matrix. Consequently, a basis for any $3D$ collineation can be arbitrarily chosen in $I\!P^3$. Given the projective space in $I\!P^3$, 5 algebraically free points (where no four of them are coplanar) form a basis.

¿From the above equations, the problem can be formulated as a conditional parameter estimation problem. In the general case we have to estimate the parameters (here the matrices $\mathbf{M}_j$ and the $3D$ coordinates of points), given noisy measurements (here the image coordinates). We assume that the measurements are obtained with a mean value equal to the observed one, and with a covariance matrix $\mathbf{C}$.

Let $\mathbf{Q}$ denote the vector of all parameters, $q_k$ denote one of its elements, $\mathbf{U}$ denote the vector of all the measurements $U_{ij}$ and $V_{ij}$, and $u_l$ denote one of its elements. If the relation between the measurements $u_l$ and the parameters $q_k$ is linear, namely, if $\mathbf{U} = \mathbf{A}\mathbf{Q}$, then the maximum likelihood estimation of the parameters is the vector $\mathbf{Q}$ which minimizes the Mahalanobis distance, i.e., the least square criterion:

$$\chi^2 = (\mathbf{U} - \mathbf{A}\mathbf{Q})^t \mathbf{C}^{-1} (\mathbf{U} - \mathbf{A}\mathbf{Q}) \tag{5}$$

In the nonlinear case, linearization may be obtained by taking the first order Taylor expansion of the nonlinear function linking $\mathbf{Q}$ with each $u_l$. Therefore, in our case, Eq (5) leads to the minimization of the sum:

$$\chi^2 = \sum_{ij} \left( \frac{U_{ij} - \frac{m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}}{\sigma_{ij}} \right)^2 + \sum_{ij} \left( \frac{V_{ij} - \frac{m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}}{\sigma_{ij}} \right)^2$$

¿From the above, the reconstruction problem turns to a nonlinear optimization problem. Therefore, an initialization is required to ensure the convergence of the nonlinear process. However, if more than five space point coordinates are given, a first step linear computation of the projection matrices becomes possible and an estimation of all the space points is then given. So to overcome the initialization problem, we suppose in the following that we are given at least 6 space points on the *reference object*. Furthermore, the *reference object* must consist of at least 8 points, not necessarily known in a 3D space, to make the epipolar geometry computation possible.

# 4 Experiments

We consider here 2 objects $RO$ and $UO$ of size $\sim 100mm \times 100mm \times 100mm$ (see Figure 3). In a first off-line stage, $RO$ is reconstructed using a calibration pattern (Figure 2) and 2D invariants characterizing $RO$ are computed. We have also reconstructed $UO$ using the calibration pattern; however, this reconstruction is used only for error computations.

In the remainder, $RO$ is the *reference object* and $UO$ is the *unknown object*. Two kinds of experiments were performed:

1. *Accuracy measurement:* the goal here is to compute the reconstruction accuracy on $UO$ relative to $RO$.

   Two images of the scene (Figure 3) are used as our input. Target centers are extracted, points of $RO$ are identified using 2D invariants, image points are matched using epipolar geometry and finally $UO$ is reconstructed.

   Table (1) shows the accuracy reached in these experiments, 0.1mm on the computed coordinates. This is a good precision which is sufficient for most industrial applications.

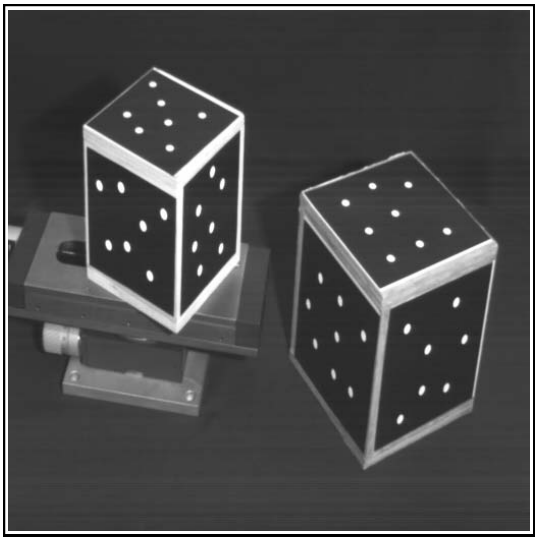


Figure 3: An image of the scene consisting of $RO$ (*reference object*) on the right and $UO$ (*unknown object*) on the left

| | $\Delta X$ | $\Delta Y$ | $\Delta Z$ |
|---|---|---|---|
| | 0.013 | 0.042 | 0.010 |
| | 0.158 | 0.063 | 0.099 |
| | 0.031 | 0.036 | 0.061 |
| | 0.041 | 0.045 | 0.013 |
| | 0.089 | 0.191 | 0.275 |
| | 0.034 | 0.026 | 0.142 |
| | 0.019 | 0.024 | 0.020 |
| | 0.051 | 0.023 | 0.089 |
| | 0.069 | 0.098 | 0.080 |
| mean_error | 0.091 | 0.082 | 0.104 |
| relative_error | 0.101% | 0.091% | 0.116% |

Table 1: Example of positioning errors

2. *Relative motion estimation :* The goal here is to estimate the motion made by $UO$ using its reconstruction relative to $RO$. A micrometric table is used to measure the object motions. $UO$ is reconstructed before and after its motion (see Figure 4). Having the 2 reconstructions of $UO$, we have computed the rigid motion between these 2 reconstructions. Note, that all our steps are based on automatic processes; the algorithm below gives an outline of how the relative motion is computed. Table (2) gives motion errors for a pure translation and a pure rotation.

   *Relative motion estimation algorithm :*
   Because $UO$ is reconstructed before and after its motion, each physical point $P$ of $UO$ has two different reconstructions (3D coordinates). Let $P_b$ be the Euclidean 3D coordinate of $P$ before the motion and $P_a$ the Euclidean 3D coordinate of $P$ after the motion. The relation between $P_b$ and $P_a$ is a rigid transformation, namely a rotation and a translation. Therefore, we have :

$$P_a = RP_b + T$$

   where, $R$ is the $3 \times 3$ rotation matrix and T is the translation vector.
   The above relation is linear; given at least 4 corresponding points, $R$ and $T$ can be computed. However, the rotation constraints (the rotation has only 3 parameters) must be added to ensure a correct estimation of the motion.

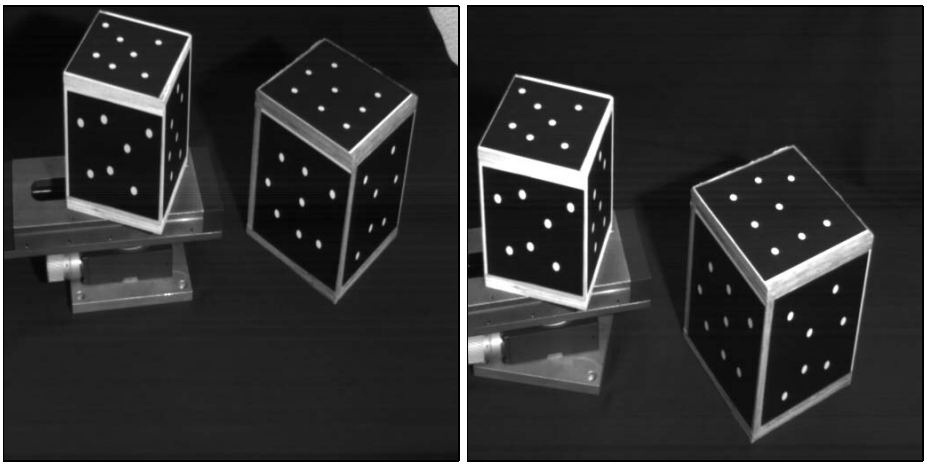
   Hence, we have computed the *unknown object* motion in 2 steps :

   (a) Linear step : the rotation and translation are computed using the above relation, this is done using the SVD method.

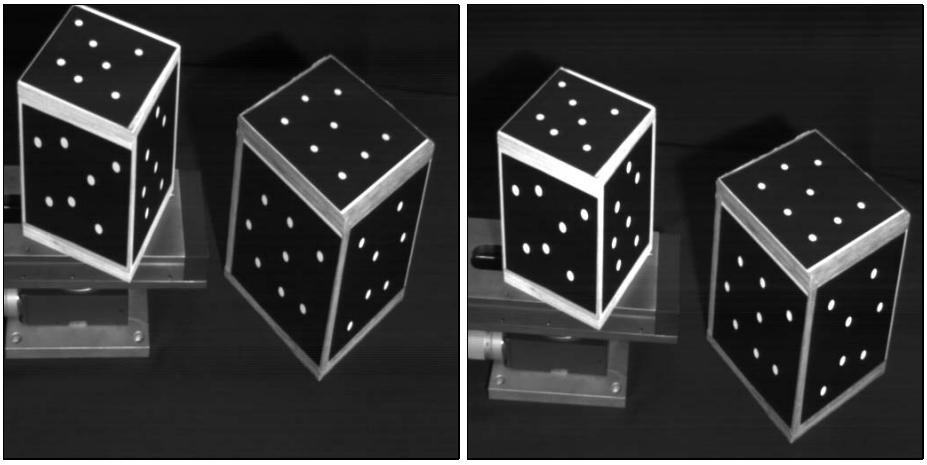
   (b) Nonlinear step : the solution obtained in the linear step is used as a starting point for a nonlinear optimization. The latter consists of the linear equations from the above relation and the nonlinear constraints on the rotation matrix (orthogonality of the matrix). This step is done using the Levenberg-Marquardt algorithm [16].

# 5 Conclusion

This paper shows the applicability of recent tools developed within the framework of projective vision. Automatic classification of points allows identification of points on a planar patch, almost without ambiguity, thanks to the use of invariants and to the accuracy with which targets are identified. It has to be noted that at



On the left, the scene before the rotation of $UO$ and
on the right the scene after a pure rotation of $UO$.



On the left, the scene before the translation of $UO$ and
on the right the scene after a pure translation of $UO$.

Figure 4: Images of the scene for motion computation.

| | *translation* | *rotation* |
|---|---|---|
| measured values | 50.00 mm | 20.00 degrees |
| computed values | 50.140mm | 20.244 degrees |
| relative errors | 0.28% | 1.2% |

Table 2: The relative motion comparison

this level of accuracy, affine approximation of the projection would be no more relevent.

Accurate reconstruction or positioning proved also to be feasible. In the environment suggested by the application, the exact position of the camera cannot be known, nor can the intrinsic parameters of the camera be known, as a zoom is used for getting full resolution in the image. Only the recent projective tools could therefore allow such a result.

One might ask how good the projective model is in the presence of distortion. We experimented with radial distortion on simulated data, using the type of distortion provided for our lens. Experiments showed few differences from data without distortion.

# 6 Acknowledgements

This work is partially supported by European Esprit BRA Viva projects which are gratefully acknowledged.

# References

[1] K. Åström. A correspondence problem in laser-guided navigation. In *Proceedings Swedish Society for Automated Image Analysis, Uppsala, Sweden*, pages 141–144, 1992.

[2] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1030–1036, June 1995.

[3] B. Boufama, R. Mohr, and F. Veillon. Euclidean constraints for uncalibrated reconstruction. In *Proceedings of the 4th International Conference on Computer Vision, Berlin, Germany*, pages 466–470, May 1993.

[4] P. Brand and R. Mohr. Accuracy in image measure. In Sabry F. El-Hakim, editor, *Proceedings of the* SPIE *Conference on Videometrics III, Boston, Massachusetts, USA*, volume 2350, pages 218–228, November 1994.

[5] S. Carlsson. Combinatorial geometry for shape representation and indexing. In J. Ponce, A. Zisserman, and M. Hebert, editors, *Proceedings of the ECCV'96 International Workshop on Object Representation in Computer Vision II, Cambridge, England*, Lecture Notes in Computer Science, pages 53–78. Springer-Verlag, April 1996.

[6] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.

[7] P. Gros. 3D projective invariants from two images. In *Proceeding of the* DARPA–ESPRIT *workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 65–85, October 1993.

[8] A.W. Gruen. Adaptative least squares correlation: a powerful image matching technique. *S. Afr. Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[9] R. Hartley. Invariants of points seen in multiple images. Technical report, G.E. CRD, Schenectady, 1992.

[10] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 579–587. Springer-Verlag, 1992.

[11] S.J. Maybank and P.A. Beardsley. Applications of invariant to model based vision. *Journal of Applied Statistics*, 1994. to appear.

[12] P. Meer, S. Ramakrishna, and R. Lenz. Correspondence of coplanar features through $p^2$-invariant representations. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*, pages A–196–202, 1994.

[13] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 543–548, June 1993.

[14] L. Morin. *Quelques contributions des invariants projectifs à la vision par ordinateur.* Thèse de doctorat, Institut National Polytechnique de Grenoble, January 1993.

[15] J.L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, Massachusetts, USA, 1992.

[16] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[17] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(1):34–46, January 1995.

[18] C.A. Rothwell. Hierarchical object descriptions using invariants. In *Proceeding of the* DARPA–ESPRIT *workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 287–303, October 1993.