

# Modelled Object Pose Estimation and Tracking by Monocular Vision.

N. Daucher, M. Dhome, J.T. Lapresté, G. Rives.

Electronics Laboratory, Blaise Pascal University

F-63177 AUBIERE CEDEX

email [lapreste@le-eva.univ-bpclermont.fr](mailto:lapreste@le-eva.univ-bpclermont.fr)

## Abstract

This paper presents a new method that permits to solve the problem of determination of a modelled 3D-object spatial attitude from a single perspective image and to compute the covariance matrix associated to the attitude parameters. Its principle is based on the interpretation of at least three segments as the perspective projection of linear ridges of the object model and on the iterative search ( using Kalman filtering) of the model attitude consistent with these projections.

The knowledge of the attitude and of the associated covariances enables to use a higher level Kalman filter to track an object along an image sequence. In the tracking process this Kalman filter is used to predict the attitude of the object and the error matrices are used to make robust automatic matches between the image segments and the model ridges.

Tracking experiments have been made that proves the validity of this approach.

*This work has been partially supported by a contract with the European Spatial Agency (ESA) in which society Sagem is the prime contractor.*

## 1 State of the art

In the literature, many methods to locate 3D objects by monocular vision can be found. All of these need some prerequisites (as the knowledge of the observed object model and matches between 2D image primitives and 3D model elements) to compensate the loss of data due to the projection of the 3D world on the image plane. These methods compute the spatial attitudes of the 3D model such that the selected model elements are projected on the corresponding 2D image primitives.

They can be classified in function of the manipulated image primitives (points, straight lines, elliptical contours, limbs of curved surfaces...) and the assumption about the projection of the real world on the image plane (orthographic or perspective). Some methods give the closed form solutions of the addressed inverse perspective problem, the other ones use iterative processes to reach the solution.

Numerous methods deal with point correspondance ([RIV-81], [HOR-87], [HUT-87], [HOR-89]). Others use line segments ([KAN-81], [LOW-85], [SHA-86], [DHO-89]).

However any efficient method must comply to the following requirements:

1. The approach must be general enough to handle any general combination of sufficiently numerous primitives (i.e. not restricted to particular configurations).
2. The model of projection must be the perspective one because, up to now it is the only tractable model that allows to compute accurate locations: orthography or scaled orthography are insufficient.

3. The used primitives must be present in the images and (easy) ways for extracting them must exist.
4. It is necessary to be able to compute the accuracy of the obtained localizations to be able to use them in real industrial environments with a minimum of confidence.

Our method tries to perform these requirements and its basics are very similar to Lowe's one. As Lowe, we suppose that the matching between 3 or more model edges with segments of the images has been obtained, and we use an iterative process to minimize a distance criterion [LOW-85]. The implementation used differs from he's in three directions:

- the pose is computed in two steps. The first step involves purely vectorial equations. The unknowns are the three rotation angles of the rigid transform and the process is properly iterative. The second is to compute the remaining unknown translation. It merely consists in the resolution of a linear least square problem, thus it only involves a linear system resolution.
- The criterion minimized differs from Lowe's one. Lowe uses a 2D criterion calculated in the image plane. Our criteria are 3D ones which permit to greatly simplify the computations involved.
- The iterative process is implemented as a Kalman filter which permits to obtain covariance matrices about the computed pose parameters.

Due to lack of space, many mathematical developments have been omitted, they can be found in [DAU-93].

## 2 The localization algorithm

In this section we will describe the problem of localization we will address, give the mathematical equations and the algorithms to solve them.

In fact two approaches are given, the first is a Newton-Raphson one and is stated for its simplicity and efficiency and also because it permits a pedagogical presentation for using the second one which consists of a Kalman filter.

In practice, to decrease the computational burden, the Newton-Raphson method is used in a first pass to obtain a good approximation of the attitude, and the Kalman filter is then run to give the associated covariance matrices.

### 2.1 General formulation.

The localization of a rigid object from an image depends on eleven parameters. The first six are the "extrinsic" ones, i.e. the values of the three rotation angles and the three components of the translation that are needed to bring the model in the observed position. The last ones are "intrinsic" to the camera system and are the coordinates of the intersection of the view axis with the image plane ( $u_0, v_0$ ), and  $(f_x, f_y)$  where  $f_x = f/dx$  and  $f_y = f/dy$  ( $f$  is the focal length and  $dx$  and  $dy$  are the digitization steps of the system of acquisition).

Thus, the problem we intend to address here can be stated in a fairly general way by the following formulation.

Given:

1. the knowledge of the intrinsic parameters of the acquisition system,
2. the accuracy of this knowledge,
3. a projective model (pinhole) for image formation,
4. the CAD-model of an object,
5. the matching of sufficiently many primitives belonging respectively to the model and to a brightness image of the object,

6. the accuracy about the extraction of the images primitives.

*Find the extrinsic parameters giving the pose of the viewed object in the acquisition reference system and the uncertainties associated to these parameters.*

To be able to solve it, we have specialized the problem in two ways:

1. the primitives involved are straight lines as well in the image and in the model,
2. the mathematical equations involving the matched straight primitives will only rely on direction or extremities of models primitives and on the normal to the interpretation plane<sup>1</sup> of the images primitives (not on particular isolated points extracted from the image)

We would like to justify these choices. The primitive likely to be chosen are points, straight lines or conics. First it must be noted that points have not been chosen as primitives for two main reasons:

- from an image to another, points of interest can more easily be subject to occlusion than curves.
- the computational process that leads to structured primitives such as line segments or ellipses involves a least square minimization that lessens the errors. (Straight lines are more accurately defined than the points they are made of.)

Moreover, if it is always easy to find in the images of an industrial environment, primitives that can be interpreted and stored as line segments, this problem is much more complex for conics or parts of conics. Also, it can be said that (with a sure loss of accuracy) curves can be transformed in straight segments chains by polygonal segmentation.

## 2.2 The mathematical equations.

Let  $n$  image segments  $l_i$  be matched respectively with  $n$  model ridges  $L_i$ .

Let us also suppose that all the vectors and points are expressed in the camera coordinate system  $Rc$  (see figure 1).

Each segment  $l_i$  of the image is characterized by its unit director vector  $v_i$  and a point  $p_i$ . Then we can easily compute the unit vector normal  $N_i$  to the interpretation plane  $\Pi_i$  of  $l_i$ .

Using the following notations:

$$v_i = (a_i, b_i, 0), \quad \overrightarrow{O_c p_i} = (x_i, y_i, f), \quad N_i = (A_i, B_i, C_i),$$

we obtain :  $N_i = v_i \times \overrightarrow{O_c p_i} / \|v_i \times \overrightarrow{O_c p_i}\| = (b_i f, -a_i f, d_i) / \sqrt{f^2 + d_i^2}$ .

Where  $d_i = a_i y_i - b_i x_i$  is the distance between the supporting line of  $l_i$  and the image center  $(u_0, v_0)$ .

The equation of the interpretation plane  $\Pi_i$  is :  $A_i x_i + B_i y_i + C_i z_i = 0$ . We are looking for the transform (rotation  $R_{\alpha\beta\gamma}$  and translation  $T_{uvw}$ ) which minimizes the sum of the distances between the extremities of each the model edges and the interpretation plane of the corresponding image segments. This minimization is made in two steps.

- To compute the rotation  $R_{\alpha\beta\gamma}$  we merely express that the image by the rotation of the director vector  $V_i$  of an edge ( $L_i$ ) of the model lies in the interpretation plane of the corresponding segment. This can be written as a scalar product:  $\langle R_{\alpha\beta\gamma} V_i, N_i \rangle = 0$ .
- When the rotation is computed, the translation is determined by expressing that a point  $P_i$  of each edge of the model, rotated and translated also lies in the interpretation plane of the corresponding image segment. This can be written:  $\langle R_{\alpha\beta\gamma} P_i + T_{uvw}, N_i \rangle = 0$ .

<sup>1</sup> the interpretation plane of an image segment is the plane containing the segment and passing through the optical center.

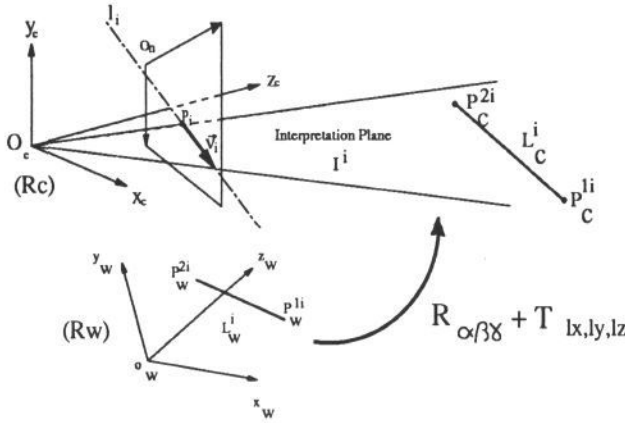


Figure 1: Interpretation plane and reference systems

The main advantage in writing these equations is that the computation is performed independently for the rotation and the translation, each of the systems having half of the number of unknown parameters. In practice, we use as points  $P_i$  the two extremities of the model ridge.

### 2.3 System resolution. The Newton-Raphson approach

*Computing the rotation:* The signed distance of a vector  $R_{\alpha\beta\gamma}V_i = V'_i = (X'_i, Y'_i, Z'_i)$  to the interpretation plane  $\Pi_i$  is given by:

$$D(\alpha, \beta, \gamma, V_i, N_i) = \langle R_{\alpha\beta\gamma}V_i, N_i \rangle = A_i X'_i + B_i Y'_i + Z'_i.$$

We are looking for a triplet  $A = (\alpha, \beta, \gamma)$  such that  $D(\alpha, \beta, \gamma, V_i, N_i) = 0, i \leq n$ .

As the system is non linear, we use a iterative Newton-Raphson approach to solve it. Moreover, we shall use a first order approximation, writing:

$$D(\alpha_k, \beta_k, \gamma_k, V_i, N_i) = \frac{\partial D}{\partial \alpha}(\alpha_k - \alpha) + \frac{\partial D}{\partial \beta}(\beta_k - \beta) + \frac{\partial D}{\partial \gamma}(\gamma_k - \gamma). \quad (1)$$

If  $(\alpha, \beta, \gamma)$  is the solution, at each step we will (as the system is overdetermined) minimize the sum of the squares of the first order approximation of the distance of the model edges director vectors to the corresponding interpretation planes.

*Computing the translation:* Once the rotation is computed, it is easy to obtain the translation as the system giving the three translation components is linear. From equation  $\langle R_{\alpha\beta\gamma}P_i + T_{uvw}, N_i \rangle = 0$ , we infer:  $\langle T_{uvw}, N_i \rangle = - \langle R_{\alpha\beta\gamma}P_i, N_i \rangle$ .

This system can be solved by any classical linear least squares technique.

### 2.4 The Kalman filter approach.

The equations are of course the same. The advantage found in using Kalman filtering is to be able of estimating errors on solutions (covariance matrix on estimated parameters) as well as solutions.

### 3 Error in computing the pose of an object

This section is devoted to the computation of the localization errors associated with our method and is crucial for the determination of the adequation of the method to practical task requirements. However, all results of this section are theoretical and the reader concerned by their application can switch to the next sections.

What we need is to be able to compute the covariance matrices about the normal vectors to the interpretation planes that are used in the measure equations of the Kalman filter (see section 2) used to locate our object.

It is the confidence about these normals that permits to compute (with the Kalman filter) the final confidence about localization parameters.

#### 3.1 Error in computing straight line equation

The process starts from pixel datas ( $p_i = (x_i, y_i)_{i=1..n}$ ) that are given by an edge detector. Following Ayache [AYA-89] it is assumed that each of the contour points encompasses a gaussian white noise.

The equation of the line that supports the points in its hessian form is  $-x \sin \theta + y \cos \theta + c = 0$ ,

It will be supposed that this noise is gaussian and oriented in the direction normal to the line. If a frame consisting of the line direction and of the normal to this direction is used, each point will have a relevant covariance matrix  $C$  in which all terms are 0 but  $C_{22} = \sigma_p^2$ .

If  $R$  is the rotation matrix that transforms the natural image frame into the frame related to the line, the covariance matrix for each point expressed in the natural frame will be :  ${}^t R C R$

The determination of line equation from points with gaussian distribution is a classical least square problem. In the computer vision literature, this problem (and also the error matrix determination involved) is treated in general by writing the line equation in the form  $y = ax + b$  (or  $x = ay + b$ ) and minimizing sums of the type  $\sum_{i=1}^n (y_i - ax_i - b)^2$ . As this method can lead to serious bias, we will use the perhaps less known following result

**Theorem.** *There exists two couples  $(\theta, c)$  minimizing*

$$S = \sum_{i=1}^n (-x_i \sin \theta + y_i \cos \theta + c)^2.$$

Let  $\bar{x}$  and  $\bar{y}$  be the mean values of the  $(x_i)$ , (resp.  $(y_i)$ ), and  $\text{Var}(x)$ ,  $\text{Var}(y)$  be the square of their respective standard deviations (variances),  $\text{Cov}(x, y)$ , their covariance. Then we have

$$\tan(2\theta) = 2 \text{Cov}(x, y) / (\text{Var}(x) - \text{Var}(y)), \quad \text{and} \quad c = \bar{x} \sin \theta - \bar{y} \cos \theta.$$

This means that two values of  $\theta$  differing of  $\pi/2$  are possible and that the line goes through the barycenter of the points.

This result can be found in many probability textbooks and also in [PAV-77].

Now it remains to determine the covariance matrix of the triplet  $(a, b, c)$ , where  $a = -\sin \theta$  and  $b = \cos \theta$ . The help of the Maple software was used to find out the next result. Let  $N = \sigma_p^2 (b^2 \text{Var}(x) - 2ab \text{Cov}(x, y) + a^2 \text{Var}(y))$

$$\text{and } D = n(\text{Var}(x) - \text{Var}(y))^2 + 4 \text{Cov}^2(x, y).$$

$$\text{Then the matrix is: Cov} = \begin{pmatrix} \frac{b^2 N}{D} & -\frac{abN}{D} & \frac{b(\bar{y}a - \bar{x}b)N}{D} \\ -\frac{abN}{D} & \frac{a^2 N}{D} & -\frac{a(\bar{y}a - \bar{x}b)N}{D} \\ \frac{b(\bar{y}a - \bar{x}b)N}{D} & -\frac{a(\bar{y}a - \bar{x}b)N}{D} & \frac{(\bar{y}a - \bar{x}b)^2 N}{D} + \frac{\sigma_F^2}{n} \end{pmatrix}.$$

### 3.2 Errors on normals

For each segment, the error in computing the normal to the plane supporting the segment through the optical center can be obtained now, in a standard way.

We will use two computations putting detail emphasis on the first one:

- computation of the covariance for one normal.
- computation of the covariance for  $n$  normals.

The second computation is very similar to the first one although more complicated. It is not possible to restrict to the first because of the dependency of all the equation to the camera intrinsic parameters. However the first computation will allow to see more easily the method.

*Error on one normal:* The equation of the line supporting a segment is  $ax + by + c = 0$  with  $a^2 + b^2 = 1$ . Thus the normal can be written:

$$\begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} \frac{af_x}{D} \\ \frac{bf_y}{D} \\ \frac{c+au_0+bv_0}{D} \end{pmatrix}, \quad \text{where } D = \sqrt{a^2 f_x^2 + b^2 f_y^2 + (c + au_0 + bv_0)^2}.$$

We will suppose that we know the covariance matrix on the coefficients  $(a, b, c)$  and also on:

- the position of the optical center projection  $(u_0, v_0)$ . We hypothesis here, that these two variables are independent of the others and mutually independent; so the corresponding matrix is diagonal with diagonal elements:  $\sigma_{u_0}^2$  and  $\sigma_{v_0}^2$ .
- the focal length of the camera  $f_x$  and  $f_y$ . We hypothesis here that these two variables are independent of the others and mutually independent; so the corresponding matrix is diagonal with diagonal elements:  $\sigma_{f_x}^2$  and  $\sigma_{f_y}^2$ .

Note that  $f_x = f/dx$ ,  $f_y = f/dy$  where  $dx$  and  $dy$  are the digitization horizontal (respectively vertical) steps.

The global covariance matrix on the normal is thus given by:

$$C(n_x, n_y, n_z) = J \begin{pmatrix} \sigma_a^2 & \text{Cov}(a, b) & \text{Cov}(a, c) & 0 & 0 & 0 & 0 \\ \text{Cov}(a, b) & \sigma_b^2 & \text{Cov}(b, c) & 0 & 0 & 0 & 0 \\ \text{Cov}(a, c) & \text{Cov}(b, c) & \sigma_c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{u_0}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_0}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{f_x}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{f_y}^2 \end{pmatrix} {}^t J,$$

where  $J$  is the Jacobian matrix of the transformation relating the unit normal vector  ${}^t(n_x, n_y, n_z)$  to the parameters  $(a, b, c, u_0, v_0, f_x, f_y)$ .

*Errors computing  $n$  normals:* Let  $C_i$  be the covariance matrix on segment  $s_i$  of the image, this for  $i = 1 \dots n$ . The covariance matrix  $\mathbf{W}_k$  on the  $n$  normals  $N_i$  corresponding to the  $n$  segments  $s_i$  can be written:

$$W_k = J \begin{pmatrix} C_1 & (0)_{3 \times 3} & (0)_{3 \times 3} & (0)_{3 \times 3} & (0)_{3 \times 3} & (0)_{3 \times 1} & (0)_{3 \times 1} & (0)_{3 \times 1} & (0)_{3 \times 1} \\ (0)_{3 \times 3} & C_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & (0)_{3 \times 3} & \ddots & (0)_{3 \times 3} & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & (0)_{3 \times 3} & C_{n-1} & (0)_{3 \times 3} & \vdots & \vdots & \vdots & \vdots \\ (0)_{3 \times 3} & (0)_{3 \times 3} & (0)_{3 \times 3} & (0)_{3 \times 3} & C_n & (0)_{3 \times 1} & (0)_{3 \times 1} & (0)_{3 \times 1} & (0)_{3 \times 1} \\ (0)_{1 \times 3} & \dots & \dots & \dots & (0)_{1 \times 3} & \sigma_{u_0}^2 & 0 & 0 & 0 \\ (0)_{1 \times 3} & \dots & \dots & \dots & \vdots & 0 & \sigma_{v_0}^2 & 0 & 0 \\ (0)_{1 \times 3} & \dots & \dots & \dots & \vdots & 0 & 0 & \sigma_{f_x}^2 & 0 \\ (0)_{1 \times 3} & \dots & \dots & \dots & (0)_{1 \times 3} & 0 & 0 & 0 & \sigma_{f_y}^2 \end{pmatrix} {}^t J,$$

where  $J$  is the Jacobian matrix of the transform relating unit normal vectors  $(N_i)_{i=1 \dots n}$  to the parameter vector:  $((s_i)_{i=1 \dots n}, u_0, v_0, f_x, f_y)$ .

The Jacobian matrix is very similar to the one of the previous paragraph. Its dimensionality is  $3n \times (3n+4)$ , the column  $\#i$  only contains 7 non null components, corresponding to partial derivatives of the  $\#i$  equation related to components  $n_{xi}$ ,  $n_{yi}$ ,  $n_{zi}$  of  $N_i$ , and  $a_i, b_i, c_i, u_0, v_0, f_x, f_y$ .

### 3.3 Errors in computing final localization of a polyhedral object

As it was seen previously, the final determination of the pose of a polyhedral object is done by iteratively improving the model attitude by minimizing the distances of the edges of the model to the interpretation planes of the corresponding image segments.

As it is quite impossible to achieve analytical computation of the covariance matrix on the six attitudes parameters, the computation of this matrix is done using the Kalman approach described in section 2, the covariance matrix about the parameters of the matched image segment and the corresponding normals to their interpretation planes being computed as stated previously in this section.

## 4 The tracking algorithm.

The location algorithm is part of a process which is used to track the 3D-location of an object along an image sequence. Each time the previous localization method finds the spatial pose of an object and the associated covariance matrix compatible with the current image content, it will pass these results to a track process that will predict the pose of the object at the time of the next shot and enable to perform robust automatic matching.

The process being a standard Kalman filter we will only give the characteristics of the filter.

**State.**  $\mathbf{A}_k = {}^t(\alpha_k, \beta_k, \gamma_k, u_k, v_k, w_k, S_{\alpha_k}, S_{\beta_k}, S_{\gamma_k}, S_{u_k}, S_{v_k}, S_{w_k})$  is the state in which:  $\alpha_k, \beta_k, \gamma_k$ , are the three Euler angles of rotation,  $u_k, v_k, w_k$  the three coordinates of the translation,  $S_{\alpha_k}, S_{\beta_k}, S_{\gamma_k}$  the angular velocities, and  $S_{u_k}, S_{v_k}, S_{w_k}$  the translational velocities.

**Measure.** The measure is given by:  $\mathbf{Y}_k = {}^t(\alpha_k^m, \beta_k^m, \gamma_k^m, u_k^m, v_k^m, w_k^m)$ .

**State prediction matrix.** The state prediction matrix is given by:

$$\Phi = \begin{pmatrix} \mathbf{I}_{6 \times 6} & \mathbf{I}_{6 \times 6} \\ (0)_{6 \times 6} & \mathbf{I}_{6 \times 6} \end{pmatrix}, \text{ this is a constant velocity model.}$$

**Measure prediction matrix.** The measure prediction matrix is given by:

$$\mathbf{M}_k = (\mathbf{I}_{6 \times 6} \quad (0)_{6 \times 6}).$$

The output of the tracking process consists in a predicted pose for the next shot and an associated covariance matrix.

Thus, the matching is performed as follows:

- the predicted pose is used to compute the perspective projections of the model ridges in the new image.
- the associated covariance matrix is used to compute the estimation of the error on the two parameters  $(c, \theta)$  (c.f. 3.1) of the supporting line of each of these projections.
- finally for each of these projections, the search for best match in the set of image segments is guided and pruned using the standard deviations on the parameters of the supporting line.

The covariance computation is straightforward.

From two model points  $P_1$  and  $P_2$  for which the pose and covariance ( $C_{loc}$ ) are known, ( $C_{c,\theta} = J_3 J_2 J_1 C_{loc} {}^t J_1 {}^t J_2 {}^t J_3$ ), where  $J_1$ ,  $J_2$  and  $J_3$  are respectively, the Jacobian matrix of the application expressing  $P_1$  and  $P_2$  in function of the pose parameters, the Jacobian matrix of the application expressing the normal vector to the interpretation plane through  $P_1$  and  $P_2$  in function of the coordinates of the two points and the Jacobian matrix of the application expressing the line parameters  $(c, \theta)$  in function of the normal vector and the camera intrinsic parameters.

## 5 Experiments and results of tracking.

The tracking process has been used on sequences of real images. The sequence presented here is composed of 82 images of a MacIntosh mouse.

The objects have been put on a rotating table, and although the motion is far from constant velocity (as well in rotation as in translation), the tracker performs its duty. In particular it can be noted on Figure 2 that the table rotation velocity has been inverted between images 70 and 72, which does not prevent the tracking.

Each image presents the superimposition of the segments extracted from the grey-level image and the model ridges projection.

## 6 Conclusion

We have presented a new method that permits to solve the problem of determination of a modelled 3D-object spatial attitude from a single perspective image and to compute the covariance matrix associated to the attitude parameters. Its principle is based on the interpretation of at least three segments as the perspective projection of linear ridges of the object model and on the iterative search of the model attitude consistent with these projections. The proposed method combines a Newton-Raphson process to rapidly obtain a correct approximation of the attitude, followed by Kalman filtering to compute the associated covariance matrix.

We have also shown that this method can be integrated in a process to track an object through a sequence of monocular images. This tracker is based on Kalman filtering to predict the attitude of the object in the next image. The covariance



matrix associated to the predicted attitude is the cue to build an efficient automatic matcher between image primitives and model ridges.

Our next purpose will be to integrate the method in a multi-camera system and also introduce new kinds of primitives, in particular ellipses. The first obtained results are very promising.

## Bibliography

- [AYA-89] N. AYACHE, "*Vision Stéréoscopique et perception multisensorielle*" InterEditions, Science Informatique 1987.
- [DAU-93] N. DAUCHER, M. DHOME, J.T. LAPRESTÉ, G. RIVES, "*Location accuracy from a Single Perspective Image*", Tech. note 3-93, Electronics Laboratory. Blaise Pascal University, France, february 1993
- [DHO-89] M. DHOME, M. RICHTIN, J.T. LAPRESTÉ, G. RIVES, "*Determination of the Attitude of 3D Objects from a Single Perspective Image*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*", Vol 11, #12, pp 1265-1278, December 1989.
- [HOR-87] R. HORAUD, "*New Methods for Matching 3-D Objects with Single Perspective View*", *IEEE Trans. on PAMI*, vol PAMI-9, n°3, May 1987, pp 401-412.
- [HOR-89] R. HORAUD, B. CONIO, O. LEBoulLEUX, B. LACOLL, "*An Analytic Solution For the Perspective 4 Point Problem*". CVGIP 1989 Academic Press.
- [HUT-87] D.P. HUTTENLOCHER & S. ULLMAN, "*Object Recognition using Alignment*", *Proc. of the First Int. Conf. on Computer Vision*, London, June 1987, pp 102-111.
- [KAN-81] T. KANADE, "*Recovery of the Three Dimensional Shape of an Object from a Single View*", *Artificial Intelligence, Special Volume on Computer Vision*, Vol 17, #1-3, August 1981.
- [LOW-85] D.G. LOWE, "*Perceptual Organization and Visual Recognition*", Kluwer, Boston, 1985, chapter 7.
- [PAV-77] T. PAVLIDIS, "*Structural Pattern Recognition*", Springer-Verlag New-York 1977.
- [RIV-81] P. RIVES, P. BOUTHEMY, B. PRASADA, E. DUBOIS, "*Recovering the Orientation and Position of a Rigid Body in Space From a Single View*." Technical Report, INRS-Télécommunications, 3, place du commerce, Ile-des-soeurs, Verdun, H3E 1H6, Québec, Canada, 1981.
- [SHA-86] T. SHAKUNAGA & H. KANEKO, "*Perspective Angle Transform and Its Application to 3-D Configuration Recovery*", *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, Miami Beach, Florida, June 1986, pp 594-601.

