2D Model-Based Tracking of Complex Shapes for Visual Servoing Tasks

Nathalie Giordana[1], Patrick Bouthemy[1], François Chaumette[1],
Fabien Spindler[1], Jean-Claude Bordas[2], Valéry Just[2]

[1]IRISA / INRIA Rennes
Campus universitaire de Beaulieu,
35042 Rennes Cedex, France

[2]DER-EdF
6, Quai Watier,
78401 Chatou Cedex, France

Abstract

Visual servoing needs image data as input to realize robotics tasks such as positioning, docking or mobile target pursuit. This often requires to track the 2D projection of the object of interest in the image sequence. To increase the versatility of visual servoing, objects cannot be assumed to carry landmarks. We have developed an original method for 2D tracking of complex objects which can be approximately modeled by a polyhedral shape. The proposed method fulfills real-time constraints as well as reliability and robustness requirements. Real experiments and results on a positioning task with respect to different objects are presented.

Keywords
Computer vision, pose computation, visual servoing, 2D tracking, deformable template, robust estimation, real-time application.

1 Introduction

The visual servoing approach, which consists in controlling movements of a robot from the estimation of image features, is attractive for industrial use in changing or hostile environments such as a nuclear power plant. In order to follow this approach, the extracted image information must be robust, accurate, and computed in real-time. Current techniques exploited in industrial environment run on marked and simple objects. Our goal is to design a method to extract relevant image features without such constraints. Several authors have proposed ways to solve tracking of features in image sequences with monocular vision [1, 7, 8, 10, 14] or stereo vision [3]. We have developed an original method for 2D tracking of complex objects which can be approximately modeled by a polyhedral shape. The efficiency of this method is demonstrated through a visual servoing homing task which consists in positioning a camera mounted on the end effector of a six d.o.f cartesian robot with respect to objects.
The paper is organized as follows. In Section 2, we briefly recall the visual servoing approach, and we specify the considered task. Section 3 will describe the initialization step of the tracking algorithm. In Section 4, we present the tracking algorithm which relies on 2D global parametric motion model and 2D deformable template. Experimental results are reported in Section 5. Section 6 contains concluding remarks.

2 Specification of the homing task

2.1 Image-based visual servoing

The image-based visual servoing approach consists in specifying a task as the regulation in the image of a set of visual features [6, 9]. An other approach consists in using a model of 2D image motion [16]. Let us denote p the visual features involved for the task. The task function is defined by:

$$e = \hat{L}^{T^+}(p(t) - p^*) \qquad (1)$$

where:
- $p(t)$ is the current value of the considered image features e.g. coordinates of the particular object points;

- p^* is the desired value of p;
- $\hat{L}^{T^+}$ is the pseudo inverse of a model or an approximation of the interaction matrix L_p^T defined by $\dot{p} = L_p^T T_c$, T_c being the camera velocity.

The goal is to minimize $\|e\|$. In order that e exponentially decreases, the desired evolution of e takes the form:

$$T_c = -\lambda e \tag{2}$$

where λ tunes the speed of convergence.

2.2 Positioning with respect to an object

We have considered a generic homing task that positions an eye-in-hand system with respect to a given object. For this application, we take as p the coordinates of an appropriate set of points on the object silhouette: $p = \{(x_j, y_j), j = 1, \ldots, k\}$, $k \geq 4$. Considering the perspective projection model, a point in the image plane with coordinates (x_j, y_j) corresponds to a 3D point (X_j, Y_j, Z_j) in the camera coordinate system with $x_j = X_j/Z_j$ and $y_j = Y_j/Z_j$.

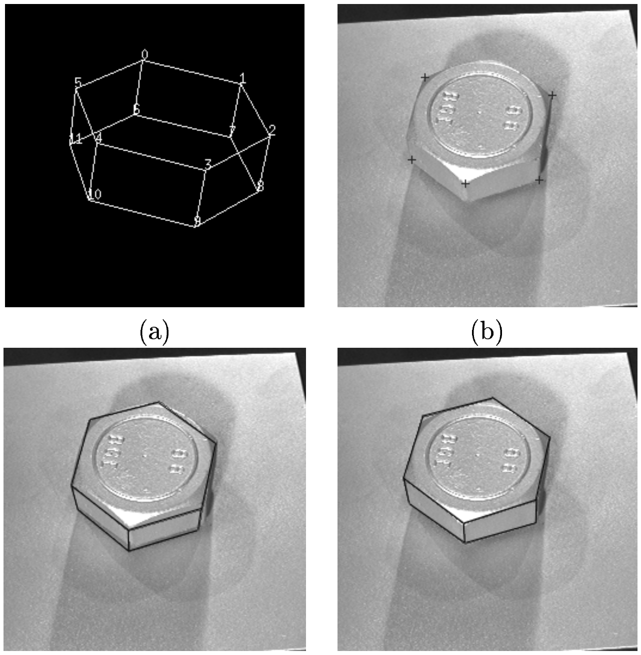
The related interaction matrix is given by:

$$L_p^T = \begin{pmatrix} -1/Z_j & 0 & x_j/Z_j & x_j y_j & -1 - x_j^2 & y_j \\ 0 & -1/Z_j & y_j/Z_j & 1 + y_j^2 & -x_j y_j & -x_j \\ & & & \vdots & & \end{pmatrix} \tag{3}$$

The model $\hat{L}^T$ of the interaction matrix chosen as $L^{T+}_{p=p^*}$, where $Z_j^*, j = 1, \ldots, k$ is obtained by a pose computation, as explained in Section 3.

3 Semi-automatic initialization

To initialize the tracking algorithm, we have to determine a number of control points on the contour of the object projection in the first image of the sequence. These points will then form a polygonal shape which is assumed to correctly model the object appearance in the image. To identify this 2D polygonal shape, we estimate the camera pose from the first image of the sequence. To this end, we exploit a CAD polyhedral 3D model of the object. We have to find the 3D rotation and the 3D translation which map the object coordinate system with the camera coordinate system. The 3D CAD model is then projected onto the image by perspective projection in order to match the silhouette of the object projection in the first image. We use the intrinsic camera parameters given by the maker. A number of methods to compute Perspective from N Points have been proposed [5, 12, 13]. We resort to the method designed by Dementhon and Davis [4]. This method calculates the rigid transformation in an iterative way from the knowledge of the coordinates of at least four non coplanar points, in the object coordinate system, and of their corresponding projections in the image. Its principle consists in approximating perspective projection by scaled orthographic projection, and then iteratively modifying the scaled orthographic projection to converge to the perspective projection. The initialization step is semi-automatic, since the correspondence of at least four non-coplanar points (typically 4 or 5) of the 3D model with image points is achieved in an interactive manual way. Since the faces of the 3D CAD model are oriented by construction, we can determine the visible parts after projection of the 3D model. In order to refine the projected contour obtained after pose calculation, we apply the tracking algorithm presented in Section 4 on the same first image. The points used in p are a subset of points characterizing the projected contour of the object (typically, the corners). An example of initialization step is presented in Figure 1, for one of the real objects we have dealt with.



(a) (b)

(c) (d)

(a) 3D CAD model of the nut
(b) Crosses represent the points selected to calculate the pose of the object
(c) Projected model superimposed on the image
(d) Projected model superimposed on the same image after the refinement step using the tracking algorithm.

Figure 1: An example of initialization step

4 2D tracking of polyhedral object

As described in the previous section, the 2D projection of the object to be tracked is characterized by points on the object contour supplied by the initialization step.

We consider that the 2D global transformation between two successive projections of the object in the image plane can be represented by a 2D affine displacement model augmented with local deformations. The aim is to estimate the parameters of the 2D global transformation.

4.1 Transformation model

Let $X^t = [X_1^t, \ldots, X_n^t]^T$ a vector composed by the image coordinates X_i^t of points along the contour of the object projection at time t, and Γ_{X^t} the contour associated with the vector X^t. Let us denote ${}^lX^t$ the optimal shape of the object projection estimated at time t, and ${}^fX^t$ a filtered version of ${}^lX^t$ (to be defined in subsection 4.2).

The optimal shape ${}^lX^{t+1}$, at time $t+1$, will be given by :

$$ {}^lX^{t+1} = {}^lX^{t+1}(\Theta, \delta) = \Psi_\Theta({}^fX^t) + \delta \qquad (4) $$

where

- Ψ_Θ is a 2D affine transformation given by :

$$ \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \qquad (5) $$

  with $\Theta^T = (a_1, a_2, a_3, a_4, T_x, T_y)$, $X = (x, y)^T$ and $X' = (x', y')^T = \Psi_\Theta(X)$.

- $\delta = (\delta_1, \ldots, \delta_n)$, with $\delta_i = (\delta_{x_i}, \delta_{y_i})$ denotes the local deformation introduced at point X_i. It will be modeled by a centered Gauss-Markov process with variance σ_i and correlation factor ϵ_{ij}.

4.2 Tracking algorithm

The tracking algorithm is articulated into five steps as outlined in fig.2. The first two steps are concerned with the estimation $\hat{\Theta}$ of the global affine parameters Θ. The third step computes the optimal shape ${}^lX^{t+1}$ by minimizing an energy function E_Θ with $\Theta = \hat{\Theta}$. In the fourth step, the model shape denoted ${}^mX^t$, undergoing only the global affine deformation, is computed. Finally, the fifth step delivers the final shape ${}^fX^{t+1}$. It is given by :

$$ {}^fX^{t+1}(\Theta_f, \delta) = \Psi_{\Theta_f}({}^fX^t) + G\,\delta $$

where Θ_f is the 2D affine deformation obtained at step 4, and G is a validation factor of local deformation.

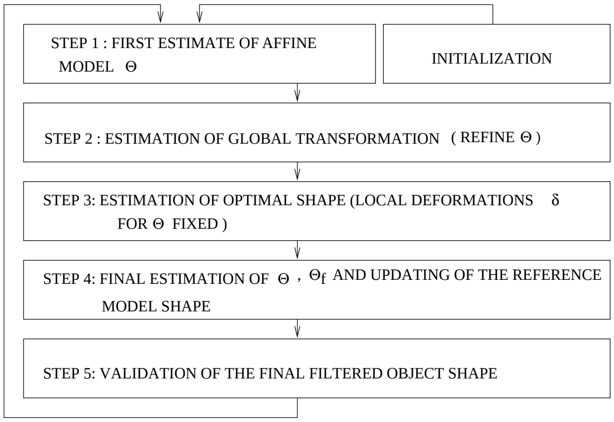

STEP 1 : FIRST ESTIMATE OF AFFINE MODEL Θ
INITIALIZATION
STEP 2 : ESTIMATION OF GLOBAL TRANSFORMATION (REFINE Θ)
STEP 3: ESTIMATION OF OPTIMAL SHAPE (LOCAL DEFORMATIONS δ FOR Θ FIXED)
STEP 4: FINAL ESTIMATION OF Θ , Θf AND UPDATING OF THE REFERENCE MODEL SHAPE
STEP 5: VALIDATION OF THE FINAL FILTERED OBJECT SHAPE

Figure 2: Outline of the tracking algorithm

4.3 Estimation of the model parameters

To estimate the optimal shape ${}^lX^{t+1}$, i.e. (Θ, δ), we adopt a Bayesian criterion, which turns out to lead to a minimization problem. More precisely, the problem is to estimate $(\hat{\Theta}, \hat{\delta})$ by minimizing an energy function E_Θ. For more details, the reader is referred to [11, 15]. This implies that $(\delta_i)_{i=1,\ldots,n}$ are supposed to be of low magnitude. This assumption is verified in our application of visual servoing.

A first step supplies an initial estimate of Θ using as input normal displacements evaluated along the object shape contour with the ECM algorithm [2, 15]. Then, this estimation will be refined as explained below.

E_Θ is decomposed in two terms E_d and E_p :

$$ E_\Theta({}^lX^{t+1}, d^{t+1}) = E_d({}^lX^{t+1}, d^{t+1}) + E_p({}^lX^{t+1}) \quad (6) $$

E_d expresses the adequacy between the variables to be estimated and the observations $d^{t+1} = \{d_s^{t+1}\}$. This is the “data-driven” energy term. The observations d^{t+1} are given by :

$$ d_s^{t+1} = -\min(\|\nabla I_s(t+1)\|, Gr_{max}) \qquad (7) $$

where ∇I_s denotes the spatial gradient of the intensity function at point s along the contour, and Gr_{max} is a threshold which permits to saturate the too high intensity gradient values. E_p represents the *a priori* information on the local deformations δ. This is the regularization term.

The optimal shape of the object projection will be given by ${}^lX^{t+1}(\hat{\Theta}, \hat{\delta})$ where

$$ (\hat{\Theta}, \hat{\delta}) = \arg\min_{(\Theta,\delta)} \{E_d({}^lX^{t+1}, d^{t+1}) + E_p({}^lX^{t+1})\} \quad (8) $$

Let us define energy terms E_d and E_p.

E_d is given by :

$$ E_d({}^lX^{t+1}, d^{t+1}) = \sum_{s \in \Gamma_{{}^lX^{t+1}}} d_s^{t+1} \qquad (9) $$

where $\Gamma_{^lX^{t+1}}$ represents the contour of the 2D shape $^lX^{t+1}$.

Concerning E_p, two deformation processes are in fact introduced. As previously mentioned, we consider the local deformation field δ with variance σ_i^2 and correlation factor ϵ_{ij}. We also take into account the "reference" shape $^mX^t$ which is provided at time t by the transformation of the initial 2D object projection model, resulting only from the combination of successive estimated 2D global affine transformations. Then, $^m\delta$ is given by $^m\delta_i = \Psi_{\hat{\Theta}}(^mX^t) - ^lX^{t+1}$ with variance $^m\sigma_i^2$ and correlation factor $^m\epsilon_{ij}$. The interest of the deformation field $^m\delta$ is to avoid undesirable deformation of the shape over time.

The expression of E_p is thus defined as follows :

$$E_p(^lX^{t+1}) = \sum_i \left(\frac{\rho(\|\delta_i\|)}{\sigma_i^2} + \frac{\|^m\delta_i\|^2}{^m\sigma_i^2} \right) + \sum_{(i,j) neighbour} \left(\frac{\rho(\|\delta_i - \delta_j\|)}{\epsilon_{ij}^2} + \frac{\|^m\delta_i - ^m\delta_j\|^2}{^m\epsilon_{ij}^2} \right) \qquad (10)$$

where ρ is a quadratic truncated function.

Two points indexed by i and j are said "neighbor" if they are located at two successive positions along the shape contour.

The criterion (8) cannot be directly solved. We resort to an alternative iterative procedure. First, we estimate Θ using :

$$\hat{\Theta} = \arg\min_{\Theta} E_d(\Psi_{\Theta}(^fX^t), d^{t+1}) \qquad (11)$$

then, for $\hat{\Theta}$ fixed, we estimate δ using :

$$\hat{\delta} = \arg\min_{\delta} E_{\hat{\Theta}}(^lX^{t+1}, d^{t+1}) \qquad (12)$$

The optimization of E_d is performed by a gradient algorithm, whereas the optimization of E_p is achieved by simulated annealing.

5 Experimental results

The complete experimental implementation and validation of the visual servoing task including initialization and tracking, have been carried out. We have conducted experiments dealing with a positioning task. Several objects of interest have been considered. This task has been performed on an experimental testbed involving a CCD camera mounted on the end effector of a six d.o.f cartesian robot.

The experiment comprises the following steps:

- In an off-line step, the camera is first positioned at the final desired position and a number of points (at least 4) on the object image are selected to specify the control law. The 2D model of the object projection is initialized as explained in Section 3.
- The camera is then positioned at the initial position. The 2D model of the object projection is then also initialized as explained in Section 3.
- At every intermediate camera position between the initial and the final ones, the contour of the object projection in the image is updated by the tracking algorithm presented in Section 4. Then, the control law is activated to reach the next position.

A first real example involving a nut as object of interest is now reported. The tracking algorithm runs on an Ultra-Sparc-1 Sun workstation, equipped with a Sunvideo image capture board, at the rate of 1Hz for images of 256 × 256 pixels. This relative low processing rate implies that the positioning task is specified in position, i.e. $\Delta r = T_c \Delta t$, where Δr is the camera displacement. Otherwise, the control could be performed on the velocity.

Figures 3 and 4 show the temporal evolution of the components of $(p - p^*)$, in pixels, and of T_c, in cm/s and dg/s. These curves show the stability and the convergence of the control law. Indeed, the error on each coordinate of the six points specifying the task and the components of the control vector T_c converge to zero. Figures 5-a and 5-b respectively contain the images delivered by the camera at its initial position and final reached one. Crosses overprinted in the image indicate the target position of the points used to specify the control law. Figure 5-b depicts the apparent trajectory in the image of these points during the achievement of the task. The initial and the final polygonal shape contours accounting for the tracking of the nut projection in the image are also drawn.

We can point out on this example that the tracking of the object contour in the image must tackle with low intensity contrast, presence of cast shadows, mirror specularities... Moreover, the object is not exactly polyhedral, and the object edges cannot be physically precisely defined. Despite these difficulties, the proposed method have proven its efficiency on different classes of objects such as box or nut. Experiments in presence of partial occlusion (fig.6) or possible false matches (fig.7) have been performed with success. However, the tracking method contains some limitations. The method cannot take into account important changes of appearance of the projection of the object in case of large displacements of the camera.

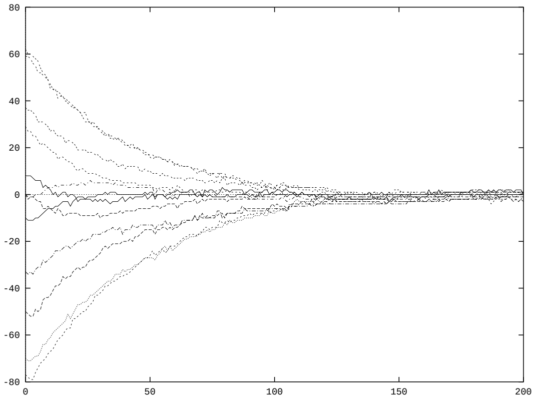


Figure 3: Temporal evolution of $(p - p^*)$ for the nut experiment

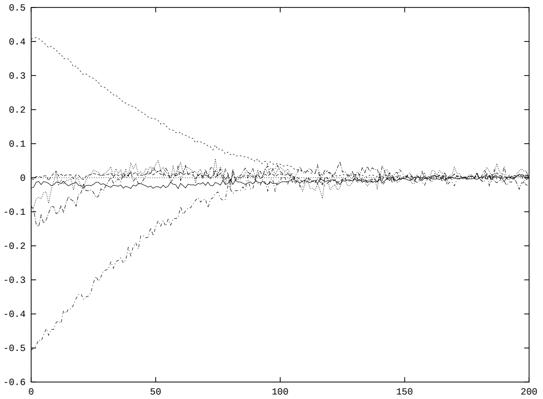
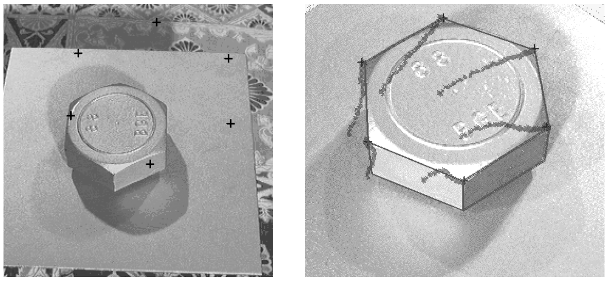


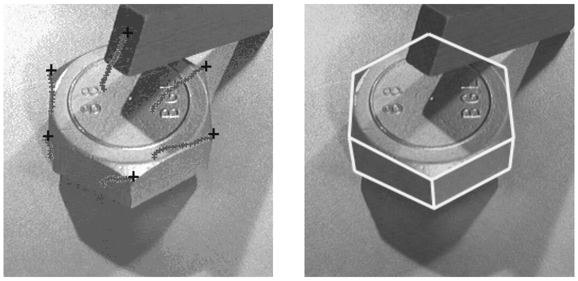
Figure 4: Temporal evolution of T_c for the nut experiment



(a) (b)

(a) Crosses indicating the desired position and plot of the polygonal model contour of the nut projection after initialization - (b) Apparent trajectories of the points used to specify the task, and plot of the contour of the nut projection model at the convergence of the task.

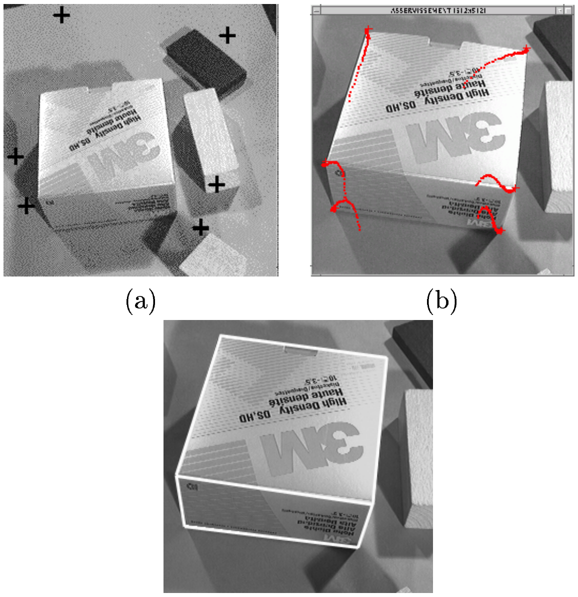
Figure 5: Example of positioning task realization for the nut experiment



(a) (b)

(a) Apparent trajectories of the points used to specify the task - (b) Contour of the nut projection model at the convergence of the task.

Figure 6: Example of positioning task with partial occlusion



(a) (b)

(c)

(a) Crosses indicating the desired position - (b) Apparent trajectories of the points used to specify the task - (c) Contour of the box projection model at the convergence of the task.

Figure 7: Example of positioning task on a box with possible false matches

6 Conclusions

We have presented an original method for tracking complex objects in an image sequence. It allows us to carry out visual servoing task of positioning with respect to real objects (without any landmarks). Initialization of the algorithm is based on pose computation while exploiting the 3D CAD model of the object of interest. The tracking is based on the estimation, between two successive images, of a global affine transformation augmented with local deformations. It is formulated within a Bayesian framework. A real practical implementation has been realized. Results on different examples of positioning task have demonstrated the robustness and the reliability of the proposed method. In order to increase the processing rate, several improvements are under investigation. For instance, the tracking stage could exploit the 3D model of the object of interest. This could avoid to estimate any local deformations, which represents the main part of the computational load. The handling of the appearance in the image of previously hidden object parts will also be considered.

Acknowledgments : This study is supported by DER-EDF under contract 1.97.C234.00.

References

[1] A. Blake, R. Curwen and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127-145, 1993.

[2] P. Bouthemy. A maximum-likelihood framework for determining moving edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp.499-511, May 1989.

[3] P. Braud, M. Dhome, J-T. Lapresté, B. Peuchot. Reconnaissance, localisation et suivi d'objets polyédrique par vision multi-oculaire. *Technique et Science Informatiques*, vol.16, no.1, pp.9-37, 1997.

[4] D. Dementhon and L. Davis. Model-Based Object Pose in 25 Lines of Codes. *International Journal of Computer Vision*, vol. 15 pp. 123-141, 1995.

[5] M. Dhome, M. Richetin, J-T. Lapresté and G. Rives. Determination of the Attitude of 3-D Objects from a Single Perspective View. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265-1278, Dec. 1989.

[6] B. Espiau, F. Chaumette, P. Rives. A new appraoch to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313-326, june 1992.

[7] D. B. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, vol. 7, no.3, pp. 243-270, 1992.

[8] G.D. Hager and K. Toyama. X Vision : A portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, vol.69, no.1, pp. 23-37, Jan. 1998.

[9] S. Hutchinson, G. D. Hager and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, vol. 12, no.5 pp. 651-670, Oct. 1996.

[10] M. Isard, A. Blake. Contour tracking by stochastic propagation of conditional density. *ECCV'96*, LNCS no. 1064, Springer-Verlag, pp. 343-356.

[11] C. Kervrann, F. Heitz. A hierarchical statistical framework for the segmentation of deformable objects in image sequences. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 724-728, Seattle, USA, June 1994.

[12] R. Kumar and A. R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding*, vol. 60, no. 3, pp. 313-342, Nov. 1994.

[13] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, vol 31, pp 355-394, 1987.

[14] C. Meilhac and C. Nastar. Robust fitting of 3D CAD models to video streams. *International Conference on Image Analysis and Processing ICIAP'97*, Florence, Italy, Sept. 1997.

[15] J-M. Odobez, P. Bouthemy, E. Fleuet. Suivi 2D de pièces métalliques en vue d'un asservissement visuel, *11ème congrès RFIA'98*, vol.2, pp. 173-182, Clermont-Ferrand, Jan. 1998.

[16] V. Sundareswaran, P. Bouthemy, F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *Int. Journal of Robotics Research*, vol. 15, no. 6, pp. 629-645, Dec. 1996.