Proceeding of the IEEE Int. Conf. on Image Processing, ICIP'99.
Kobe, Japan, October 1999

Robust visual tracking by coupling 2D motion and 3D pose estimation

Éric Marchand[1], Patrick Bouthemy[1], François Chaumette[1], Valérie Moreau[2]

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

[2]EDF - Pôle Industrie -
Division Recherche et Développement
6, Quai Watier
78401 Chatou Cedex, France

Abstract

We present an original method for tracking in an image sequence complex objects which can be modeled approximately by a polyhedral shape. The approach relies on the estimation of the object image motion as well as the computation of the object pose. The proposed method fulfills real-time constraints along with reliability and robustness requirements.

1 Introduction

This paper describes an original method for a robust and fast tracking of complex objects in an image sequence. Most of the available tracking techniques can be classified as feature-based or model-based ones. The former approach tracks features such as geometrical primitives (points, segments [2, 8], circle,...), object contours [1, 9], regions of interest [8],... The latter explicitly uses a model of the tracked objects. This model can be a CAD model [6, 11, 12, 4] or a 2D template of the object [10]. This second class of methods usually provides a more robust tracking (for example it can cope with partial occlusion of the objects). Both approaches may use Kalman filters to predict and estimate the position of the tracked primitives over time.

We have developed a method involving an estimation of the 2D object motion and an estimation of the 3D pose of the object. It supplies a fast and robust tracking of complex objects which can be approximately modeled by a polyhedral shape. A 2D affine motion model is estimated, using a robust statistical method, from normal displacements computed along the object shape contours with the algorithm described in [3]. The 2D affine motion model does not always match the real displacement of the object. A second step that consists in fitting the projection of the object shape on the intensity gradients in the image is necessary. This is achieved using an iterative minimization of a non-linear energy function with respect to 3D pose parameters.

The main advantages of this two-steps method can be summarized as follows. The motion estimation step allows us to handle large displacements of the object and to avoid a prediction step. The result of this step is exploited to provide an appropriate initialization to the pose estimation. Our model-based tracking only requires a coarse calibration of the camera and a rough model of the object. Both 2D motion estimation and 3D pose estimation do not involve edge detection (we only consider gray level images). Both are robust to partial occlusions of the object. Finally, the algorithm supplies a real-time tracking (currently 10Hz). The efficiency of this method is demonstrated through various real experiments.

This algorithm has been proposed in order to extract reliable information from an image sequence for visual servoing purpose. Such systems are of interest for the research and development division of EdF (Électricité de France) to achieve intervention and monitoring task in hostile environment (nuclear power plant).

2 Motion-based tracking

We first consider that the global transformation between two successive projections of the object in the image plane can be represented by a 2D affine motion model. The goal of this first step is to estimate the parameters of this 2D transformation while being able to handle large 2D displacements of the object image. Compared to usual Kalman filter methods used to predict the new position of the object [6, 10, 11], this motion-based method does not require the introduction of a state model evolution (*e.g.*, a constant velocity model), nor the initialization of the variance of the noise on the state and measurement models.

Affine transformation model. Let $\mathcal{X}^t = [X_1^t, \ldots, X_n^t]^T$ be a vector formed by the image coordinates X_i^t of points along the boundaries of the

object model projection at time t. The object image shape $\mathcal{X}^{t+1}$ at time $t+1$ will be given by:

$$\mathcal{X}^{t+1} = \Psi_\Theta(\mathcal{X}^t) \tag{1}$$

where Ψ_Θ is a 2D affine transformation expressed as:

$$\begin{bmatrix} x_i^{t+1} \\ y_i^{t+1} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x_i^t \\ y_i^t \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} = \mathbf{W}(X_i^t)\Theta \tag{2}$$

with $\Theta = (a_1, a_2, a_3, a_4, T_x, T_y)^T$, $X_i^t = (x_i^t, y_i^t)^T$, $X_i^{t+1} = \Psi_\Theta(X_i^t)$, and

$$\mathbf{W}(X) = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix}$$

This transformation is linear wrt. Θ, and displacement $d_i(X_i) = X_i^{t+1} - X_i^t$ can be written as:

$$d_i(X_i) = \mathbf{W}(X_i)\Theta' \tag{3}$$

where $\Theta' = \Theta - (1,0,0,1,0,0)^T$.

The part of the tracking algorithm concerned with the estimation of the parameters Θ', is articulated into two sub-steps. The first one computes normal displacements evaluated along the projections of the object shape contours using the so-called Moving Edges algorithm (ME) [3], while the second one exploits this normal displacement field to estimate $\widehat{\Theta}'$ using an extension of the robust multiresolution estimation technique introduced in [13]. We now describe these two sub-steps.

Computing normal displacements. One of the advantages of the ME method is that it does not require any edge extraction. Furthermore, it can be implemented with convolution efficiency and leads to real-time computation [2, 3].

We aim at designing an algorithm that is fast, reliable, robust to partial occlusions and to false matches. We consider a list L^t of pixels along the projection of the object model contours (model fitting in the first image is performed in a semi-automatic mode). The process consists in searching for the "correspondent" P_i^{t+1} in image I^{t+1} of each point $P_i^t \in L^t$. We determine a 1D search area $Q_i^j, j \in [-J, J]$ in the direction δ of the normal to the contour. For each point P_i^t of the list L^t, and for each position Q_i^j lying in the direction δ, we compute the square root of a log-likelihood ratio ζ^j which is nothing but the absolute sum of convolution values computed at P_i^t and Q_i^j using a predetermined mask M_δ function of the orientation of the contour. New position P_i^{t+1} is given by:

$$Q_i^{j^*} = \arg \max_{j \in [-J,J]} \zeta^j \text{ with } \quad \zeta^j = \mid I^t_{\nu(P_i)} * M_\delta + I^{t+1}_{\nu(Q_i^j)} * M_\delta \mid$$

providing that ζ^{j^*} is greater than a threshold. $\nu(.)$ is the neighborhood of the considered pixel. Then pixel P_i^{t+1} given by $Q_i^{j^*}$ is stored in L^{t+1}.

At this step we have a list of k pixels as well as their displacement component orthogonal to the object model contour: $(P_i^t, d_i^\perp)_{i=1\ldots k}$. This local approach allows us to be robust to partial occlusions of the object and to missing measurements.

Affine transformation estimation. Using $(P_i^t, d_i^\perp)_{i=1\ldots k}$, we can estimate the 2D affine transformation Θ'. From equation (3), we have:

$$d_i^\perp = \mathbf{n}_i^T \mathbf{d}(P_i) = \mathbf{n}_i^T \mathbf{W}(P_i)\Theta' \tag{4}$$

where n_i is a unit vector orthogonal to the object shape contour at point P_i. From (4), we can use a robust estimator (an M-estimator ρ) to obtain $\widehat{\Theta}'$ as follows:

$$\widehat{\Theta}' = \arg\min_{\Theta'} \sum_{i=1}^{k} \rho(d_i^\perp - \mathbf{n}_i^T \mathbf{W}(P_i)\Theta')$$

This robust statistical approach enables not to be affected by locally incorrect measures (due to shadows, miss-matching, occlusions, etc.)

3 Model-based tracking

Knowing the positions $\mathcal{X}^t$ of the projection of the contours of the tracked object at time t and the estimation $\widehat{\Theta}'$ of the global affine motion parameters between t and $t+1$, we are able to compute the position of points $\mathcal{X}^{t+1}$ at time $t+1$:

$$\mathcal{X}^{t+1} = \Psi_{\widehat{\Theta}}(\mathcal{X}^t)$$

However, the 2D affine transformation cannot completely account for the real transformation undergone by the projection of the object (due to perspective effects, important rotations, non shallow environment), and after a few iterations tracking may fail. To alleviate this problem, in a first version of this algorithm [7], the 2D affine displacement model was augmented with local 2D deformations. However, when adding local deformations, we cannot ensure rigidity constraints. Moreover, this was highly time consuming. Therefore, we prefer to exploit explicitly a rough CAD polyhedral model of the object. We have to find the 3D rotation and the 3D translation (*i.e.*, pose Φ) that map the object coordinate system with the camera coordinate system. Once the pose parameters are available, it is easy to determine visible and invisible faces.

A number of methods to compute pose (perspective from N points) have been proposed. We need to estimate the pose of the object wrt the camera from the

positions $\mathcal{X}^{t+1}$ obtained after the first step of the algorithm, described in Section 2. In practice, we use the method designed by Dementhon [5] followed by Lowe's method [12]. We therefore get a first estimate of the pose parameters Φ_{init}^{t+1} that has to be still updated to correspond as well as possible to the real new aspect of the object. This further step consists in fitting the projection of the object model shape on the intensity gradients in the image. This is achieved using an iterative minimization of a non-linear energy function wrt Φ, using Φ_{init}^{t+1} as initialization.

More precisely, we estimate the pose parameters $\widehat{\Phi}$ as follows:

$$\widehat{\Phi} = \arg\min_{\Phi} \left\{ E(d_{\Phi}^{t+1}) \right\} \tag{5}$$

where the energy function $E(d_{\Phi}^{t+1})$ is defined as:

$$E(d_{\Phi}^{t+1}) = -\int_{\Gamma_{\Phi}} \|\nabla I_{\pi_{\Phi}(s)}(t+1)\| ds \tag{6}$$

with

- Γ_{Φ} representing the contours of the visible part of the 3D object model for the pose Φ.
- $\nabla I_{\pi_{\Phi}(s)}$ denoting the spatial gradient of the intensity function at 2D point $\pi_{\Phi}(s)$ along the contour $\pi_{\Phi}(\Gamma_{\Phi})$ where π_{Φ} is the perspective projection function.

The projection function π_{Φ} depends on the camera intrinsic parameters $\mathcal{I}$. The minimization of the energy function (5) requires that the camera calibration is available. Nevertheless a rough knowledge of the camera parameters is sufficient. These parameters can also be estimated on-line. In that case, the function to be minimized can be rewritten as follows:

$$\left(\widehat{\Phi}, \widehat{\mathcal{I}}\right) = \arg\min_{(\Phi,\mathcal{I})} \left\{ E(d_{(\Phi,\mathcal{I})}^{t+1}) \right\} \tag{7}$$

In the general case, we have 11 parameters to estimate (if we consider the radial distortion). In practice, we have only performed experiments dealing with the on-line estimation of the radial distortion.

We have paid particular attention to the discretization problem of E (especially from a real-time implementation point of view). The optimization of E is performed by an appropriate search algorithm in the pose parameters space.

4 Experimental results

Most of the experiments reported here involve a nut as the object to be tracked. We can point out that the tracking of the nut silhouette in the considered image sequences has to cope with low intensity contrast, presence of cast shadows, mirror specularities,... Moreover, the nut is not exactly polyhedral, since it presents no physically precisely defined ridges. Besides, camera calibration is not precisely known.

Figure 1 contains results of the tracking of the nut along a sequence of 44 images. Figure 1.a shows the results of the tracking if we consider only the 2D motion estimation step. In that case, tracking is performed at video rate (25Hz). However, after a few images the algorithm is no longer able to track accurately the object shape. The failure is mainly due to the fact that 2D affine motion model cannot account for the perspective effects. Figure 1.b reports the results of the tracking using both 2D motion estimation and 3D pose computation. In that case, the tracking is performed at 10 Hz and results are quite satisfactory.

In order to demonstrate the robustness and the efficiency of the proposed tracking method, we have performed real experiments involving various typical difficulties. In Figure 3.a, a camera motion is performed around the y axis[1]. A face of the nut appears while another disappears. In Figure 3.b, the main difficulty is the very important rotation around the x axis. The pose estimation becomes complex. Furthermore, the illumination conditions are not constant along the sequence. In Figure 3.c, the difficulties lie in the multiple occlusions of the nut. In Figure 3.d, the nut is tracked within a highly textured environment. We have also tried to estimate on-line the radial lens distortion. We have considered a simple object (a box) and a camera with an important distortion (3.5mm lens) with initial value set to 0 (see Figure 2). We estimate on-line the distortion. It decreases toward its reference value when the object projection moves toward the image border (since then a better estimation of this parameter is then required). In that case, the tracking is performed at 1 Hz.

5 Conclusion

We have presented an original method for tracking complex objects in an image sequence at a high processing rate. The tracking is based on the estimation, between two successive images, of a global affine transformation augmented with an estimation of the object pose, formulated as an energy minimization process. To achieve this last step, a 3D approximate polyhedral model of the object is sufficient. Appearance and disappearance of hidden faces of the object can be straightforwardly handled. Both steps of the algorithm are robust to partial occlusions. The proposed method can deal with real objects (without any

[1] z axis follows the optical axis while x axis is parallel to the image rows and y axis is parallel to image columns.

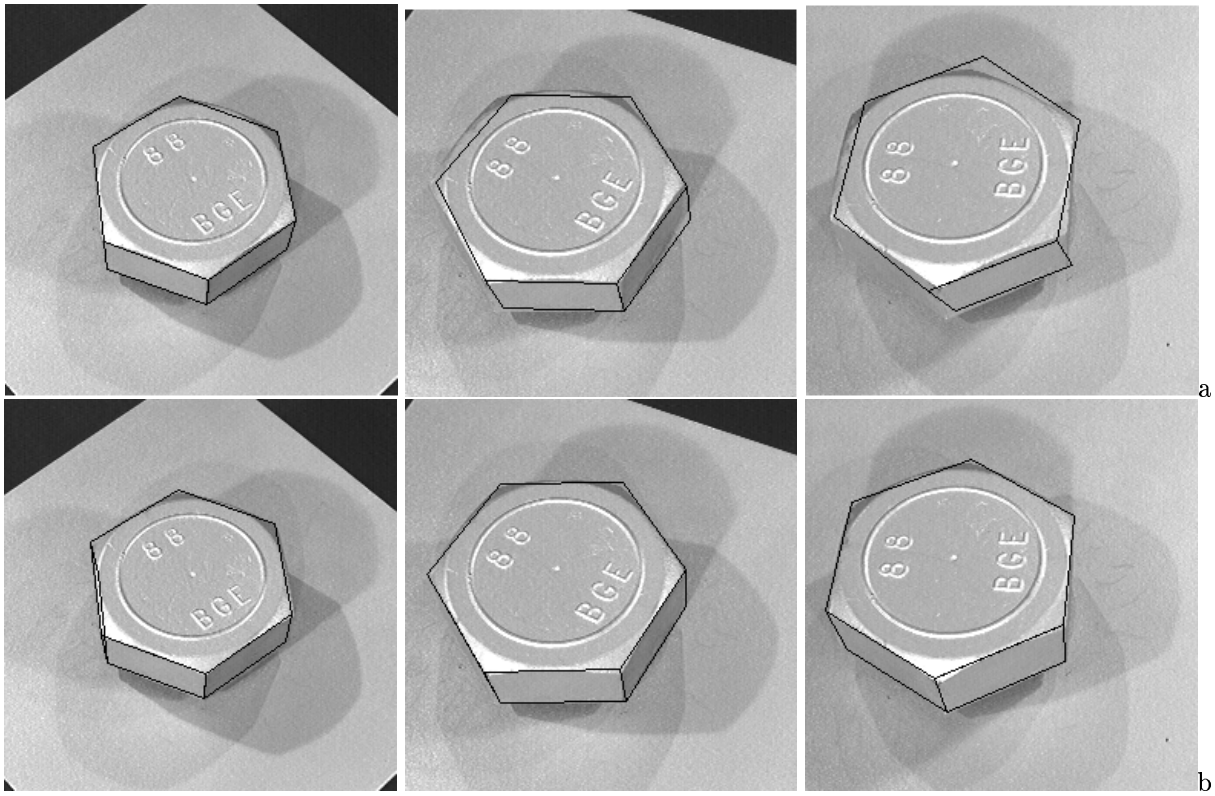

Figure 1: Nut tracking: (a) tracking with only 2D motion estimation (b) tracking with both 2D motion estimation and 3D pose computation

landmarks) in complex situations.

Acknowledgments. This study is supported by EDF (Pôle Industrie – Division Recherche et Développement) under contract 1.97.C234.00.

References

[1] B. Bascle, P. Bouthemy, N. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. In *ICPR'94*, pp. 426–431, Jerusalem, October 1994.

[2] S. Boukir, P. Bouthemy, F. Chaumette, and D. Juvin. A local method for contour matching and its parallel implementation. *Machine Vision and Application*, 10(5/6):321–330, April 1998.

[3] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. on PAMI*, 11(5):499–511, May 1989.

[4] N. Daucher, M. Dhome, J.T. Lapreste, and G. Rives. Modelled object pose estimation and tracking by monocular vision. In *British Machine Vision Conference, BMVC'93*, pages 249–258, Guildford, UK, September 1993.

[5] D. Dementhon, L. Davis. Model-based object pose in 25 lines of codes. *Int. J. of Computer Vision*, 15:123-141, 1995.

[6] D.B. Gennery. Visual tracking of known three-dimensional objects. *Int. J. of Computer Vision*, 7(3):243–270, 1992.

[7] N. Giordana, P. Bouthemy, F. Chaumette, F. Spindler, J.-C. Bordas, V. Just. 2D model-based tracking of complex shapes for visual servoing tasks. In G. Hager and M. Vincze, editors, *IEEE Workshop on Robust Vision for Vision-Based Control*, Leuven, May 1998.

[8] G. Hager, K. Toyama. The X-Vision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, Jan. 1998.

[9] M. Isard, A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV'96, LNCS 1064,* Springer-Verlag, pp. 343–356, Cambridge, 1996.

[10] C. Kervrann, F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60(3):173–195, May 1998.

[11] H. Kollnig, H.-H. Nagel. 3D pose estimation by fitting image gradients directly to polyhedral models *IEEE Int. Conf. on Computer Vision*, pp. 569-574, Boston, May 1995.

[12] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Int. J. of Computer Vision*, 8(2):113–122, 1992.

[13] J.-M. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348-365, December 1995.

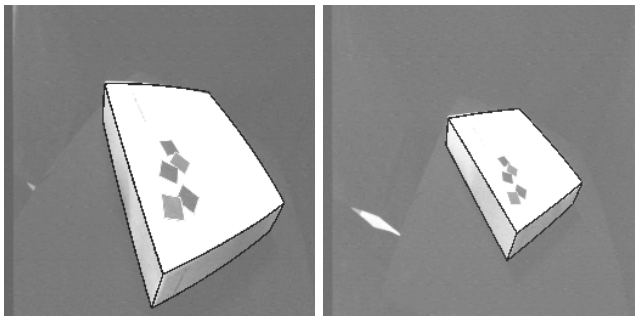

Figure 2: Box tracking: Distortion is very important due to a 3.5mm lens

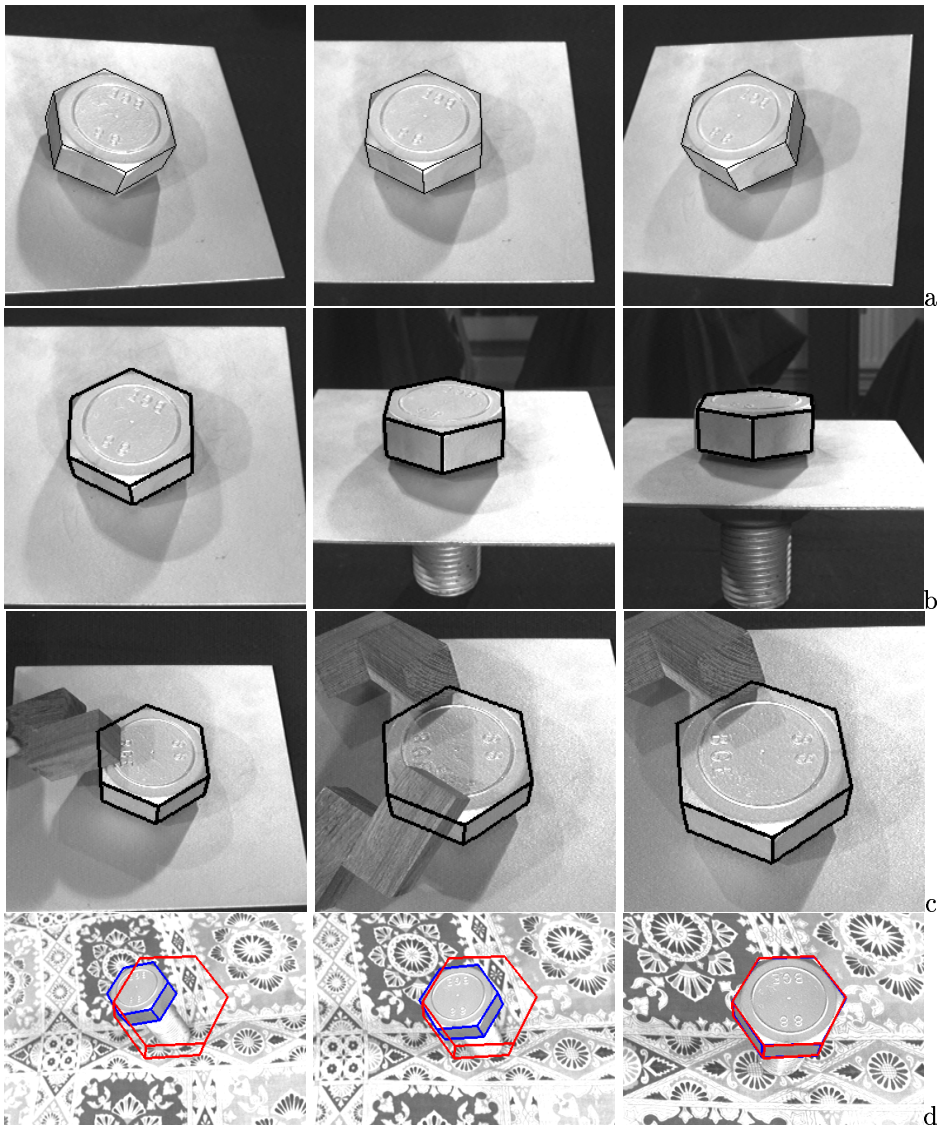

Figure 3: For different experiments (a, b, c, d) nut tracking featuring various difficulties (see text for details), quite satisfactory results are obtained (three distinct instants of the sequence are only displayed for each example).