A MODEL FREE HYBRID ALGORITHM FOR REAL TIME TRACKING

Muriel Pressigout and Éric Marchand

IRISA-INRIA and University of Rennes I
Rennes, France

ABSTRACT

Robustness and accuracy are major issues in real-time tracking. This paper describes a reliable tracking for markerless planar objects based on the fusion of visual cues and on the estimation of a 2D transformation. Its parameters are estimated by a non-linear minimization of an unique criterion that integrates information on both texture and edges. The efficiency and the robustness of the proposed method are tested on image sequences as well as during a robotic application.

1. INTRODUCTION

Elaboration of real-time object tracking algorithms in image sequences is an important issue for numerous applications related to computer vision, robotic, etc. For the time-being, most of the available tracking techniques that don't require a 3D model can be divided into two main classes: edge-based and texture-based tracking. The edge-based tracking relies on the high spatial gradients outlining the contour of the object or some geometrical features of its pattern (points, lines, circles, distances, splines, ...). When 2D tracking is considered, such edge points enable to defined the parameters of some geometrical features (such as lines, splines,...) and the position of the object is defined by the parameters of these features [6]. Snakes or active contours are also based on high gradients and can be used to outline a complex shape [3]. These edge-based techniques have proved to be very effective for applications that required a fast tracking approach. On the other hand, they have also often proved to fail in the presence of highly textured environments. Previous approaches rely mainly on the analysis of intensity gradients in the images. When the scene is too complex, other approaches are required. Another possibility is to directly consider the image intensity and to perform 2D matching on a part of the image without any feature extraction by minimizing a given correlation criterion: we then refer to template-based tracking or motion estimation (according to the problem formulation). It is possible to solve the problem using efficient minimization techniques that are able to consider quite complex 2D transformations [2, 5, 9]. Let it be noted that these methods are closely related to classical image motion estimation algorithms [12]. Such tracking techniques are also fast and reliable when a "good" texture is present on the tracked object but fail otherwise.

As mentionned above, these two classes of approaches have complementary advantages and drawbacks. In order to develop algorithms robust to aberrant measurements and to potential occlusions it is interesting to take into account visual information related to these different types. They can be used sequentially, in order to combine robustness and accuracy, as in [1, 4, 10]. Other approaches rely on probabilistic frameworks. In [14], the authors consider a texture-based approach to find the projected contour of a 3D object. Some classical single-cue trackers, such as the CONDENSATION algorithm, have been extended to multi-cue tracking [8].

The method presented here however integrates simultaneously both approaches. Since both edge and texture tracking algorithms can be seen as optimization algorithms, our goal was to define a unique state vector that describes both the appearance of the template as well as its edge boundaries. Considering this state vector, we will be able to compute the parameters of a 2D transformation that minimizes the error between a current multi-cue template and the displaced reference one. A similar approach has been proposed by [11]. In this latter work, the template matching algorithm is handled using a texture-based tracking algorithm [9] (the Jacobian is then learned) whereas ours is related to Hager and Belhumeur's algorithm [5] (we have an explicit formulation of the Jacobian). In [11], edge and texture points are classified according to the eigen-values of the signal autocorrelation matrix. Sharp edges of the texture are then likely to be classified as edge points and as a consequence, the remaining points that are classified as texture points hold little information since they belong to smooth gradient varying regions.

The tracker is presented in section 2. The general framework of the object tracking based on a 2D transformation estimation is developped in subsection 2.1 and details about the edge-based and texture-based features are given respectively in subsection 2.2 and 2.3. Finally, some experiments in section 3 will illustrate the behavior of the tracker on real image sequences.

0-7803-9134-9/05/$20.00 ©2005 IEEE

2. TRACKING BASED ON 2D TRANSFORMATION ESTIMATION

2.1. Hybrid tracking: general framework

The 2D transformation of the object between image $\mathbf{I}^{t-1}$ and image $\mathbf{I}^t$ is such as if $\mathbf{x}_{\mu_t} = (x_{\mu_t}, y_{\mu_t})^T$ is a point in $\mathbf{I}^t$ belonging to the object and $\mathbf{x}_{\mu_{t-1}}$ its corresponding point in $\mathbf{I}^{t-1}$, then :

$$\mathbf{x}_{\mu_t} = \mathbf{\Psi}_{\mu_t}(\mathbf{x}_{\mu_{t-1}}) \tag{1}$$

where $\mathbf{\Psi}_{\mu_t}$ is the 2D transformation which can be described by M parameters. For an homography, using homogeneous coordinates, one has :

$$\mathbf{x}_t^h = \begin{pmatrix} \mu_0 & \mu_1 & \mu_2 \\ \mu_3 & \mu_4 & \mu_5 \\ \mu_6 & \mu_7 & \mu_8 \end{pmatrix} \mathbf{x}_{t-1}^h \tag{2}$$

Therefore the parameters to be estimated are :

$$\mu = \begin{pmatrix} \mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8 \end{pmatrix}^T \tag{3}$$

Let $\mathbf{m}_\mu$ denote the column vector of dimension N that stored the value of the image features according to the 2D transformation parameters μ: $\mathbf{m}_\mu = (m_\mu^1, \ldots, m_\mu^N)^T$. Its ground truth value in $\mathbf{I}^t$ is noted $\mathbf{m}_{\mu_t^*}$ and its current value in $\mathbf{I}^t$ estimated according to the 2D transformation estimation μ_t is denoted by $\mathbf{m}_{\mu_t}$.

The key idea is to estimate the 2D transformation that verifies (1). This is achieved by minimizing the error between the current value $\mathbf{m}_{\mu_t}$ and its value $\mathbf{m}_{\mu_t^*}$ observed in the current image $\mathbf{I}^t$ to be estimated :

$$\widehat{\mu}_t = \underset{\mu_t}{\operatorname{argmin}} \parallel \mathbf{m}_{\mu_t} - \mathbf{m}_{\mu_t^*} \parallel^2 \tag{4}$$

Motion continuity ensures that $\mu_t = \mu_{t-1} + \delta\mu$. The problem is then to estimate the value $\widehat{\delta\mu}$ that minimizes the error $\mathbf{e}$ defined by: $\mathbf{e} = \parallel \mathbf{m}_{\mu_{t-1}+\widehat{\delta\mu}} - \mathbf{m}_{\mu_t^*} \parallel^2$.

A non-linear minimization based on a first order approximation minimizes iteratively the error $\mathbf{e}$ by :

$$\widehat{\delta\mu} = -\lambda \mathbf{J}^+_{\mathbf{m}_{\mu_t}} \mathbf{e} \tag{5}$$

$\mathbf{J}_{\mathbf{m}_{\mu_t}}$ is the Jacobian matrix of $\mathbf{m}$ with respect to the current 2D transformation parameters. It is a $N \times M$ matrix storing the N Jacobian matrix $\mathbf{J}_{\mathbf{m}^i_{\mu_t}}$ of each visual feature $\mathbf{m}^i_{\mu_t}$:

$$\mathbf{J}_{\mathbf{m}_{\mu_t}} = (\mathbf{J}_{\mathbf{m}^1_{\mu_t}}, \ldots, \mathbf{J}_{\mathbf{m}^N_{\mu_t}})^T \qquad \text{with} \qquad \mathbf{J}_{\mathbf{m}^i_{\mu_t}} = \frac{\partial \mathbf{m}^i_{\mu_t}}{\partial \mu_t} \tag{6}$$

In the video sequence, noise or occlusion can occur. Since the minimization process is sensible to such outliers, M-estimators are introduced in (5) to eliminate these data:

$$\widehat{\delta\mu} = -\lambda (\mathbf{D}\mathbf{J}_{\mathbf{m}_{\mu_t}})^+ \mathbf{D}\mathbf{e} \tag{7}$$

where $\mathbf{D}$ is a $N \times N$ diagonal matrix such as:

$$\mathbf{D} = diag(w_1, ..., w_N) \tag{8}$$

The N weights w_i reflect the confidence of each visual feature $m^i_{\mu_t}$ and are computed using M-estimators [7]. To do so, various influence function are used in the literature. The Tukey's hard re-descending function is here considered as it completely rejects outliers.

The approach described above is valid for any visual features $\mathbf{m}$ if the Jacobian matrix $\mathbf{J}_\mathbf{m}$ is available. Two features are exploited in the hybrid tracker. The first one is based on the contour points extracted in the current image, the second one on the grey levels of the pattern. If there are N_c contour points and N_t texture points, the feature vector $\mathbf{m}$ will be of size $N = N_c + N_t$, storing both of them :

$$\mathbf{m}_\mu = (m_\mu^1, ..., m_\mu^{N_c}, m_\mu^{N_c+1}, ..., m_\mu^{N_c+N_t})^T \tag{9}$$

where $\left(m^i_\mu\right)_{i \leq N_c}$ is the feature associated with the i-th contour point and $\left(m^i_\mu\right)_{i=N_c+j}$ is the feature associated with the j-th texture point. However, the error associated with a texture point and the one associated with the contour points being of a different order of magnitude, a normalization must be performed to take into account the information given by the different cues. The weights in (8) are now :

$$w'_i = \begin{cases} \frac{w_i}{max_c(error)} & \text{if} \quad i \leq N_c \\ \frac{w_i}{max_t(error)} & \text{if} \quad i > N_c \end{cases} \tag{10}$$

where $max_c(error)$ (resp. $max_t(error)$) is the maximal absolute value stored in the error vector associated with the contour points (resp. texture points) and w_i is the weight computed by the M-estimators.

The two following subsections are dedicated to the two kinds of features presentation.

2.2. Edge-based features

The considered edges are those outlining the contours of the object but also the geometrical features in the pattern of the object. These edges are tracked from an image to another using a search along the contour normal. The points tracked by such a low-level process are called the contour points and are denoted $\mathbf{x}_t^i$ in $\mathbf{I}^t$(see Figure 1).

The aim of the edge-based features is to enable the current contour estimated using parameters μ_t to lie on the contour points extracted in the current image. As there is no matching correspondence between the contour points in $\mathbf{I}^{t-1}$ and those in $\mathbf{I}^t$, the point-to-contour distance $d_\perp\left(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i\right)$ is the feature used in the minimization process (7), where $\mathcal{C}_{\mu_t} = \mathbf{\Psi}_{\mu_t}(\mathcal{C}_{\widehat{\mu_{t-1}}})$ is the contour estimated from the current parameters of the 2D transformation μ_t . Referring to (4),

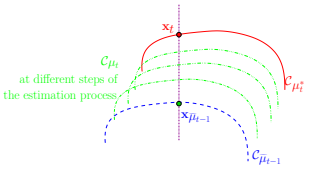


Fig. 1. Tracking point along the normal.

$\mathbf{m}_{\mu_t} = d_\perp(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i)$ whereas $\mathbf{m}_{\mu_t^*} = d_\perp(\mathcal{C}_{\mu_t^*}, \mathbf{x}_t^i)$ which obviously is equal to zero since $\mathbf{x}_t^i$ is located on the contour $\mathcal{C}_{\mu_t^*}$ that denotes the contour in $\mathbf{I}^t$ (although $\mathcal{C}_{\mu_t^*}$ is never really computed).

For the $\mathbf{J}_{d_\perp(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i)}$ computation, let express $d_\perp(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i)$ as a function of the current contour parameters ϵ_j that depend on μ_t. One then has :

$$\mathbf{J}_{d_\perp(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i)} = \sum_j \frac{\partial d_\perp(\mathcal{C}_{\mu_t}, \mathbf{x}_t^i)}{\partial \epsilon_j} \frac{\partial \epsilon_j}{\partial \mu_t} \quad (11)$$

From this latter equation, one can obtain the analytical form of the Jacobian matrix.

The framework has been applied to objects outlined by lines as well as by a NURBS (Non Uniform Rationnal B-Spline) [13] which is invariant to perpective transformation. The distance between a point and the curve is approximated by the distance between the point and the line tangent to the NURBS. The minimization problem is then similar to the polygonal object case since a distance between a point and a line is considered. In the result section, both cases are presented.

If only such edge-based features are used in the minimization process (7), the output of such a tracker is accurate when the tracked object is not textured but not stable. However, it requires a good initialization and is sensitive to texture/cluttered environment and to large 2D transformations.

2.3. Texture-based features

The feature considered here is the classical one in template-based matching, the grey-levels of the object pattern :

$$\mathbf{m}_{\mu_t}^j = I^t(\mathbf{x}_{\mu_t}^j) \quad (12)$$

where $I^t(\mathbf{x}_{\mu_t}^j)$ is the current grey levels sub-sampled at the location $\mathbf{x}_{\mu_t}^j = \mathbf{\Psi}_{\mu_t}(\mathbf{x}^j_{\widehat{\mu_{t-1}}})$. The locations $\mathbf{x}^j$ are called the texture points. With the constant illumination assumption, one has $\mathbf{m}_{\mu_t^*} = \mathbf{m}_{\mu_0^*}$ where $\mathbf{m}_{\mu_0^*}$ is the template sub-sampled in the first image, such as $\mathbf{x}_{\mu_0}^j$ are Harris points.

The Jacobian matrix of $\mathbf{m}_{\mu_t}^j$ is [5] :

$$\mathbf{J}_{I_{\mu_t}^j} = \frac{\partial I^t(\mathbf{x}_{\mu_t}^j)}{\partial \mu_t} = \nabla \mathbf{I}^t(\Psi_{\mu_t}(\mathbf{x}^j_{\widehat{\mu_{t-1}}}))^T \frac{\partial \Psi_{\mu_t}(\mathbf{x}^j_{\widehat{\mu_{t-1}}})}{\partial \mu_t} \quad (13)$$

where $\nabla \mathbf{I}^t(\mathbf{x})$ is the spatial gradient of $\mathbf{I}^t$ at the location $\mathbf{x}$. From (1), one gets easily $\partial \Psi_{\mu_t}(\mathbf{x}^j_{\widehat{\mu_{t-1}}})/\partial \mu_t$ (see [5] for the complete derivation and speed-up computing).

If only such texture-based features are used in the minimization process (7), the output of such a tracker is robust to large 2D transformation and occlusion and it is very smooth as the tracker uses information about the whole object. On the other side, the drawbacks of such an approach are that it requires a well-textured object and it does not always give a very accurate position of the object in the image.

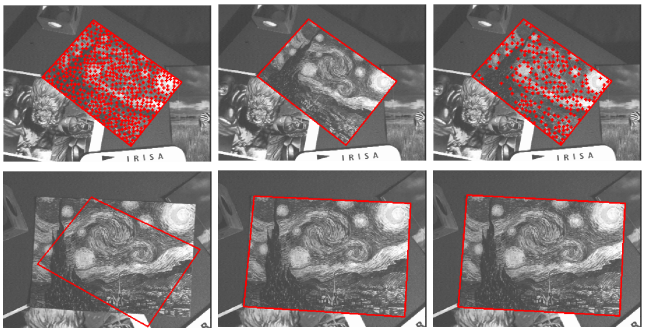
3. RESULTS

The two following subsections present some tracking results on video sequences. The hybrid tracker is compared to a edge-based one and a texture-based one. These two latter ones are similar to the hybrid tracker but using only the kind of feature associated with. The same amount of data is used for each tracker: if $2n$ features are tracked using a single cue tracker, then n of each kind of features are tracked using the hybrid tracker. In the first image of these tracking experiments, the contour and texture points used in the minimization process are displayed (red crosses for inliers and green ones for outliers). The object position in each image is given by the current contour in red.

In the last subsection, an example of a tracking perfor-mad during a robotic application is presented.

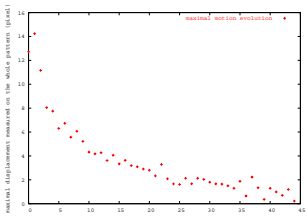
3.1. The Van Gogh "starry night" sequence

In this first sequence (about 45 images), large displacements are considered. Inter-frame displacement may reach about 14 pixels as shown in Figure 3(a). The initial and final images for each tracker are shown in Figure 2. The texture based tracker loses the object and the edge-based one gives quite good results but some drifts are sometimes observed on the left side (see Figure 3(b)). The only tracker that gives a good position of the object is the hybrid one. The hybrid tracker runs at an average rate of 13 Hz.

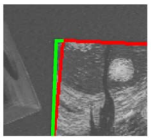


(a) Texture-based (b) Edge-based (c) Hybrid

Fig. 2. Van Gogh sequence. Initial and final images



(a)

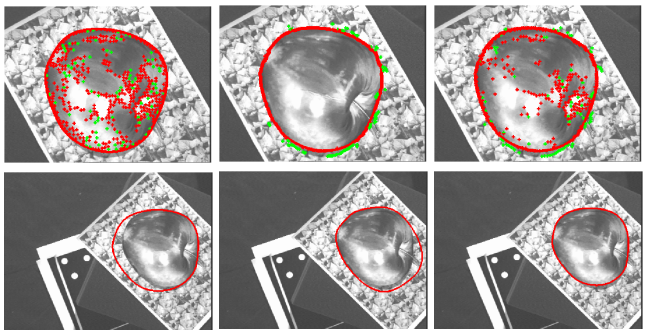


(b)

Fig. 3. Van Gogh sequence. (a) Evolution of the maximum motion between two successive frames. (b) Detail of the last image of Figure 2b : the left side is not precisely estimated (red lines). The real one is underlined in green.

3.2. The apple sequence

The tracked object in this sequence of 140 images is a picture of an apple. The challenge here is to obtain an accurate contour, which is quite difficult because of the background and the shadow. This experiment also illustrates the complementarity of the information given by each kind of visual features. Indeed, although the same amount of data is used, the single cue trackers fail whereas the hybrid one succeeds. The initial and final images of the sequence for each tracker are showned in Figure 4.



(a) Texture-based (b) Edge-based (c) Hybrid

Fig. 4. Apple sequence. Initial and final images

3.3. The robotic application

The hybrid tracker has been used successfully in visual servoing positioning tasks. Here is an example of the tracking performed during such a task where the robot has to move such as the object is located at a desired position in the image. Note that occlusions occur. The green crosses are points associated with features considered as outliers (due to noise, occlusions and shadow) and the red ones are for in inliers ones. Hidden contour points are represented in blue.

4. CONCLUSION

In this paper a reliable 2D tracker has been presented. It is based on a multi-cue template matching where an object

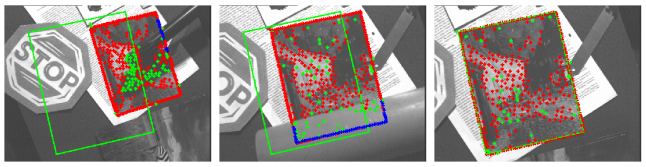


Fig. 5. Visual servoing experiment. Green rectangle : desired position of the object in the image, red rectangle : its current position

is represented by the most relevant points of its intensity pattern and a regular sampling of its contour points.

This tracker is then more robust to important motions, to occlusions and to the nature of the tracked object.

To handle generic 3D object motion, we are now interested in hybrid 3D tracking, by fusing pose computation and motion estimation.

5. REFERENCES

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

[2] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IROS'04*, Sendai, Japan, October 2004.

[3] A. Blake and M. Isard. *Active Contours*. Springer Verlag, April 1998.

[4] N. Chiba and T. Kanade. A tracker for broken and closely-spaced lines. In *ISPRS'98*, pages 676 – 683., Hakodate, 1998.

[5] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. In *PAMI'98*, 20(10):1025–1039, October 1998.

[6] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *CVIU'98*, 69(1):23–37, January 1998.

[7] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[8] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *ECCV*, pages 893–908–yy, 1998.

[9] F. Jurie and M. Dhome. Hyperplane approximation for template matching. In *PAMI'02*, 24(7):996–1000, July 2002.

[10] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *ICCV'99*, volume 1, pages 262–268, Kerkira, Greece, September 1999.

[11] L. Masson, F. Jurie, and M. Dhome. Contour/texture approach for visual tracking. In *SCIA 2003*, volume 2749 of *Lecture Notes in Computer Science*, pages 661–668. Springer, 2003.

[12] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. In *JVCIR'95*, 6(4):348–365, December 1995.

[13] L. Piegl and W. Tiller. *The NURBS book (2nd ed.)*. Springer-Verlag New York, Inc., 1997.

[14] A. Shahrokni, T. Drummond, and P. Fua. Texture boundary detection for real-time tracking. In *ECCV'04*, volume 2, pages 566–577, Prague, Czech Republic, May 2004.