Robust Shape Tracking in the Presence of Cluttered Background

Jacinto C. Nascimento, *Member, IEEE,* and Jorge S. Marques

***Abstract*—Many object-tracking algorithms are based on low-level features detected in the image. Typically, the object shape and position are estimated to fit the observed features. Unfortunately, image analysis methods often produce invalid features (outliers) which do not belong to the object boundary. These features have a strong influence on the shape estimates, leading to meaningless tracking results. This paper proposes a robust tracking algorithm which is able to deal with outliers, inspired in the probabilistic data association filter proposed in the context of point tracking. The algorithm is based on two key concepts. First, middle level features (strokes) are used instead of low-level ones (edge points). Second, two labels (valid/invalid) are considered for each stroke. Since the stroke labels are unknown all labeling sequences are considered and a probability (confidence degree) is assigned to each of them. In this way, all the strokes contribute to track the moving object but with different weights. This allows a robust performance of the tracker in the presence of outliers. Experimental tests are provided to assess the performance of the proposed algorithm in lip and gesture tracking and surveillance applications.**

***Index Terms*—Data association, deformable contours, object tracking, robust filtering, shape analysis.**

I. Introduction

ACTIVE contours have been thoroughly investigated in the context of shape tracking. They represent the object boundary by an elastic contour which is attracted toward image features. The estimation of the model parameters is often performed by Kalman filtering [1], [2]. Unfortunately, feature detection algorithms used to extract information from the image produce many invalid features (outliers) which do not belong to the boundary of the object to be tracked (Fig. 1). Furthermore, the outliers have a strong influence on the performance of active contours algorithms.

Several methods have been proposed to alleviate this difficulty. Some of them impose additional restrictions to the object shape, e.g., by using rigid templates or eigen shapes learned from the data [1], [3]. This prevents the model boundary from having unpredictable shapes caused by the outliers. Temporal restrictions have also been considered by representing the evolution of the motion and shape parameters using dynamic models, e.g., stochastic difference equations. In general, dynamic models may be specified by the user or learned from the video sequences using standard system identification methods [1].

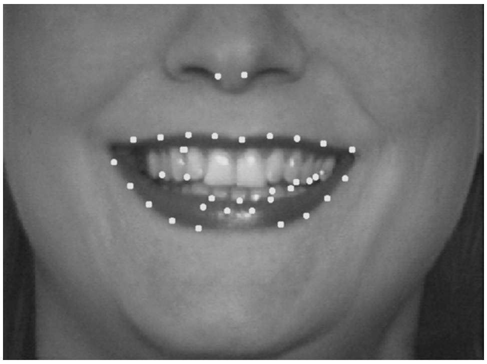


Fig. 1. Detected features.

Despite all this work, none of these methods is able to solve the segmentation problem, i.e., none of them is able to discriminate valid data from the outliers which hamper the performance of Kalman trackers. An *ad hoc* procedure to improve Kalman estimates consists of using a validation gate, computed from the predicted object boundary [3]. The regions where each boundary samples are located in the next frame are computed and all the features outside these regions are discarded. This method works well if the object motion is slow and highly predictable but it fails in more complex situations.

The estimation of random signals from multiple noisy observations with outliers was extensively studied in the context of target tracking using radar measurements [4], [5]. Several techniques were proposed in this context. First, a validation gate is usually adopted to restrict the search area associated with a given target. This procedure is also used in active contours as explained before. Second and most important, robust filtering methods have been considered since multiple returns can still be observed, most of them being false alarms. A variety of techniques were proposed for target tracking ranging from the naive Nearest Neighbor Filter, which considers the closest return only and discards the rest, to the optimal Track Splitting Filter leading to a combinatorial explosion of hypothesis. The probabilistic data association filter (PDAF) proposed by Bar-Shalom and Fortmann [5] is a good compromise since it takes into account multiple data association hypothesis without the combinatorial explosion. Nonparametric filtering techniques have also been adopted in the context of object tracking, using the Condensation algorithm [6].

This paper describes a robust estimation algorithm for shape tracking, called Shape PDAF (S-PDAF) [7]. This algorithm is an extension of the PDAF filter proposed in the context of point tracking. The algorithm considers middle level image features (strokes) detected in the vicinity of the object to be tracked. Each

Manuscript received December 12, 2000; revised February 20, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Faouzi Kossentini.

The authors are with the Instituto Superior Técnico (IST), 1049-001 Lisbon, Portugal.

Digital Object Identifier 10.1109/TMM.2004.837253

feature can be either a valid observation of the object boundary or an outlier. Since we do not know *a priori* which features are valid, all possible sequences of valid/invalid stroke labels are considered and a probability (confidence degree) is assigned to each labeling sequence. For convenience, each label sequence is denoted as a data interpretation.

The stroke probabilities are not assigned by *ad hoc* criteria. They are computed using a data model, an outlier model and the predicted shape estimate and uncertainty. This leads to a Kalman type of recursion for the update of the state estimates and uncertainties. All the strokes contribute to update shape and motion estimates, however, each stroke has a different confidence degree and therefore a different influence on the estimates. For example, a stroke far away from the object boundary will have a negligible influence on the estimates (the opposite behavior is observed in Kalman filtering: strokes far from the object dominate the estimation process). Furthermore, since the algorithm considers stroke sequences, it is able to detect false strokes near the object boundary if the stroke is not compatible with the others.

The differences between the S-PDAF proposed in this paper and PDAF filter of Bar-Shalom and Fortmann concern: 1) different sensor models (we are no longer dealing with radar measurements but with image strokes instead) and 2) new update equations for the shape and motion estimates which will be presented in the paper. The paper is organized as follows: Section II describes the problem; Section III presents the S-PDAF; Section IV applies the S-PDAF to shape tracking and describes the computation of the association probabilities; Section V presents results obtained the S-PDAF method in lip and gesture tracking, as well as in surveillance applications. Finally, Section VI concludes the paper.

II. Problem Statement

This paper addresses the estimation of moving objects in image sequences in the presence of outliers. In Kalman-based methods, object tracking is performed in three steps [1]: contour prediction, image measurement and contour update. The first step predicts the object position and shape in the next image. The second step computes image features in the vicinity of the predicted contour and the third step uses these measurements to update the contour estimates. The main difficulty concerns the presence of type I and type II errors: false alarms and detection failures. Both of them produce undesirable effects which significantly hamper the performance of Kalman-based trackers.

In order to estimate the object position and deformation, image features are detected. First, a predicted object boundary is computed using the information from previous frames. Then, the predicted boundary is sampled at equally spaced points and a set of features is detected for each point by searching along a line orthogonal to the predicted contour [see Fig. 2(a)]. These features can be considered either as true (belonging to the object boundary) or false (produced by the background), but this information is not available in advance. In general, it is not possible to classify each image feature as being true or false. This would lead to 2^N interpretations of the data, where N is the number of detected features (typically higher than 50). A different approach is adopted in this paper. The observations are associated in strokes. Strokes are obtained by matching feature points detected at consecutive measurement lines. This is accomplished by using the mutual favorite pairing method [8]. In this method, each feature from the first line selects the best match from the set of features detected in the second line. The procedure is performed backward for each feature of the second line. Two features are then linked when they both choose the other feature as its best match. The matching criterion is the distance between feature points. In this way, the number of data interpretations is drastically reduced to 2^M, where M is the number of strokes (typically $M < 10$). An example is shown in Fig. 2(b).

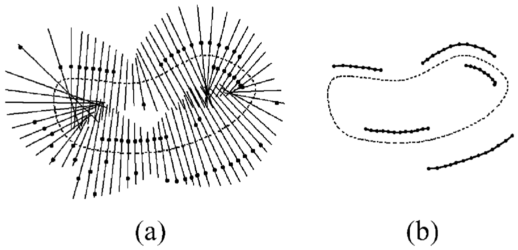


Fig. 2. Stroke detection: (a) edge detection and (b) linking (●—denote the features obtained in the measurement lines orthogonal to the predicted contour).

We shall assume that the object shape and position in the image are described by a vector of parameters x, described by a stochastic difference equation

$$x(k+1) = Ax(k) + w(k) \tag{1}$$

where $w(k) \sim \mathcal{N}(0, Q)$ is white Gaussian noise, A is a square dynamic matrix, and k denotes the time instant. The covariance matrix Q accounts for the motion uncertainty. Q is often defined by the user, assuming independence of the shape coefficients or estimated from the available data for a given tracking problem, using system-identification techniques [9].

After computing the strokes, each of them may be classified either as true or false. A stroke interpretation I_i is defined as a binary sequence $I_i = \{I_i^1, \ldots, I_i^M\}$, where $I_i^j \in \{0, 1\}$ is the label of the jth stroke in the interpretation I_i, ($I_i^j = 1$ means valid stroke; $I_i^j = 0$ means invalid stroke).

Let $y(k)$ be the vector of all image features detected at time instant k and let $y_i(k)$ be a vector with the coordinates of all valid features in the interpretation I_i.

Let x be a set of parameters defining the object boundary (e.g., the control points of a spline curve). It will be assumed that y_i and the curve parameters $x(k)$ are related by

$$y_i(k) = C_i x(k) + v_i(k) \tag{2}$$

where C_i is the observation matrix associated to the ith interpretation and $v_i(k) \sim \mathcal{N}(0, R_i)$ is a white Gaussian noise (measurement noise). In general, the observation matrices C_i, C_j associated with two interpretations I_i, I_j are different since the observation vectors y_i, y_j contain different data features and often have different dimensions.

The problem to be solved can be formulated as follows: *how to estimate the object parameters $x(k)$ assuming that we do not know which interpretation is right ?*

III. S-PDAF

A nonlinear filtering approach is adopted in this paper. The estimation of the state vector requires the propagation of the *a posteriori* density $p(x(k) \mid Y^k)$, where Y^k is a set with the current and past observations (visual features). Since there are multiple data interpretations (hypothesis), the *a posteriori* density is not Gaussian: it is a mixture of Gaussians, where the number of modes increases exponentially with k [10]. Therefore, the exact propagation of the *a posteriori* density is difficult. A suboptimal approach is adopted instead, inspired in the PDAF [5] developed in the context of point target tracking.

The main hypothesis is the following: it will be assumed that the state distribution given past observations is Gaussian i.e.,

$$p[x(k) \mid Y^{k-1}] = \mathcal{N}[x(k); \hat{x}(k \mid k-1), P(k \mid k-1)] \quad (3)$$

where $\hat{x}(k \mid k-1)$, $P(k \mid k-1)$ are the mean and covariance of $x(k)$ given past observations Y^{k-1}.

Let us now consider the computation of the state estimate and uncertainty (state mean and covariance matrix) given current and past observations, i.e.,

$$\hat{x}(k \mid k) \triangleq E[x(k) \mid Y^k] \quad (4)$$

$$P(k \mid k) \triangleq E\left\{[x(k) - \hat{x}(k \mid k)] \times [x(k) - \hat{x}(k \mid k)]^T \mid Y^k\right\}. \quad (5)$$

Since we do not know which interpretation is valid, they all have to be considered as follows:

$$\begin{aligned}\hat{x}(k \mid k) &= E[x(k) \mid Y^k] = \int x(k) p(x(k) \mid Y^k) dx(k) \\ &= \int x(k) \sum_i p\left(x(k), I_i(k) \mid Y^k\right) dx(k) \\ &= \sum_i \int x(k) p\left(x(k) \mid I_i(k), Y^k\right) \\ &\quad \times p\left(I_i(k) \mid Y^k\right) dx(k). \end{aligned} \quad (6)$$

Equation (6) can be rewritten as

$$\hat{x}(k \mid k) = \sum_{i=0}^{m_k} \alpha_i(k) \hat{x}_i(k \mid k) \quad (7)$$

where $\alpha_i(k) \triangleq p\left(I_i(k) \mid Y^k\right)$ is the *a posteriori* probability of the ith interpretation and

$$\hat{x}_i(k \mid k) = E\left\{x(k) \mid I_i(k), Y^k\right\}. \quad (8)$$

The state estimate $\hat{x}(k \mid k)$ is a weighted sum of the state estimates $\hat{x}_i(k \mid k)$ obtained for each interpretation $I_i(k)$ and updated by Kalman filtering

$$\hat{x}_i(k \mid k) = \hat{x}(k \mid k-1) + K_i(k)\nu_i(k) \quad (9)$$

where $K_i(k)$, $\nu_i(k)$ are the Kalman gain and innovation for the interpretation $I_i(k)$, computed by

$$K_i(k) = P(k \mid k-1) C_i^T S_i(k)^{-1} \quad (10)$$

$$S_i(k) = C_i P(k \mid k-1) C_i^T + R_i \quad (11)$$

$$\nu_i(k) = y_i(k) - C_i \hat{x}(k \mid k-1). \quad (12)$$

Replacing (9) in (7), we have

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + \sum_{i=1}^{m_k} \alpha_i(k) K_i(k) \nu_i(k). \quad (13)$$

A recursive equation can also be derived for the covariance matrix (see the Appendix).

$$\begin{aligned} P(k \mid k) = &\left[I - \sum_{i=1}^{m_k} \alpha_i(k) K_i(k) C_i\right] P(k \mid k-1) \\ &+ \sum_{i=0}^{m_k} \alpha_i(k) \hat{x}_i(k \mid k) \hat{x}_i(k \mid k)^T \\ &- \hat{x}(k \mid k) \hat{x}(k \mid k)^T. \end{aligned} \quad (14)$$

Equations (13) and (14) update the mean and covariance of the state vector using all data interpretations. They all contribute to the propagation of the *a posteriori* distribution but with different confidence degrees. This algorithm will be denoted as the S-PDAF.

IV. Application to Tracking

Three aspects have to be considered before we can apply the S-PDAF in tracking: 1) the computation of the association probabilities which requires a probabilistic model for the contour strokes; 2) a dynamic model for the motion and deformation of the object boundary; and 3) the detection of image features. These issues are addressed in Sections IV-A–IV-C.

A. Association Probabilities

To use the S-PDAF filter in object tracking, a set of data association probabilities $\alpha_i(k)$ have to be computed. This requires a probabilistic model for the strokes detected in the image. The following assumptions are made: high association probabilities will be assigned to the interpretations with the following characteristics:

- long valid strokes;
- valid strokes close to the predicted shape;
- nonoverlapping strokes.

Interpretations with overlapping strokes, which assign multiple observations to a single contour sample, will have zero probability. Next, a stroke generation model is described.

Let us consider the example in Fig. 3. It will be assumed that the image strokes are characterized by the following variables: M—number of strokes; b^j e^j—first and last index of the jth stroke; I_i—label sequence; and $y(k)$—a vector containing the coordinates of the visual features detected in the kth image.

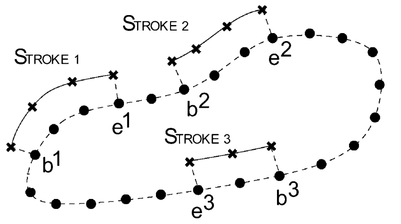


Fig. 3. Stroke parameters (•—samples of the predicted contour; ×—edge points).

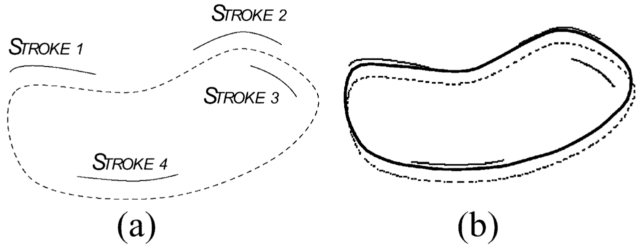


Fig. 4. (a) Detected strokes. (b) S-PDAF estimate (solid line).

The data model is characterized by the Joint distribution $P\left(y(k), I_i(k), b, e, M \mid Y^{k-1}\right)$ which can be factorized as follows:

$$\begin{aligned} P&\left(y(k), I_i(k), b, e, M \mid Y^{k-1}\right) \\ &= p\left(y(k) \mid I_i(k), b, e, M, Y^{k-1}\right) \\ &\quad \times p\left(I_i(k) \mid b, e, M, Y^{k-1}\right) \\ &\quad \times p(b, e \mid M, Y^{k-1}) p(M \mid Y^{k-1}). \end{aligned} \tag{15}$$

Since all these variables are observed, except the interpretation I_i

$$\begin{aligned} \alpha_i(k) &= p\left(I_i(k) \mid y(k), b, e, M, Y^{k-1}\right) \\ &= \frac{p\left(y(k), I_i(k), b, e, M \mid Y^{k-1}\right)}{p(y(k), b, e, M \mid Y^{k-1})} \\ &= \beta p\left(y(k) \mid I_i(k), b, e, M, Y^{k-1}\right) \\ &\quad \times p\left(I_i(k) \mid b, e, M, Y^{k-1}\right) \end{aligned} \tag{16}$$

where β is a normalization constant.

Probabilities $p\left(y(k) \mid I_i(k), b, e, M, Y^{k-1}\right)$, $p(I_i(k) \mid b, e, M, Y^{k-1})$ are problem dependent. They characterize the data features observed in valid strokes and the outliers, as well as the *a priori* probabilities of the data interpretations. The choice of these distributions is addressed below. First we assume that all image features are independently generated, i.e.,

$$\begin{aligned} p&\left(y(k) \mid I_i(k), b, e, M, Y^{k-1}\right) \\ &= \prod_{j=1}^{M} \prod_{n=b^j}^{e^j} p\left(y^j(s_n, k) \mid I_i^j(k)\right) \end{aligned} \tag{17}$$

where $y^j(s_n, k)$ is the feature point belonging to the jth stroke and detected in the vicinity of s_n. The independence of features obtained in different search lines is a typical assumption in active contours [1]. More general models can easily be adopted by learning the covariance matrices different from the data [9], [11]. Second, it will be assumed that the visual features have uniform distribution in the search area (validation gate) if they are classified as unreliable $\left(I_i^j = 0\right)$ and they are generated with a Gaussian distribution if they are classified as reliable. Therefore

$$\begin{aligned} p&\left(y^j(s_n, k) \mid I_i^j(k)\right) \\ &= \begin{cases} V(s_n, k)^{-1}, & \text{if } I_i^j(k) = 0 \\ \rho^{-1} \mathcal{N}\left(\nu^j(s_n, k); 0, S(s_n, k)\right), & \text{otherwise} \end{cases} \end{aligned} \tag{18}$$

where $V(s_n, k)$ is the length of the search line; ρ is the normalization constant; $\nu^j(s_n, k)$ is the innovation associated to the jth stroke; and $S(s_n, k) = C(s_n)P(k \mid k-1)C(s_n)^T + R(s_n)$ is the covariance of the innovation vector where $C(s_n)$ and $R(s_n)$ are the output matrix and noise covariance associated to the nth sample.

Let us now consider the second term $P(I_i(k) \mid b, e, M)$. Assuming independence of the stroke labels, we have

$$P(I_i(k) \mid b, e, M) = p\left(I_i^1 \mid b, e, M\right) \dots p\left(I_i^M \mid b, e, M\right). \tag{19}$$

Since it is assumed that long valid strokes have higher probability than short strokes, a linear model is adopted to represent the dependence of the stroke probability with length

$$p\left(I_i^j = 1\right) = ml^j + c \quad p\left(I_i^j = 0\right) = 1 - ml^j - c \tag{20}$$

where

$$c = P_A \quad m = \frac{P_A - P_B}{L} \tag{21}$$

and where l^j is the length of the jth stroke, and L is the number of sampling points. P_A and P_B are constants.

Using (20)

$$P(I_i(k) \mid b, e, M) = \prod_{j: I_i^j = 1} (ml^j + c) \prod_{j: I_i^j = 0} (1 - ml^j - c). \tag{22}$$

Thus, each interpretation has a data association probability α_i given by (16)–(22). The interpretation with highest probability is called the dominant interpretation.

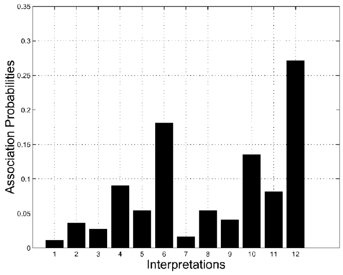
Fig. 4(a) shows the predicted boundary and four strokes detected in the image. The strokes position and orientation suggest that a translation must be applied to the predicted contour in order to approximate the detected features. However, stroke S_3 overlaps S_2 producing a data conflict. Table I shows all possible data interpretations and their association probabilities computed by the model described in this section. It is stressed that this reduces the number of interpretations to 12. The dominant interpretation (I_{12}) correctly neglects the contribution of stroke S_3. Therefore S-PDAF solves the data conflict by assigning high confidence degrees to the strokes (S_1, S_2, S_4) and low confidence degree to S_3. The S-PDAF estimate of the object boundary according to (13) is shown in Fig. 4(b) (solid line).

B. Dynamic Shape Model

To represent a moving object in a given frame k, it is assumed that the object boundary is a transformed version of a reference shape plus an additional deformation.

TABLE I
DATA INTERPRETATIONS AND ASSOCIATION PROBABILITIES

Interpretations	*S1*	*S2*	*S3*	*S4*	α
I_1	0	0	0	0	0.0110
I_2	0	0	0	1	0.0365
I_3	0	0	1	0	0.0271
I_4	0	0	1	1	0.0903
I_5	0	1	0	0	0.0543
I_6	0	1	0	1	0.1809
I_7	1	0	0	0	0.0164
I_8	1	0	0	1	0.0547
I_9	1	0	1	0	0.0406
I_{10}	1	0	1	1	0.1354
I_{11}	1	1	0	0	0.0814
I_{12}	1	1	0	1	0.2712



Let $r(s) : I \to \Re^2$ be a parametric representation of the object boundary[1]. It is assumed that[2]

$$r(s) = \mathcal{G}r_r(s) + d(s) + v(s) \tag{23}$$

where $\mathcal{G}$ is a geometric transformation (e. g., affine transformation), r_r, d and v are the parametric descriptions of the reference shape, deformation and measurement noise, respectively. For the sake of simplicity it is assumed that these curves are described by B-splines. It will be assumed that $\mathcal{G}$, d can be expressed in terms of a small number of parameters which are updated by the S-PDAF algorithm and v is a white noise process.

Several transforms can be considered (e.g., translation, Euclidean similarities, affine transform) [1]. For example, in the case of the affine transform, the object boundary is

$$\begin{cases} r_1(s_i) = a_1 r_{r1}(s_i) + a_2 r_{r2}(s_i) + a_3 + d_1(s_i) + v_1(s_i) \\ r_2(s_i) = a_4 r_{r1}(s_i) + a_5 r_{r2}(s_i) + a_6 + d_2(s_i) + v_2(s_i) \end{cases} \tag{24}$$

where $r(s) = (r_1(s), r_2(s))$, $r_r(s) = (r_{r1}(s), r_{r2}(s))$, and $a_1, \ldots, a_6$ are the motion parameters at instant k; $v(s) = (v_1(s), v_2(s))$ is the measurement noise curve. Furthermore, it will be assumed that shape deformation is given by

$$d(s) = \sum_{k=1}^{N_c} d_k \phi_k(s) \tag{25}$$

where $\phi_k(s)$ are known basis functions and $d_1, \ldots, d_{N_c}$ are two-dimensional vectors. Quadratic B-spline basis function are used in this paper [12].

Let $x(k)$ denote the vector of unknown shape and motion parameters

$$x = [a_1, \ldots, a_6, d_{x1}, \ldots, d_{xN_c}, d_{y1}, \ldots, d_{yNc}]^T \tag{26}$$

and let y be a $2L \times 1$ vector obtained by sampling the object boundary at L equally spaced points

$$y = [r_1(s_1), \ldots, r_1(s_L), r_2(s_1), \ldots, r_2(s_L)]^T. \tag{27}$$

Equation (24) can be written as follows:

$$y(k) = Cx(k) + v(k) \tag{28}$$

where

$$C = \begin{bmatrix} M & O_{L\times 3} & \mathcal{B}_{L\times N_c} & O_{L\times N_c} \\ O_{L\times 3} & M & O_{L\times N_c} & \mathcal{B}_{L\times N_c} \end{bmatrix} \tag{29}$$

with

$$M = \begin{bmatrix} r_{r1}(s_1) & r_{r2}(s_1) & 1 \\ r_{r1}(s_2) & r_{r2}(s_2) & 1 \\ \vdots & \vdots & \vdots \\ r_{r1}(s_L) & r_{r2}(s_L) & 1 \end{bmatrix} \tag{30}$$

$$\mathcal{B} = \begin{bmatrix} \phi_1(s_1) & \phi_2(s_1) & \ldots & \phi_{N_c}(s_1) \\ \phi_1(s_2) & \phi_2(s_2) & \ldots & \phi_{N_c}(s_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(s_L) & \phi_2(s_L) & \ldots & \phi_{N_c}(s_L) \end{bmatrix}. \tag{31}$$

In (29), $\mathcal{B}$ is a $L \times N_c$ $\mathbf{B}$-spline interpolation matrix [12], $O_{L\times 3}$ $O_{L\times N_c}$ are null matrices. Similar expressions can be derived for other types of models.

C. Feature Detection

Feature detection is guided by the predicted contour and by its uncertainty. Instead of detecting the edge points in the whole image, only feature points in the vicinity of the predicted contour are considered. Furthermore, these points are obtained by a directional search procedure similar to the ones presented in [1], [3], and [11].

The length of the inspection interval depends on the predicted contour uncertainty in the search direction and is given by

$$\rho(s_i, k) = \delta\sqrt{n(s_i)^T S(s_i, k) n(s_i)} \tag{32}$$

where $n(s_i)$ is the unit normal at s_i, $S(s_i, k) = C(s_i)P(k \mid k-1)C(s_i)^T + R(s_i)$ is the covariance matrix of the predicted contour point and $C(s_i)$ is a matrix formed by lines i and $i + L$ of matrix C, δ is a constant.

The predicted contour is sampled at equally spaced points and for each point a set of features is obtained by searching for transitions along the direction orthogonal to the contour [see

[1] the time instant k is omitted for the sake of simplicity.

[2] other works assume that $r_k = \mathcal{G}_k(r_r + d) + v$, [3], [11]. This model is nonlinear and allows a better representation of shape deformation, e.g., using eigen shapes. However, parameter estimation is usually more complex being performed in two steps: pose estimation and shape estimation.

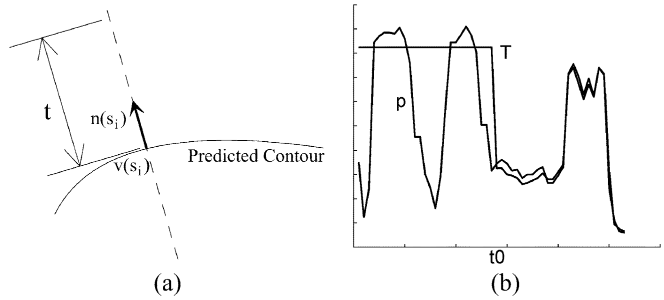


Fig. 5. Feature detection: (a) directional search and (b) image profile p, and shifted template T taken at t_0.

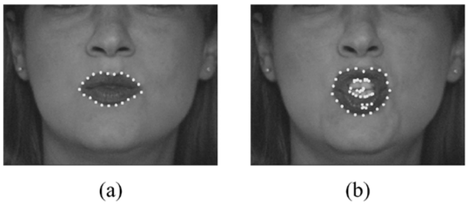


Fig. 6. (a) Detected features on the object boundary. (b) Detected features with outliers.

Fig. 5(a)]. This provides enough information to estimate the object motion and deformation provided that enough samples are considered.

Feature detection along the ith direction is performed by computing the local minima of the function

$$\mathcal{J}(t_0) = \int_t |p_i(t) - T_i(t, t_0)|^2 dt \tag{33}$$

where $p_i(t)$ is the image profile along the ith direction, $\mathbf{t}$ denotes the distance to the object boundary, and $T_i(t, t_0)$ is a known template. In this paper, the template T_i is defined as follows: $T_i(t)$ is equal to the average intensity of the object for $t \leq t_0$ and $T_i(t)$ is equal to the background image profile or to the average background color for $t > t_0$ [see Fig. 5(b)]. The first hypothesis is used in surveillance examples and the second is used in the lip and gesture-tracking experiments (see Section V).

Fig. 6 shows the feature-detection results. The first image illustrates an ideal situation where all the features are located on the boundary of the object. The second image shows a more difficult case in which the features are located inside of the object boundary as well. These are false alarms (outliers) which hamper the performance of classic tracking algorithms.

V. Experimental Results

The S-PDAF tracker was tested with a large number of sequences. Three examples will be presented: hand, lip, and vehicle tracking. All these problems have been extensively studied in the last decade [13]–[15]. Robust lip tracking is useful for automatic speech recognition systems. Speech recognition is a major goal which has been thoroughly pursued for more than two decades [16]–[18]. The use of visual features in such systems is useful for improving the accuracy and robustness of speech recognition system specially in noisy environments. Recently much work has been done on audio-visual speech recognition [19]–[21], where the aim is to use the lip movements and configuration to be jointly processed with audio features. However this visual information requires accurate lip tracking. The S-PDAF can be used to perform this task.

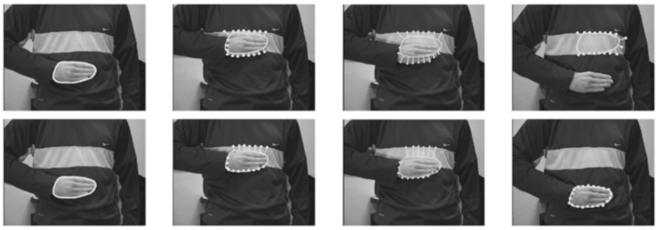


Fig. 7. Tracking results obtained with Kalman filter (first row) and S-PDAF filter (second row), (frames 1, 7, 9, 18).

Human gestures and surveillance sequences will also be shown to illustrate the performance of the algorithm in other contexts.

A. Human Gesture

Fig. 7 illustrates the performance of the proposed algorithm in gesture estimation. A hand is represented by a B-spline curve with 12 control points. The solid white line shows the estimated contour and the dots are the detected features. The lines between the contour and the features can be interpreted as a spring forces which attract the contour toward the features. Two algorithms are considered: the Kalman filter and the S-PDAF. The Kalman filter fails to track the hand. Since it assumes all the data as valid it tries to match all the visual features (valid data and outliers) with the shape model. This produces strong shape distortions if we allow shape to deform during tracking or wrong pose estimates if the object shape is considered as rigid (see Fig. 7, frame 9). On the contrary, the S-PDAF manages to distinguish the outliers from valid data and tracks the hand well.

B. Lip Tracking

The images used in lip tracking were obtained with a fixed camera and digitized at a sampling rate of 12 fps. In these experiments we used a B-spline curve with 12 control points in grayscale images, and eight control points in color images. It is assumed that the object shape in the first image is manually defined by the user. This section shows results obtained with the S-PDAF and Kalman filters in two cases: during a speech and while the person is singing. These situations are quite different. During a speech, lip motion is smoother while it presents sudden shape changes when the person is singing. These experiments allow us to evaluate the robustness of the S-PDAF filter to sudden changes in the lip contour.

Figs. 8 and 9 show the results obtained by the Kalman tracker and by the S-PDAF while the person is speaking. The Kalman filter is able to deal with a small number of outliers close to the lip contour (see Fig. 8) but it becomes lost when the number of outliers increases (see Fig. 9) and it is not able to recover after that instant of time. The S-PDAF performs well in both cases providing accurate boundary estimates, even in the presence of a large number of outliers.

The next example is based on a sequence of 100 images of a person singing. Fig. 10 shows the tracking results obtained with

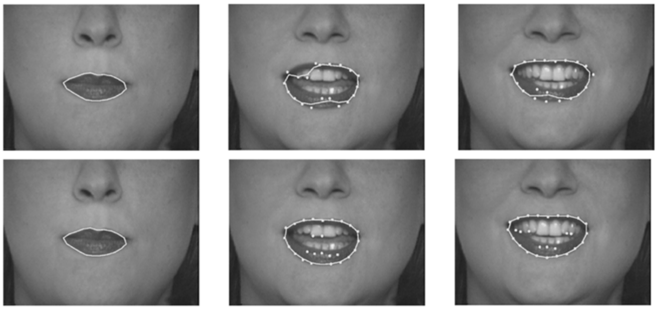


Fig. 8. Lip tracking (speaking) with Kalman filter (first row) and S-PDAF filter (second row), (frames 1, 7 10).

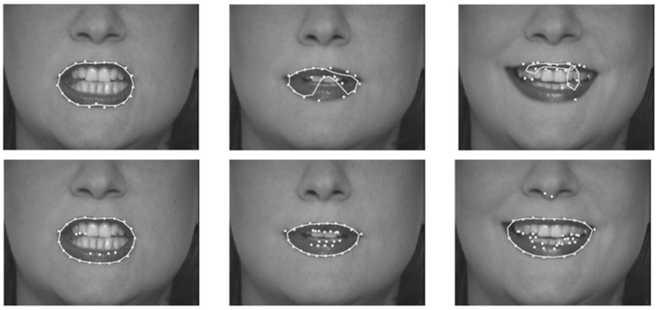


Fig. 9. Lip tracking (speaking) with Kalman filter (first row) and S-PDAF (second row), (frames 18, 28, 30).

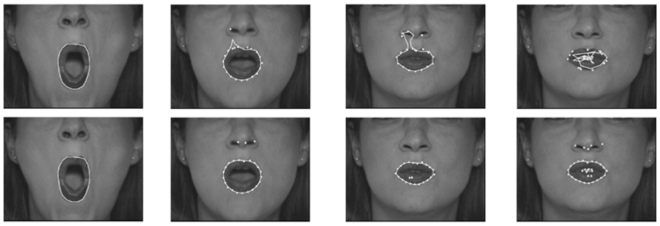


Fig. 10. Lip tracking (singing) with Kalman filter (first row) and S-PDAF (second row) (frames 1, 6, 8, 12).

the Kalman filter and with the S-PDAF. Again, Kalman estimates are attracted by spurious data, the S-PDAF filter manages to discriminate the true data from false alarms and produces correct estimates.

Finally, Fig. 11 shows pairs of consecutive frames corresponding to sudden changes of the lip contour. The ability of S-PDAF to deal with these changes is remarkable.

The S-PDAF tracker was also used with color images. Color images were first converted into gray scale images, by computing the Fisher linear discriminant between lip and skin color distributions [22].

Fig. 12 shows results obtained with the S-PDAF for different speakers. The S-PDAF can also be used for tracking other face features such as the eyebrows. This is useful to estimate facial expressions or to animate face models, e.g., using the model described in [23]. Fig. 13 shows some tracking results obtained with the S-PDAF, (dots are the observations). Three decoupled S-PDAF filters were used in this experiment.

C. Vehicle Tracking

This section presents tracking results obtained in traffic sequences. It is assumed that $x(k)$ is defined by a stochastic difference equation

$$x(k) = \begin{bmatrix} A & O_{2N_f \times 2N_c} \\ O_{2N_c \times 2N_f} & I_{2N_c \times 2N_c} \end{bmatrix} x(k-1) + w(k) \quad (34)$$

with

$$A = \begin{bmatrix} I_{N_f \times N_f} & I_{N_f \times N_f} \\ O_{N_f \times N_f} & I_{N_f \times N_f} \end{bmatrix} \quad (35)$$

where $x(k) = [a_1, \ldots, a_{N_f}, \dot{a}_1, \ldots, \dot{a}_{N_f}, d_{x1}, \ldots, d_{x_{N_c}}, d_{y1}, \ldots, d_{y_{N_c}}]^T$ is a state vector containing the shape-space parameters as well as its derivatives and deformation, I is the identity matrix, O is the null matrix, N_f is the dimension of the shape-space and N_c the number of control points.

The S-PDAF filter was used to track cars in traffic sequences. Two examples are shown in Figs. 14 and 15. In both examples, a B-spline curve with 12 control points was used. In Fig. 14 a translation model with two degrees of freedom was used. In this case, (2) is written as

$$y_i(k) = C_i x(k) + D_i + v_i(k) \quad (36)$$

with matrices C and D given by

$$C = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (37)$$

$$D = [r_{r1}(s_1), \ldots, r_{r1}(s_L), r_{r2}(s_1), \ldots, r_{r2}(s_L)]^T \quad (38)$$

where $\mathbf{1} = \{1, 1, \ldots, 1\}^T$ and $\mathbf{0} = \{0, 0, \ldots, 0\}^T$ are $L \times 1$ vectors.

Fig. 15 shows another example in which the car performs a left turn. In this case, it was assumed that the car motion is described by an Euclidean similarity (four degrees of freedom). Therefore

$$C = \begin{bmatrix} r_{r1}(s) & \mathbf{1} & -r_{r2}(s) & O_{L\times 1} & O_{L\times 4} \\ r_{r2}(s) & O_{L\times 1} & r_{r1}(s) & \mathbf{1} & O_{L\times 4} \end{bmatrix} \quad (39)$$

and D is the null matrix. This model allows rotation, translation and scaling. However, no shearing is allowed.

The S-PDAF manages to correctly track the vehicle in both examples. The Kalman filter fails to track the car in these examples due to the clutter. The results are not shown.

The S-PDAF method was 1–3 times slower than the Kalman filter in the experiments presented in this section, if we do not consider the computational time associated with feature detection. This is not critical issue for real time implementations since feature detection remains the most complex operation that the tracker has to perform.

VI. Conclusions

Feature detection is an ill-posed problem. Many features detected in the image are not generated by the boundary of the object to be tracked. They are produced either by noise or by other objects present in the scene. This raises a major difficulty in the design of feature-based tracking algorithms:

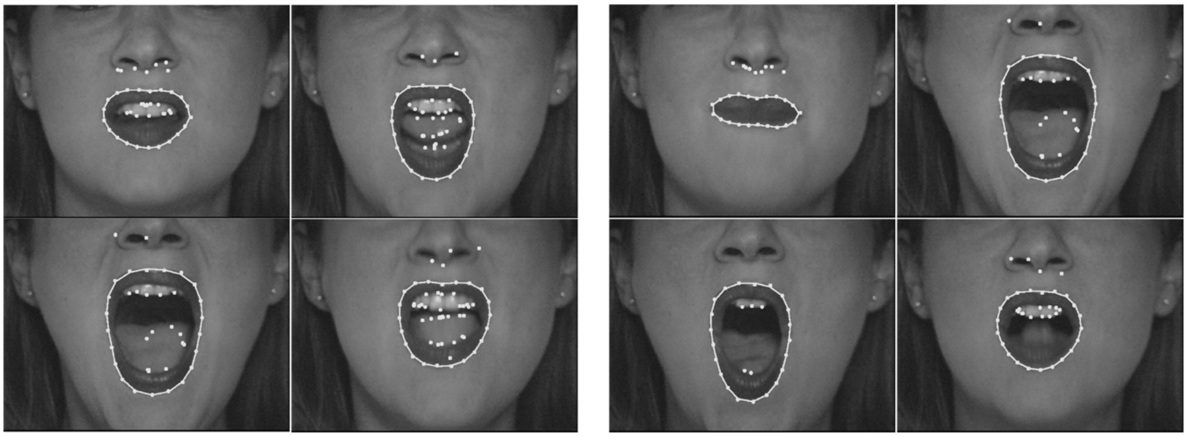


Fig. 11. Lip tracking (singing) with S-PDAF. Frames 45, 46, 60, 61 (first row) and frames 66, 67, 87, 88 (second row).

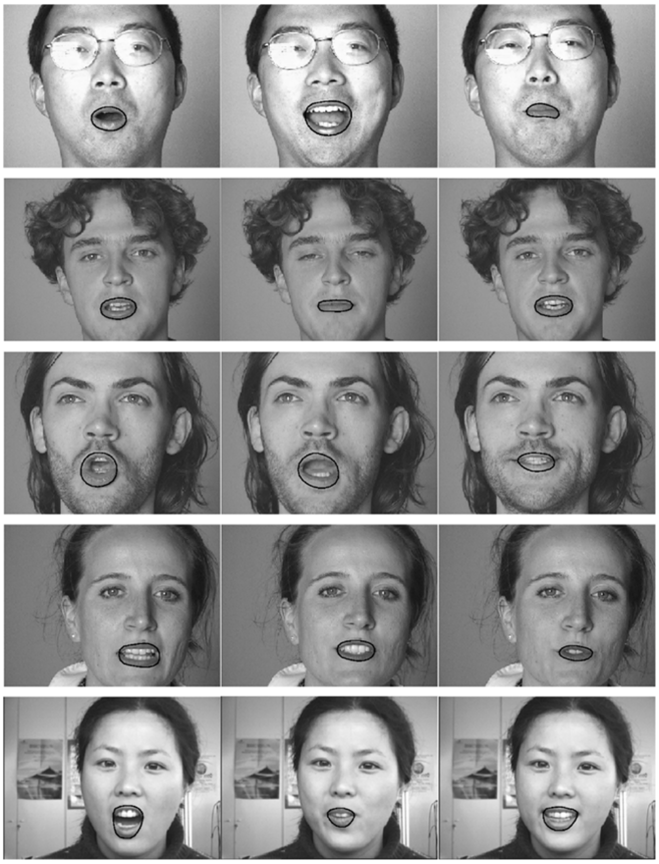


Fig. 12. Lip tracking in RGB images.

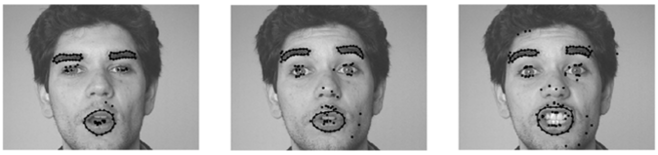


Fig. 13. Tracking of multiple objects (frames 10, 32, 44).

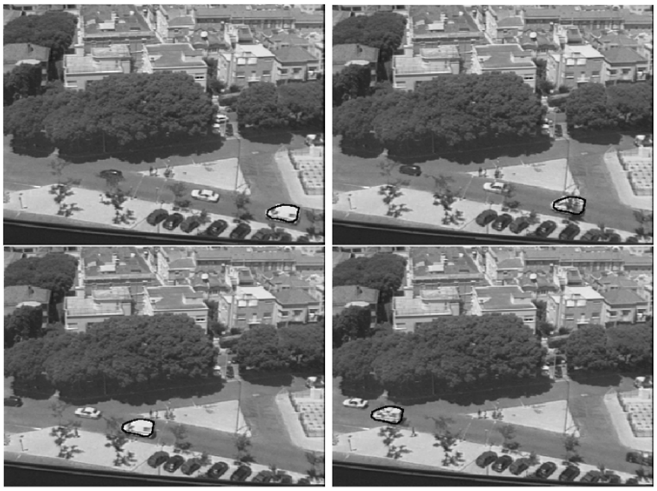


Fig. 14. Tracking results with S-PDAF. Frames 14, 20 (first row) and frames 34, 46 (second row).

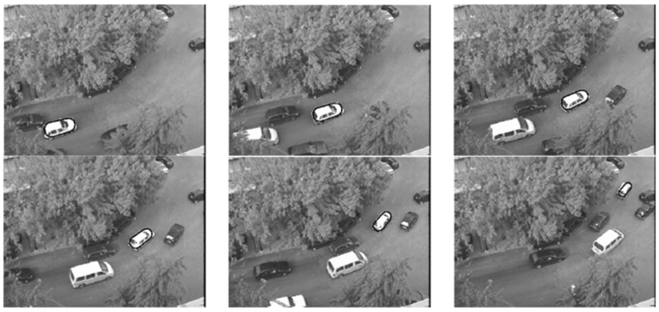


Fig. 15. Tracking results with S-PDAF. Frames 5, 19, 28 (first row) and frames 37, 47, 61 (second row).

how to discriminate valid data from outliers. This paper avoids a hard decision using middle level features (strokes) and by considering all possible hypothesis of valid/invalid label sequences which are denoted data interpretations. All interpretations are used to update the shape and motion estimates in each new image. However each interpretation has its own confidence degree (association probability). The most probable interpretations have larger influence on the shape and motion estimates than the least probable ones. The proposed algorithm was denoted as S-PDAF since it is an extension of the PDAF method proposed by Bar-Shalom and Fortmann [5] in the context of target tracking from radar measurements. The algorithm proposed in this paper was tested in several shape estimation problems (e.g., gesture estimation, lip tracking, and surveillance applications). A robust performance was observed, even in the presence of a large number of outliers and sudden shape changes.

Robust shape tracking has also been considered using other types techniques such as particle filtering [24], probabilistic graph matching [25], and concave optimization [26]. The work presented in this paper has been recently extended for tracking with multiple dynamic models [27].

Appendix

Covariance Update: The covariance of the state estimate is

$$\begin{aligned} P(k \mid k) &= E\left\{[x(k) - \hat{x}(k \mid k)][x(k) - \hat{x}(k \mid k)]^T \mid Y^k\right\} \\ &= E\left\{\left(\underbrace{x(k)x(k)^T}_{P^1} - \underbrace{x(k)\hat{x}(k \mid k)^T}_{P^2} \right.\right. \\ &\qquad - \underbrace{\hat{x}(k \mid k)x(k)^T}_{P^{2^T}} \\ &\qquad \left.\left. + \underbrace{\hat{x}(k \mid k)\hat{x}(k \mid k)^T}_{P^3}\right) \mid Y^k\right\} \end{aligned} \tag{40}$$

where

$$\begin{aligned} P^1 &\triangleq E\{x(k)x(k)^T \mid Y^k\} \\ &= \sum_{i=0}^{m_k} E\left\{x(k)x(k)^T \mid I_i(k), Y^k\right\}\alpha_i(k). \end{aligned} \tag{41}$$

Attending that

$$cov\{x(t)\} = E\{x(t)x(t)^T\} - \hat{x}(t \mid t)\hat{x}(t \mid t)^T. \tag{42}$$

The first term is

$$P^1 = \sum_{i=0}^{m_k} \alpha_i(k)\left[P_i(k \mid k) + \hat{x}_i(k \mid k)\hat{x}_i(k \mid k)^T\right]. \tag{43}$$

The second term in (40) is

$$\begin{aligned} P^2 &\triangleq -\sum_{i=0}^{m_k} E\left\{x(k)\hat{x}(k \mid k)^T \mid I_i(k), Y^k\right\}\alpha_i(k) \\ &= -\left(\sum_{i=0}^{m_k} E\left\{x(k) \mid I_i(k), Y^k\right\}\alpha_i(k)\right)\hat{x}(k \mid k)^T \\ &= -\left(\sum_{i=0}^{m_k} \hat{x}_i(k \mid k)\alpha_i(k)\right)\hat{x}(k \mid k)^T \\ &= -\hat{x}(k \mid k)\hat{x}(k \mid k)^T = P^{2^T}. \end{aligned} \tag{44}$$

The third term is

$$\begin{aligned} P^3 &\triangleq \hat{x}(k \mid k)\hat{x}(k \mid k)^T \sum_{i=0}^{m_k} \alpha_i(k) \\ &= \hat{x}(k \mid k)\hat{x}(k \mid k)^T = -P^2. \end{aligned} \tag{45}$$

Combining (41) and (44) into (40) yields

$$\begin{aligned} P = \sum_{i=0}^{m_k} \alpha_i(k)\left[P_i(k \mid k) + \hat{x}_i(k \mid k)\hat{x}_i(k \mid k)^T\right] \\ -\hat{x}(k \mid k)\hat{x}(k \mid k)^T \end{aligned} \tag{46}$$

with the conditional covariance

$$P_i(k \mid k) = (I - K_i(k)C_i)P(k \mid k-1) \tag{47}$$

we can write (46) in the form

$$\begin{aligned} P = &\left[I - \sum_{i=1}^{m_k} \alpha_i(k)K_i(k)C_i\right]P(k \mid k-1) \\ &+ \sum_{i=0}^{m_k} \alpha_i(k)\hat{x}_i(k \mid k)\hat{x}_i(k \mid k)^T \\ &- \hat{x}(k \mid k)\hat{x}(k \mid k)^T. \end{aligned} \tag{48}$$

Acknowledgment

The authors thank J. P. Costeira and J. Maciel of ISR for suggesting the experiments on facial estimation with multiple trackers. They also provided the color sequences used in these experiments.

The robust tracker was used by J. Costeira and J. Maciel to animate a two-dimensional model of a human face with good visual results. The face model used in these experiments was described in [23].

References

[1] A. Blake and M. Isard, *Active Contours*. New York: Springer, 1998.

[2] D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, ch. 1, pp. 3–20.

[3] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[4] J. Evans and R. Evans, "Image-enhanced multiple model tracking," *Automatica*, vol. 35, pp. 1769–1786, 1999.

[5] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

[6] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching ," in *Proceedings of the 6th International Conference on Computer Vision (ICCV-98).* Bombay, India: Narosa, 1998, pp. 107–112.

[7] J. Nascimento and J. Marques, "Robust shape tracking in the presence of cluttered background on image processing," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Vancouver, BC, Canada, 2000, pp. 82–85.

[8] D. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Comput. Vis.*, vol. 5, pp. 195–212, 1990.

[9] B. North and A. Blake, "Learning dynamical models using expectation-maximization," in *Proc. 6th Int. Conf. Computer Vision*, 1998, pp. 384–389.

[10] J. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, no. 5, pp. 607–615, 1982.

[11] A. Baumberg and D. Hogg, "Learning deformable models for tracking the human body," in *Motion Based Recognition*, R. Jain and M. Sha, Eds. Norwell, MA: Kluwer, 1997, pp. 39–60.

[12] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[13] S. Lucey, S. Sridharan, and V. Chandran, "Initialized eigenlip estimator for fast lip tracking using linear regression," in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 3, 2000, pp. 182–185.

[14] A. Rajagopalan and R. Chellappa, "Vehicle detection and tracking in video," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2000, pp. 351–355.

[15] F. D. la Torre, J. Vitria, P. Radeva, and J. Melenchon, "Eigenfiltering for flexible eigentracking (efe)," in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 3, 2000, pp. 1118–1121.

[16] T. Faruquie, A. Majumdar, N. Rajput, and L. Subramaniam, "Large vocabulary audio-visual speech recognition using active shape models," in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 3, 2000, pp. 110–113.

[17] H. Ney, “Stochastic modeling: From pattern classification to speech recognition and translation,” in *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 3, 2000, pp. 25–32.

[18] L. Rabiner, “A tutorial on hidden Markov models and selected applications,” in *Proc. IEEE Speech Recognition*, vol. 77, 1989, pp. 257–286.

[19] S. Basu, C. Neti, N. Rajput, A. Senior, L. Subramaniam, and A. Verma, “Audio-visual large vocabulary continuous speech recognition in the broadcast domain,” in *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Nagoya, Japan, Sept. 1999, pp. 475–481.

[20] G. Potamianos and H. P. Graf, “Discriminative training of hmm stream exponents for audio-visual speech recognition,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 6, 1998, pp. 3733–3736.

[21] A. Verma, T. Faruquie, C. Neti, S. Basu, and A. Senior, “Late integration in audio-visual continuous speech recognition,” in *Auto. Speech Recognit. and Understanding*, vol. 1, 1999, pp. 71–74.

[22] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[23] J. Maciel and J. Costeira, “Holistic synthesis of human face images,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 6, Phoenix, AZ, 1999, pp. 3545–3548.

[24] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” in *Proc. European Conf. Computer Vision*, vol. 1, 1996, pp. 343–356.

[25] G. Mori and J. Malik, “Estimating human body configuration using shape context matching,” in *Proc. European Conf. Computer Vision*, vol. 3, 2002, pp. 666–680.

[26] J. Maciel and J. Costeira, “A global solution to sparse correspondence problems,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 187–199, Feb. 2003.

[27] J. Nascimento and J. S. Marques, “Robust multi-model filter for shape tracking in the presence of outliers,” *Pattern Recognit.*, vol. 35, pp. 2711–2718, Dec. 2002.

Jacinto C. Nascimento (M’03) received the B.S. and E.E. degrees from the Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal, in 1993 and 1995, respectively, and the M.Sc. and Ph.D. degrees from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 1998 and 2003, respectively.

Presently, he is a researcher at IST, where he is affiliated with Institute for Systems and Robotics. His research interests are image processing, shape tracking and surveillance systems.

Jorge S. Marques received the E.E., M.Sc., and Ph.D. degrees from the Technical University of Lisbon, Lisbon, Portugal, in 1981, 1984, and 1990, respectively, and the Aggregation title in 2002.

Presently, he is Associate Professor at the Electrical Engineering Department, Instituto Superior Tecnico, Lisbon, and a Researcher at the Institute for Systems and Robotics. He has published over 90 papers in international journals and conferences and is author of the book *Pattern Recognition: Statistical and Neural Methods* (IST Press (Editions), September 1999, in Portuguese) . His research interests are in the areas of statistical image processing, shape analysis, and pattern recognition.

Dr. Marques was President of the Portuguese Association for Pattern Recognition (APRP) during 2001–2003 and a member of the IAPR Governing Board.