Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 295–307

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Using middle level features for robust shape tracking ☆

Jacinto C. Nascimento [a,*], Arnaldo J. Abrantes [b], Jorge S. Marques [a]

[a] *Institute Superior Tecnico, IST/ISR, Torre Norte, Av. Rovisco Pais, 1049-001 Lisbon, Portugal*
[b] *ISEL, R. Conselheiro Emídio Navarro, 1949-014 Lisbon, Portugal*

Received 13 November 2001; received in revised form 3 June 2002

Abstract

Shape tracking with low level features (e.g., edge points) often fails in complex environments (e.g., in the presence of clutter, inner edges, or multiple objects). Two alternative methods are discussed in this paper. Both methods use middle level features (data centroids, strokes) which are more informative and reliable than edge transitions used in most tracking algorithms. Furthermore, it is assumed in this paper that each feature can be either a valid measurement or an outlier. A confidence degree is assigned to each feature or to a given interpretation of all visual features. Features/interpretations with high degrees of confidence have large influence on the shape estimates while features/interpretations with low degrees of confidence have negligible influence on the shape estimates. It is shown in this paper that both items (middle level features and confidence degrees) lead to a significant improvement of the tracker robustness and performance in the presence of clutter and abrupt shape and motion changes.

Keywords: Middle level features; Strokes; Centroid; Gaussians mixture; EM; Shape probabilistic data association filter

1. Introduction

Object tracking is a challenging problem which has been extensively studied for more than three decades (Nagel, 2000). Active contours are among the most popular approaches (Kass et al., 1987). They represent the object boundary by an elastic model attracted by low level features, e.g., edge points (Cohen and Cohen, 1993) or transition points, obtained by directional search (Tagare, 1997). The estimation of the model parameters is often performed by Kalman filtering when linear state models are adopted to describe the evolution of the unknown parameters and the visual features detected in the image. Non-linear filtering approaches (e.g., particle filter, Blake and Isard, 1998; mixture of Gaussians, Marques and Lemos, 2001) are used when the observation noise is non-Gaussian or when multiple models are used to track the object motion and shape evolution. Unfortunately, feature detection is an ill-posed problem and the low level features detected in the image often contain outliers which limit the performance of shape tracking algorithms (see Fig. 1).

☆ This work was partially supported by POSI and by FCT under the scope of project LTT.

[*] Corresponding author.

E-mail addresses: pcjan@pop.isr.ist.utl.pt (J.C. Nascimento), aja@isel.pt (A.J. Abrantes), jsm@isr.ist.utl.pt (J.S. Marques).

PII: S0167-8655(02)00243-X

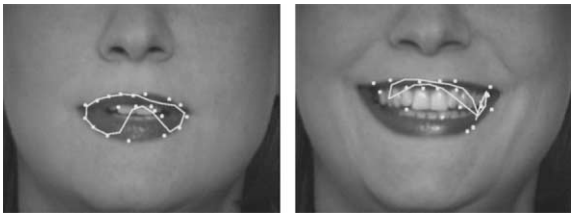

Fig. 1. Typical difficulties of shape tracking based on low level features.

Several methods were proposed to alleviate this difficulty. Some impose additional restrictions to the object shape by adopting rigid templates (Blake et al., 1993), or eigenshapes learned from the data (Cootes et al., 1994). Temporal restrictions have also been considered by describing the evolution of the motion and shape parameters using stochastic difference equations. The parameters of the dynamic model can also be trained from video sequences using standard identification methods (Baumberg and Hogg, 1997; Blake et al., 1995).

All these methods improve the tracker performance. However, none of these approaches solves the segmentation problem i.e., none of them allows to discriminate valid data from the outliers which hamper the performance of the shape trackers. An ad hoc procedure used to improve Kalman estimates consists of using a validation gate computed from the predicted object boundary, discarding the observations falling outside (Cootes et al., 1994). This method works well if the object motion is slow and predictable but fails in more complex situations. Non-linear filtering methods based on non-Gaussian noise distributions have also been used to address this problem (Isard and Blake, 1998).

This paper considers two robust filtering methods: a centroid based method and a stroke based method. The first algorithm performs a fuzzy labeling of low level features (edge points) detected in the image, considering them either as reliable or as outliers. A degree of confidence is assigned to each feature using an object model and a noise model. The object model is a mixture of Gaussians which attempts to represent the boundary of the object in every new frame while the outliers are represented by a single Gaussian distribution with a large covariance matrix. The mixture parameters are interpreted as middle level features which summarize the image data useful for the estimation of the object boundary. This type of approach has been adopted in other works (e.g., see Abrantes and Marques, 1996; Wu et al., 2000) in which intermediated representations of the data are used for compression and robustness purposes.

The second algorithm is based on edge strokes. Instead of assigning a degree of confidence to each stroke, this method considers sequences of stroke labels and computes a degree of confidence for each sequence. Each sequence corresponds to a classification of all the strokes detected in the image as valid or as outliers (data interpretations). The update of the model parameters is inspired in the probabilistic data association filter (PDAF) proposed by Bar-Shalom and Fortmann in the context of target tracking in clutter (Bar-Shalom and Fortmann, 1988), providing a fast and robust filtering algorithm. The shape tracker derived under these assumptions is therefore denoted as shape PDAF (S-PDAF) and it was described before in (Nascimento and Marques, 2000).

Both methods use middle level features and robust estimation techniques. The middle level features used in this paper do not depend on the object to be tracked. Other approaches have been considered in which higher level features are used and trained (e.g., blobs, Oliver et al., 1997) for specific types of objects. These methods achieve good tracking results although they are adapted to specific tracking problems.

This paper is organized as follows. Sections 2 and 3 describe the robust tracking algorithms in detail. Section 4 presents the shape model used in this paper. Section 5 presents results obtained by both methods in the presence of clutter and Section 6 concludes the paper.

2. Centroid based method

Let I denote an image and $y = \{y_1, \ldots, y_N\}$ the coordinates of the edges points detected in I. We wish to approximate a subset of y by a parametric curve $c(s)$, where s is the arc length.

Two difficulties have to be considered: first we need to know which edge points are valid i.e., which edge points belong to the object boundary and which are outliers (segmentation problem). Second we need to match each valid feature with a contour point (matching problem).

To overcome these difficulties it will be assumed in this section that the edge points y_i are independent random variables. Their distribution is described by a mixture of $N+1$ Gaussians

$$p(y_i) = \sum_{k=1}^{N+1} \alpha_k \mathcal{N}(y_i; \mu_k, R_k) \tag{1}$$

where $\mathcal{N}(y_i; \mu, R)$ denotes a normal distribution with mean μ and covariance matrix R; α_k, μ_k, R_k denote the weight, mean and covariance matrix of the kth mixture component. The first N Gaussians are used to represent the object boundary and their means are located at equally spaced points belonging to the curve $c(s)$. The $(N+1)$th Gaussian has a large covariance matrix and it is used to represent the invalid data points (outliers). The object boundary and the $N+1$ Gaussians are shown in Fig. 2.

To solve the problems which were described above (data segmentation and matching) all we need is to estimate which Gaussian generated each edge point y_i. Since it is not possible to obtain an error free classification of the edge points, a set of probabilities (confidence degrees) will be assigned to each feature. The estimation of the mixture parameters and confidence degrees is recursively performed using the EM algorithm (Dempster et al., 1977).

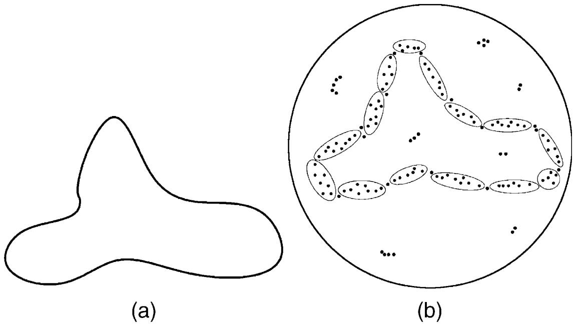

Fig. 2. (a) Object boundary and (b) Gaussian mixture and low level features.

This approach is useful for the analysis of still images. In tracking problems we have to merge past information available about the object shape with the new information obtained from the current frame. This can be done by using Kalman filtering. It will be assumed that the object boundary c_t depends on a vector of parameters $x_t \in \Re^n$ which evolves in time, i.e., $c_t(s) = c(s, x_t)$. [1] The evolution of x_t is usually described by a discrete state model with random input

$$x_t = Ax_{t-1} + w_t \tag{2}$$

where A is a matrix of dynamics and $w_t \sim \mathcal{N}(0, Q)$ is a white Gaussian noise with zero mean and covariance matrix Q.

Since B-spline curves and geometric transformations are used in this paper, the state vector x contains either the coordinates of the control points or the parameters of the geometric transformation. The algorithm described in this section is more general however and can be used with other type of models (e.g., articulated models, Cheng and Moura, 1999; Hogg, 1983).

In the proposed algorithm, the Kalman filter is not driven by low level features directly, but with mixture parameters instead. The means of the first N Gaussians are considered as noisy observations of the object boundary at N sample points. Therefore, it is assumed that

$$\mu_i = c(s_i, x) + v_i \tag{3}$$

where v_i is an observation noise with zero mean and covariance R_i estimated by the EM algorithm. Table 1 summarizes the proposed algorithm.

3. Shape probabilistic data association filter

3.1. Shape tracking

The S-PDAF is a robust tracking algorithm recently proposed in (Nascimento and Marques, 2000). The main ideas are simple. First, low level

[1] The shape model used in this work is described in Section 4.

Table 1
Centroid based tracker: steps performed for each new frame

1. *Prediction:* Predict the state vector $\hat{x}_t^-$ and covariance matrix P_t^- using the prediction step of the Kalman filter

$$\hat{x}_t^- = A\hat{x}_{t-1}$$

$$P_t^- = AP_{t-1}A^{\mathrm{T}} + Q$$

2. *Image analysis:* Detect edge points y in the image I_t
3. *Mixture initialization:* Initialize $N+1$ centroids and covariances as follows: the means μ_i, $i = 1, \ldots, N$, are obtained by sampling the predicted contour at equally spaced points. The 2D covariance matrices, R_i, $i = 1, \ldots, N$ are initialized in such a way that the main principal direction is tangent to the object boundary at each sampling point (see Fig. 2(b)). The last Gaussian μ_{N+1} is located at the center of the predicted contour with a large covariance R_{N+1}. The initial values of the parameters are therefore

$$\theta = [\mu_1^-, \ldots, \mu_N^-,\ R_1^-, \ldots, R_N^-,\ \mu_{N+1}^-,\ R_{N+1}^-]$$

4. *EM algorithm:* update the mixture parameters μ_i, R_i, $i = 1, \ldots, N+1$ by using the EM algorithm:
   - E-step: compute the weights

$$\omega_{ij} = P\{l_j = i \mid y, \theta^-\}$$

where l_j denotes the label of the jth feature and $P\{l_j = i \mid y, \theta^-\}$ is the a posteriori probability of assigning y_j to the model sample μ_i.

   - M-step: update the Gaussian parameters θ^-.

5. *Filtering:* Update $\hat{x}_t$, P_t, using the Kalman filter

$$\hat{x}_t = \hat{x}_t^- + K_t(\mu_t - C\hat{x}_t^-)$$

$$K_t = P_t^- C^{\mathrm{T}}(CP_t^- C^{\mathrm{T}} + R_t)^{-1}$$

$$P_t = (I - K_t C)P_t^-$$

where the ith blocks of μ_t, R_t are given by

$$\mu_{it} = \frac{\sum_j w_{ij} y_j}{\sum_j w_{ij}} \qquad R_{it} = \frac{\sum_j w_{ij}(y_j - \mu_i)(y_j - \mu_i)^{\mathrm{T}}}{\sum_j w_{ij}}$$

features (intensity transitions obtained by directional search along lines orthogonal to the object boundary) are detected and organized in M strokes. [2]

Strokes are obtained by matching feature points detected at consecutive search lines. This is accomplished by using the mutual favorite pairing method (Huttenlocher and Ullman, 1990). 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. Small strokes are discarded. Fig. 3(a) shows the contour samples (dots) and the detected features (crosses). Fig. 3(b) displays the strokes obtained by edge linking. Isolated points are discarded.

Each stroke is classified as valid or invalid. The stroke is classified as valid if it belongs to the boundary of the object to be tracked. Otherwise it is classified as an outlier. Since each stroke is either valid or invalid, 2^M hypothesis (interpretations) have to be considered. A data interpretation I_i is defined as a binary sequence $I_i^1, \ldots, I_i^M$, where $I_i^j \in \{0, 1\}$ is the label of the jth stroke in the interpretation I_i.

[2] Since the object boundary is usually unknown, an initial (predicted) boundary estimate is used instead.

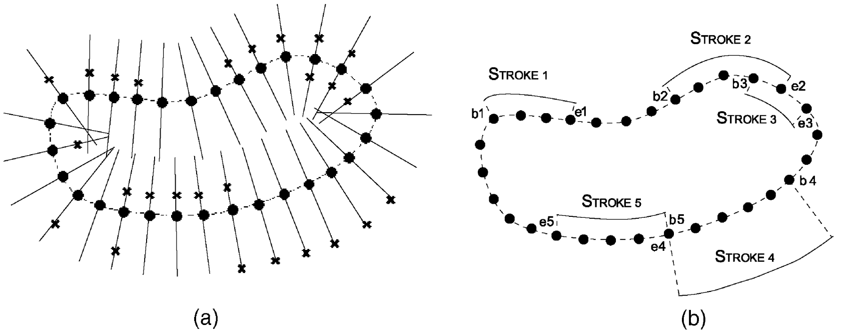

Fig. 3. (a) Detected features and (b) strokes ((•): model points; (×): detected features; (- - -): predicted contour).

The object shape and position are described as in Section 2, however, the output equation depends on the interpretation i.e., we have a bank of output equations each one associated to a different data interpretation

$$y_{it} = C_{it}x_t + \eta_t \tag{4}$$

where C_i is the observation matrix associated to the ith interpretation and $\eta_t \sim \mathcal{N}(0, R_i)$ is a white Gaussian measurement noise. The observation matrices C_i, C_j associated with two interpretations I_i, I_j are always different since they represent different data points (outliers are not described by (4)). For example, in Fig. 3(b) a possible interpretation is $I_i = (11\,000)$. In this case, the matrix C_i includes the rows associated with the two first strokes.

Since there are multiple interpretations the propagation of the a posteriori density of the state vector $p(x_t|Y^t)$ given the current and the past observations Y^t, is a mixture of Gaussians, with a number of modes increasing with t (Tugnait, 1982). To avoid the combinatorial explosion associated with the use of multiple interpretations, it will be assumed that the distribution of the unknown parameters x_t, given past observations Y^{t-1}, is Gaussian, i.e.,

$$p[x_t|Y^{t-1}] = \mathcal{N}[x_t; \hat{x}_t^-, P_t^-] \tag{5}$$

where $\hat{x}_t^-$, P_t^- are the mean and covariance of x_t given Y^{t-1}. This assumption replaces a complex prior by a simpler one with the same mean and covariance matrix, discarding higher order statistics.

The computation of the state estimate and uncertainty given the current and the past observation are given by,

$$\hat{x}_t \triangleq E[x_t|Y^t] \tag{6}$$

$$P_t \triangleq E\left\{[x_t - \hat{x}_t][x_t - \hat{x}_t]^{\mathrm{T}}|Y^t\right\} \tag{7}$$

Since it is possible to have multiple data interpretations, they all have to be considered. Eq. (6) can be written as

$$\hat{x}_t = \sum_i \alpha_{it} \int x_t p(x_t|I_{it}, Y^t)\mathrm{d}x_t \tag{8}$$

$$\hat{x}_t = \hat{x}_t^- + \sum_{i=1}^{m_k} \alpha_{it} K_{it} v_{it} \tag{9}$$

where $\alpha_{it} \triangleq p(I_{it}|Y^t)$ is the a posteriori probability of the ith interpretation, also denoted as the data association probability, $v_{it} = y_{it} - C_i\hat{x}^-, K_{it}$ are the innovation and the Kalman gain respectively.

A recursive equation can also be derived for the covariance matrix (see Nascimento and Marques, 2000).

$$P_t = \left[I - \sum_{i=1}^{m_k} \alpha_{it} K_{it} C_i\right] P_t^- + \sum_{i=0}^{m_k} \alpha_{it}\hat{x}_{it}\hat{x}_{it}^{\mathrm{T}} - \hat{x}_t\hat{x}_t^{\mathrm{T}} \tag{10}$$

These equations are closely related to the PDAF method proposed by Bar-Shalom, Fortmann in the context of target tracking (Bar-Shalom and Fortmann, 1988). The main differences between both methods are twofold. First the S-PDAF estimates the evolution of shape and motion parameters instead of a single point target. Second it uses a different observation model tailored to the shape tracking problem.

The Gaussian hypothesis assumed in the S-PDAF is not accurate. However, it leads to fast update equations without a combinatorial explosion of the number of modes and captures the main properties (mean and uncertainty) of the unknown parameters to be tracked.

3.2. Association probabilities

Since we do not know which interpretation is valid, a probability α_{it} (confidence degree) is computed for each data interpretation. This requires a probabilistic model for the image strokes. Three items will be considered in this model: the stroke lengths, the distances from the stroke points to the best estimate of the object boundary and stroke superposition. The following variables are used to describe the association probability: M, number of strokes detected in the image; b, e, vectors with the first and last indices of all data strokes; I_i, interpretation. The a posteriori probability of the ith interpretation is

$$\alpha_{it} = P\{I_{it}|y_t, b, e, M, Y^{t-1}\} \tag{11}$$

which can be decomposed as follows

$$\alpha_{it} = cp(y_t|I_{it}, b, e, M, Y^{t-1})p(I_{it}|b, e, M, Y^{t-1}) \tag{12}$$

where y_t denotes the data observed at instant t and c is a normalization constant.

It is assumed that all features in y_t are independently generated with Gaussian distributions if they belong to a valid stroke and uniform distribution if they belong to an invalid stroke, according to ith interpretation I_i. Therefore

$$p(y_t|I_{it}, b, e, M, Y^{t-1}) = \prod_{j=1}^{M} \prod_{n=b^j}^{e^j} p(y_t^j(s_n)|I_{it}^j) \tag{13}$$

where

$$p(y_t^j(s_n)|I_{it}^j, Y^{t-1}) = \begin{cases} P_G^{-1}\mathcal{N}(v_t^j(s_n); 0, S_{it}^j(s_n)) \\ V_t^j(s_n)^{-1} \quad \text{if } I_{it}^j = 0 \end{cases} \tag{14}$$

$V_t^j(s_n, t)$ is the area of the search region; P_G is a normalization constant; and $S_{it}^j(s_n) = C_i\hat{P}_tC_i^T + R_i$.

To define $p(I_{it}|b, e, M, Y^{t-1})$ it is assumed that the stroke labels are independent,

$$p(I_{it}|b, e, M) = \prod_j p(I_{it}^j|b, e, M) \tag{15}$$

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

$$p(I_{it}|b, e, M) = \begin{cases} (k/L)l^j + c & \text{if } I_{it}^j = 1 \\ 1 - (k/L)l^j - c & \text{otherwise} \end{cases} \tag{16}$$

where k, c are constants, L is the contour length, l^j is the length of the jth stroke.

Fig. 4 shows a simple example with two strokes. The association probabilities are displayed in Table 2. In this example, the most probable interpretation ($\alpha_4 = 0.5006$) is the one which considers both strokes as valid. Finally, the a posteriori distribution of the unknown parameters is computed, considering all possible interpretations weighted by their associated probabilities.

Table 3 summarizes the S-PDAF algorithm.

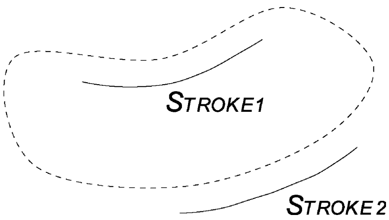

Fig. 4. Feature detection: predicted contour (- - -) and detected strokes (—).

Table 2
Data interpretations and association probabilities

	I_i^1	I_i^2	α_i
1	0	0	0.0852
2	0	1	0.1915
3	1	0	0.2226
4	1	1	0.5006

Table 3
Shape PDAF

1. *Prediction:* Predict the state vector $\hat{x}_t^-$ and covariance matrix P_t^- using the prediction step of the Kalman filter

$$\hat{x}_t^- = A\hat{x}_{t-1}$$

$$P_t^- = AP_{t-1}A^{\mathrm{T}} + Q$$

2. *Image analysis:* Detect features along lines orthogonal to the predicted contour and organize them in Strokes
3. *Association probabilities:* For each data interpretations I_i, compute the association probability α_{it}, using (13) and (16)

$$\alpha_{it} = cp(y_t|I_{it}, b, e, M, Y^{t-1})p(I_{it}|b, e, M, Y^{t-1})$$

4. *Filtering:* Update the state estimate and uncertainty, according to

$$\hat{x}_t = \hat{x}_t^- + \sum_{i=1}^{m_k} \alpha_{it}K_{it}v_{it}$$

$$P_t = \left[I - \sum_{i=1}^{m_k} \alpha_{it}K_{it}C_i\right]P_t^- + \sum_{i=0}^{m_k} \alpha_{it}\hat{x}_{it}\hat{x}_{it}^{\mathrm{T}} - \hat{x}_t\hat{x}_t^{\mathrm{T}}$$

4. Dynamic shape modelling

To estimate the evolution of the shape and motion parameters, we first need to represent shape and motion evolution as the output of stochastic difference equations. To represent a moving object in a frame t, it is assumed that the object is a transformed version of a reference shape plus an additional deformation. By shape it is meant either the boundary of the object or a set of detected strokes in the image. Let $c_t : I \rightarrow \Re^2$ be a parametric representation of the reference curve, where $I \in \Re$ and c^{r} a reference shape of the object (which can be obtained in the first image of the sequence). It is assumed that

$$c_t = \mathscr{T}c^{\mathrm{r}} + v \tag{17}$$

where v is a noise curve and $\mathscr{T}$ is a geometric transformation which conveys motion and distortion information. In this paper several transforms are considered depending on the application. It is assumed that $\mathscr{T}$ is either a translation, an Euclidean transformation or an affine map. Therefore, the boundary model $c_t(s) = (c_{1t}, c_{2t})$ is given by

Translation:

$$\begin{aligned} c_{1t}(s) &= c_1^{\mathrm{r}}(s) + x_{1t} + d_1(s) + v_1(s) \\ c_{2t}(s) &= c_2^{\mathrm{r}}(s) + x_{2t} + d_2(s) + v_2(s) \end{aligned} \tag{18}$$

Euclidean similarity:

$$\begin{aligned} c_{1t}(s) &= x_{1t}c_1^{\mathrm{r}}(s) - x_{3t}c_2^{\mathrm{r}}(s) + x_{2t} + d_1(s) + v_1(s) \\ c_{2t}(s) &= x_{3t}c_1^{\mathrm{r}}(s) + x_{1t}c_2^{\mathrm{r}}(s) + x_{4t} + d_2(s) + v_2(s) \end{aligned} \tag{19}$$

Affine transform:

$$\begin{aligned} c_{1t}(s) &= x_{1t}c_1^{\mathrm{r}}(s) + x_{2t}c_2^{\mathrm{r}}(s) + x_{3t} + d_1(s) + v_1(s) \\ c_{2t}(s) &= x_{4t}c_1^{\mathrm{r}}(s) + x_{5t}c_2^{\mathrm{r}}(s) + x_{6t} + d_2(s) + v_2(s) \end{aligned} \tag{20}$$

where $s \in I$ is a parameter defining the location of a point in the stroke, $v(s) = (v_{1t}, v_{2t})$ is the measurement noise curve, $x_{1t} \ldots x_{6t}$ are unknown parameters to be estimated and $d(s) = (d_1(s), d_2(s))$ is the deformation curve, defined by

$$d_i(s) = \sum_{k=1}^{L} d_{ik}\phi_k(s) \tag{21}$$

where $\phi_k(s)$ are known basis functions and $d_{i1} \ldots d_{iN}$ are 2D deformation vectors. A popular solution consists of using B-spline basis functions although other approaches can be considered (Cootes et al., 1994; Dias, 1999).

Let us assume that the object moves during the acquisition interval. In this case, dynamic equations must be considered to represent the evolution of the model parameters. Let x_t denote the vector of unknown shape, motion and deformation parameters

$$x_t = [x_{1t}, \ldots, x_{Dt},\ \dot{x}_{1t}, \ldots \dot{x}_{Dt},\ d_{11}, \ldots, d_{1L},\ d_{21}, \ldots, d_{2L}]^{\mathrm{T}} \tag{22}$$

where $D = 2$, 4 or 6, in the case of translation, Euclidean similarity or affine motion respectively, and d_{iL} is the deformation parameters. Let y be a $2N \times 1$ vector obtained by sampling the object boundary at N equally spaced points

$$y = [c_{1t}(s_1), \ldots, c_{1t}(s_N),\ c_{2t}(s_1), \ldots, c_{2t}(s_N)]^{\mathrm{T}} \tag{23}$$

Eqs. (18)–(20) can be rewritten as in (3) and (4) with

Translation:

$$C = \begin{bmatrix} \mathbf{1}_{N\times1} & \mathbf{O}_{N\times1} & \mathbf{O}_{N\times2} & \mathbf{B}_{N\times L} & \mathbf{O}_{N\times L} \\ \mathbf{O}_{N\times1} & \mathbf{1}_{N\times1} & \mathbf{O}_{N\times2} & \mathbf{O}_{N\times L} & \mathbf{B}_{N\times L} \end{bmatrix} \tag{24}$$

Euclidean similarities:

$$C = \begin{bmatrix} M_x & -M_y & \mathbf{O}_{N\times4} & \mathbf{B}_{N\times L} & \mathbf{O}_{N\times L} \\ M_y & M_x & \mathbf{O}_{N\times4} & \mathbf{O}_{N\times L} & \mathbf{B}_{N\times L} \end{bmatrix}$$

with

$$M_x = \begin{bmatrix} c_1(s_1) & 1 \\ \vdots & \vdots \\ c_1(s_N) & 1 \end{bmatrix}, \quad M_y = \begin{bmatrix} c_2(s_1) & 0 \\ \vdots & \vdots \\ c_2(s_N) & 0 \end{bmatrix} \tag{25}$$

Affine transform:

$$C = \begin{bmatrix} M & \mathbf{O}_{N\times3} & \mathbf{O}_{N\times6} & \mathbf{B}_{N\times L} & \mathbf{O}_{N\times L} \\ \mathbf{O}_{N\times3} & M & \mathbf{O}_{N\times6} & \mathbf{O}_{N\times L} & \mathbf{B}_{N\times L} \end{bmatrix}$$

with

$$M = \begin{bmatrix} c_1(s_1) & c_2(s_1) & 1 \\ c_1(s_2) & c_2(s_2) & 1 \\ \vdots & \vdots & \vdots \\ c_1(s_N) & c_2(s_N) & 1 \end{bmatrix} \tag{26}$$

where $\mathbf{B}$

$$\mathbf{B} = \begin{bmatrix} \phi_1(s_1) & \phi_2(s_1) & \cdots & \phi_L(s_1) \\ \phi_1(s_2) & \phi_2(s_2) & \cdots & \phi_L(s_2) \\ \vdots & \ddots & \ddots & \vdots \\ \phi_1(s_N) & \phi_2(s_N) & \cdots & \phi_L(s_N) \end{bmatrix} \tag{27}$$

is an interpolation B-spline matrix, M is the matrix containing the coordinates of the object boundary $\mathbf{O}_{pq}$ is a $p \times q$ null matrix.

Remark that in case of S-PDAF algorithm we often have subsets of the stroke samples $c_t(s_1), \ldots, c_t(s_N)$ depending on the data interpretation considered (see Fig. 3(a)). Therefore the shape matrices C_i are obtained from C defined in (24)–(26) by removing the rows which are not considered as valid according to the ith interpretation.

Equations (22), (24)–(26) accounts for shape motion and deformation. Special cases can be considered. For example, if only rigid motion of the object is considered, the deformation parameters d_{ij} can be removed (in (22)), and the last columns of C (containing $\mathbf{B}$) are discarded. If the object velocity is to be ignored, the derivatives of x can be removed (in (22)), and the corresponding columns of the C matrix should also be eliminated.

5. Results

Experimental tests were performed to evaluate the algorithms described in the previous sections and to compare their performance with a Kalman tracker, i.e., a tracking method which considers all the data as valid features. Synthetic and real images were used to assess the performance of these methods.

Example 1. Fig. 5 shows a synthetic example. The goal is to estimate the object boundary from a set of detected features (strokes) represented as solid lines. For simplicity, the strokes are labeled as in Fig. 3(b). Looking at Fig. 5(a), it is easily concluded that the best interpretation classifies S_1, S_2, S_5 as valid data and S_3, S_4 as outliers, assuming that the object undergoes a translation motion. The results obtained by the three algorithms using a translation model without deformation are shown in Fig. 5(b)–(d). The shape estimates obtained with the robust methods described in this paper are much better than the estimates obtained with the Kalman tracker since they manage to neglect the influence of the outlier strokes and provide correct estimates of the translation using S_1, S_2 and S_5. This happens since these two methods use higher level features and robust estimation methods.

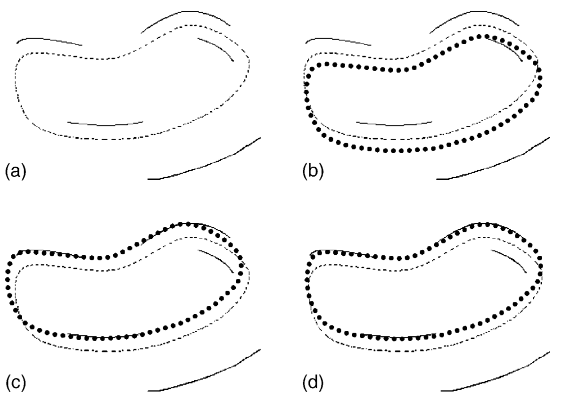

Fig. 5. Shape translation. (a) Initialization, results obtained with (b) Kalman filter, (c) centroid based tracker and (d) S-PDAF.

Example 2. Fig. 6 shows a more difficult example. In this case, the object boundary undergoes a translation and a rotation motion. The motion model adopted in this example is based on the group of Euclidean similarities with four degrees of freedom (translation, rotation and scaling without deformation). The centroid based tracker and the S-PDAF solve this problem well, estimating the rotation and discarding the influence of the outliers strokes. However, the Kalman filter gives a wrong answer. The object boundary is attracted towards S_4.

To understand better the behaviour of the algorithms, the same data was processed by allowing shape deformation. The results of this experiment are given in the Fig. 7. The Kalman tracker tries to match all the data points, even the outliers, by distorting the object shape. Better results are obtained with two middle level algorithms. The centroid based tracker also allows strong shape deformations close to the strokes S_2 and S_3, since this method does not consider overlapping strokes as invalid. The best results are obtained with the S-PDAF which manages to solve this problem well. The previous examples illustrate synthetic problems in which the noise model (data interpretations in the case of the S-PDAF and distribution noise in the centroid based tracker) play a key role in the performance of the algorithms.

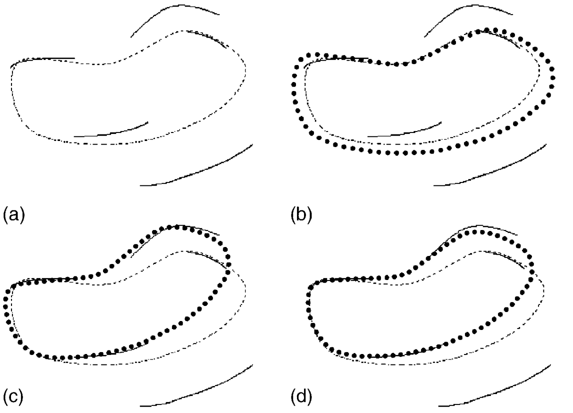

Fig. 6. Shape rotation. (a) Initialization, results obtained with (b) Kalman filter, (c) centroid based tracker and (d) S-PDAF.

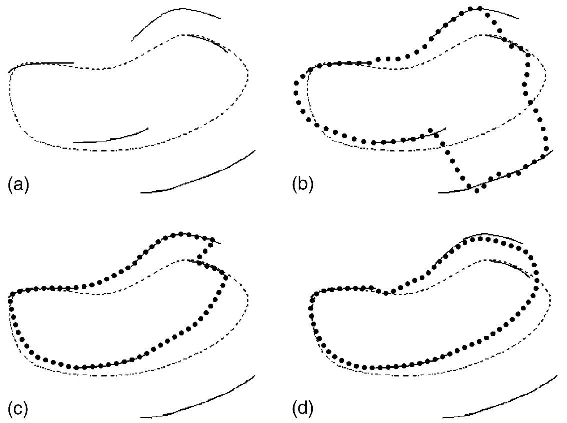

Fig. 7. Shape rotation with deformation. (a) Initialization, results obtained with: (b) Kalman filter, (c) centroid based tracker and (d) S-PDAF.

Example 3. Fig. 8 shows some results using real images. In this example the goal is to track the boundary of a car in a traffic sequence. The motion model used in this examples was the Euclidean similarities with 4 degrees of freedom ($x \in \Re^4$) which allows to represent shape translations, rotations and scaling. It is concluded that the Kalman filter, Fig. 8 (left column), provides wrong estimates of the rotation angle and translation vector. The shape estimates is being attracted towards the inner edges of the car to be tracked. This behaviour was already observed before in Fig. 6. The results obtained by the robust trackers are much better. Once again the confidence degrees assigned to the features play a crucial role to achieve robust estimates of the object boundary.

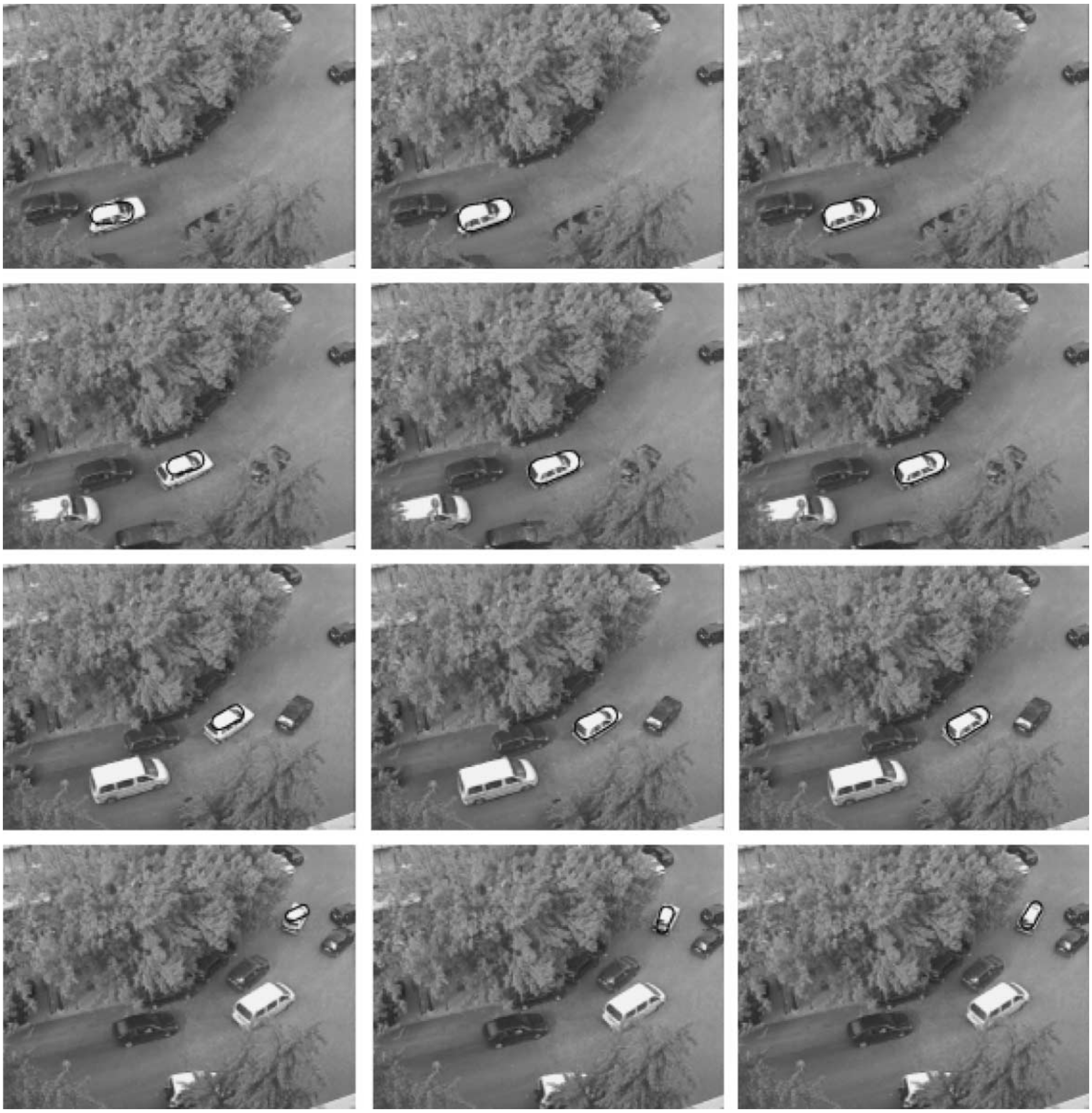

Fig. 8. Tracking results with Kalman (left), centroid based (center) and S-PDAF (right) (frames 8, 21, 32, 57) (dark line is the output of the algorithms).

Example 4. Figs. 9–11 show the performance of the algorithms in lip tracking. These experiments illustrate the estimation of the lips boundaries while a person is talking and singing. The sampling rate was 12 fps in both examples. In these examples the dots (in Kalman and S-PDAF trackers) and the thin white lines (in the centroid tracker) are the detected features. In the first example (woman talking) the centroid based tracker and the S-PDAF are able to track the lips during the whole sequence while the Kalman tracker becomes lost after the first 25 frames (see Fig. 9 for some details).

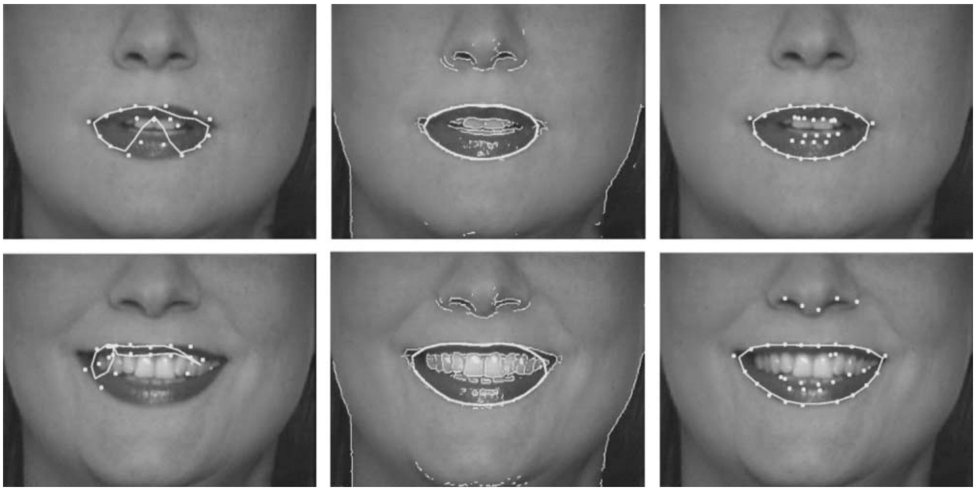

Fig. 9. Tracking results with Kalman (left), centroid based (center) and S-PDAF (right) (frames 28, 38). White points represent low level features.

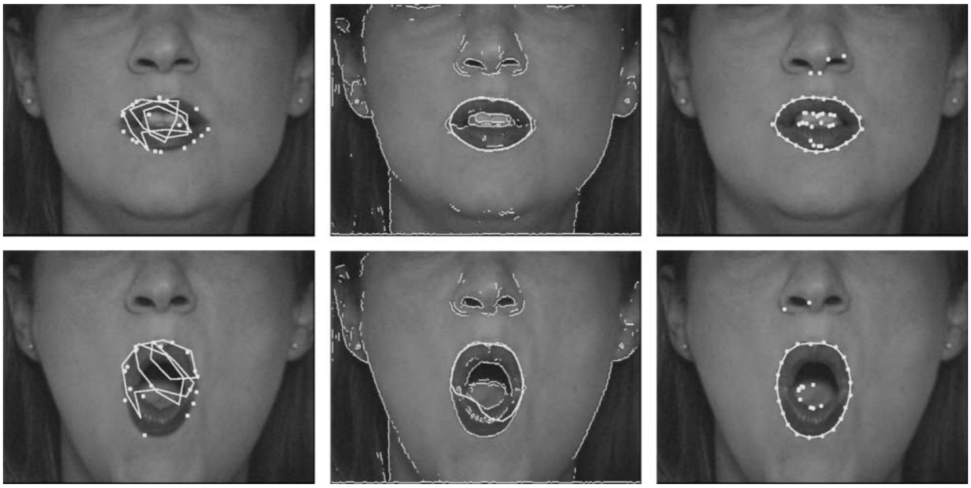

Fig. 10. Tracking of abrupt transitions with Kalman (left) centroid based (center) and S-PDAF (right) (frames 31, 32). White points represent low level features.

The three algorithms were also applied to sequences of a woman singing a song. The Kalman tracker loses the lips boundaries as before (see Figs. 10 and 11 (left)). The centroid based tracker tends to lose track during short intervals specially in the presence of abrupt motion changes but manages to recover after a few frames. These difficulties are shown in Figs. 10 and 11 (center). The S-PDAF always provides accurate estimates of the lips boundaries even in the case of abrupt shape deformations during the whole sequences (see Figs. 10 and 11 (right)).

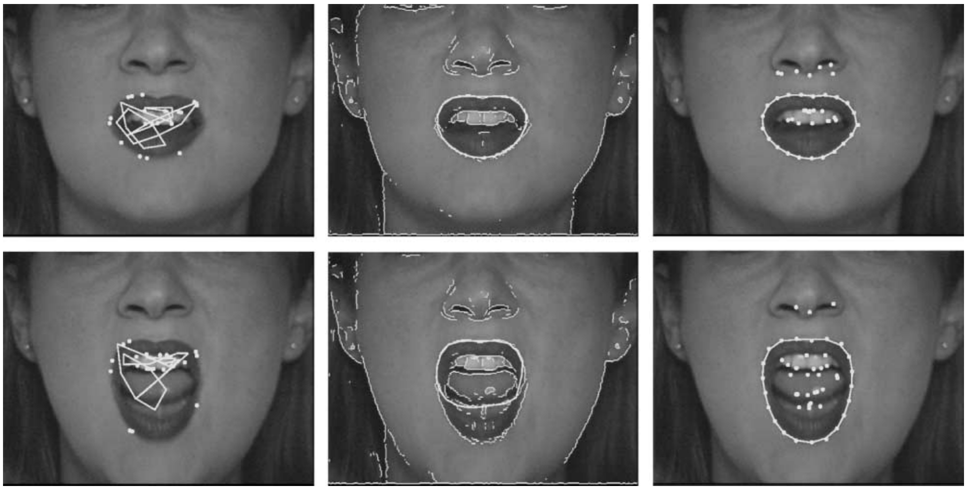

Fig. 11. Tracking of abrupt transitions with Kalman (left) centroid based (center) and S-PDAF (right) (frames 45, 46). White points represent low level features.

6. Conclusions

Three algorithms were studied in this paper a Kalman based tracker and two robust methods based on middle level features (a centroid based tracker and the S-PDAF). The Kalman tracker assumes that the features detected in the image are all reliable, therefore providing poor estimates of the object boundary when invalid features are detected in the image (outliers). The robust trackers assign a confidence degree to the features detected in the image therefore achieving much better results. The centroid based tracker and the S-PDAF assume that the features detected in the image may not be reliable. Each feature can be either valid data or outlier. Confidence degrees are computed which influence the contribution of each feature to the estimation of the object boundary. The centroid based tracker performs a fuzzy labeling of edge points (low level features) using an object model and a noise model. This allows to compute data centroids as weighted averages of the features locations. The S-PDAF uses strokes instead of edge points since they are more reliable. In this case, the middle level features (strokes) are directly computed from the image by image analysis techniques. The algorithm however assumes that some strokes are outliers. Since we do not know which are the good ones, all combinations of valid/invalid strokes are considered, each of them being denoted as a data interpretation, and a confidence degree is assigned to each data interpretations. In this case the confidence degrees are not assigned to individual features as in the centroid based method but to interpretations of all the detected features. Experimental results show that both algorithms perform better than the Kalman based tracker. The S-PDAF achieves the best results and shows a remarkable performance even in the presence of clutter and abrupt motion and shape changes.

Acknowledgements

We thank the suggestions made by the anonymous reviewers which helped to improve the paper.

References

Abrantes, A.J., Marques, J.S., 1996. A class of constrained clustering algorithms for object boundary detection. IEEE Trans. Image Process., 1507–1521.

Bar-Shalom, Y., Fortmann, T., 1988. Tracking and Data Association. Academic Press, London.

Baumberg, A., Hogg, D., 1997. Learning deformable models for tracking the human body. In: Jain, R., Sha, M. (Eds.), Motion Based Recognition. Kluwer, Dordrecht, pp. 39–60.

Blake, A., Isard, M., 1998. Active Contours. Springer, Berlin.

Blake, A., Curwen, R., Zisserman, A., 1993. A framework for spatiotemporal control in the tracking of visual contours. Int. J. Comput. Vis. 11 (2), 127–145.

Blake, A., Isard, M., Reynard, D., 1995. Learning to track the visual motion of contours. Artif. Intell. 78, 179–212.

Cheng, J.-C., Moura, J.M.F., 1999. Capture and representation of human walking in live video sequences. IEEE Trans. Multimed. 1 (2), 144–156.

Cohen, L., Cohen, I., 1993. Finite element methods for active contour models and ballons for 2-D and 3-D images. IEEE Trans. Pattern Anal. Machine Intell. 15 (11), 1131–1147.

Cootes, T., Hill, T., Taylor, C., Haslam, J., 1994. The use of active shape models for locating structures in medical images. Image Vis. Comput. 12 (6), 355–366.

Dempster, A., Laird, M., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM-algorithm. J. Royal Stat. Soc. B (39), 1–38.

Dias, J., 1999. Bayesian contour estimation: A subspace representation approach. In: Hancock, E., Pelillo, M. (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition—EMMCVPR'99. Springer, Berlin, pp. 157–172.

Hogg, D., 1983. Model-based vision: A program to see a walking person. Image Vis. Comput. 1 (1), 5–20.

Huttenlocher, D., Ullman, S., 1990. Recognizing solid objects by alignment with an image. Int. J. Comput. Vis. 5, 195–212.

Isard, M., Blake, A., 1998. A mixed-state condensation tracker with automatic model-switching. In: International Conference on Computer Vision, pp. 107–112.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: Active contour models. Int. J. Comput. Vis. 1, 259–268.

Marques, J.S., Lemos, J.M., 2001. Optimal and suboptimal shape tracking based on switched dynamic models. Image Vis. Comput. 4 (June), 539–550.

Nagel, H., 2000. Image sequence evaluation; 30 years and still going strong. ICPR, 149–158.

Nascimento, J., Marques, J.S., 2000. Robust shape tracking in the presence of cluttered background. In: Proceedings IEEE International Conference on Image Processing, Vancouver, vol. 3, pp. 82–85.

Oliver, N., Berard, F., Pentland, A., 1997. LAFTER: Lips and face real time tracker. In: Proceedings IEEE International Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 123–129.

Tagare, H., 1997. Deformable 2D Template Matching using Orthogonal Curves. Trans. Med. Imaging, 108–117.

Tugnait, J., 1982. Detection and estimation for abruptly changing systems. Automatica 18, 607–615.

Wu, Y., Tian, Q., Huang, T., 2000. Integrating unlabeled images for image retrieval based on relevance feedback. ICPR, 21–24.