Robust Multi-Model Filter for Shape Tracking in the Presence of Outliers [*]

Jacinto Nascimento
pcjan@pop.isr.ist.utl.pt

Jorge S. Marques
jsm@isr.ist.utl.pt

IST/ISR, Torre Norte, Av. Rovisco Pais, 1049-001, Lisbon Portugal

Abstract

This paper addresses the problem of tracking objects with complex motion dynamics or shape changes. It is assumed that some of the visual features detected in the image (e.g. edge strokes) are outliers i.e., they do not belong to the object boundary. A robust tracking algorithm is proposed which allows to efficiently track an object with complex shape or motion changes in clutter environments. The algorithm relies on the use of multiple models, i.e., a bank of stochastic motion models switched according to a probabilistic mechanism. Robust filtering methods are used to estimate the label of the active model as well as the state trajectory.

1 Introduction

Object tracking in complex video sequences is a difficult operation. Several reasons contribute to explain this difficulty: the diversity of the object shapes and motion regimes; the time varying illumination conditions which may change along the sequence; the changes in the object pose with respect to the camera; the presence of invalid image features produced by other objects or by the background. Some of these difficulties have been recently addressed by using multiple shape and motion models, tailored to different shape configurations and motion regimes [1, 2].

[*] This work was partially supported by FCT under project TMO

Most tracking algorithms assume a constant motion model with additive noise [3]. This assumption often leads to inaccurate estimates of the object to be tracked since it cannot cope with different motion trajectories and accelerations. To represent the observed data, the tracker should accommodate different motion regimes. This task can be accomplished by using model switching techniques incorporated in a tracking framework. This allows to choose the most appropriate dynamic model among several candidates.

A tracking algorithm based on multiple hybrid models was proposed in [1] in the scope of the condensation method. This algorithm propagates the uncertainty of the unknown parameters using non parametric methods: the *a posteriori* distribution of the object shape is characterized by a set of random contours which are recursively updated by the condensation algorithm.

Another tracker based on multiple model which deals with sudden changes in shape or motion was recently proposed in [2]. This algorithm describes the data as a output of a bank of linear filters equiped with a switching mechanism. This method has advantages: it allows an efficient choice of the best model for the object to be tracked based on the object shape and motion parameters; it uses a parametric representation of the unknown variables, by using mixtures of Gaussians whose parameters are updated by a tree of Kalman filters. The parametric techniques allow the tracker to work in high dimensional spaces. Despite these advantages the method has several weaknesses: the performance of the algorithm is hampered by the presence of incorrect boundary points detected in the image (outliers). These outliers have a strong influence on the shape estimates, leading to meaningless tracking results. This paper addresses this difficulty in the context of multiple model tracking algorithms.

This paper presents a robust method to update the shape and motion parameters in a switched framework which is able to deal with outliers. The algorithm is an extension of the S-PDAF tracker (Shape PDAF) recently proposed in [4]. Two concepts play an instrumental role in this approach. First, middle level features (strokes) are used instead of low level ones (edge points) used in most trackers. Second, two labels (valid/invalid) are considered for each stroke. Since the stroke labels are unknown all labeling sequences are used and a probability (confidence degree) is assigned to each labeling sequence (data interpretation). This is performed according to three probabilistic models: a valid data model; an outlier model and a motion and deformation model. Observations far from the object boundary will have low confidence degree and a negligible contribution to the position and shape estimates. In this way, all the strokes contribute to track the object but with different weights. This allows a robust performance of the tracker in the presence of outliers.

2 Multiple Dynamical Models

In order to estimate the object position and deformation, three steps are considered [3]: contour prediction, image measurement and contour update. The first step predicts the position of the object boundary in the next image. The second step computes image features in the vicinity of the predicted contour e.g., by sampling the predicted contour at equally spaced points. The third step uses the image measurements to update the contour estimate. It is assumed that image features (edges points) either belong to the boundary of the object to be tracked or they are produced by the background (outliers). The main difficulty lies in the presence of false alarms or detection failures which produce undesirable effects. One way to deal with this situation is by considering that each feature is either valid or invalid. This approach is not practical since it involves 2^N hypothesis (data interpretations), N being the number of detected features (sometimes hundreds). As suggested in [4] we adopt a different approach to reduce the number of hypothesis. The edge points are linked in M strokes, and each stroke is classified either as valid or invalid edge points. This dramatically reduces the number of hypothesis to 2^M, with $M \ll N$.

Let $x(t) \in \Re^n$ be a vector containing the shape parameters of the object to be tracked (e.g. control points of a spline curve). We assume that the state vector verifies a stochastic difference equation [5]

$$x(t) = A_{k(t-1),k(t)}x(t-1) + w(t) \tag{1}$$

$w(t) \sim \mathcal{N}(0, Q_{k(t-1),k(t)})$ is a white Gaussian noise, $k(t) \in \{1, \ldots, m\}$ is the label of the active model at instant t and m is a number of models (see Fig. 1). Matrices A, Q depend on $k(t)$ and $k(t-1)$. It is assumed that the label sequence $k(t)$ is a first order Markov process with the transition probability

$$T_{rq} = p(k(t) = q \mid k(t-1) = r) \tag{2}$$

where $r, q \in \{1, \ldots, m\}$, and m the number of models.

Switched dynamic models were studied in control theory and aeronautics to deal with abrupt changes in dynamic systems (e.g., see [5], [6]). The available observations are the strokes detected in the image. However, we do not know which strokes belong to the object boundary and should therefore be considered as valid. Since this information is not available a label (valid/invalid) will be assigned to each stroke and all the label sequences will be considered. Each label sequence is denoted as a *data interpretation*. An 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 j-th stroke in the i-th interpretation.

Let $y(t)$ be the vector of all image features detected at instant t and let $y_i(t)$ be the vector of the valid features according in the i-th interpretation, $(y_i(t) \subset y(t))$. It will be assumed that the sensor model for the i-th interpretation is given by

$$y_i(t) = C_i x(t) + \eta(t) \tag{3}$$

where $\eta(t) \sim \mathcal{N}(0, R_i)$ is a white Gaussian noise and C_i is the shape matrix associated to the i-th interpretation.

Fig.2 shows an example in which there are 2^3 interpretations. A possible interpretation is

$I_i = (S_1 = 1, S_2 = 1, S_3 = 0)$. In this case, matrix C_i includes the rows associated with the indexes $\{b^1, ..., e^1, b^2, ..., e^2\}$ of the image features considered as true. However for the interpretation $I_j = (S_1 = 1, S_2 = 0, S_3 = 1)$, C_j contains the rows with the indexes $\{b^1, ..., e^1, b^3, ..., e^3\}$. Thus, 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 have different dimensions.

The state process of switched multiple model is characterize by the transition density $p(x(t), k(t) \mid x(t-1), k(t-1))$, which can be split as follows

$$p(x(t), k(t) \mid x(t-1), k(t-1)) = p\big(x(t) \mid k(t), x(t-1), k(t-1)\big)\; p\big(k(t) \mid x(t), k(t-1)\big) \qquad (4)$$

The first factor depends on the dynamic equation (1) while the second is an element of the transition matrix of the Markov chain $T_{k(t-1),k(t)}$.

3 Density Propagation

The problem to be solved can be formulated as follows: given a set of observations $Y^t = \{y(1), \dots y(t)\}$ which may contain outliers, what is the best estimate of the state and model label $\hat{x}(t)$, $\hat{k}(t)$. This is a nonlinear filtering problem. If the joint probability density function, conditioned on the observations is evaluated $p(x(t), k(t) \mid Y^t)$, estimates of the unknown parameters $(\hat{x}(t), \hat{k}(t))$ can be obtained by using the maximum *a posteriori* (MAP) method

$$(\hat{x}(t), \hat{k}(t)) = \arg \max_{x(t),k(t)} p(x(t), k(t) \mid Y^t) \qquad (5)$$

Using the law of total probabilities, the *a posteriori* density becomes

$$
\begin{aligned}
p(x(t), k(t) \mid Y^t) &= \sum_{K^{t-1}} p(x(t), k(t), K^{t-1} \mid Y^t) \\
&= \sum_{K^{t-1}} p(x(t) \mid K^t, Y^t) p(K^t \mid Y^t) \\
&= \sum_{K^{t-1}} c_{K^t} p(x(t) \mid K^t, Y^t) \qquad (6)
\end{aligned}
$$

where $c_{K^t} = p(K^t \mid Y^t)$ and $K^t = \{k(1), \ldots, k(t)\}$ is the model label sequence up to instant t. Since $p(x(t) \mid K^t, Y^t)$ is a Gaussian density, the joint density $p(x(t), k(t) \mid Y^t)$ defined in (6) is a mixture of Gaussians, each of them being associated to different label sequence K^t.

The computation of the mixture modes depends on the method being used. If all the observations were valid, each $p(x(t) \mid K^t, Y^t)$ (Gaussian component) could be updated by Kalman filtering and this would be the optimal solution [2]. However, when $y(t)$ is contaminated with outliers robust filtering methods must be derived. In fact, assuming that the model sequence K^t is known, the mean and covariance matrix can be computed using the S-PDAF method, recently proposed in [4] and inspired in the work of Bar-Shalom and Fortmann [7] in the context of target tracking. To update the coefficients c_{K^t} a new update law is required leading to (see appendix A)

$$
c_{K^t} = \gamma \; c_{K^{t-1}} \; T_{k(t-1)k(t)} \sum_i {}^k\alpha_i(t) \prod_{j=1}^{M} \prod_{n=b^j}^{e^j} {}^k\mathcal{E}_i^j(s_n, t) \qquad (7)
$$

where γ is the normalization constant; $c_{k(t-1)}$ is the predicted mixture coefficient; $T_{k(t-1)k(t)}$ is an element of the transition matrix of the Markov chain; $\alpha_i(t)$ is the association probability assigned to the data interpretation $I_i(t)$; M is the number of strokes; b^j, e^j are the indexes of the j-th stroke; $\mathcal{E}$ is a normal or uniform distribution, depending if the stroke j is considered as valid/invalid on the interpretation $I_i(t)$.

The Kalman filter is a particular case of S-PDAF (see appendix A) since a single model is used and all the data is considered as valid. Therefore, $\mathcal{E}$ becomes independent of j and i. In this case, equation (7) can be written in this case as

$$c_{K^t} = \gamma \; c_{K^{t-1}} \; T_{k(t-1)k(t)} \prod_{n=1}^{L} {}^{k}\mathcal{E}(s_n, t) \tag{8}$$

The state mean and covariance matrix estimates are updated by S-PDAF, and given by (see [4] for details)

$$\hat{x}_{K^t} = \hat{x}(t \mid t-1) + \sum_{i=1}^{m_t} \alpha_i(t) K_i(t) \nu_i(t) \tag{9}$$

$$\begin{aligned} P_{K^t} &= \left[I - \sum_{i=1}^{m_t} \alpha_i(t) K_i(t) C_i \right] P(t \mid t-1) \\ &+ \sum_{i=0}^{m_t} \alpha_i(t) x_i(t) x_i(t)^T - \hat{x}(t \mid t) \hat{x}(t \mid t)^T \end{aligned} \tag{10}$$

where $K_i(k)$, $\nu_i(k)$ are the Kalman gain and innovation associated to the interpretation $I_i(k)$. The filter defined in (6-10) will be denoted as RMM *Robust Multi Model tracker.*

The computation of (7,9,10) is organized in a tree structure, each branch being characterized by (see Fig. 3), x_{K^t}, P_{K^t} and c_{K^t}. The structure illustrated in Fig. 3 suggests that the number of leaves (Gaussian mixtures) increases as time goes by. Assuming that we have m models, the mixture will have m^t modes at time t. It is crucial to limit the growth, in order to obtain a practical solution. Several strategies can be applied to achieve this goal, e.g., by using mode merging and elimination [8]. In this paper the second method is adopted by discarding the mixture components with small enough coefficients.

Let us now consider the estimation of the unknown variables $x(t)$, $k(t)$. The model label is estimated the MAP method as follows

$$\begin{aligned}\hat{k}(t) &= \arg\max_q P\{k(t)=q \mid Y^t\} \qquad (11)\\ &= \arg\max_q \int p(k(t)=q, x(t) \mid Y^t)dx(t)\\ &= \arg\max_q \sum_{K^t:k(t)=q} c_{K^t}\int p(x(t) \mid K^t, Y^t)dx(t) \qquad (12)\end{aligned}$$

In the case of the state vector, the mean square method was used instead for the sake of simplicity (see appendix B).

$$^q\hat{x}_{K^t} = \gamma \sum_{K^t:k(t)=q} c_{K^t} \sum_i {}^k\alpha_i(t) \ \ {}^kx_i(t \mid t) \qquad (13)$$

The state estimate is a weighted sum of the estimates associated to the tree path which end with a q-leave.

4 Object Tracking

4.1 Contour Shape Representation

To represent a moving object in a given frame t, it is assumed that the object boundary is a transformed version a reference shape plus shape deformation [3]. Let $r(s) : I \to \Re^2$ be a parametric representation of the object boundary. It is assumed that [1]

$$r(s) = \mathcal{T}r_r(s) + d(s) + v(s) \qquad (14)$$

where $\mathcal{T}$ 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 these curves are described by B-splines. It will be assumed that $\mathcal{T}$, d can be expressed in terms of a small number of parameters which are updated by the robust MM filter and v is a white noise process.

[1] other works assume that $r_k = \mathcal{G}_k(r_r + d) + v$, [9, 10]. Both approaches have advantages and disadvantages.

Several transforms can be considered (e.g., translation, Euclidean similarities, affine transform) [3]. The affine transform is a flexible solution, since it allows to represent the motion of planar objects in 3D space. In this case the object boundary is

$$\begin{cases} r_1(s_i) = a_1 r_{r1}(s_i) + a_2 r_{r2}(s_i) + a_3 + d_1(s_i) + v_1(s_i) \\ r_2(s_i) = a_4 r_{r1}(s_i) + a_5 r_{r2}(s_i) + a_6 + d_2(s_i) + v_2(s_i) \end{cases} \tag{15}$$

where $r(s) = (r_1(s), r_2(s))$, $r_r(s) = (r_{r1}(s), r_{r2}(s))$, $a_1, \ldots, a_6$ are the motion parameters at instant t; $v(s) = (v_1(s), v_2(s))$ is the measurement noise curve. Furthermore, it will be assumed that shape deformation is a linear combination of N_c deformation modes i.e.,

$$d(s) = \sum_{k=1}^{N_c} d_k \phi_k(s) \tag{16}$$

where $\phi_k(s)$ are known deformation curves and $d_1, \ldots, d_{N_c}$ are 2D vectors.

Assuming that the object moves during the acquisition process, dynamical equations have to be devised to describe the evolution of shape and motion parameters. Let $x(t)$ denote the vector of unknown motion and shape parameters

$$x = [a_1, \ldots, a_6, d_{x1}, \ldots, d_{xN_c}, d_{y1}, \ldots, d_{yNc}]^T \tag{17}$$

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

$$y = [r_1(s_1), \ldots, r_1(s_L), r_2(s_1), \ldots, r_2(s_L)]^T \tag{18}$$

Equation (15) can be written as follows

$$y(t) = Cx(t) + v(t) \tag{19}$$

where

$$C = \begin{bmatrix} M & O_{L\times 3} & B_{L\times N_c} & O_{L\times N_c} \\ O_{L\times 3} & M & O_{L\times N_c} & B_{L\times N_c} \end{bmatrix} \tag{20}$$

$$M = \begin{bmatrix} r_{r1}(s_1) & r_{r2}(s_1) & 1 \\ r_{r1}(s_2) & r_{r2}(s_2) & 1 \\ \vdots & \vdots & \vdots \\ r_{r1}(s_L) & r_{r2}(s_L) & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \phi_1(s_1) & \dots & \phi_{Nc}(s_1) \\ \phi_1(s_2) & \dots & \phi_{Nc}(s_2) \\ \cdots & \cdots & \cdots \\ \phi_1(s_L) & \dots & \phi_{Nc}(s_L) \end{bmatrix} \tag{21}$$

In (20) B is a $L \times N_c$ B-spline interpolation matrix [11], $O_{L\times 3}$ $O_{L\times N_c}$ are null matrices. Similar expressions can be derived for other type of models.

4.2 Feature Detection

Feature detection is performed by searching along the normal direction at specific contour points as suggested in [3, 9, 10]. The length of the inspection interval depends on the uncertainty predicted contour and given by

$$\rho(s_i, t) = \delta \sqrt{n(s_i)^T S(s_i, t) n(s_i)} \tag{22}$$

where $n(s_i)$ is the unit normal at s_i, and

$$S(s_i, t) = C(s_i) \left(\sum_{k(t)} c_{k(t)|k(t-1)} P(k(t) \mid k(t-1)) \right) C(s_i)^T + R(s_i) \tag{23}$$

is a covariance matrix of the predicted boundary point $s = s_i$. In (23), $C(s_i)$ is a matrix formed by lines i and $i + L$ of C.

Each feature is detected by comparing the image profile with shifted versions of a profile template T. This procedure is based on the minimization of a cost function given by

$$\mathcal{J}(t_0) = \int_t |p(t) - T(t - t_0)|^2 dt \tag{24}$$

where $p(t)$ is the image profile, along a direction orthogonal to the object boundary where t denotes the distance to the object boundary and $T(t)$ is the template.

4.3 Experimental Results

The robust multi-model was tested to track objects with significant shape changes. An example of lip tracking will be presented. A comparison between the proposed method and Kalman multi-model algorithm is given.

The dynamic models can be defined by the user or learned from the data. The former approach was used to track the lips using two models: model 1 performs a vertical contraction of the object shape estimated in the previous frame; model 2 expands the object contour (see Fig. 4). The first model corresponds to closing the mouth while the second corresponds to opening it. This is accomplished by introducing a vertical scale factor in matrix A.

Fig. 5 shows the results obtained using KMM tracker presented in [2]. The first row shows the predicted contours obtained by both models, the second row shows the estimated contour (line) and the observations (dots). We can see that the transition (frame 5 to 6) is well modeled by exchanging from model 2 to model 1, being the model 1 the closest to the object boundary in the following frame. Fig. 6 shows the results by using RMM tracker, the meaning of the fist and second row are the same as before. We conclude that in the absence of outliers the two methods have a similar behaviour, the choice of the models and the evolution of the predicted contours (fist row) are the same. In the absence of outliers the KMM filter is equivalent to the RMM filter.

Fig. 7 shows the performance of the KMM and RMM trackers using the same input data. The typical difficulties of the Kalman filter are clear, the presence of outliers in frame 8, strongly influences the predicted contours. In this example the Kalman filter still manages to overcome this situation. However, when the number of outliers increases (see frame 13) the KMM filter loses the object contour. The robust tracker described in this paper overcomes this difficulties and exhibits good tracking performance.

A more difficult situation is presented in Fig. 8. In this case, the KMM tracker loses the boundary of the lips and fails to estimate the correct dynamic model. Fig. 8 shows the results given by RMM filter (second row) showing a remarkable robustness with respect to outliers. We have even increased the search area during the feature detection phase, therefore allowing more outliers. The algorithm selects the expansion model in these frames since it is the one which describes best the opening of the mouth. It is shown the robustness of the RMM even in the presence of a large number of clutter features.

Figure 9 show the performance of the RMM algorithm in the presence of sudden shape changes. Three consecutive frames are shown in this figure. The use of multiple models allows to track sudden changes of motion or shape deformation. The expansion model is selected in this example to track the opening of the mouth. It is also displayed the predicted contours obtained by both models showing that the expansion model performs better in these three frames.

5 Conclusions

A new algorithm has been described for tracking of moving objects from video sequences. It allows the use of multiple dynamic models, modeled by a bank of stochastic difference equations. Furthermore, it is assumed that the visual features detected in the image contain outliers, i.e., invalid features which do not belong to the object boundary. A robust filtering algorithm is proposed which is able to deal with multiple dynamics and invalid observations. This is accomplished by computing the propagation of the *a posteriori* density using Gaussian mixtures. Experimental results presented in the paper show that significant improvements are achieved, comparing to the results obtained by the Kalman MM filter which was recently proposed in [2]. The algorithm was tested in lip tracking operations. It was experimentally observed that the proposed method efficiently copes with the presence of abrupt shape changes and noisy measurements corrupted by outliers. This is clearly seen in some test sequences in which the mouth changes from completely closed to completed open in consecutive images and the robust MM tracker still manages to estimate the lips contours well.

Appendix A

Mixture coefficients for S-PDAF model

$$
\begin{aligned}
c_{K^t} &\triangleq \frac{p(K^t, Y^t)}{p(Y^t)} = \frac{1}{P(Y^t)} \int p\big(y(t) \mid K^t, Y^{t-1}, x(t)\big) p\big(K^t, Y^{t-1}, x(t)\big) dx(t) \\
&= \frac{1}{p(Y^t)} \int \sum_i p\big(y(t) \mid I_i(t), K^t, Y^{t-1}, x(t)\big) p\big(I_i(t) \mid k(t), Y^{t-1}, x(t)\big) p\big(K^t, Y^{t-1}, x(t)\big) dx(t) \\
&= \frac{1}{p(Y^t)} \int \sum_i {}^k\alpha_i(t) p\big(y(t) \mid I_i(t), K^t, Y^{t-1}, x(t)\big) p\big(k(t) \mid K^{t-1}, Y^{t-1}, x(t)\big) \\
&\quad p\big(K^{t-1}, Y^{t-1}, x(t)\big) dx(t) \\
&= \frac{1}{p(Y^t)} T_{k(t-1)k(t)} \sum_i {}^k\alpha_i(t) \int p\big(y(t) \mid I_i(t), k(t), Y^{t-1}, x(t)\big) p\big(x(t) \mid K^{t-1}, Y^{t-1}\big) \\
&\quad p\big(K^{t-1}, Y^{t-1}\big) dx(t) \\
&= \gamma T_{k(t-1)k(t)} c_{K^{t-1}} \sum_i {}^k\alpha_i(t) \int p\big(y(t) \mid I_i(t), k(t), Y^{t-1}, x(t)\big) p\big(x(t) \mid K^{t-1}, Y^{t-1}\big)
\end{aligned}
\tag{25}
$$

with $\gamma = \dfrac{p(Y^{t-1})}{p(Y^t)}$.

Since $y(k)$ may contain some gaps along the contour, it depends on the localization of the strokes detected in the image. Thus the probability $p(y(t) \mid I_i(t), k(t), Y^{t-1})$ can be written as

$$p(y(t) \mid I_i(t), k(t), b, e, M, Y^{t-1}) \tag{26}$$

where $b = \{b^1, \dots b^M\}, e = \{e^1, \dots e^M\}$ defines the beginning and the end of the strokes. Assuming that all features are independently generated, i.e.

$$p(y(t) \mid I_i(t), k(t), b, e, M, Y^{t-1}) \quad = \quad \prod_{j=1}^{M} \prod_{n=b^j}^{e^j} p({}^k y^j(s_n, t) \mid I_i^j(t)) \tag{27}$$

where ${}^k y^j(s_n, t)$ is the feature point belonging to the j-th stroke detected in the vicinity of s_n given the model k, since the observations depend on the model being selected. It is assumed that visual features have uniform distributions in the search area if they are classified as unreliable ($I_i^j = 0$) and they are generated with a Gaussian distribution if they are classified as reliable. We define

$$^k\mathcal{E}_i(s_n,t) = p(^ky^j(s_n,t) \mid I_i^j(t)) = \begin{cases} ^kV^j(s_n,t)^{-1} & if\ I_i^j(t) = 0 \\ \rho^{-1}\mathcal{N}\Big(\,^k\nu^j(s_n,t);0,S^j(s_n,t)\Big) & \text{otherwise} \end{cases} \tag{28}$$

$^kV^j(s_n,t)$ is the volume of the search area; ρ is the normalization constant; $^k\nu^j(s_n,t) = {}^ky^j(s_n,t) - C^j(s_n)x(t \mid t-1)$ is the innovation associated to the j-th stroke and $S^j(s_n,t) = C^j(s_n)P(t \mid t-1)C^j(s_n)^T + R^j(s_n)$ is the covariance of the innovation vector where C^j and R^j are the output matrix and noise covariance associated to the j-th stroke. Replacing (28) in (27) into (25)

$$c_{K^t} = \gamma\ c_{K^{t-1}}\ T_{k(t-1)k(t)} \sum_i {}^k\alpha_i(t) \prod_{j=1}^{M} \prod_{n=b^j}^{e^j} {}^k\mathcal{E}_i^j(s_n,t) \tag{29}$$

Mixture coefficients for Kalman model

The Kalman model is a particular case of S-PDAF, since there is a single interpretation for the data. Equation (25) can be written as

$$c_{K^t} = \gamma T_{k(t-1)k(t)} c_{K^{t-1}} \int p\big(y(t) \mid k(t), Y^{t-1},\, x(t)\big) p\big(x(t) \mid K^{t-1}, Y^{t-1}\big) \tag{30}$$

Assuming independence of the L features along the contour we can write

$$c_{K^t} = \gamma\ c_{K^{t-1}}\ T_{k(t-1)k(t)} \prod_{n=1}^{L} {}^k\mathcal{E}(s_n,t) \tag{31}$$

$\mathcal{E}(s_n,t)$ is similar to (28) and defined as

$$^k\mathcal{E}(s_n,t) = \begin{cases} ^kV(s_n,t)^{-1} & if \text{ no features detected} \\ \rho^{-1}\mathcal{N}\Big(\,^k\nu(s_n,t);0,S(s_n,t)\Big) & \text{otherwise} \end{cases} \tag{32}$$

${}^{k}\nu(s_n,t)$, $S(s_n,t)$ have the same meaning as before, however the superscript j and subscript i are suppressed since we do not have interpretations of strokes.

Appendix B

State update assuming the model $k(t) = q$

$$\begin{aligned}
{}^{q}\hat{x}_{K^t} &\triangleq E\{x(t) \mid Y^t, k(t) = q\} \\
&= \int x(t) p(x(t) \mid Y^t, k(t) = q)\, dx(t) \\
&= \int \frac{x(t)p(x(t), k(t) = q \mid Y^t)}{p(k(t) = q)}\, dx(t) \\
&= \frac{1}{p(k(t)=q)} \int x(t) \sum_{K^{t-1}} p(x(t), k(t) = q, K^{t-1} \mid Y^t) dx(t) \qquad (33) \\
&= \frac{1}{p(k(t)=q)} \int x(t) \sum_{K^t : k(t)=q} c_{K^t} \sum_i p({}^{k}x(t),\ {}^{k}I_i(t) \mid Y^t) dx(t) \\
&= \gamma \sum_{K^t : k(t)=q} c_{K^t} \sum_i \int x(t) p({}^{k}x(t) \mid\ {}^{k}I_i(t), Y^t) p({}^{k}I_i(t) \mid Y^t) dx(t) \qquad (34) \\
&= \gamma \sum_{K^t : k(t)=q} c_{K^t} \sum_i {}^{k}\alpha_i(t) \int x(t) p({}^{k}x(t) \mid\ {}^{k}I_i(t), Y^t)\, dx(t) \qquad (35)
\end{aligned}$$

where ${}^{k}\alpha_i(t) \triangleq p({}^{k}I_i(t) \mid Y^t)$ is the *a posteriori* association probability of the i-th interpretation assigned to the model k. Since

$$ {}^{k}x_i(t \mid t) = E\{x(k) \mid\ {}^{k}I_i(t), Y^t\} \qquad (36)$$

Replacing (36) in (35)

$$ {}^{q}\hat{x}_{K^t} = \gamma \sum_{K^t : k(t)=q} c_{K^t} \sum_i {}^{k}\alpha_i(t)\ \ {}^{k}x_i(t \mid t) \qquad (37)$$

References

[1] M. Isard and A. Blake, "A Mixed-State Condensation Tracker with Automatic Model-Switching", in *Int. Conference on Computer Vision*, pp. 107-112, 1998.

[2] J. S. Marques, J. M. Lemos, "Optimal and Suboptimal Shape Tracking Based on Switched Dynamic Models". Image and Vision Computing, accepted for publication, 2001.

[3] A. Blake and M. Isard, "Active Contours". Springer, 1998.

[4] J. Nascimento, J. S. Marques, "Robust Shape Tracking in the Presence of Cluttered Background", in *Proc. IEEE Int. Conf. on Image Processing* vol. 3, pp. 82-85, Vancouver, 2000.

[5] J. Tugnait, "Detection and Estimation for Abruptly Changing Systems", in *Automatica*, vol. 18, pp. 607-615, 1982.

[6] C. Chang, M. Athans, "State Estimation for Discrete Systems with Switching Parameters", in IEEE Trans. Aerospace Electr. Syst. 14 (1978) 418-425.

[7] Bar-Shalom,T. Fortmann, "Tracking and Data Association" Academic Press, 1988.

[8] J. S. Marques, J. M. Lemos, "Shape tracking Based on Switched Dynamical Models", in *Proc. IEEE Int. Conf. on Image Processing*, pp. 954-958, Kobe, 1999.

[9] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active Shape Models - Their training and Application", in *Computer Vision and Image Understanding* , 61(1): 38-59, 1995.

[10] A. Baumberg and D. Hogg, "Learning Deformable Models for Tracking the Human Body", in *Motion Based Recognition*, R. Jain, M. Sha Ed., pp. 39-60. Kluwer, 1997.

[11] A. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, New Jersey, 1989.

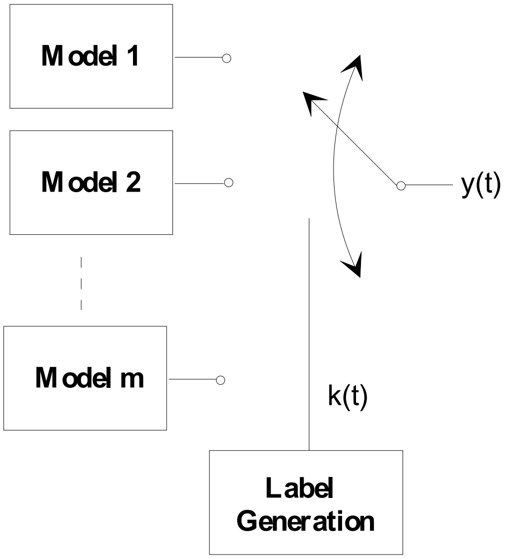

Figure 1: Bank of switched models.

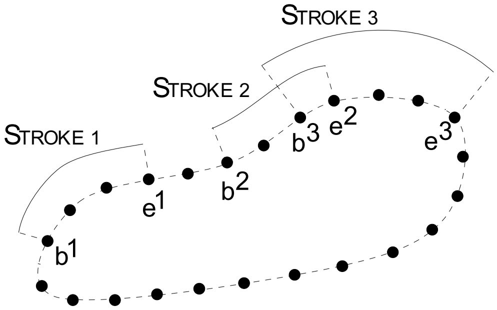

Figure 2: Predicted shape and image strokes.

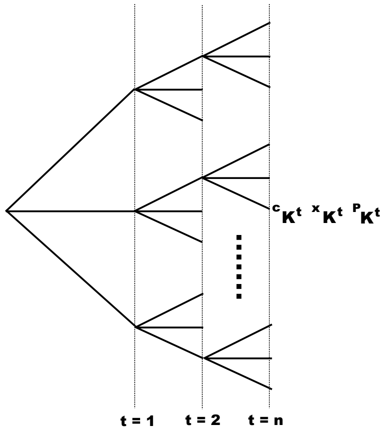

Figure 3: Tree structure of RMM tracker ($m = 3$).

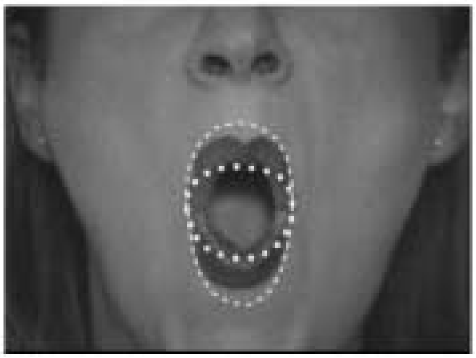

Figure 4: Multi-model initialization. Model 1 (dots); Model 2 (dashed line).

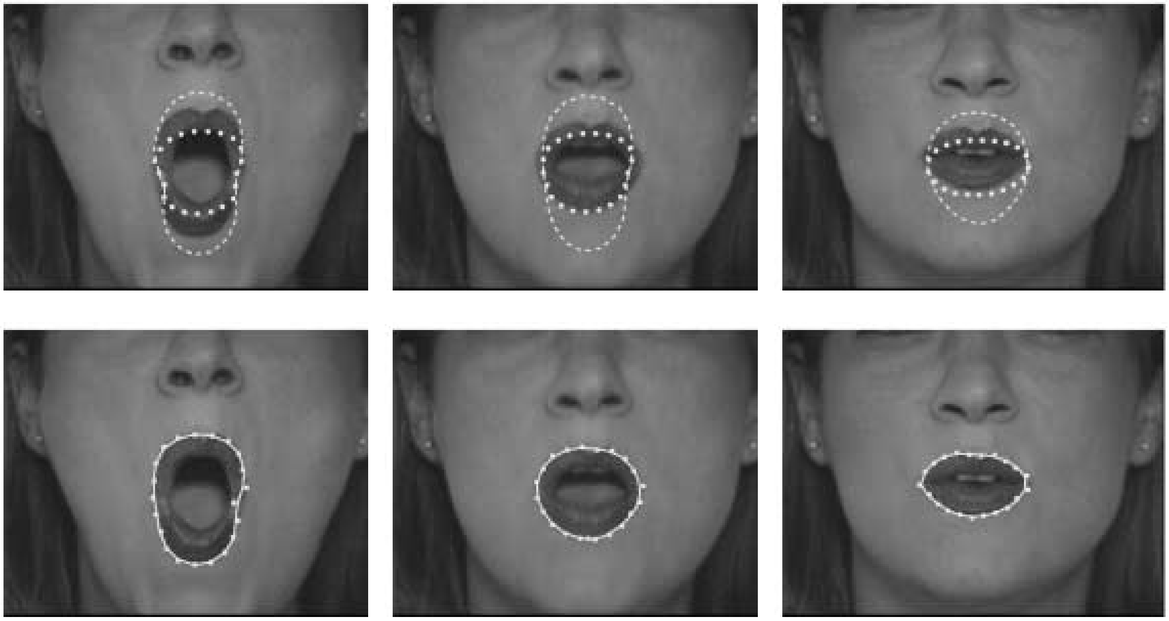

Figure 5: Lip tracking with KMM tracker. First line: predicted contours, second line: estimated contours. Frames 5, 6, 7. Active model: 2 1 1.

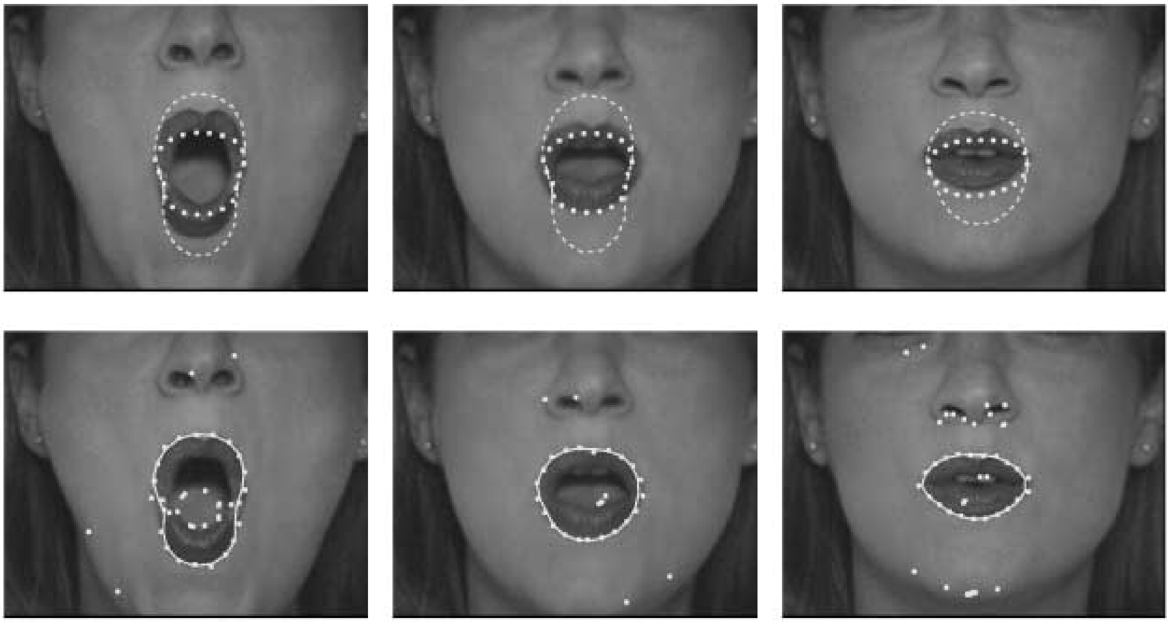

Figure 6: Lip tracking with RMM tracker. First line: predicted contours, second line: estimated contours. Frames 5, 6, 7. Active model: 2 1 1.

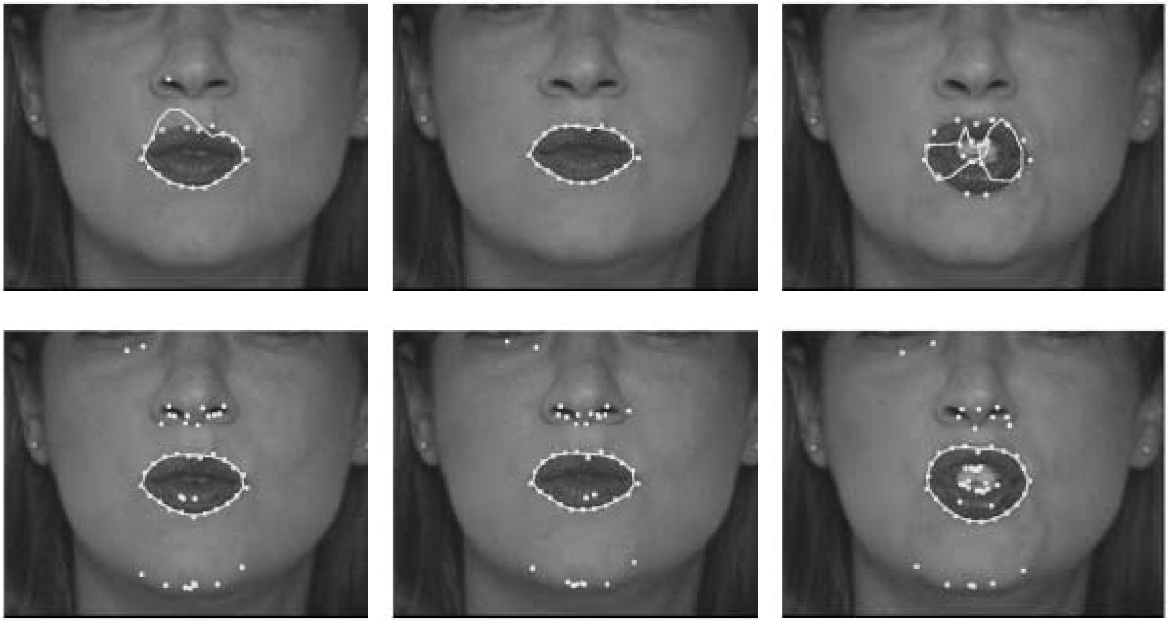

Figure 7: Lip tracking with KMM tracker (first row, active model: 2 1 1) and RMM tracker (second row, active model: 2 1 2), (frames 8, 9, 13).

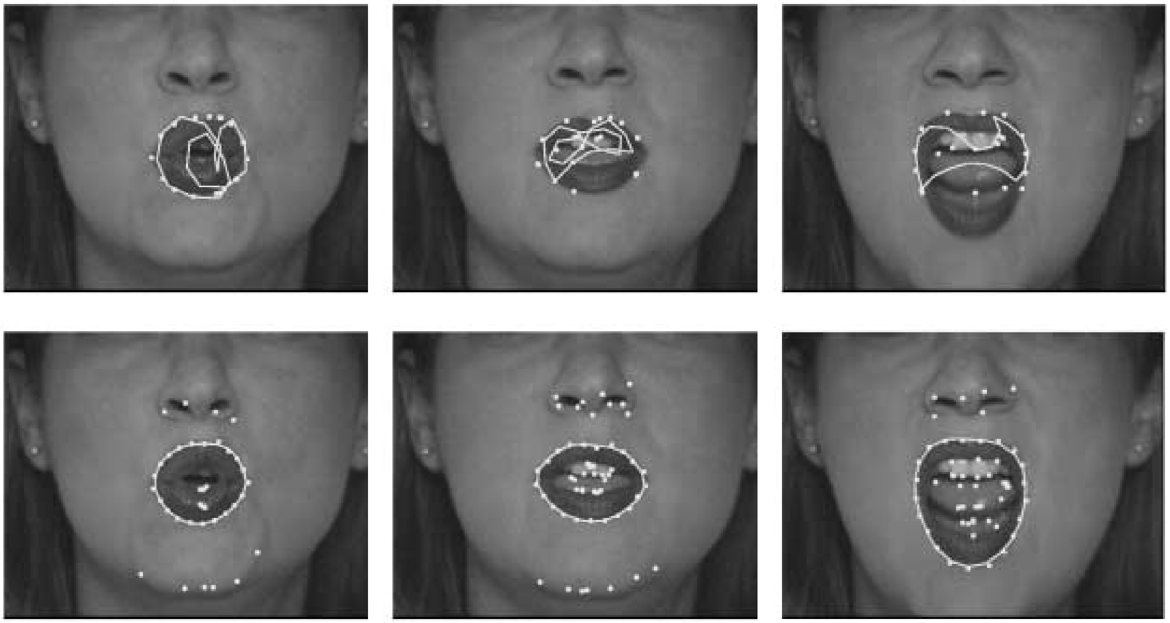

Figure 8: Lip tracking with KMM tracker (first row, active model: 2 1 1) and RMM tracker (second row, active model: 2 2 2), (frames 16, 27, 46).

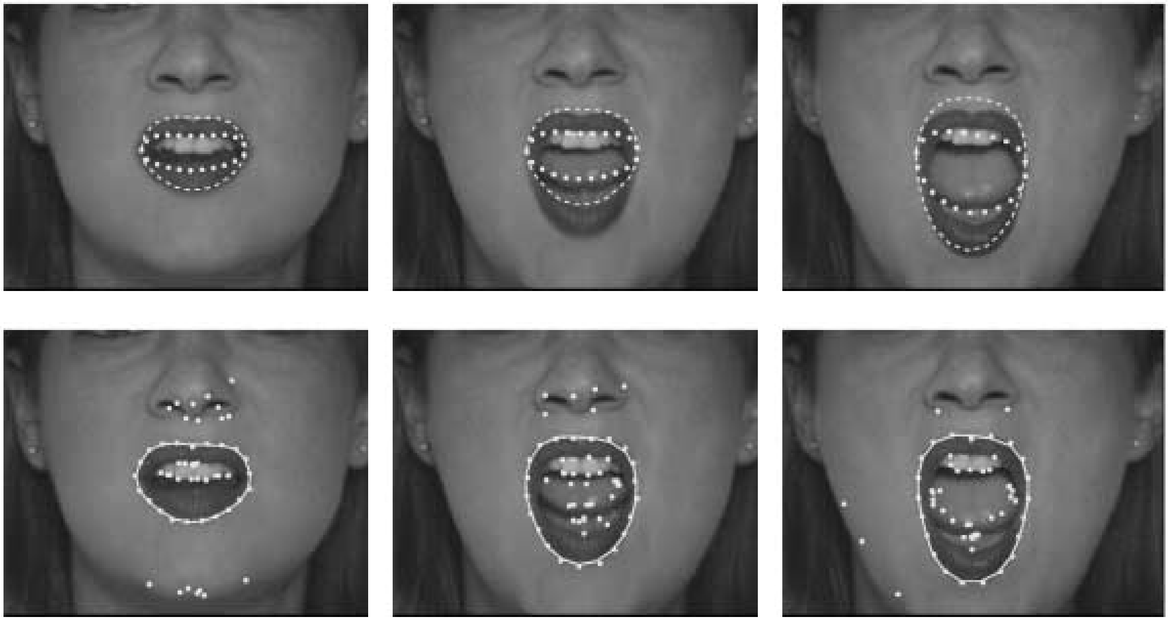

Figure 9: Lip tracking with RMM: predicted contours (first row) and estimated contours (second row), (frames 45, 46, 47), active model: 2 2 2).