Robust Deformable Models for 2D and 3D Shape Estimation

Jorge S. Marques, Jacinto C. Nascimento and Carlos Santiago

Abstract Deformable models are useful tools to extract shape information from images and video sequences. However, the model has to be initialized in the vicinity of the object boundary, in order to foster convergence towards the desired features. This chapter describes four methods which alleviate this restriction. Despite their differences, they share three common features: (i) they use middle level features (edge segments) instead of low level ones; (ii) they explicitly assume that the measured features contain outliers and assign confidence degrees to the detected features and (iii) they adopt robust model updates, taking the confidence degrees into account. These four methods are reviewed and their performance is illustrated with selected examples.

1 Introduction

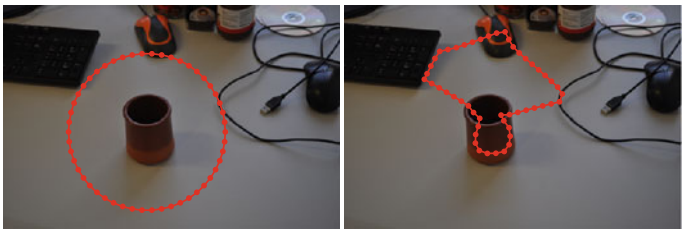
The automatic estimation of object boundaries in images and video sequences is a key operation in many computer vision systems. However, several factors make this task difficult namely, the high variety of objects shape and appearance, partial occlusion and the presence of clutter in the background image. Deformable models are popular techniques for dealing with this kind of problems [3, 11, 12]. They assume that object boundaries can be approximated by elastic curves or surfaces. The model is

J. S. Marques (✉) · J. C. Nascimento · C. Santiago
Institute for Systems and Robotics, Instituto Superior Tecnico,
av. Rovisco Pais, 1049-001 Lisboa, Portugal
e-mail: jsm@isr.ist.utl.pt; jsr@isr.ist.utl.pt

J. C. Nascimento
e-mail: jan@isr.ist.utl.pt

C. Santiago
e-mail: carlos.santiago@ist.utl.pt

M. González Hidalgo et al. (eds.), *Deformation Models*, Lecture Notes in Computational Vision and Biomechanics 7, DOI: 10.1007/978-94-007-5446-1_7,



(a) **(b)**

Fig. 1 Convergence difficulties in the estimation of the mug boundary with poor initialization: **a** initial contour and **b** final contour estimate obtained with the Snake algorithm

initialized close to the object boundary and it is automatically attracted towards the boundary by features detected in the image.

Despite the simplicity of this approach, model adaptation is not an easy task due to the following difficulties: (i) boundary cues (e.g., edge lines) are often subtle and incomplete, (ii) they are mixed with cues produced by other objects and by the cluttered background and (iii) the object is often partially occluded. Therefore, image noise, occlusion and cluttered background degrade the performance of deformable models. Figure 1 shows typical difficulties found in practice.

Classic deformable models deal with such difficulties by initializing the deformable contour in the vicinity the object boundary and by reducing its attraction range, to decrease the influence of outliers. This is often considered as a myopic behavior, since in such approaches, the model only "sees" the features located in a small neighborhood of the deformable contour [10].

Several attempts have been made to improve classic methods by using more informative features such as color [15], motion [6, 19] or gradient vector field [23]. Other approaches use prior information about the object shape by characterizing their statistics e.g., mean shape and main deformation modes [5]. The main problem remains unsolved, however: these algorithms are unable to separate valid features from outliers. They require a careful contour initialization, close to the object boundary, and myopic estimation algorithms.

The estimation of models in the presence of invalid data (outliers) is an old challenge in Computer Vision and has been addressed using robust estimation methods [14]. A popular approach (RANSAC) is based on the generation of multiple hypotheses, each of them based on different subset of data, considered as valid [7]. The model is estimated from the subset of data considered as valid and applied to all the data. The model which achieves the best fit is selected. This approach performs well if the number of parameters to estimate is small (<10) but it cannot be extended to flexible models that depend on tens or hundreds of parameters.

This chapter reviews four methods developed by the authors (Adaptive Snakes, S-PDAF, MMDAT, 3DS-PDAF) which allow a robust estimation of deformable curves and surfaces from image data, corrupted by outliers [8, 13, 16–18]. Instead of assuming that all the features are valid, we assume that some of them are outliers and

Table 1 Structure of robust adaptation methods

initialization: curve or surface initialization in the vicinity of the object boundary;
cycle
feature extraction
weights calculation
model update
end

assign a confidence degree to each detected feature. Model update is accomplished by taking the confidence degrees into account. The second main idea concerns feature extraction: the proposed methods extract middle level features (curve segments or surface patches) from the image instead of low level ones (e.g., edge points). Both strategies contribute to improve the robustness of the algorithms. Table 1 summarizes the main steps of the robust model update algorithms. The methods discussed in this paper fit into this structure with one exception. In Adaptive Snakes, feature extraction is done once for all at the beginning and it is not repeated in each iteration.

The paper is organized as follows. Section 2 reviews two classic algorithms which fail in the presence of outliers. Section 3 describes Adaptive Snakes for the estimation of static objects. Section 4 describes the S-PDAF method for object estimation and tracking. Sections 5 and 6 extend S-PDAF to cope with multiple dynamics and 3D images. Section 7 concludes the paper.

2 Two Classic Algorithms

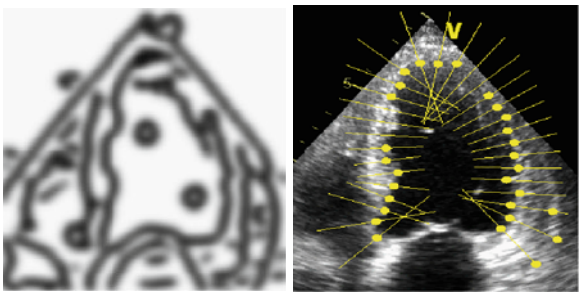
The Snake algorithm proposed in [11] estimates a deformable curve, called snake, $x(s) : [0, 1] \rightarrow \mathbb{R}^2$ by minimizing an energy functional $E(x) = E_{int}(x) + E_{img}(x)$, where $E_{int}(x)$ is an internal energy which prevents the snake from taking unusual configurations, and $E_{img}(x)$ is an image energy that attracts the model towards the object boundary. The image energy is defined by

$$E_{img}(x) = \int P(x(s))ds \tag{1}$$

where $P(x) : \mathbb{R}^2 \rightarrow \mathbb{R}$ stands for an *image potential* function, chosen in such a way that some of its valleys are located on the object boundary.

Several image potential functions have been proposed e.g., $P(x) = \|\nabla I(x)\|$ where $I(x)$ denotes the image intensity at point x and $\nabla I(x)$ is the image gradient [11]. Another alternative is the edge-based potential [4]

$$P(x) = -\sum_k G(x - e_k) \tag{2}$$



(a) **(b)**

Fig. 2 *Bottom-up* and *top-down* image analysis: **a** image potential and **b** model guided feature extraction

where $G(x)$ is a Gaussian kernel and e_k denotes the kth edge point detected in the image. This potential function attracts the contour towards regions with high concentration of edge points. An example of an edge-based potential is shown Fig. 2a. The Snake algorithm equipped with this potential function performs well, if the edge points are located at the object contour and there are no other edges in the vicinity of the object. However, it performs poorly otherwise.

The Snake algorithm is slow and tends to get trapped in textured regions. In order to speed up convergence, top down algorithms were proposed [3]. Instead of computing a potential function, they extract feature points in the vicinity of the deformable curve i.e., the model is used to extract information from the image in a top-down way. Typically, the deformable contour is sampled at equally spaced points. Features points are detected by searching intensity transitions along lines orthogonal to the contour (see Fig. 2b). In tracking applications, the contour model and the set of measurements are characterized by vectors $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, respectively, whose evolution is described by a linear dynamical system

$$x(t) = Ax(t-1) + w(t) \tag{3}$$

$$y(t) = Cx(t) + v(t) \tag{4}$$

where t denotes the frame number, $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{m \times n}$ characterize the dynamical behavior of the curve and the relationship between the curve parameters and the observations; $w(t) \sim N(0, Q)$, $v(t) \sim N(0, R)$ are uncorrelated random perturbations, with zero mean and covariance matrices Q, R. The estimation of the curve parameters from the observed measurements $y(t)$ can be performed by Kalman filtering and this method is called Kalman Snakes [21]. This approach is fast and well suited to real-time tracking applications. However the robustness difficulties remain unsolved: after a few frames, the deformable contour gets stuck in the outlier features.

It should be mentioning that several models are used to represent deformable curves. In some works, the deformable curve is represented by a sequence of 2D points (curve samples), $(x_1, \ldots, x_N)$, or by a spline curve, $x(s)$, whose shape is

modified by a set of control points. Some works, however assume that the observed shape, $x(s)$, is obtained from a known reference shape, $x_r(s)$, modified by a global transformation T_θ (e.g., affine transform) and local deformation [3]. Therefore,

$$x(s) = T_\theta[x_r(s)] + x_d(s)\,, \tag{5}$$

where θ is the set of global parameters and $x_d(s)$ is the deformation spline. Both the global parameters, θ, and the deformation parameters have to be estimated from the input images.

3 Adaptive Snakes

As discussed in Sect. 2, the classic Snake algorithm gets easily trapped in wrong image features i.e., features associated to other objects or to the textured background, which are considered as *outliers*. Ideally, we would like to assign a binary label (valid/invalid) to each feature (edge point) and let the model be attracted by valid features only. This strategy inspired the Adaptive Snake algorithm proposed in [18]. It must be stressed, however, that some difficulties must be addressed as the feature labels are unknown.

The Adaptive Snakes algorithm follows a bottom-up approach to shape estimation. It starts by detecting edge points in the image using a standard edge detection algorithm. The edge points are then organized in connected line segments, called strokes in this paper. The set of strokes is denoted by $E = (e^1, \ldots, e^M)$ where the jth stroke $e^j = (e^j_1, \ldots, e^j_{L^j})$ is a sequence of edge points $e^j_p \in \mathbb{R}^2$ and L^j is the stroke length. Let $K = (k^1, \ldots, k^M)$ be a sequence of binary labels associated to the strokes in which $k^j \in \{0, 1\}$ is the label associated with the jth stroke e^j.

The ideal potential function is the sum of valid and invalid stroke potentials. We will assume that the potential function associated to valid strokes is given by (2) while the potential associated to the outliers is constant C. Therefore,

$$P(x, E, K) = -\sum_j \left(k^j \sum_p G(x - e^j_p) + (1 - k^j) L^j C \right). \tag{6}$$

Unfortunately, we do not know the labels k^j and cannot use this ideal potential function to directly estimate the deformable contour. To overcome this difficulty, the problem will be stated in a probabilistic framework.

Let $X = (x_1, \ldots, x_N)$, $x_i \in \mathbb{R}^2$, be N samples of the deformable contour. We will assume that the strokes and labels, E, K, given the contour samples X, follow a Gibbs distribution,

$$p(E, K|X) = \alpha \prod_{i=1}^{N} e^{-P(x_i, E, K)}. \tag{7}$$

We want to estimate X given the observed strokes E. This is a statistical inference problem which can be tackled by the Maximum Likelihood method. However, the binary labels are not observed and make this problem difficult since we must resort to $p(K \mid X)$ and a marginalization of the joint distribution $p(E, K \mid X)$ is required.

This difficulty can be circumvented by using the Expectation-Maximization (EM) method. The expressions for the E and M steps are derived in [18] and will not be repeated here. The E-step computes the confidence degrees associated to each stroke

$$w^j = Pr(k^j = 1 | e^j, \hat{X}) \tag{8}$$

where $\hat{X}$ denotes the most recent deformable model estimate and the M-step minimizes the snake energy

$$E(X) = E_{int}(X) + \sum_{i=1}^{N} P_a(x_i) \tag{9}$$

where

$$P_a(x) = -\sum_j w^j \sum_p G(e_p^j - x), \tag{10}$$

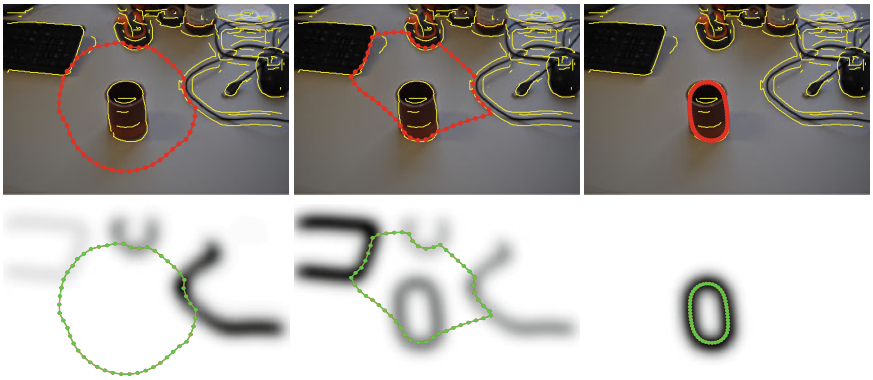
changes in every iteration of the EM method and is called an *adaptive potential function*. The EM method leads to a recursive estimation algorithm similar to the Classic Snakes. However, the stroke potentials are multiplied by their confidence degrees, updated in each new iteration. All the strokes contribute to the estimation of the object contour. Although, the strokes with low confidence degrees have a negligible influence on the final contour estimates.

Figures 3, 4, illustrate the performance of Classic and Adaptive Snakes in the estimation of the mug boundary, using a deformable model represented by a sequence of 2D points. While in Classic Snakes, the potential function remains invariant during the convergence process and the contour is attracted towards outlier valleys, in Adaptive Snakes the potential changes as the contour deforms. After 20 iterations (see Fig. 4c), only the potential valleys associated to the mug receive a high confidence degree and the model converges towards the mug boundary.

Adaptive Snakes have been successfully used in several problems, e.g., in the segmentation of skin lesions displayed in dermoscopic images. A comparison among several segmentation algorithms was carried out in [20], using a database of dermoscopic images annotated by a medical expert. The Adaptive Snakes were selected as the best method (ex-aequo) in this study. Figure 5 shows an example of the Adaptive Snakes performance in the segmentation of a melanocytic skin lesion.

(a) (b) (c)

Fig. 3 Classic Snakes performance: **a** edge points and initial contour, **b** edge potential and **c** final snake configuration



(a) (b) (c)

Fig. 4 Convergence of Adaptive Snakes; Snake configuration at iterations 1, 8 and 20 (*1st row*) and corresponding adaptive potential functions (*2nd row*)

4 Robust Tracking with S-PDAF

As mentioned before (Sect. 2), Kalman Snakes extract image features in the vicinity of the current contour estimate using directional search [3]. This strategy is very fast and tailored to object tracking in video sequences. However, some of the features detected by directional search are outliers and jeopardize the contour estimates after a few frames. Figure 2b illustrates feature extraction guided by the model estimate.

The method described in this section tries to alleviate this difficulty by using two complementary strategies. First, it replaces feature points by strokes i.e., sequences of points detected by directional search at consecutive samples of the deformable contour and such that their distances to the model boundary change smoothly (see Fig. 6a). Strokes are more informative and reliable than isolated points. Second, we will explicitly assume that some of the strokes are outliers (we do not know which) and should not be considered by the tracker.

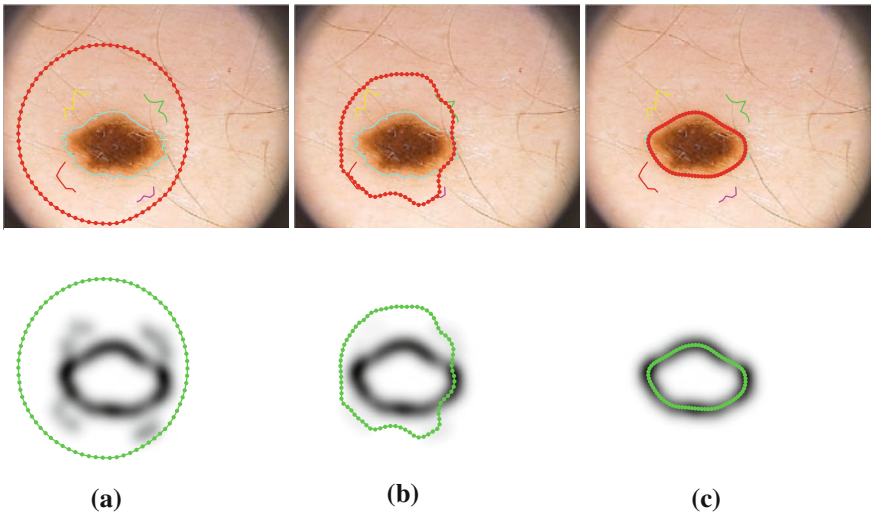


Fig. 5 Convergence of Adaptive Snakes in skin lesion segmentation: image features and evolution of the elastic curve in iterations 1, 8 and 20 (*1st line*); evolution of the adaptive potential (*2nd line*)

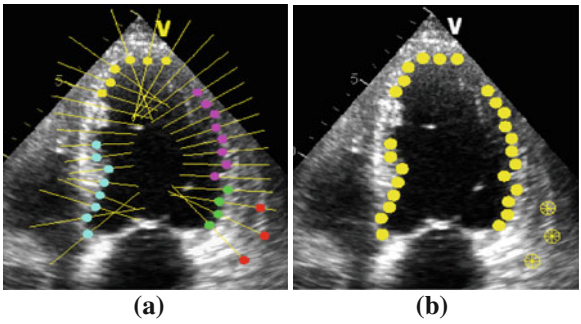


Fig. 6 **a** Strokes detected in an ultrasound image of the left ventricle (each color represents a different stroke). In this example 32 interpretations are possible. One of these interpretations is illustrated in **b** where strokes with the *filled dot lines* are considered as valid and the *dashed* stroke as invalid

We will assume that there is a binary label associated to each stroke (valid/invalid) and consider all the admissible label sequences. Each binary sequence will be denoted as a *data interpretation*. Thus, the ith data interpretation is defined by $I_i = (I_i^1, \ldots, I_i^M)$ where $I_i^j \in \{0, 1\}$ is the label of the jth stroke in the ith interpretation.

If all the observations were valid, we could use the dynamical model (3, 4) to describe shape and observations evolution. However, this model does not hold in the case of invalid observations. Specifically, if the ith interpretation, I_i, is true, only a subset of the observations, y_i, is valid and all the other measurements are considered as outliers. Therefore, the model associated to the ith interpretation is

$$x(t) = Ax(t-1) + w(t) \tag{11}$$

$$y_i(t) = C_i x(t) + v_i(t)\,, \tag{12}$$

where matrix C_i is obtained from matrix C by removing the rows associated to the invalid observations. This makes the inference problem more difficult since we cannot apply Kalman filtering based on the correct interpretation of the data, as we do not know which interpretation is true. There are multiple observation models and we do not know which is the correct one at time t.

The *a posteriori* distribution of the state vector $x(t)$, given all the observations until the time instant t, $Y^t = \{y(1), \ldots, y(t)\}$, is a mixture of Gaussians. Unfortunately the number of Gaussians exponentially increases with t and exact inference is unfeasible.

A similar problem occurs in the tracking of moving point targets from Radar measurements. Although the radar system detects multiple echoes at each instant of time, only one of them, at most, corresponds to the target to be tracked. All the others echoes are produced by clutter or by other targets. In this case, exact inference is also unfeasible, since the number of hypotheses exponentially grows with the operation time and the number of detected echoes in each frame. This difficulty has been addressed by using approximate methods. Remarkable results have been obtained by the probabilistic data association filter (PDAF) [1, 2], used to compute the *a posteriori* distribution of the state vector $x(t)$, given a sequence of observations, most of them being outliers. This filter approximates the distribution of the state vector $x(t)$, given past observations $Y^{t-1} = (y(1), \ldots, y(t-1))$, by a normal distribution

$$p(x(t)|Y^{t-1}) = N(x(t); \hat{x}(t|t-1), P(t|t-1)) \tag{13}$$

where $\hat{x}(t|t-1)$, $P(t|t-1)$ are the mean vector and covariance matrix of $x(t)$ given past observations until time $t-1$, Y^{t-1} [1].

The PDAF can be modified to cope with multiple observation models in the context of robust shape tracking and the presence of multiple sensor models [17]. Using (13), it can be shown that the *a posteriori* distribution of $x(t)$, given Y^t, with multiple observation models is still Gaussian, i.e., $p(x(t)\,|Y^t) = N(x(t); \hat{x}(t\,|t), P(t\,|t))$, with mean vector being updated by

$$\hat{x}(t|t) = \sum_{i=1}^{M} \alpha_i(t)\hat{x}_i(t|t) \tag{14}$$

where $\hat{x}_i(t|t)$ is the mean vector obtained by Kalman filtering tailored to the ith interpretation of data, and $\alpha_i = p(I_i(t)\,|Y^t)$ is the corresponding data association probability. Similarly, the covariance matrix is updated by combining the Kalman filters associated to each interpretation of the data

$$P(t|t) = \left[I - \sum_{i=1}^{M} \alpha_i(t) K_i(t) C_i \right] P(t|t-1) + \sum_{i=1}^{M} \alpha_i(t) \hat{x}_i(t|t) \hat{x}_i(t|t)^T - \hat{x}(t|t)\hat{x}(t|t)^T \tag{15}$$

where K_i is the Kalman gain associated to the ith interpretation. These equations show that the contour estimate considers all data interpretations, each of them weighted by the corresponding association probability.

The association probabilities play an important role in this algorithm and they are computed using the Bayes law,

$$\alpha_i(t) = \beta p(y(t)|I_i(t), Y^{t-1}) p(I_i(t)) , \tag{16}$$

where β is a normalization constant. Equation (16) requires the distribution of the data associated to the ith interpretation, given previous observations, and the prior distribution for the interpretations. The first factor is computed as follows. Assuming that the observed features $y(t)$ are conditionally independent, we obtain

$$p(y(t), |I_i(t), Y^{t-1}) = \prod_{j=1}^{M} \prod_{p=1}^{L^j} p(y_p^j | I_i(t), Y^{t-1}) . \tag{17}$$

where $p(y_p^j | I_i(t), Y^{t-1})$ is a Gaussian density function, if the observation is valid, and a uniform density function, if the observation is invalid [17]. The prior, $p(I_i(t))$, is defined according to two main assumptions: (i) interpretations with long valid strokes are more probable than interpretations with small valid strokes, and (ii) if two valid strokes overlap, the interpretation has zero probability.

Figure 7 shows the operation of the S-PDAF tracker in the analysis of facial expressions (lips and eyebrows). A large number of outliers are detected since the search window along each direction is large. The tracker successfully manages to discard the outliers and to correctly update the deformable contours in this example.

5 Dealing with Multiple Motion Models

Sometimes, the object shape deforms according to different motion regimes (e.g., vehicles moving in streets, heart deformation in systole and diastole phases) and a single dynamical system (3, 4) is not enough to the describe the evolution of an object shape in the video sequence. In this case we should resort to multiple motion models

$$x(t) = A_{k(t)} x(t-1) + w(t) \tag{18}$$

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

where $k(t) \in \{1, \ldots, O\}$ is the label of the active model at time t, matrices A_1, $\ldots$, $A_O \in \mathbb{R}^{n \times n}$ define O dynamical behaviors of the state vector $x(t) \in \mathbb{R}^n$

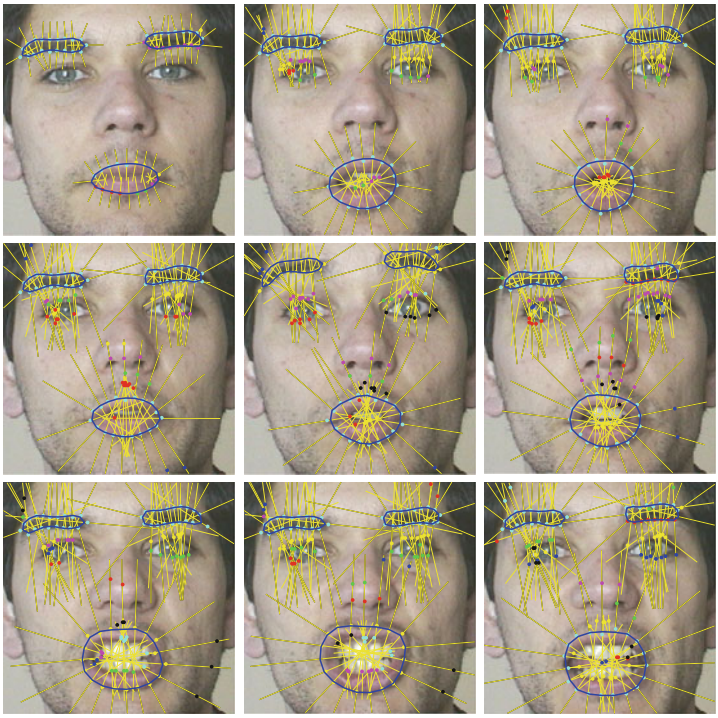


Fig. 7 Robust model tracking with S-PDAF. The tracker manages to discard the influence of outliers

and matrix $C \in \mathbb{R}^{m \times n}$ characterizes the relationship between the state vector and the observations $y(t) \in \mathbb{R}^m$; $w(t), v(t)$ are uncorrelated random sequences with normal distributions, $w(t) \sim N(0, Q_{k(t)}), v(t) \sim N(0, R)$. Furthermore, it is assumed that the label sequence $k(t)$ is a first order Markov process with transition probabilities

$$P(k(t) = j | k(t-1) = i) = T_{ij} \,. \tag{20}$$

Equations (18–20) define a switched dynamical model with a hybrid state $(x(t), k(t))$.

The most probable estimate of the hybrid state given the observations until time t, is obtained by solving the optimization problem

$$(\hat{x}(t), \hat{k}(t)) = \arg \max_{x(t), k(t)} p(x(t), k(t) | Y^t) \,. \tag{21}$$

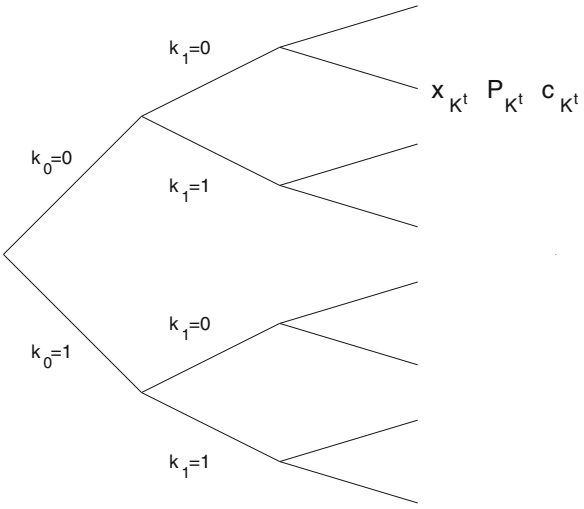


Fig. 8 Multi-model tracking: tree of Kalman filters

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

$$p(x(t), k(t)|Y^t) = \sum_{K^{t-1}} p(x(t), K^t|Y^t) \tag{22}$$

$$= \sum_{K^{t-1}} p(x(t)|K^t, Y^t) p(K^t|Y^t) \tag{23}$$

$$= \sum_{K^{t-1}} c_{K^t} p(x(t)|K^t, Y^t) \tag{24}$$

where $c_{K^t} = p(K^t|Y^t)$ is one mixture coefficient and $p(x(t)|K^t, Y^t)$ is a normal density function whose parameters can be obtained by Kalman filtering. Equation (24) shows that the *a posteriori* distribution of the hybrid state is a mixture of Gaussians each of them associated to a different label sequence K^t. Although, each Gaussian can be computed by Kalman filtering on a tree structure (see Fig. 8), the number of hypothesis, K^t exponentially increases, making a direct computation unfeasible.

This difficulty can be overcome by mode merging and elimination. If the probability c_{K^t}, for a specific label sequence K^t, becomes too small, the associated mode will be eliminated. In addition, if a pair of Gaussians have similar parameters (the similarity being measured by the Kullback-Leibler divergence) the two modes are merged and replaced by a single mode [13]. After performing mode merging and elimination, the estimation of the best shape estimate is determined by maximizing the *a posteriori* distribution, according to (21). This involves finding the maximum value of a multivariate mixture of Gaussians.

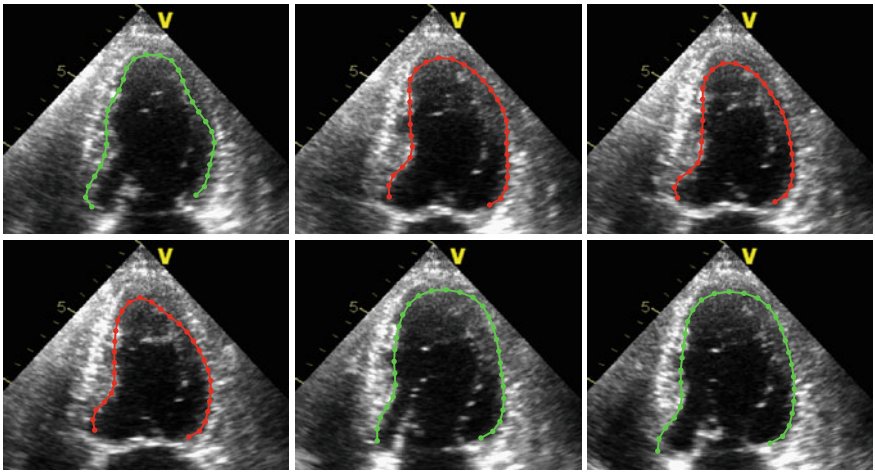


Fig. 9 Tracking of the *left ventricle* using the MMDAT, during systole and diastole phases. The contour color displays the label of the selected model: contraction (*red line*) or expansion (*green line*) models. Notice that whenever the mitral valve is closed (open) the cardiac cycle is in systole (diastole) phase

This algorithm can be used to track moving objects under different motion regimes and will be denoted as Multi-model tracker (MMT). However, the MMT is not robust since the state estimates are strongly affected by outlier measurements. To deal with this problem, we will extend the robust S-PDAF tracker to this multi-model context. We assume that the observations are organized in strokes and some of them may be outliers. A binary label will be associated to each stroke as we did in Sect. 4. The sequence of all binary labels, $I_i = (I_i^1, \ldots, I_i^M)$, is called a data interpretation.

The dynamical model associated to the model label $k(t)$ and data interpretation I_i is given by

$$x(t) = A_{k(t)}x(t-1) + w(t) \tag{25}$$

$$y_i(t) = C_i x(t) + v_i(t) \tag{26}$$

where $y_i(t)$ denotes the vector of valid observations according to interpretation I_i. The true data interpretation is unknown, however, and the estimation of the hybrid state based on this model can be done in a similar way to the one used in the MMT. However, the tree of Kalman filters is replaced by a tree of robust S-PDAF filters. This multi-model tracker with robust state estimate will be denoted Multi-model Data Association Tracker (MMDAT).

Figure 9 shows the application of the MMDAT in a sequence of ultrasound images of the heart. Two dynamical models are used to represent the evolution of the left ventricle during systole (contraction) and diastole (expansion) phases. Figure 9 shows the contour estimates obtained with this algorithm and the contour color represents the label of the active model, $k(t)$, which was automatically selected by the algorithm in each frame (see (21)).

6 From 2D to 3D

This section considers the segmentation of 3D objects in 3D images, $I : [0, 1]^3 \rightarrow \mathbb{R}$ e.g., ultrasound, MRI or PET images. The previous methods can be used in this context, specially the S-PDAF method which combines fast feature extraction, low memory requirements and robust model update. Memory and computational time play important roles in this context since the models typically depend on hundreds or thousands of parameters.

Four issues must be considered in order to extend the S-PDAF method to 3D shape estimation: (i) the surface model, (ii) model initialization, (iii) feature extraction and (iv) robust model update. Each of these issues is discussed in the sequel.

Surface model: we wish to approximate the object boundary by a deformable surface in 3D space. The choice of the surface model is an important issue since there is a tradeoff between model accuracy and number of free parameters. Popular solutions range from parametric models, such as superquadrics, with a small number of parameters and limited representation capability, to non-parametric models with many hundreds of parameters [22]. We adopted a non-parametric simplex mesh [9] which consist of a set of nodes, associated to 3D points, and edges among neighboring nodes. The model is constructed in such a way that each node has 3 neighbors which define a tangent plane to the surface. The normal vector is therefore available for each node.

Model initialization: this is an important step since we wish to create a simplex mesh close to the object boundary as a starting point for adaptation. We adopt a space carving approach. We discretize the region of interest (data volume) and intersect the region of interest by several inspection planes (e.g., three orthogonal planes as shown in Fig. 10). We then define a 2D contour in each plane using a graphical editor. All the voxels in the data volume are projected onto the inspection plane and those outside the contour are zeroed. At the end, we obtain a binary 3D mask which is then approximated by a simplex mesh. The simplex mesh is initialized with a spherical shape and deformed until it fits the binary mask. This is an easy task for the deformable simplex mesh since we are dealing with a binary object without clutter or noise.

Patch extraction: this operation is done in two steps. First, we detect transitions along straight lines orthogonal to the surface model. This task is repeated for each node and may lead to multiple detections in each direction (see Fig. 11a). Then the nodes are grouped in surface patches. Neighboring nodes receive the same label if their distances to the surface are similar. Patch construction depends on the order by which nodes are visited. Figure 11b shows an example of two patches.

Robust model update: model update is done using the S-PDAF Eqs. (14, 15) but there are two important differences. First the dimension of the state space is much higher. Instead of a few tens of parameters needed to represent a deformable curve, we typically need a few thousands of coordinates to represent the nodes positions. The update iteration is therefore much more demanding and slow. Second, the middle level features (surface patches) also have a large number of observations (sometimes

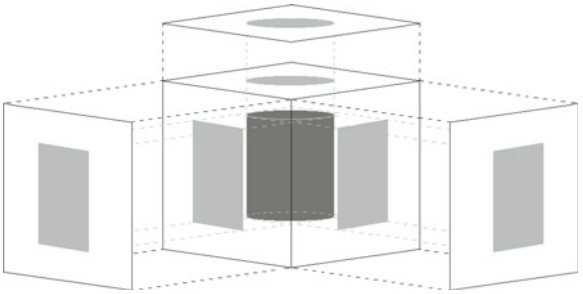


Fig. 10 Model initialization by space carving: image data is observed along three or more planes and the object boundary is approximated by a 2D contour in each plane. All voxels are projected onto the planes and those who project outside the contour are zeroed

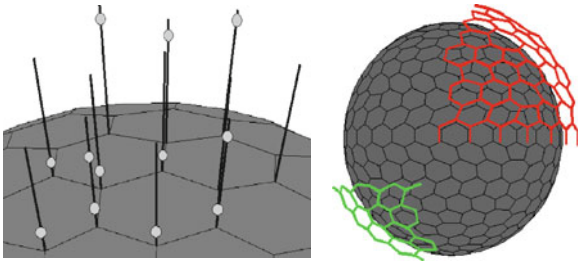


Fig. 11 Feature extraction: **a** directional search; **b** patch creation by region growing

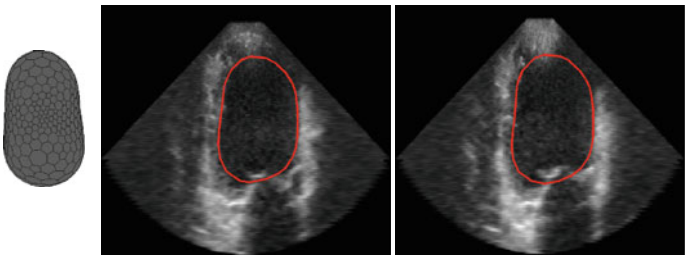


Fig. 12 Segmentation of the *left ventricle* in an ultrasound 3D volume: **a** estimated 3D model using 3DS-PDAF; **b, c** pair of observed images and model cross sections obtained by intersecting the deformable mesh by the inspection planes

hundreds of transitions are detected in the 3D volume of data). It is no longer possible to assume that all the observations in the same patch are statistically independent. The confidence degrees under these hypotheses are almost binary: patches closest to the surface have confidence degrees equal to 1, while other patches have 0 confidence degree. We therefore adopted a different probabilistic model which takes into account

the average distance of the valid features to the surface along the normal direction, d_{av}. The association probabilities are given by (16) as before but with

$$p(y(t)|I_i(t), Y^{t-1}) = Ce^{-\gamma d_{av}^2}. \tag{27}$$

This algorithm will be called the 3DS-PDAF. Figure 12 shows the results obtained by the 3DS-PDAF method in the segmentation of the left ventricle in an ultrasound volume of data. This is a difficult problem due to the low signal-to-noise ratio, the presence of speckle noise and lack of visual boundary information (edge dropout) in some regions of the heart boundary. Figure 12a shows an example of a deformable surface estimated by the 3DS-PDAF method. Figure 12b, c shows cross sections of the 3D volume of data and of the estimated model.

7 Conclusions

This chapter presents a set of robust shape estimation algorithms based on robust feature extraction and robust model update techniques. These methods can be applied in a variety of problems ranging from static shape estimation, to object tracking in 2D and 3D applications.

The proposed algorithms exhibit an improved performance when compared with the classical deformable model techniques. The key ideas explored in these approaches are: (i) the use of more reliable features (line segments or surface patches) and (ii) explicit modeling of outlier observations by assigning a binary label to each of them. Formulated in this way, the estimation of the object shape is an inference problem with unobserved variables (labels) which can be tackled by the Expectation-Maximization method or by Data Association filtering algorithms.

Acknowledgments Part of the work on multi-model tracking was done in collaboration with Profs. Gilles Celeux (Paris-Sud University) and João M. Lemos (IST). The dermoscopic image was kindly provided by Dr. Jorge Rozeira of Hospital Pedro Hispano. This work was partially supported by FCT under projects PTDC/EEA-CRO/098550/2008, PTDC/SAU-BEB/103471/2008 and PTDC/EEA-CRO/103462/2008.

References

1. Bar-Shalom Y (1987) Tracking and data association. Academic Press, Boston
2. Bar-Shalom Y, Daum F, Huang J (2009) The probabilistic data association filter. IEEE Control Syst 29:82–100
3. Blake A, Michael I (1998) Active contours. Springer, London
4. Cohen LD, Cohen I (1993) Finite element methods for active contour models and balloons for 2-D and 3-D images. IEEE Trans Pattern Anal Mach Intell 15:1131–1147
5. Cootes TF, Cooper D, Taylor CJ, Graham J (1995) Active shape models—their training and application. Comput Vis Image Underst 61(1)38–59

6. DeCarlo D, Metaxas D (1996) The integration of optical flow and deformable models: applications to human face shape and motion estimation. IEEE computer vision and pattern recognition, pp 231–238
7. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(1981)
8. Gilles C, Marques JS, Nascimento JC (2004) Learning switching dynamic models for objects tracking. Pattern Recognit 37(9):1835–1840
9. Hervé D (1999) General object reconstruction based on simplex meshes. Int J Comput Vision 32:111–146
10. Jain AK, Zhong Y, Marie-Pierre Dubuisson-Jolly (1998) Deformable template models: a review. Signal Process 71:109–129
11. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vis 1(4):321–331
12. Lei H, Zhigang P, Bryan E, Xun W, Chia YH, Kenneth LW, William GW (2008) Comparative study of deformable contour methods on medical image segmentation. Image Vis Comput 26:141–163
13. Marques JS, Lemos JM (2001) Optimal and Suboptimal Shape Tracking Based on Multiple Switched Dynamic Models, Image and Vision Comput 19:539–550
14. Meer P, Mintz D, Rosenfeld A, Kim DY (1991) Robust regression methods for computer vision: a review. Int J Comput Vision 6:59–70
15. Moreno-Noguer F, Sanfeliu A, Samaras D (2007) Integration of deformable contours and a multiple hypotheses Fisher color model for robust tracking in varying illuminant environments. Image Vis Comput 25:285296
16. Nascimento JC, Marques JS (2002) Improving the robustness of parametric shape tracking with switched multiple models. Pattern Recognit 35:2711–2718
17. Nascimento JC, Marques JS (2004) Robust shape tracking in the presence of cluttered background. IEEE Trans Multimedia 6(6):852–861
18. Nascimento JC, Marques JS (2005) Adaptive snakes using the EM algorithm. IEEE Trans Image Process 14(11):1678–1686
19. Nilanjan R, Acton ST (2004) Flow motion gradient vector: an external force for tracking rolling leukocytes with shape and size constrained active contour. IEEE Trans Med Imaging 23:1466–1477
20. Silveira M, Nascimento JC, Marques JS, Marçal AR, Mendonça T, Yamauchi S, Maeda J, Rozeira J (2009) Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. IEEE J Sel Top Sign Proces 3:35–45
21. Terzopoulos D, Szeliski R (1992) Tracking with kalman snakes. In: Blake A, Yuille AL (eds) Active vision. MIT Press, Cambridge, pp 3–20
22. Xu C, Pham DL, Prince JL (2000) Medical image segmentation using deformable models. In: Handbook of medical imaging, vol 2. SPIE Press, Bellingham, pp 129–174
23. Xu C, Prince J (1998) Snakes, shapes, and gradient vector flow. IEEE Trans Image Process 7:359–369