

Adaptive Snakes Using the EM Algorithm

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

Abstract—Deformable models (e.g., snakes) perform poorly in many image analysis problems. The contour model is attracted by edge points detected in the image. However, many edge points do not belong to the object contour, preventing the active contour from converging toward the object boundary. A new algorithm is proposed in this paper to overcome this difficulty. The algorithm is based on two key ideas. First, edge points are associated in strokes. Second, each stroke is classified as valid (inlier) or invalid (outlier) and a confidence degree is associated to each stroke. The expectation maximization algorithm is used to update the confidence degrees and to estimate the object contour. It is shown that this is equivalent to the use of an adaptive potential function which varies during the optimization process. Valid strokes receive high confidence degrees while confidence degrees of invalid strokes tend to zero during the optimization process. Experimental results are presented to illustrate the performance of the proposed algorithm in the presence of clutter, showing a remarkable robustness.

Index Terms—Adaptive potential, contour estimation, deformable models, expectation maximization (EM) algorithm, robust estimation, snakes.

I. INTRODUCTION

DEFORMABLE models (e.g., “snakes”) have been widely used as a way to estimate the object boundary when the object shape is partially unknown. The deformable curve is an elastic curve which is initialized close to the object boundary and is able to elastically deform during the estimation process, under the influence of image (external) forces and internal forces. The ability of the curve to approximate the desired image features depends on the action of the external forces, whereas the internal forces impose smoothness constraints [1].

The design of image forces has been extensively studied see [2]–[5]) as a way to improve the convergence of the active curve toward the object boundary. The main difficulty concerns the invalid features detected in the image, i.e., features which are not located at the object boundary, but they are caused by other objects or by inner edges. The presence of such features (outliers) often attract the elastic model toward wrong shape configurations.

A great amount of research has been done to improve the performance of active contours in the presence of outliers, e.g., by using a validation gate to reduce the search region [6], nonlinear filtering techniques with non-Gaussian distributions (e.g., particle filters [7]), the use of geometric and dynamic constraints

Manuscript received June 9, 2004; revised August 30, 2004. This work was supported in part by POCTI and FCT under project LTT. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christophe Molina.

The authors are with the Instituto Superior Técnico/Instituto de Sistemas e Robótica (IST/ISR), 1049-001 Lisboa, Portugal (e-mail: jan@isr.ist.utl.pt).

Digital Object Identifier 10.1109/TIP.2005.857252



Fig. 1. Typical difficulties in the estimation of the lip boundary. (a) Original image. (b) Corresponding edges obtained with Prewitt operator. The presence of pixels that do not belong to the object (e.g., face, nose, and tongue edge pixels) often hampers the shapes estimates.

to reduce shape and motion variability [4], [8], and robust estimation techniques which are able to reduce the influence of outliers on the final shape estimate [9]. Fig. 1 shows an example in which the elastic model aims to estimate the outer boundary of the lips using the edge points obtained by the Prewitt algorithm. Although the edge points located at the lips boundary are correctly detected, there is a large number of outliers which make shape estimation a difficult task.

This paper proposes a robust algorithm for the estimation of the object boundary in the presence of outliers denoted as “*Adaptive Snakes*.” The proposed method is based on two key ideas. First, image strokes are used instead of edge points. Middle-level features, such as strokes, are more reliable than edge points. Strokes have been successfully used by several authors in [9] and [10]–[12]. Second, new external forces are used to update the contour model. In classic approaches, all the image features are considered as valid and used to compute the potential function. A different strategy is adopted in this paper. We assume that the feature extraction algorithm produces valid features, belonging to the object boundary (inliers), as well as invalid features, produced by other objects or by inner edges (outliers). Active contours should be able to discriminate both types of features in order to discard the influence of outliers and attract the contour model toward the inliers. This is done by assigning a confidence degree to each feature. The confidence degrees are not assigned by heuristic rules. They are defined as the probability of each stroke being valid given the best contour shape, and they are recursively computed using the *expectation maximization* (EM) method [13]. These probabilities are updated in each iteration using the EM algorithm. The relationship between the proposed method and the unified framework of shape analysis methods described in [14] is also studied. It is concluded that the adaptive snakes belong to the unified framework and can be implemented using its equations.

The paper is organized as follows. Section II describes related work. Section III describes the estimation of active contours,

assuming that we know which features are valid and which are outliers (ideal case of known labeling). Section IV extends these ideas to the case in which the labeling information is unknown. Section V deals with optimization issues, and Section VI presents the experimental results. Section VII concludes the paper.

II. RELATED WORK

The EM method was proposed in the 1970s [15] as a tool to solve maximum likelihood (ML) problems when some observations are missing (incomplete observations). Since then, it has been used in a wide variety of applications, e.g., in mixture estimation [16], neural networks [17], [18], hidden Markov models (HMMs) [19], [20], system identification [21], graphical models [22], [23], and image restoration [24].

The EM method has been recently used in object tracking to cope with hidden variables, e.g., shape and motion parameters (e.g., [25]–[27]). In [25], appearance models are used to describe the visual data. The EM method is employed to estimate the parameters of the appearance models from the data. In [27], a new EM contour algorithm is introduced. The E-step estimates the posteriori probability density function over a discrete set of shape deformations and the M-step estimates the contour location. The EM method has also been used in the context of object tracking using multiple motion and shape models [21], [28]. Multiple model trackers have been proposed to deal with sudden changes in shape or motion variables, which cannot be easily tackled by a single dynamical model. Models identification [21], [28] can be achieved using EM method. Another application concerns the representation of image features (edge points) by a mixture of Gaussians as an intermediate step in the estimation of the object contour. This approach is a way to perform data compression and to overcome the correspondence problem between the model and the data [26]. In this work, the outlier features are represented by a normal distribution with a large covariance matrix.

A related problem in which the EM algorithm has been used concerns three-dimensional shape and motion estimation from a sequence of images, assuming a rigid shape [29] (shape from motion problem [30]). In [29], the EM algorithm is used to solve the correspondence problem, i.e., the association between image features and model points. This is done by computing a confidence degree for each pair of feature/model point. The object shape is then estimated under the rigidity assumptions, taking all the data points into account, with different confidence degrees.

Another class of methods relying on confidence degrees, are the data association methods [e.g., the probabilistic data association filter (PDAF) and the joint probabilistic data association filter (JPDAF) [31]], widely used in target tracking for surveillance applications. These methods have been used to track isolated and multiple targets in the presence of clutter. The PDAF tracker provides robust estimates of the targets position and velocity in the presence of outliers and misdetections since it computes the association probabilities between the observed data and the estimated trajectory of the target, assuming that there

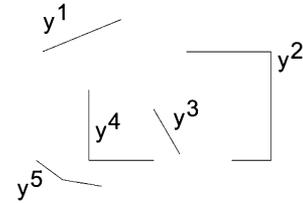


Fig. 2. Image strokes: Valid strokes (y^2, y^4) and outliers (y^1, y^3, y^5).

are detection errors (false positives and false negatives). However, the association probabilities are not obtained using the EM algorithm. They are computed by a sub optimal approximation of the *a posteriori* density of the state variables using a Gaussian distribution.

The method proposed in this paper is different from the above methods since it uses the EM algorithm to estimate the object shape in the presence of outliers, and assuming that shape is unknown and it may deform. Furthermore, the proposed method is based on middle level features (strokes) which are more reliable and informative than edge points.

III. IDEAL CASE

This section addresses shape estimation in the presence of outliers, assuming that the feature labels are known, i.e., assuming we know which features are valid and which are outliers. This is, of course, an unrealistic hypothesis, since we usually do not know which strokes are invalid. However, it is a useful step toward the final solution for the problem. For the sake of simplicity, no regularization forces will be considered in this section.

Let y be the set of all the features detected in an image organized in strokes $y = \{y^1, \dots, y^N\}$, y^j being the set of observations (edge points) belonging to the j th stroke.¹ Fig. 2 shows the organization of the data in strokes. In this figure, the object to be estimated is a rectangle, the strokes y^2, y^4 belong to the object boundary (inlier strokes), and the strokes y^1, y^3, y^5 do not belong to the rectangle, thus being outliers.

Let v be a contour model defined by a sequence of two-dimensional (2-D) points $v_i, i = 1, \dots, M$. The goal is to approximate the data contained in y , or a subset of it, by the contour model v .

To accomplish this, we shall consider the potential function

$$P(v_i; y, k) = - \sum_{j,n} \Phi(v_i; y_n^j, k^j) \quad (1)$$

where v_i is the i th model unit, y_n^j is the n th observation of the j th stroke, and $k = \{k^1, \dots, k^N\}$ is a set of stroke labels defined as follows: $k^j = 1$ if the j th stroke is valid, $k^j = 0$; otherwise, $\Phi(v_i; y_n^j, k^j)$ measures the influence of y_n^j on the model unit v_i . In this paper, we assume that Φ is a gaussian kernel $\mathcal{N}(y_n^j; v_i, \sigma^2 I)$ with mean v_i and covariance $\sigma^2 I$ when $k^j = 1$ and the outliers have a constant potential L within the analysis interval. Therefore

$$\Phi(v_i; y_n^j, k^j) = \begin{cases} \mathcal{N}(y_n^j; v_i, \sigma^2 I), & \text{if } k^j = 1 \\ L, & \text{if } k^j = 0. \end{cases} \quad (2)$$

¹A standard edge-linking algorithm is used to compute the image strokes.

In this way, the invalid features only influence the dc value of the potential function,² not its minima.

If y, k were known, the contour model could be obtained by minimizing the contour energy as follows:

$$\hat{v} = \arg \min_v E \quad (3)$$

where

$$E = \sum_i P(v_i; y, k). \quad (4)$$

Equation (4) is equivalent to the snake algorithm with the Cohen potential [3] if we assume that all data are valid, i.e., $k^j = 1, \forall j$.

The problem may also be addressed in a probabilistic framework, by assuming that y, k are random variables with probability density function

$$p(y, k | v) = \alpha e^{-\sum_i P(v_i; y, k)}. \quad (5)$$

In this case, the log likelihood function is

$$l(v; y, k) = \log p(y, k | v) = C - \sum_i P(v_i; y, k) \quad (6)$$

and the maximization of the log likelihood function leads to the optimization problem defined in (4).

In practice, we do not know which features are valid and which are outliers. The labels k^j are unknown and we need additional techniques to tackle this problem. This difficulty is addressed in the next section.

IV. ADAPTIVE POTENTIAL

Since the stroke labels are unknown in practice, the object contour should be estimated by maximizing the joint probability density function

$$\begin{aligned} \log p(y, v) &= \log \sum_k p(y, k, v) \\ &= \log \sum_k p(y, k | v) + \log p(v). \end{aligned} \quad (7)$$

where $p(v)$ is a prior distribution. Using (5), we obtain

$$\log p(y, v) = \log \left[\alpha \sum_k e^{-\sum_i P(v_i; y, k)} \right] + \log p(v). \quad (8)$$

The maximum *a posteriori* (MAP) estimates are obtained by solving the optimization problem

$$\hat{v} = \arg \max_v \log p(y, v). \quad (9)$$

This is, however, a difficult problem since we have to take into account every possible combination of labels k and it is not possible to obtain a closed form expression for (8) nor to analytically optimize it.

The EM method proposed in [15] can be used to solve this difficulty. In its original formulation, EM is presented as an al-

gorithm to perform ML estimates [13], [15], [32], [33]. The EM method can be used if there is an auxiliary variable k such that $\log p(y, k, v)$ has a simpler expression than $p(y, v)$ and can be optimized in an analytical way. The function $p(y, k, v)$ is denoted complete log probability distribution.

If k was observed, the unknown variables v would be estimated by

$$\hat{v} = \arg \max_v \log p(y, k, v). \quad (10)$$

Since k is not known, we cannot estimate v using (10). The EM algorithm replaces $\log p(y, k, v)$ by its expected value with respect to the hidden variables k , i.e.,

$$Q'(v; \hat{v}^{(t)}) = E_k \left\{ \log p(y, k, v) \mid y, \hat{v}^{(t)} \right\} \quad (11)$$

where $\hat{v}^{(t)}$ is the best estimate of v . An improved estimate is obtained by optimizing $Q'(v; \hat{v}^{(t)})$ with respect to v . A formal proof of the EM method can be found in [13]. This procedure is recursively performed until convergence is achieved. Equation (11) can also be written as

$$Q'(v; \hat{v}^{(t)}) = Q(v; \hat{v}^{(t)}) + \log p(v) \quad (12)$$

where $p(v)$ is a prior density and

$$Q(v; \hat{v}^{(t)}) = E \left\{ \log p(y, k | v) \mid y, \hat{v}^{(t)} \right\}. \quad (13)$$

The EM method approximates the MAP estimate \hat{v} by a sequence of contour estimates $\hat{v}^{(t)}, t = 0, 1, 2, \dots$, obtained as follows.

- **E-step:** Calculation of the auxiliary function

$$Q(v; \hat{v}^{(t)}) \triangleq E_k \left\{ \log p(y, k | v) \mid y, \hat{v}^{(t)} \right\}. \quad (14)$$

- **M-step:** Optimization of Q'

$$\hat{v}^{(t+1)} = \arg \max_v \left\{ Q(v; \hat{v}^{(t)}) + \log p(v) \right\}. \quad (15)$$

The E-step in the MAP method (14) is the same as in the ML approach. However, the M-steps are different since $Q(v; \hat{v}^{(t)})$ is augmented by an additional term depending on prior density $p(v)$ [see (12)]. The presence of this term leads to the appearance of the so called internal energy, which is a regularization term used in deformable contours methods.

Equation (14) can be rewritten as follows:

$$\begin{aligned} Q(v; \hat{v}^{(t)}) &\triangleq E_k \left\{ \log p(y, k | v) \mid y, \hat{v}^{(t)} \right\} \\ &= \sum_j E_{k_j} \left\{ \log p(y^j, k^j | v) \mid y^j, \hat{v}^{(t)} \right\} \\ &= \sum_j \sum_{l=0}^1 p(k^j = l \mid y^j, \hat{v}^{(t)}) \log p(y^j, k^j = l | v) \\ &= \sum_j w^j \log p(y^j, k^j = 1 | v) \\ &\quad + (1 - w^j) \log p(y^j, k^j = 0 | v) \end{aligned} \quad (16)$$

²The dc level of the potential function (1) is $-mL$ where m is the number of invalid features (outliers)

where

$$w^j = p(k^j = 1 \mid y^j, \hat{v}^{(t)}). \quad (17)$$

The second term in (16) is the outlier potential which does not depend on the contour model v . Thus

$$\begin{aligned} Q(v; \hat{v}^{(t)}) &= \sum_j w^j \log p(y^j, k^j = 1 \mid v) + \text{const} \\ &= \sum_j w^j (\log p(y^j \mid k^j = 1, v) \\ &\quad + \log p(k^j = 1 \mid v)) + \text{const}. \end{aligned} \quad (18)$$

Using (1) and (6), Q can be written as (the constant is discarded)

$$Q(v; \hat{v}^{(t)}) = \sum_j w^j \left(- \sum_{i,n} P(v_i; y_n^j, k^j = 1) + \log p(k^j = 1) \right). \quad (19)$$

Since all strokes are considered as valid ($k^j = 1$), a Gaussian distribution is used to model the potential in (1)

$$Q(v; \hat{v}^{(t)}) = \sum_{i,j,n} w^j \mathcal{N}(y_n^j; v_i, \sigma^2 I) + \sum_j w^j c^j \quad (20)$$

where $c^j = \log p(k^j = 1)$ does not depend on v . Furthermore, we shall assume that $\log p(v) \propto E_{\text{int}}(v)$ where $E_{\text{int}}(v)$ is an internal energy. Therefore, $p(v)$ is a Gibbs distribution $p(v) = (1/V_{\text{int}} \exp^{-E_{\text{int}}(v)})$. The regularization term used in this paper is similar to the one proposed in [34] which tries to keep the distance between consecutive model points close to an average distance l_0 , i.e., $E_{\text{int}}(v) = \sum_i (l_i - l_0)^2$, where $l_i = \|v_{i+1} - v_i\|$ is the distance between consecutive model points and l_0 is the average distance specified by the user.

Therefore, the function Q' becomes

$$Q'(v; \hat{v}^{(t)}) = C_0 + C_1 - \sum_i \mathcal{P}_a(v_i, y) - \sum_i (l_i - l_0)^2 \quad (21)$$

where $C_0 = \sum_j w^j c^j$, $C_1 = \log(1/V_{\text{int}})$, and

$$\mathcal{P}_a(v_i, y) = - \sum_j w^j \left(\sum_n \mathcal{N}(y_n^j; v_i, \sigma^2 I) \right) \quad (22)$$

is denoted as an *adaptive potential* since it depends on the confidence degrees of the image strokes w^j which vary during the estimation process.

Let us consider the estimation of the weights w^j in the E-step of the EM algorithm. Using (1), (2), and (5)

$$\begin{aligned} p(y, k = 1 \mid \hat{v}^{(t)}) &= \alpha e^{\sum_i \sum_{j,n} \mathcal{N}(y_n^j; v_i, \sigma^2 I)} \\ &= \alpha \prod_j \prod_i e^{\sum_n \mathcal{N}(y_n^j; v_i, \sigma^2 I)}. \end{aligned} \quad (23)$$

Assuming that y^j, k^j are independent random variables

$$p(y, k = 1 \mid \hat{v}^{(t)}) = \prod_j p(y^j, k^j = 1 \mid \hat{v}^{(t)}). \quad (24)$$

Therefore, comparing (23) and (24)

$$p(y^j, k^j = 1 \mid \hat{v}^{(t)}) = \alpha^j \prod_i e^{\sum_n \mathcal{N}(y_n^j; v_i, \sigma^2 I)}. \quad (25)$$

Using (17)

$$w^j = p(k^j = 1 \mid y^j, \hat{v}^{(t)}) = \beta^j \prod_i e^{\sum_n \mathcal{N}(y_n^j; v_i, \sigma^2 I)} \quad (26)$$

where β^j is a normalization constant, β^j can be obtained by computing

$$p(k^j = 0 \mid y^j, \hat{v}^{(t)}) = \beta^j \prod_i e^{\sum_n L} \quad (27)$$

since

$$p(k^j = 0 \mid y^j, \hat{v}^{(t)}) + p(k^j = 1 \mid y^j, \hat{v}^{(t)}) = 1 \quad (28)$$

then

$$\beta^j = \left[e^{\sum_{i,n} \mathcal{N}(y_n^j; v_i, \sigma^2 I)} + e^{Mn^j L} \right]^{-1}. \quad (29)$$

where M is the number of model units and n^j is the length of the j th stroke, and $L = (1/n^j)$.

V. CONTOUR ESTIMATION

The contour estimation is performed by the EM algorithm described in the previous section. We shall now consider some additional details. The M-step involves the minimization of an energy with two terms: a regularization term which tries to keep the distance between consecutive model points close to an average distance l_0 and an image dependent term given by (22). Therefore

$$Q'(v; \hat{v}^{(t)}) = -E_{\text{int}}(v) - \sum_i \mathcal{P}_a(v_i, y). \quad (30)$$

The minimization of (30) is performed in the M-step as follows:

$$\hat{v}^{(t+1)} = \arg \max_v Q'(v; \hat{v}^{(t)}). \quad (31)$$

Using the gradient algorithm

$$\hat{v}^{(t+1)} = \hat{v}^{(t)} - \gamma \nabla_v Q'(v; \hat{v}^{(t)}) \quad (32)$$

where ∇_v is the gradient operator defined by $\nabla_v Q'(v; \hat{v}^{(t)}) = [\nabla_{v_1} Q', \dots, \nabla_{v_M} Q']^T$. The algorithm is a generalized EM (GEM) [13], since it does not attempt to find the value of v that globally maximizes the function $Q'(v; \hat{v}^{(t)})$.

Equation (32) can be rewritten as follows:

$$\hat{v}^{(t+1)} = \hat{v}^{(t)} - \gamma_i F_{\text{int}} + \gamma_e F_{\text{ext}} \quad (33)$$

where γ_i, γ_e are gains, $F_{\text{int}} = \{F_{\text{int}}(v_1), \dots, F_{\text{int}}(v_N)\}$, $F_{\text{ext}} = \{F_{\text{ext}}(v_1), \dots, F_{\text{ext}}(v_N)\}$ are interpreted as internal and external forces applied to the model unit v_i , respectively. A gain $\gamma_i = (1/\|F_{\text{int}}\|)$ is used to normalize the internal force F_{int} and γ_e is an adaptive gain which tries to avoid abrupt changes of the model units in a single iteration (details can be found in [14]). After straightforward manipulations, we obtain the following expressions for the internal and external forces:

$$F_{\text{int}}(v_i) = -2\frac{l_i - l_0}{l_i}(v_{i+1} - v_i) + 2\frac{l_{i-1} - l_0}{l_{i-1}}(v_i - v_{i-1}) \quad (34)$$

$$F_{\text{ext}}(v_i) = \frac{1}{\sigma^2} \sum_j w^j \sum_n (y_n^j - v_i) \mathcal{N}(y_n^j; v_i, \sigma^2 I). \quad (35)$$

The external forces (35) can also be written in a different way. If we define a weight function $\vartheta_i(y_n^j) = \mathcal{N}(y_n^j; v_i, \sigma^2 I)$, the external forces are given by

$$\begin{aligned} F_{\text{ext}}(v_i) &= \alpha \sum_j w^j \sum_n (y_n^j - v_i) \vartheta_i(y_n^j) \\ &= \alpha \left\{ \sum_j w^j \sum_n y_n^j \vartheta_i(y_n^j) - v_i \sum_j w^j \sum_n \vartheta_i(y_n^j) \right\} \\ &= \alpha \sum_j w^j \sum_n \vartheta_i(y_n^j) \left[\frac{\sum_j w^j \sum_n y_n^j \vartheta_i(y_n^j)}{\sum_j w^j \sum_n \vartheta_i(y_n^j)} - v_i \right]. \end{aligned} \quad (36)$$

Therefore, the external force applied to v_i becomes

$$F_{\text{ext}}(v_i) = \mu_i(\xi_i - v_i) \quad (37)$$

where

$$\begin{aligned} \mu_i &= \sum_j w^j \sum_n \vartheta_i(y_n^j) \\ \xi_i &= \frac{\sum_j w^j \sum_n y_n^j \vartheta_i(y_n^j)}{\sum_j w^j \sum_n \vartheta_i(y_n^j)} \end{aligned} \quad (38)$$

are denoted as the mass and centroid of the i th unit.

Equation (37) has an intuitive meaning. The external force attracts each model unit v_i toward a data centroid ξ_i computed as a weighted sum of edge points. The force magnitude is proportional to the distance from the model point to the centroid. These expressions are similar to the ones derived in [14] which considers several methods sharing the same structure and denoted as a unified framework for active contours. The methods belonging to this framework share a set of common properties namely: In each iteration, the model units are attracted toward data centroids using different choices for $\vartheta_i(y_n^j)$.

Table I summarizes the adaptive snake algorithm. The next section presents experimental results and performs a comparison between the snake and adaptive snake algorithms.

TABLE I
ADAPTIVE SNAKES

Adaptive Snakes	
Strokes detection: Detect edge points in the image and organize them in strokes, by performing an edge-linking operation	
iterate	
1) E-step: For each stroke compute the weight	$w^j = p(k^j = 1 y^j, \hat{v}) = \beta^j \prod_i e^{\sum_n \mathcal{N}(y_n^j; v_i, \sigma^2 I)}$
2) M-step: Update the contour model by	$\hat{v}^{(t+1)} = \hat{v}^{(t)} - \gamma_i F_{\text{int}}(v) + \gamma_e F_{\text{ext}}(v)$,
where γ_i, γ_e are gains defined as in [14], [2]. The internal and external forces applied to each model unit are given by	
$F_{\text{int}}(v_i) = -2\frac{l_i - l_0}{l_i}(v_{i+1} - v_i) + 2\frac{l_{i-1} - l_0}{l_{i-1}}(v_i - v_{i-1})$, $F_{\text{ext}}(v_i) = \mu_i(\xi_i - v_i)$,	
with	
$\mu_i = \sum_j w^j \sum_n \vartheta_i(y_n^j)$, $\xi_i = \frac{\sum_j w^j \sum_n y_n^j \vartheta_i(y_n^j)}{\sum_j w^j \sum_n \vartheta_i(y_n^j)}$,	
where $\vartheta_i(y_n^j) = \mathcal{N}(y_n^j; v_i, \sigma^2 I)$.	

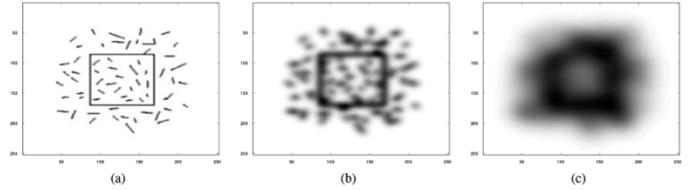


Fig. 3. Image potentials obtained with (a) $\sigma = 1$, (b) $\sigma = 5$, and (c) $\sigma = 15$.

VI. EXPERIMENTAL RESULTS

This section presents experimental results illustrating the performance of the proposed algorithm in the estimation of object boundaries in synthetic and real images. We compare the proposed algorithm with the classical snakes which is a special case of the proposed method, assuming that all image features are valid. A statistical study is provided to assess the performance of both methods when the number of outliers is increased, in synthetic images. Then, we illustrate the performance of both methods in real images. To improve the performance of the algorithm, and its convergence rate, the following procedures are adopted: in each iteration, the boundary model is resampled at equally spaced points; a gain $\gamma_i = (1/\|F_{\text{int}}\|)$ is used to normalize the internal forces; independent gain factors γ_e acting on each model unit are used as described in [14].

The first example illustrates the estimation of a square shape in a cluttered environment. Fig. 4 shows the results obtained with the snakes in a synthetic image. The size of the image is 256×256 and $\sigma = 5$ pixel was used [see (2)]. This parameter has a key role in the design of the external forces. If the value of σ is set too high, the centroids [see (38)] will be located close to the mass center of the data. This happens, since the potential valleys are very broad and overlap. On the contrary, if σ is set too small, the potential valleys are narrow. Thus, the model units

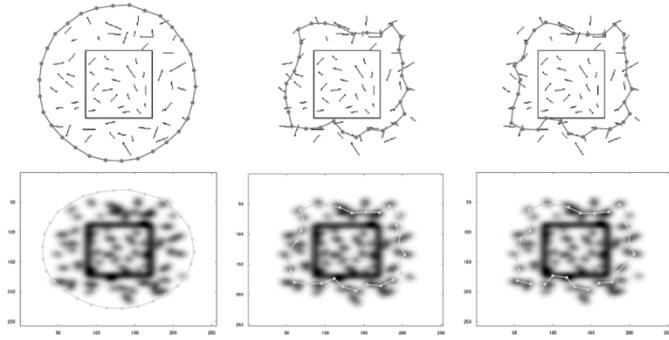


Fig. 4. (Top row) Results obtained with snakes and (bottom row) the correspondent attraction valleys. Each row shows initial, intermediate, and final results.

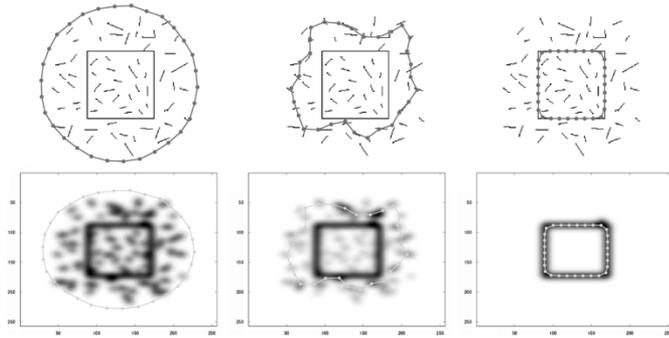


Fig. 5. (Top row) Results obtained with adaptive snakes potential and (bottom row) the correspondent attraction valleys. Each row shows initial, intermediate, and final results.

v_i are only attracted by a small amount of data points in the vicinity of the units. In this case, it is not possible to merge the influence of different edge points. This means that the object boundary may not be correctly estimated. Fig. 3 illustrates the potential function for different values of σ . In the next examples, we assume that $\sigma \in [5, 10]$.

Fig. 4 shows the performance of snakes in this example. The first row shows the image features and the contour model at $t = 0, 20, 50$. It is shown that the model gets stuck in the potential valleys associated with outliers. The second row shows the potential function of the snake algorithm which remains constant during the estimation process. One can see that the valleys remain invariant. The valleys associated with the outliers are not removed and produce a strong attraction of the contour model. This explains why the estimates obtained with the snake potential do not converge toward the object boundary. The model units get stuck in the valleys (local minima) associated to the outliers.

Fig. 5 shows the results obtained with adaptive snakes at the same iterations. The adaptive potential changes during the optimization process managing to assign larger weights to the true strokes and discarding the influence of the outliers, since they have smaller lengths and are farther from the contour. It is concluded that the removal of the outlier valleys is crucial to achieve accurate estimates of the object boundary. The ability of the potential function to change during the iteration process is a key issue for the success of this method. In this way, it is possible to modify the external forces applied to the contour model.

Monte Carlo tests were performed to evaluate the performance of both methods in the estimation of the square object.

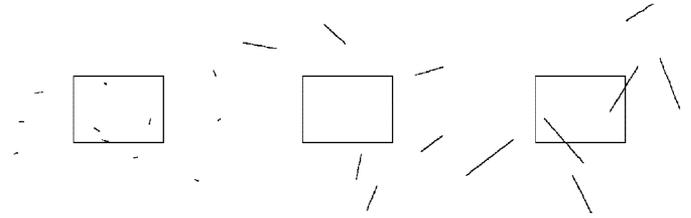


Fig. 6. Randomly generated data with (left) $p = 30\%$, $\mu = 3$, (center) $p = 60\%$, $\mu = 10$, and (right) $p = 100\%$, $\mu = 20$.

The test images were obtained by adding random strokes to the original image. The number of outlier strokes is randomly generated and the length of each stroke is a random variable with gamma distribution. The parameters of the gamma distribution are $\alpha = \mu/\beta$, $\beta = \sigma^2/\mu$, where μ and σ are the mean and variance of the stroke length. The initial point of each stroke is randomly generated with uniform distribution in the image domain.

In these experiments, two parameters were changed: the number of strokes (stroke percentage p) and the average length μ . The stroke percentage p is defined as the sum of stroke lengths divided by the object perimeter. For each value of μ , the stroke percentage was changed from 10% to 100%. Three values were considered for the average stroke length: $\mu = 3, 10, 20$. Fig. 6 shows three sample images used in the tests with $p = 30\%, 60\%, 100\%$.

The final shape estimates were evaluated using two distances: the average distance and the maximum distance defined as

$$\mathcal{D}_{av} = \frac{1}{2}((d_{av}(v, y) + d_{av}(y, v))) \quad (39)$$

$$\mathcal{D}_{max} = \frac{1}{2}((d_{max}(v, y) + d_{max}(y, v))) \quad (40)$$

where

$$d_{av}(v, y) = \frac{1}{M} \sum_{n=1}^M \min_{y_i \in y} \|v_n - y_i\| \quad (41)$$

is the average distance from the contour model p to the true boundary (ideal contour) defined by the set y , and

$$d_{max}(v, y) = \max_n \min_{y_i \in y} \|v_n - y_i\| \quad (42)$$

is the largest distance from the contour model to the true boundary. The other two measures $d_{av}(y, v)$, $d_{max}(y, v)$ are obtained by changing the role of y and v in (41) and (42).

Fig. 7 shows the performance of both methods for several values of μ and p . The solid line corresponds to the estimates obtained with the adaptive snake and the dashed line corresponds to the snake algorithm with the Cohen potential. These results were computed by performing ten experiments for each test condition. The first row shows the values of $\overline{\mathcal{D}_{av}}$ while the second shows the results of $\overline{\mathcal{D}_{max}}$. It is concluded from Fig. 7 that the proposed algorithm is robust against the presence of outliers while the snake algorithm shows a significant degradation when the noise level increases, specially in the case of small strokes. When the average length is small, a large number of strokes are generated filling the whole image plane. In this case,

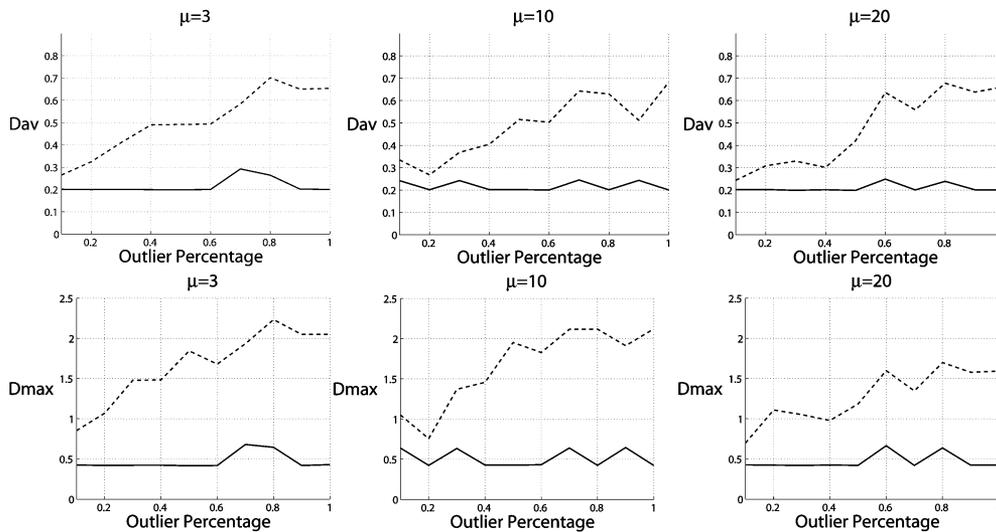


Fig. 7. (Dashed line) Results obtained with snake and (solid line) adaptive snake. (First row) $\overline{D_{av}}$. (Second row) $\overline{D_{max}}$. From left to right, $\mu = 3$, $\mu = 10$, $\mu = 20$.

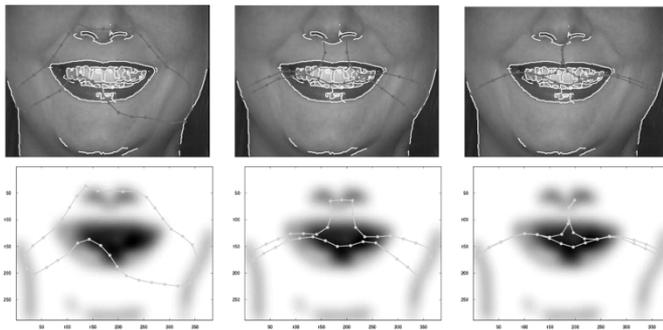


Fig. 8. (First row) Shape estimates and (second row) snake potential.

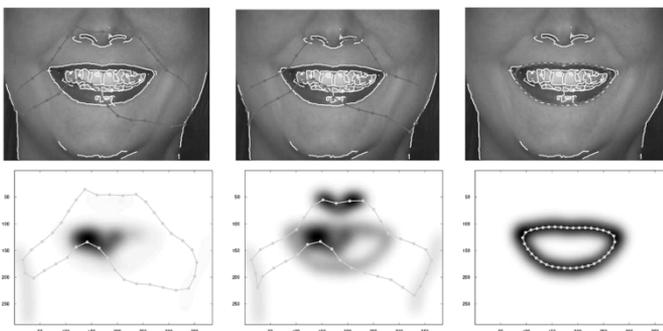


Fig. 9. (First row) Shape estimates and (second row) adaptive potential.

the deformable model gets easily stuck during the convergence process.

Figs. 8 and 9 show an example concerning the estimation of the lips boundary. The size of the image is 200×250 , and $\sigma = 10$. The strokes detected in the image are represented as thin yellow lines. The initialization is far from the lips boundary; therefore, the model units are attracted by outliers located at the face contour. The poor convergence of the snakes is notorious [see Fig. 8] since it is sensitive to invalid strokes, as well. The results obtained with adaptive snakes are much better [Fig. 9]

since the method manages to provide correct estimates of the lips contour. The final estimates accurately represent the lips boundary.

From this point on, we present experimental results obtained with adaptive snakes only. Fig. 10 shows five artistic images,³ the strokes detected in the image (yellow lines), the initial contour (red line), and the evolution of the contour model estimated using the adaptive potential.

The sizes of images, the number of detected strokes, and the values of the parameter σ are summarized in Table II.

Fig. 11, shows the evolution of the adaptive potential proposed herein, as well as the position of the contour model (green line). Dark valleys have the strongest influence on the contour estimates. A similar behavior is observed in the first two examples (first and second row of the Fig. 11). However, the second example has much more strokes than the first. Therefore, the convergence of the algorithm is faster in the first example than in the second case (compare the second column of Fig. 11).

In the sea stack example (third row), the clouds and the sea strokes have a strong influence in the first few iterations. When the contour model approximates the stack boundary, the longest stroke starts to have a stronger influence and the converge to the correct boundary is achieved.

In the cabbage example, there are several strokes which belong to the cabbage boundary. This example is different from the previous ones since the object of interest is described by a multiple strokes. During the first iterations, the upper right stroke has a strong influence on the model configuration, providing an initial attraction valley. In this way, it is possible to estimate the model units in this region well (see left column in the fourth row). Then, the left stroke starts to create its own valley, which helps the contour model to estimate the left boundary. Finally, on the right column we see the final model. Although the left stroke is the only visible valley in Fig. 11, its influence is not enough to attract all the model units located far from this valley.

³These images were obtained from <http://www.billatkinson.com/>—original works of art by Bill Atkinson.

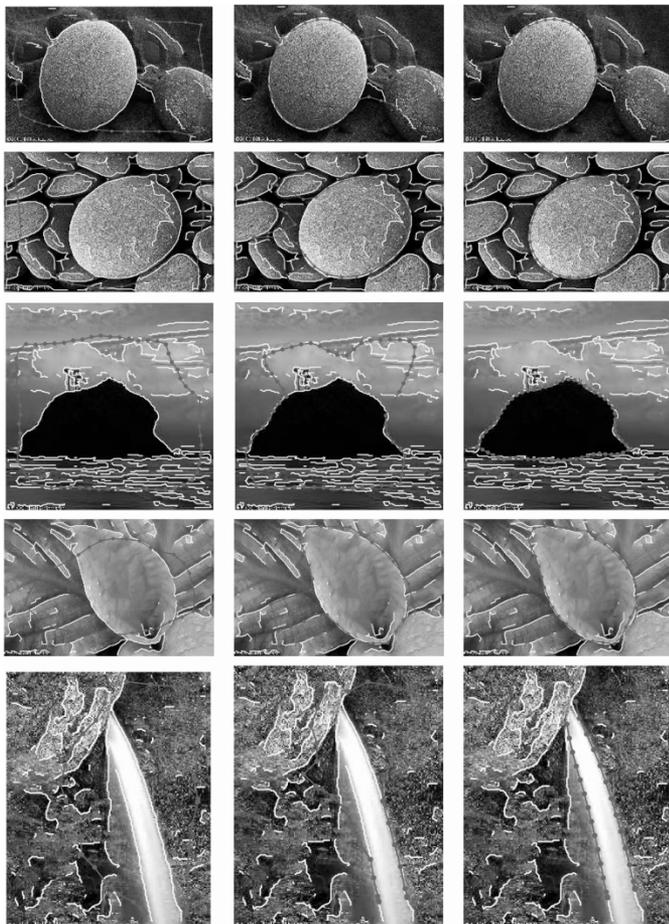


Fig. 10. Shape estimates using adaptive snakes: (first row) round rock, (second row) round beach stones, (third row) sea stack at sunset, (fourth row) cabbage, and (fifth row) waterfall.

TABLE II
EXPERIMENTAL RESULTS OBTAINED WITH ADAPTIVE SNAKES

Image	Size	Strokes number	σ
Round Rock	176 × 268	18	7
Round Beach Stones	176 × 268	47	10
Cabbage	177 × 268	46	6
Sea Stack at Sunset	216 × 214	74	6
Waterfall	216 × 171	86	8

In this example, we see the importance of the individual contribution of the strokes taken at different iterations of the estimation process.

The last example (waterfall) is similar to the previous one. In this case we have used two open lines to estimate the object shape. These examples, illustrate that the algorithm manages to solve situations where the object is described by a single stroke and by multiple strokes detected in the image. Fig. 12 shows the detected strokes, as well as the initializations.

VII. CONCLUSION

This paper describes a new algorithm for the estimation of objects boundaries in the presence of outliers. The object boundary is approximated by a deformable contour as in snakes, defined by a sequence of 2-D points (model units). The

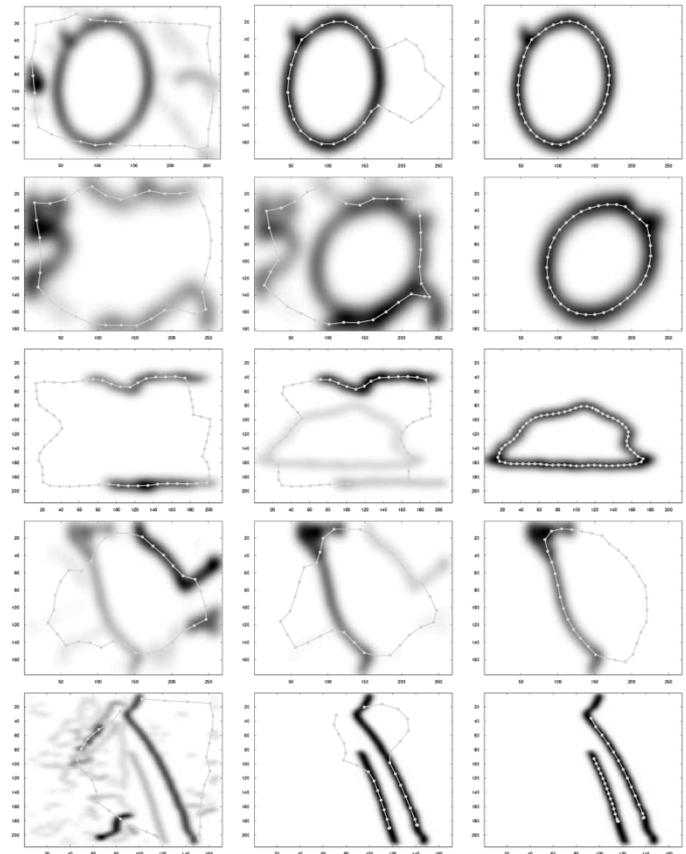


Fig. 11. Evolution of the adaptive potential through the iterative process. Circles denote the convergence of the model.

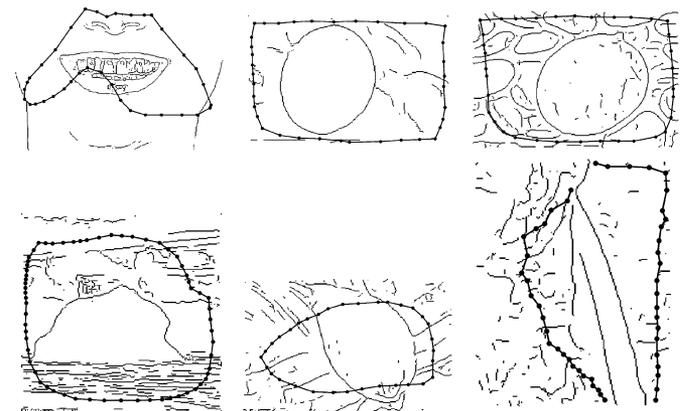


Fig. 12. Strokes detected in the images (circles are the initial position of the model).

model units are deformed by the action of internal forces and external forces computed using an image potential. However, instead of using a classic potential function which remains invariant during the shape estimation process, an adaptive potential is proposed which is able to discard the influence of outliers. This is achieved as follows. Image features (edges points) are organized in strokes and a binary label variable is associated to each stroke. The label should be one if the stroke belongs to the object boundary and zero otherwise. Since this information is not available, a confidence degree is assigned to each stroke, being updated during the estimation process.

Therefore, all strokes contribute to the image potential but with different weights. The image potential and the contour model are recursively obtained by the MAP method using the EM algorithm.

Experimental tests show that the proposed algorithm is robust and provides much better results than classic methods in the presence of clutter, achieving significant robustness. It was also observed that the adaptive snakes are robust with respect to initialization specially when the outlier strokes are smaller than the object strokes.

ACKNOWLEDGMENT

The authors would like to thank B. Atkinson for the artistic nature images used in Fig. 10. They would also like to thank the anonymous reviewers for their helpful comments which improved the paper.

REFERENCES

- [1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 14, pp. 321–331, 1987.
- [2] L. Cohen, "On active contour models and balloons," *Image Understand.*, vol. 53, no. 2, pp. 211–218, 1991.
- [3] L. Cohen and I. Cohen, "Finite element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1131–1147, Nov. 1993.
- [4] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.
- [5] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.
- [6] A. Blake and M. Isard, *Active Contours*. New York: Springer, 1998.
- [7] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *Proc. 6th Int. Conf. Computer Vision*, N. P. House, Ed., Bombay, Maharashtra, India, 1998, pp. 107–112.
- [8] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *J. Comput. Vis.*, vol. 11, no. 2, pp. 127–145, 1993.
- [9] J. Nascimento and J. Marques, Robust shape tracking in the presence of cluttered background, in *IEEE Trans. Multimedia*, to be published.
- [10] H. Gu, Y. Shirai, and M. Asada, "Mdl-based segmentation and motion modeling in a long sequence of scene with multiple independently moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 58–64, Jan. 1996.
- [11] S. Kalitzin, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever, "Image segmentation and object recognition by bayesian grouping," in *IEEE Int. Conf. Image Processing*, vol. 3, 2000, pp. 580–583.
- [12] X. Zhang and H. Burkhardt, "Grouping edge points into line segments by sequential hough transformation," in *Int. Conf. Pattern Recognition*, vol. 3, Vancouver, BC, Canada, 2000, pp. 676–679.
- [13] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
- [14] A. Abrantes and J. Marques, "A class of constrained clustering algorithms for object boundary detection," *IEEE Trans. Image Process.*, vol. 5, no. 11, pp. 1507–1521, Nov. 1996.
- [15] A. Dempster, M. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em-algorithm," *J. Roy. Stat. Soc. B*, no. 39, pp. 1–38, 1977.
- [16] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
- [17] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neur. Comput.*, vol. 3, pp. 79–87, 1991.
- [18] S. Ng and G. McLachlan, "Using the EM Algorithm to train neural networks: Misconceptions and a new algorithm for multiclass classification," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 738–749, May 2004.
- [19] X. D. Huang, Y. Arika, and M. A. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh, U.K.: Edinburgh Univ. Press, 1990.
- [20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [21] B. North and A. Blake, "Using expectation-maximization to learn dynamical models from visual data," *Image Vis. Comput.*, vol. 17, no. 8, pp. 611–616, 1999.
- [22] E. Ghahramani, "An introduction to hidden markov models and bayesian networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 5, no. 1, pp. 9–42, 1991.
- [23] M. Jordan, *Learning in Graphical Models*, M. Jordan, Ed. Cambridge, MA: MIT Press, 1997.
- [24] M. A. T. Figueiredo and R. D. Nowak, "An em algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
- [25] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. I, Kauai, HI, 2001, pp. 415–422.
- [26] J. Nascimento, A. J. Abrantes, and J. S. Marques, "Using middle level features for robust shape tracking," *Pattern Recognit. Lett.*, vol. 24, pp. 295–307, 2003.
- [27] A. E. C. Pece and A. D. Worrall, "Tracking with the EM contour algorithm," in *Proc. 7th Eur. Conf. Comp. Vision*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds, 2002, pp. 3–17.
- [28] G. Celeux, J. Nascimento, and J. S. Marques, "Learning switching dynamic models for objects tracking," *Pattern Recognit.*, to be published.
- [29] S. T. F. Dellaert, S. M. Seitz, and C. Thorpe, "Feature correspondence: A Markov chain monte carlo approach," presented at the Advances in Neural Information Processing Systems 13, 2001.
- [30] D. A. Forsyth, S. Ioffe, and J. Haddon, "Bayesian structure from motion," in *Proc. 7th Int. Conf. Computer Vision*, vol. 2, Greece, 1999, pp. 660–665.
- [31] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.
- [32] R. Little and D. Rubin, *Statistical Analysis with Missing Data*. New York: Wiley, 1987.
- [33] M. Tanner, *Tools for Statistical Inference*. New York: Springer-Verlag, 1993.
- [34] F. Leymarie and M. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 617–634, Jun. 1993.

Jacinto C. Nascimento (M'05), photograph and biography not available at the time of publication.

Jorge S. Marques, photograph and biography not available at the time of publication.