

# A Framework for Image Segmentation Using Shape Models and Kernel Space Shape Priors

Samuel Dambreville, *Student Member, IEEE*, Yogesh Rathi, *Member, IEEE*, and Allen Tannenbaum, *Member, IEEE*

**Abstract**—Segmentation involves separating an object from the background in a given image. The use of image information alone often leads to poor segmentation results due to the presence of noise, clutter, or occlusion. The introduction of shape priors in the geometric active contour (GAC) framework has proven to be an effective way to ameliorate some of these problems. In this work, we propose a novel segmentation method combining image information with prior shape knowledge using level sets. Following the work of Leventon et al., we propose revisiting the use of principal component analysis (PCA) to introduce prior knowledge about shapes in a more robust manner. We utilize kernel PCA (KPCA) and show that this method outperforms linear PCA by allowing only those shapes that are close enough to the training data. In our segmentation framework, shape knowledge and image information are encoded into two energy functionals entirely described in terms of shapes. This consistent description permits us to fully take advantage of the KPCA methodology and leads to promising segmentation results. In particular, our shape-driven segmentation technique allows for the simultaneous encoding of multiple types of shapes and offers a convincing level of robustness with respect to noise, occlusions, or smearing.

**Index Terms**—Kernel methods, shape priors, active contours, principal component analysis, level sets.

## 1 INTRODUCTION

SEGMENTATION consists of extracting an object from an image, a ubiquitous task in computer vision applications. It is quite useful in applications ranging from finding special features in medical images to tracking deformable objects; see [1], [2], [3], [4], and the references therein. The active contour methodology has proven to be very effective for performing this task. However, the use of image information alone often leads to poor segmentation results in the presence of noise, clutter, or occlusion. The introduction of shape priors in the contour evolution process has been shown to be an effective way to address this issue, leading to more robust segmentation performances.

A number of methods that use a parameterized or an explicit representation for contours have been proposed [5], [6], [7] for active contour segmentation. In [8], the authors use the B-spline parameterization to build shape models in the kernel space [9]. The distribution of shapes in kernel space was assumed to be Gaussian and a Mahalanobis distance was minimized during the segmentation process to provide a shape prior.

The geometric active contour (GAC) framework (see [10] and the references therein) involves a parameter-free

representation of contours, that is, a contour is represented implicitly by the zero level set of a higher dimensional function, typically a signed distance function [11]. In [1], the authors obtain the shape statistics by performing linear principal component analysis (PCA) on a training set of signed distance functions (SDFs). This approach was shown to be able to convincingly capture small variations in the shape of an object. It inspired other schemes to obtain the shape prior described in [2], [12], notably where SDFs were used to learn the shape variations.

However, when the object considered for learning undergoes complex or nonlinear deformations, linear PCA can lead to unrealistic shape priors by allowing linear combinations of the learned shapes that are unfaithful to the true shape of the object. Cremers et al. [13] successfully pioneered the use of kernel methods to address this issue within the GAC framework using a Parzen estimator to model the shape distribution in kernel space.

The present work builds on the methods and results outlined by the authors in [14]. We propose using kernel PCA (KPCA) to introduce shape priors for GACs. KPCA was proposed by Mika et al. [9] and allows one to combine the precision of kernel methods with the reduction of dimension in training sets. This is the first time, to our knowledge, that KPCA has been explicitly used to introduce shape priors in the GAC framework. We also propose a novel intensity-based segmentation method specifically tailored to meaningfully allow for the inclusion of a shape prior. Image and shape information are described in a consistent fashion that allows us to combine energies to realize meaningful trade-offs.

We now outline the contents of this paper. In Section 2, we briefly recall generalities concerning active contours using level sets. In Section 3, we propose a consistent

• S. Dambreville and A. Tannenbaum are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive, Atlanta, GA 30332-0250.

E-mail: samuel.dambreville@bme.gatech.edu, tannenbu@ece.gatech.edu.  
 • Y. Rathi is with the Psychiatry Neuroimaging Lab, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02115.  
 E-mail: yogesh.rathi@gmail.com.

Manuscript received 25 Oct. 2006; revised 17 June 2007; accepted 23 July 2007; published online 6 Sept. 2007.

Recommended for acceptance by B.S. Manjunath.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0754-1006. Digital Object Identifier no. 10.1109/TPAMI.2007.70774.

method to introduce shape priors within the GAC framework, using linear PCA and KPCA. Next, in Section 4, we propose a novel intensity-based energy functional separating an object from the background in an image. This energy functional has a strong shape interpretation. Then, in Section 5, we present a robust segmentation framework, combining image cues and shape knowledge in a consistent fashion. The performances of linear PCA and KPCA are compared and the performance and robustness of our segmentation method are demonstrated on various challenging examples in Section 6. Finally, in Section 7, we make our conclusions and describe possible future research directions.

## 2 LEVEL-SET EVOLUTION

Level-set representations were introduced by Osher and Sethian [15], [16] to model interface motion and became a popular tool in the fields of image processing and computer vision. The idea consists of representing a contour by the zero-level set of a smooth Lipschitz continuous function. A common choice is to use an SDF for embedding the contour. The contour is propagated implicitly by evolving the embedding function to decrease a chosen energy functional. Implicit representations present the advantage of avoiding dealing with complex resampling schemes of control points. Moreover, the contour represented implicitly can naturally undergo topological changes such as splitting and merging.

In what follows, the SDF representing the contour of interest will be denoted by  $\phi$ . The SDF  $\phi$  is a function  $\phi: \Omega \rightarrow \mathbf{R}$  and the contour corresponds to the zero-level set of  $\phi$ , that is,  $\{(x, y) \in \Omega / \phi(x, y) = 0\}$ . The convention  $\phi(x, y) \geq 0$  will be taken for points  $(x, y)$  belonging to the interior of the contour.

Typically, the energy functional is minimized via a gradient descent approach. Accordingly, one defines an energy functional in terms of  $\phi$ , namely,  $E(\phi)$  (see, for example, [17], [16], [15], and references therein), and, then, the contour evolution is derived via the following gradient descent flow:

$$\frac{d\phi}{dt} = -\nabla_{\phi} E. \quad (1)$$

This amounts to deforming the contour from an initial contour  $\phi_0$  in order to minimize the energy  $E(\phi)$ .

## 3 LINEAR AND KERNEL PCA FOR SHAPE PRIORS

In this section, we recall a general formulation allowing one to perform linear PCA, as well as KPCA, on any data set [18], [19]. Then, we present specific kernels allowing us to perform linear or nonlinear PCA on training sets of shapes. Finally, we propose an energy functional allowing us to introduce shape priors obtained from either linear PCA or KPCA within the GAC framework. Recent studies compared other methods such as locally linear embedding (LLE) or kernel LLE to perform shape analysis and learning (with better results usually observed on the tested sequences for kernel LLE as compared to LLE; see [20]). Comparable levels of performance were observed for KPCA and for kernel LLE. Although the present framework is

general enough to use other learning methods, we focus on KPCA in what follows.

### 3.1 KPCA

KPCA can be considered to be a generalization of linear PCA. This technique was introduced by Mika et al. [9] and has proven to be a powerful method to extract nonlinear structures from a data set. The idea behind KPCA consists of mapping a data set from an input space  $\mathcal{I}$  into a feature space  $F$  via a nonlinear function  $\varphi$ . Then, PCA is performed in  $F$  to find the orthogonal directions (principal components) corresponding to the largest variation in the mapped data set. The first  $l$  principal components account for as much of the variance in the data as possible by using  $l$  directions. In addition, the error in representing any of the elements of the training set by its projection onto the first  $l$  principal components is minimal in the least squares sense.

The nonlinear map  $\varphi: \mathcal{I} \rightarrow F$  typically does not need to be known through the use of Mercer kernels. A  *Mercer kernel* is a function  $k(\cdot, \cdot)$  such that, for all data points  $\chi_i$ , the kernel matrix  $\mathbf{K}(i, j) = k(\chi_i, \chi_j)$  is symmetric positive definite [9]. According to Mercer's Theorem (see [21]), computing  $k(\cdot, \cdot)$  as a function of  $\mathcal{I} \times \mathcal{I}$  amounts to computing the inner scalar product in  $F: k(\chi_a, \chi_b) = (\varphi(\chi_a) \cdot \varphi(\chi_b))$ , with  $(\chi_a, \chi_b) \in \mathcal{I} \times \mathcal{I}$ . This scalar product in  $F$  defines a distance  $d_F$  such as

$$\begin{aligned} d_F^2(\varphi(\chi_a), \varphi(\chi_b)) &= \|\varphi(\chi_a) - \varphi(\chi_b)\|^2 \\ &= k(\chi_a, \chi_a) - 2k(\chi_a, \chi_b) + k(\chi_b, \chi_b). \end{aligned}$$

We now briefly describe the KPCA method [9]. Let  $\tau = \{\chi_1, \chi_2, \dots, \chi_N\}$  be a set of training data. The centered kernel matrix  $\tilde{\mathbf{K}}$  corresponding to  $\tau$  is defined as

$$\begin{aligned} \tilde{\mathbf{K}} &= ((\varphi(\chi_i) - \bar{\varphi}) \cdot (\varphi(\chi_j) - \bar{\varphi})) \\ &= (\tilde{\varphi}(\chi_i) \cdot \tilde{\varphi}(\chi_j)) = \tilde{k}(\chi_i, \chi_j), \quad \text{for } i \in [1, N], \end{aligned} \quad (2)$$

with  $\bar{\varphi} = \frac{1}{N} \sum_{i=1}^N \varphi(\chi_i)$ ,  $\tilde{\varphi}(\chi_i) = \varphi(\chi_i) - \bar{\varphi}$  being the centered map corresponding to  $\chi_i$  and  $\tilde{k}(\cdot, \cdot)$  denoting the centered kernel function. Since  $\tilde{\mathbf{K}}$  is symmetric, it can be decomposed as

$$\tilde{\mathbf{K}} = \mathbf{U} \mathbf{S} \mathbf{U}^t, \quad (3)$$

where  $\mathbf{S} = \text{diag}(\gamma_1, \dots, \gamma_N)$  is a diagonal matrix containing the eigenvalues of  $\tilde{\mathbf{K}}$ .  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N]$  is an orthonormal matrix. The column vectors  $\mathbf{u}_i = [u_{i1}, \dots, u_{iN}]^t$  are the eigenvectors corresponding to the eigenvalues  $\gamma_i$ s. Besides, it can easily be shown that  $\tilde{\mathbf{K}} = \mathbf{H} \mathbf{K} \mathbf{H}$ , where  $\mathbf{H} = \mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^t$ .  $\mathbf{1} = [1, \dots, 1]^t$  is an  $N \times 1$  vector.

Let  $\mathbf{C}$  denote the covariance matrix of the elements of the training set mapped by  $\tilde{\varphi}$ . Within the KPCA methodology, the covariance matrix  $\mathbf{C}$ , which is possibly of very high dimension, does not need to be computed explicitly. Only  $\tilde{\mathbf{K}}$  needs to be known to extract features from the training set since the eigenvectors of  $\mathbf{C}$  are simple functions of the eigenvectors of  $\tilde{\mathbf{K}}$  [18]. The subspace of the feature space  $F$  spanned by the first  $l$  eigenvectors of  $\mathbf{C}$  will be referred to as the *KPCA space* in what follows: The KPCA space is the subspace of  $F$  obtained from learning the training data.

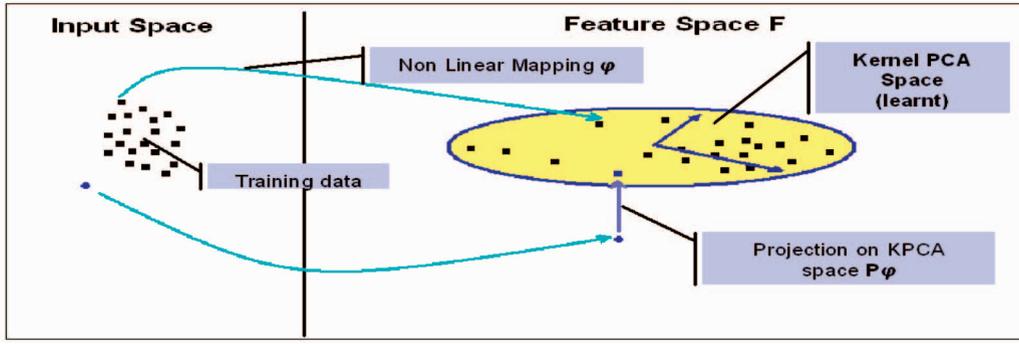


Fig. 1. Kernel PCA methodology. A training set is mapped from input space  $\mathcal{I}$  to feature space  $F$  via a nonlinear function  $\varphi$ . PCA is performed in  $F$  to determine the principal directions defining the kernel PCA space (learned space): oval area. Any element of  $\mathcal{I}$  can then be mapped to  $F$  and projected on the kernel PCA space via  $P^l\varphi$ .

Let  $\chi$  be *any* element of the input space  $\mathcal{I}$ . The projection of  $\chi$  on the KPCA space will be denoted by  $P^l\varphi(\chi)$ .<sup>1</sup> The projection  $P^l\varphi(\chi)$  can be obtained as described in [9]. The squared distance  $d_F^2$  between a test point  $\chi$  mapped by  $\varphi$  and its projection on the KPCA space is given by

$$\begin{aligned} d_F^2[\varphi(\chi), P^l\varphi(\chi)] &= \|\varphi(\chi) - P^l\varphi(\chi)\|^2, \\ &= k(\chi, \chi) - 2(\varphi(\chi) \cdot P^l\varphi(\chi)) + (P^l\varphi(\chi) \cdot P^l\varphi(\chi)). \end{aligned}$$

This distance measures the discrepancy between a (mapped) element of  $\mathcal{I}$  and the elements of the learned space and will be minimized to introduce shape knowledge in the contour evolution process in Section 3.3. Using some matrix manipulations, this squared distance can be expressed only in terms of kernels as

$$\begin{aligned} d_F^2[\varphi(\chi), P^l\varphi(\chi)] &= \\ &k(\chi, \chi) + \frac{1}{N^2} \mathbf{1}^t \mathbf{K} \mathbf{1} - \frac{2}{N} \mathbf{1}^t \mathbf{k}_\chi + \tilde{\mathbf{k}}_\chi^t \mathbf{M} \tilde{\mathbf{K}} \mathbf{M} \tilde{\mathbf{k}}_\chi - 2\tilde{\mathbf{k}}_\chi^t \mathbf{M} \tilde{\mathbf{k}}_\chi, \end{aligned} \quad (4)$$

where  $\mathbf{k}_\chi = [k(\chi, \chi_1) \ k(\chi, \chi_2) \ \dots \ k(\chi, \chi_N)]^t$ ,  $\tilde{\mathbf{k}}_\chi = \mathbf{H}(\mathbf{k}_\chi - \frac{1}{N} \mathbf{K} \mathbf{1})$ , and  $\mathbf{M} = \sum_{i=1}^l \frac{1}{\gamma_i} \mathbf{u}_i \mathbf{u}_i^t$ .

Fig. 1 recapitulates the KPCA methodology, as well as the projection operation on the learned space.

## 3.2 Kernels for Linear and Nonlinear PCA

### 3.2.1 Linear PCA

In [1], a method is presented to learn shape variations by performing PCA on a training set of shapes (closed curves) represented as the zero level sets of SDFs. Using the following kernel in the formulation of the KPCA presented above amounts to performing linear PCA on the given SDFs:

$$k_{id}(\phi_i, \phi_j) = (\phi_i \cdot \phi_j) = \int \int \phi_i(u, v) \phi_j(u, v) du dv \quad (5)$$

for all SDFs  $\phi_i$  and  $\phi_j : \mathbf{R}^2 \rightarrow \mathbf{R}$ . The subscript *id* stands for the identity function: When performing linear PCA, the kernel used is the inner scalar product in the input space; hence, the corresponding mapping function  $\varphi = id$ .

A different representation for shapes is based on the use of binary maps, that is, one sets to 1 the pixels located inside the shape and to 0 the pixels located outside (see Fig. 4).

1. In this notation,  $l$  refers to the first  $l$  eigenvectors of  $\mathbf{C}$  used to build the KPCA space.

One can change the shape representation from SDFs to binary maps using the Heaviside function

$$H(\phi) = H\phi = \begin{cases} 1 & \text{if } \phi \geq 0, \\ 0 & \text{else.} \end{cases}$$

Note that, in this case, the kernel allowing one to perform linear PCA is given by

$$k_{id}(H\phi_i, H\phi_j) = (H\phi_i \cdot H\phi_j). \quad (6)$$

In numerical applications, a smooth version  $H_\epsilon\phi$  of  $H\phi$  can be obtained by taking  $H_\epsilon\phi = \frac{1}{2} + \frac{1}{\pi} \arctan(\frac{\phi}{\epsilon})$  for  $\epsilon$  small. The derivative of  $H$  will be noted  $\delta$  in the rest of this paper. In numerical applications, a smooth version  $\delta_\epsilon\phi$  of  $\delta\phi$  can be obtained by taking  $\delta_\epsilon\phi = \frac{1}{\pi} \frac{1}{1+(\frac{\phi}{\epsilon})^2}$ .

### 3.2.2 Nonlinear PCA

Choosing a nonlinear kernel function  $k(\cdot, \cdot)$  is the basis of nonlinear PCA. The exponential kernel has been a popular choice in the machine learning community and has proven to nicely extract nonlinear structures from data sets; see, for example, [19]. Using SDFs for representing shapes, this kernel is given by

$$k_{\varphi_\sigma}(\phi_i, \phi_j) = e^{-\frac{\|\phi_i - \phi_j\|^2}{2\sigma^2}}, \quad (7)$$

where  $\sigma^2$  is a variance parameter estimated a priori and  $\|\phi_i - \phi_j\|^2$  is the squared  $L_2$ -distance between two SDFs  $\phi_i$  and  $\phi_j$ . The subscript  $\varphi_\sigma$  stands for the nonlinear mapping corresponding to the exponential kernel; this mapping also depends on the choice of  $\sigma$ . If the shapes are represented by binary maps, the corresponding kernel is

$$k_{\varphi_\sigma}(H\phi_i, H\phi_j) = e^{-\frac{\|H\phi_i - H\phi_j\|^2}{2\sigma^2}}. \quad (8)$$

This exponential kernel is one among many possible choices of Mercer kernels. Other kernels may be used to extract other specific features from the training set; see [9].

## 3.3 Shape Prior for GAC

In order to include prior shape knowledge in the GAC framework, we propose using the projection on the KPCA space as a model and minimizing the following energy functional:

$$E_{\text{shape}}^F(\chi) := d_F^2[\varphi(\chi), P^l\varphi(\chi)]. \quad (9)$$

The superscript  $F$  in  $E_{\text{shape}}^F$  denotes the fact that the shape knowledge is expressed as a distance in feature space. A similar idea was presented in [22] for the purpose of pattern recognition and denoising. In (9),  $\chi$  is a test shape represented using either an SDF ( $\chi = \phi$ ) or a binary map ( $\chi = H\phi$ ) and  $\varphi$  refers to either  $id$  (linear PCA) or  $\varphi_\sigma$  (KPCA). Minimizing  $E_{\text{shape}}^F$  as in (1) amounts to driving the test shape  $\chi$  toward the KPCA space computed a priori from a training set of shapes using (3). This contour evolution involving only the minimization of  $E_{\text{shape}}^F$  (no image information) will be referred to as “warping” in what follows.

A number of researchers have proposed minimizing the distance between the current shape and the mean shape obtained from a training set. The assumption is indeed often made that the underlying distribution of familiar “shapes,” in either the input or feature space is Gaussian [1], [7], [8]. Following this assumption, driving the curve toward the mean shape is a sensitive choice. Here, however, we deliberately chose to use the projection of the (mapped) current SDF to drive the evolution because we would like to deal with objects of different geometry in the training set (see, for example, Fig. 4c; a training set of four words was used for the experiments). When dealing with objects of very different shapes, the underlying distribution can be quite non-Gaussian (for example, multimodal).

Thus, the average shape would not be meaningful in this case since it would amount to mixing shapes belonging to different clusters. As a consequence, driving the (mapped) current shape toward its projection on the KPCA space appears to be a more sensible choice for our purposes. In addition, choosing the projection as a model of shape allows for comparing the given learning methods without image information, that is, by warping the same initial contour for each method and comparing the final results in terms of their resemblance to the elements of the training set. The final shape obtained can be interpreted as “the most probable shape of the initial contour given what is known from the data” (see the warping experiments in Section 6.2). If the mean shape is chosen as a shape model, warping would result in the initial contour converging to this mean shape and no comparison would be possible among learning methods. Thus, in this latest case, image information must be included and it is difficult to conclude whether differences between segmentation results are due to differences in the performances of the learning methods or simply to poor balancing between image and shape information.

The gradient of  $E_{\text{shape}}$  can be computed by applying the calculus of variations on (9), using the expression of  $d_F^2(\varphi(\xi), P^l\varphi(\xi))$  in (4). The energy  $E_{\text{shape}}^F(\chi)$  can be minimized as follows:

$$\frac{d\phi}{dt} = -\nabla_\phi E_{\text{shape}}^F = -\nabla_\chi E_{\text{shape}}^F \cdot \frac{d\chi}{d\phi}. \quad (10)$$

The minimization of  $E_{\text{shape}}$  for any arbitrary contour (no image information) results in the deformation of the contour toward a familiar shape (as presented in Section 6.2).

For the exponential kernel involving SDFs and given in (7), the following result is obtained:

$$\nabla_\phi E_{\text{shape}}^F = -\frac{\sum_{i=1}^N g_i(\phi) \cdot k_{\varphi_\sigma}(\phi, \phi_i) \cdot [\phi - \phi_i]}{\sigma^2} \quad (11)$$

with  $[g_1(\phi), \dots, g_N(\phi)] = -\frac{2}{N} \mathbf{1}^t + 2\tilde{\mathbf{k}}_\phi^t \mathbf{M} \tilde{\mathbf{K}} \mathbf{M} \mathbf{H} - 4\tilde{\mathbf{k}}_\phi^t \mathbf{M} \mathbf{H}$  and  $\tilde{\mathbf{k}}_\phi$ ,  $\mathbf{M}$ , and  $\tilde{\mathbf{K}}$  computed for  $k_{\varphi_\sigma}$ .

For the exponential kernel involving binary maps and given in (8), one derives that

$$\nabla_\phi E_{\text{shape}}^F = -\frac{\sum_{i=1}^N g_i(\phi) \cdot k_{\varphi_\sigma}(H\phi, H\phi_i) \cdot \delta\phi \cdot [H\phi - H\phi_i]}{\sigma^2}, \quad (12)$$

where  $\tilde{\mathbf{k}}_\phi$ ,  $\mathbf{M}$ , and  $\tilde{\mathbf{K}}$  are computed for the kernel  $k_{\varphi_\sigma}(H, H)$ .

For the kernel given in (5), corresponding to linear PCA on SDFs, the following result is obtained:

$$\nabla_\phi E_{\text{shape}}^{\text{linear}} = 2\phi + \sum_{i=1}^N g_i(\phi) \cdot \phi_i, \quad (13)$$

where  $\tilde{\mathbf{k}}_\phi$ ,  $\mathbf{M}$ , and  $\tilde{\mathbf{K}}$  are computed for the kernel  $k_{id}$ .

Finally, for the kernel given in (6), corresponding to linear PCA on binary maps, one finds that

$$\nabla_\phi E_{\text{shape}}^{\text{linear}} = 2\delta\phi + \sum_{i=1}^N g_i(H\phi) \cdot H\phi_i \cdot \delta\phi, \quad (14)$$

where  $\tilde{\mathbf{k}}_\phi$ ,  $\mathbf{M}$ , and  $\tilde{\mathbf{K}}$  are computed for the kernel  $k_{id}(H, H)$ .

#### 4 INTENSITY-BASED SEGMENTATION

Different models [23], [24], [25], [26], [27] which incorporate geometric and/or photometric (color, texture, intensity) information have been proposed to perform region-based segmentation using level sets. Other methods such as those in [28], [29] use local information, that is, edges. Most of the region-based models have been inspired by the region competition technique proposed in [30] and usually offer a higher level of robustness to noise than models based on local information. Morel and Solimini’s book [31] is a nice reference on the various variational segmentation methods.

We now formulate our region-based segmentation approach aimed at separating an object from the background in a given image  $I$ . As with most region-based approaches, we assume that the object and background are characterized by statistical properties which are visually consistent and distinct from each other. The main idea behind the proposed method is to build an “image-shape model” (denoted by  $G_{[I, \phi(t)]}$ ) by extracting a binary map from the image  $I$ , based on the estimates of the intensity statistics of the object (and background) available at each step  $t$  of the contour evolution.

Let  $P_{\text{in}}(I(x, y)) = p(I(x, y) | (x, y) \in \text{Inside})$  denote the probability of the image intensity taking the value  $I(x, y)$ , knowing that  $(x, y)$  belongs to the interior of the contour. Similarly, let  $P_{\text{out}}(I(x, y)) = p(I(x, y) | (x, y) \in \text{Outside})$  denote the probability of the image intensity taking the value  $I(x, y)$ , knowing that  $(x, y)$  belongs to the exterior of the contour.<sup>2</sup> Estimates of these probabilities can be computed from the image intensities corresponding to pixels inside and outside the contour at each step  $t$  of the contour evolution. This will be detailed further in Sections 4.1 and 4.2. Given a meaningful initialization of the evolving

2.  $P_{\text{in}}(I(x, y))$  and  $P_{\text{out}}(I(x, y))$  obviously depend on  $t$  since these densities respectively involve the “inside” and “outside” of the evolving contour. However, we do not include this dependence in our notation for simplicity.

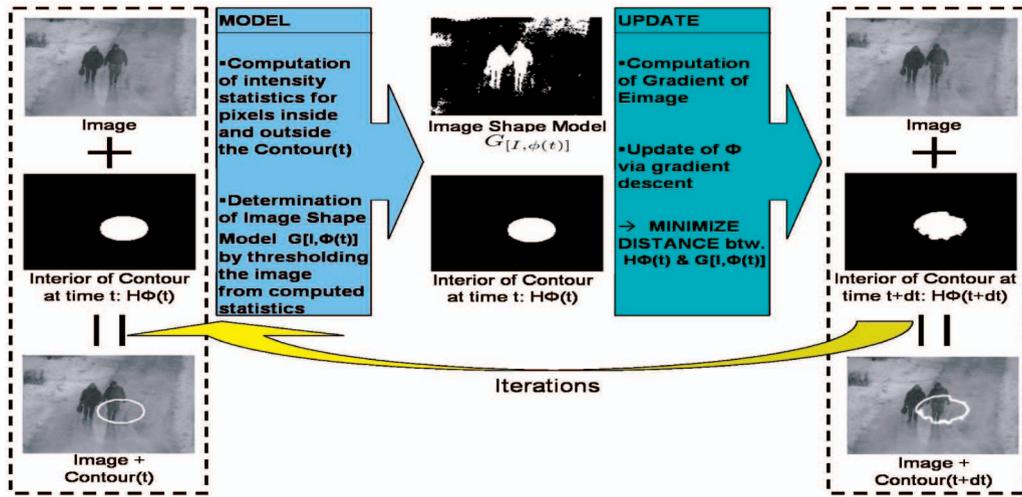


Fig. 2. Model phase: An image-shape model is built from the statistics of the image. Update phase: The contour is evolved to reduce the distance between the shape it underlines and the image-shape model.

curve, these statistics provide valuable information about the statistics of the object and the background. Intuitively, the conditional densities  $P_{in}$  and  $P_{out}$  are, respectively, the best available estimates of the *unknown* conditional densities  $P_O = p((x, y) \in \text{Object} | I(x, y))$  (density of the pixel intensities belonging to the object) and  $P_B = p((x, y) \in \text{Background} | I(x, y))$  (density of the pixel intensities belonging to the background). The image-shape model  $G[I, \phi(t)]$  is built in the following manner:

$$G_{[I, \phi(t)]}(x, y) = \begin{cases} 1 & \text{if } P_{in}(I(x, y)) \geq P_{out}(I(x, y)) \\ 0 & \text{if } P_{in}(I(x, y)) < P_{out}(I(x, y)) \end{cases} \quad (15)$$

Hence, the image-shape model  $G_{[I, \phi(t)]}$  can be interpreted as the most likely shape of the object of interest based on these statistics: Pixels set to 1 in  $G_{[I, \phi(t)]}$  correspond to pixels of the image that are deemed to belong to the object of interest, while pixels set to 0 are believed to not be part of the object.

To perform image segmentation, the contour at time  $t$  is deformed toward this image-shape model by minimizing the following energy:

$$E_{\text{image}}(\phi, I) := \|H\phi - G_{[I, \phi(t)]}\|^2, \quad (16)$$

$$= \int_{\Omega} (H\phi - G_{[I, \phi(t)]})^2 dx dy.$$

This energy amounts to measuring the distance between two binary maps, namely,  $H\phi$  and  $G_{[I, \phi(t)]}$ . This is quite valuable in the present context, where shapes can be represented using binary maps, as in earlier sections. Thus, when the shape energy described in (9) is combined with the image energy defined in (16), all of the elements can be expressed in terms of shapes and meaningful trade-offs can be realized. The gradient of  $E_{\text{image}}$  can be computed, assuming that the image-shape model  $G_{[I, \phi(t)]}$  undergoes little variation during one step of the contour evolution:<sup>3</sup>

3. This amounts to supposing that the variation in the statistics of the pixels inside and outside the contour varies slowly from one step of the contour evolution to the other. Similar hypotheses were used in, for example, [24], where mean intensities were assumed to be constant between two successive evolution steps.

$$\nabla_{\phi} E_{\text{image}} = 2\delta\phi \cdot (H\phi - G_{[I, \phi(t)]}). \quad (17)$$

Here, the gradient “points” in the direction of the image-shape model, driving the contour toward it.

Fig. 2 offers a visualization of the proposed segmentation algorithm: First, the image-shape model is built and, then, the contour is evolved toward the shape model.

#### 4.1 Object and Background with Different Mean Intensities

When the object of interest is consistently darker (or brighter) than the background, a reasonable segmentation may be obtained by considering only the mean intensities of the object and of the background; see [23] and [24]. Such images can be treated within the proposed framework by assuming that the intensities of the object and the background have a Gaussian distribution of different means and identical variances.<sup>4</sup> Following this assumption, at each step  $t$  of the contour evolution,  $P_{in}(I(x, y))$  and  $P_{out}(I(x, y))$  can be approximated as

$$P_{in}(I(x, y)) = \beta e^{-\frac{(I - \mu_{in})^2}{2\sigma_0^2}} \quad \text{and} \quad P_{out}(I(x, y)) = \beta e^{-\frac{(I - \mu_{out})^2}{2\sigma_0^2}}, \quad (18)$$

with

$$\mu_{in} = \frac{\int I(x, y) H\phi dx dy}{\int H\phi dx dy},$$

respectively,

$$\mu_{out} = \frac{\int I(x, y) (1 - H\phi) dx dy}{\int (1 - H\phi) dx dy},$$

denoting the mean intensity of the pixels located inside (respectively, outside) the contour and  $(\sigma_0, \beta) \in \mathbf{R} \times \mathbf{R}$ . The optimal threshold separating the object from the background based on the knowledge of the image statistics at time  $t$  is  $\frac{\mu_{in} + \mu_{out}}{2}$ . Following the definition of  $G_{[I, \phi(t)]}$  in (15), two cases need to be distinguished to compute the image-shape model at time  $t$ :

4. The case involving different variances can also be dealt with using the proposed method, as presented in [14].

$$\text{if } \mu_{in} > \mu_{out}, G_{[I,\phi(t)]}(x, y) = \begin{cases} 1 & I(x, y) \geq \frac{\mu_{in} + \mu_{out}}{2}, \\ 0 & \text{else,} \end{cases}$$

$$\text{if } \mu_{in} \leq \mu_{out}, G_{[I,\phi(t)]}(x, y) = \begin{cases} 1 & I(x, y) \leq \frac{\mu_{in} + \mu_{out}}{2}, \\ 0 & \text{else.} \end{cases}$$

Notice that  $G_{[I,\phi(t)]}$  is the image-shape model (binary map) obtained from thresholding the image intensities so that values closer to  $\mu_{in}$  are classified as object (set to 1) and others are classified as background (set to 0). For numerical experiments, the function  $G_{[I,\phi(t)]}$  is calculated as follows:

$$\text{if } \mu_{in} > \mu_{out},$$

$$G_{[I,\phi(t),\varepsilon]}(x, y) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{I(x,y) - \mu_{in} + \mu_{out}}{\varepsilon}\right)$$

$$\text{else } G_{[I,\phi(t),\varepsilon]}(x, y) = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\frac{I(x,y) - \mu_{in} + \mu_{out}}{\varepsilon}\right),$$

where  $\varepsilon$  is a parameter such that  $G_{[I,\phi(t),\varepsilon]} \rightarrow G_{[I,\phi(t)]}$  as  $\varepsilon \rightarrow 0$ .

## 4.2 Object and Background with Different Global Statistics

When the image statistics of the object of interest (or background) have more complicated distributions than the one alluded to above, a more involved scheme needs to be applied to perform segmentation. A few methods were recently presented aiming at performing segmentation using global considerations about the statistics of both the object and the background in an unsupervised fashion. For instance, in [32], contour evolution is undertaken in order to reduce the mutual information between the object and background statistics, while, in [33], the Bhattacharyya distance between the histograms of the object and background is minimized. Promising segmentation results were obtained on a wide array of objects and backgrounds having diverse intensity distributions.

The image-shape model approach presented above can be generalized to take into account the global statistics of both the image and the background. At each step  $t$  of the contour evolution, the probability density functions  $P_{in}(I(x, y))$  and  $P_{out}(I(x, y))$  can be estimated, using Parzen windows [34], in the following manner:

$$P_{in}(I(x, y)) = \frac{\int K(I(x, y) - I(x', y')) H\phi(x', y') dx' dy'}{\int H\phi(x, y) dx dy} \quad \text{and}$$

$$P_{out}(I(x, y)) = \frac{\int K(I(x, y) - I(x', y')) [1 - H\phi(x', y')] dx' dy'}{\int [1 - H\phi(x, y)] dx dy},$$

where  $K$  can be chosen to be the Gaussian kernel with zero mean and chosen variance.<sup>5</sup>

The image-shape model can be constructed by plugging the estimates of  $P_{in}$  and  $P_{out}$  into (15). In numerical applications, a smoothed version of  $G_{[I,\phi(t)]}$  may be computed (for  $\varepsilon$  small) as

5. In numerical applications,  $P_{in}$  and  $P_{out}$  are normalized to ensure that their respective sum over the span of the intensity values is 1.

$$G_{[I,\phi(t)]}(x, y) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{P_{in}(I(x, y)) - P_{out}(I(x, y))}{\varepsilon}\right).$$

## 5 COMBINING SHAPE PRIOR AND INTENSITY INFORMATION

### 5.1 Balancing Energy Functionals

In this section, we combine shape knowledge obtained by performing nonlinear PCA on binary maps with image information obtained by building an “image-shape model.” Classically, image and shape information are combined by defining a total energy  $E_{total}$ :

$$E_{total}(I, \phi) = E_{shape}^F(\phi) + \beta E_{image}(I, \phi). \quad (19)$$

Using Bayes’ formula, it can be shown that minimizing the above energy functional amounts to maximizing the posterior probability of the curve given current image information and shape knowledge. The coefficient  $\beta$  represents the level of trust given to image information (higher  $\beta$ s result in emphasizing image information). Considering  $-\nabla_{\phi} E_{total}$  in (19), two forces are obtained: One is the image force  $f_{image} = -\beta \nabla_{\phi} E_{image}$ , the other is the shape force  $f_{shape}^F = -\nabla_{\phi} E_{shape}^F$ . The gradient descent process described in (1) converges to a local minimum where  $f_{image}$  and  $f_{shape}^F$  balance each other (same amplitude and opposite directions).

However,  $E_{image}$  is a squared  $L_2$ -distance in the input space, whereas  $E_{shape}^F$  is a squared distance in the feature space. As a consequence, a meaningful balance between  $f_{image}$  and  $f_{shape}^F$  would be hard to reach. The gradient  $\nabla_{\phi} E_{shape}^F$  exhibits a nonlinear behavior due to the exponential terms it involves. To illustrate this point, one can consider a one-dimensional static version of  $E_{image}$  and  $E_{shape}^F$ .<sup>6</sup> Let  $E_{image}^{1d} = (x - x_{image})^2$  be the one-dimensional analogous of  $E_{image}$ , with  $x_{image}$  being the one-dimensional equivalent to the image-shape model, and let  $E_{shape}^{1d} = 2 - 2e^{-\frac{(x - x_{shape})^2}{2\sigma^2}}$  be the one-dimensional analogous of  $E_{shape}^F$ , with  $x_{shape}$  being the one-dimensional equivalent to the shape model obtained from learning.<sup>7</sup> Although this is usually not the case during the actual evolution process, the models  $x_{image}$  and  $x_{shape}$  will be considered constant throughout this discussion for simplicity. The one-dimensional forces  $f_{image}^{1d}$  and  $f_{shape}^{1d}$  corresponding, respectively, to  $E_{image}^{1d}$  and  $E_{shape}^{1d}$  can easily be computed as

6. This discussion involving one-dimensional functions also holds for multiple dimensions since the different forces enjoy radial symmetry.

7. Recall that  $E_{shape}^F$  is a squared distance  $d_{\phi}^2$  in the feature space. We assume here for the purpose of illustration that  $P^l \varphi(x)$  has a preimage  $x_{shape}$  in the input space, that is,  $\varphi(x_{shape}) = P^l \varphi(x)$ . Thus, the one-dimensional energy  $E_{shape}^{1d} = 2 - 2k(x, x_{shape}) = 2 - 2e^{-\frac{(x - x_{shape})^2}{2\sigma^2}}$ . This assumption is not true in general; see further remarks in what follows.

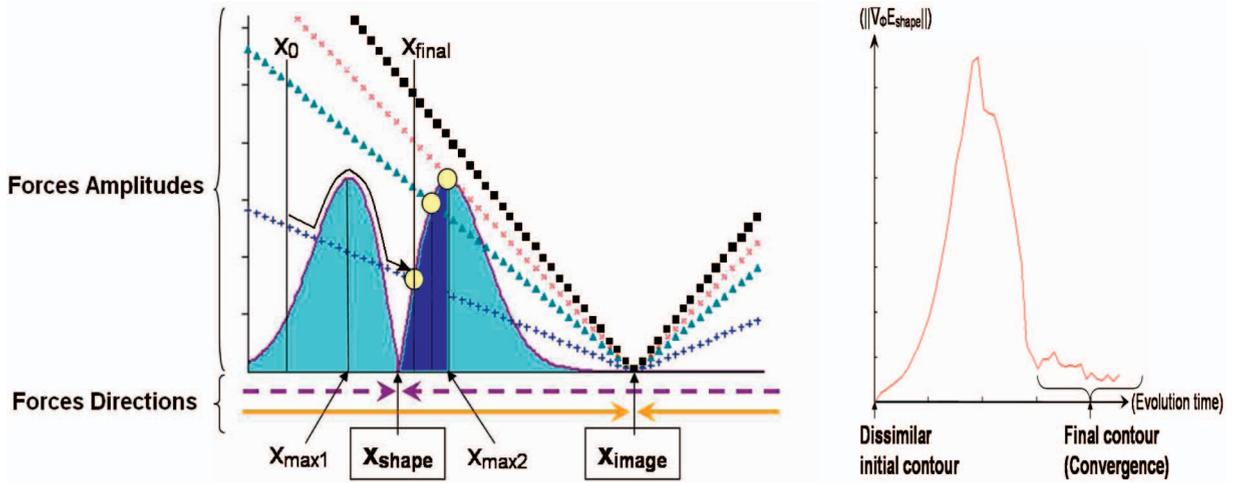


Fig. 3. (a) Theoretical forces corresponding to  $E_{image}$  and  $E_{shape}^F$ . The double-bell-shaped curve represents  $|f_{shape}^{1d}|$ . The two-segment curves represent  $|f_{image}^{1d}|$  for diverse values of  $\beta$  (higher absolute slopes correspond to higher  $\beta$ s). The respective directions of both forces are materialized by the two pairs of opposite horizontal arrows at the bottom of the figure. The darker area under the  $|f_{shape}^{1d}|$  curve represents the locus of possible convergence points that realize a trade-off between  $|f_{image}^{1d}|$  and  $|f_{shape}^{1d}|$ : for the +, ×, and △ lines, the corresponding possible point of convergence is indicated by a circle (no trade-off is realizable between  $|f_{image}^{1d}|$  and  $|f_{shape}^{1d}|$  for the ■ line). (b) Experimental result highlighting the nonlinear behavior of the  $L_2$  norm of  $\nabla_{\phi} E_{shape}^F$ . An initial contour “far away” from the learned shapes was warped (note the similarity between the experimental curve and the theoretical solid-line curve in the leftmost graph).

$$f_{image}^{1d} = -2\beta \cdot (x - x_{image}) \quad \text{and}$$

$$f_{shape}^{1d} = -\frac{2}{\sigma^2} (x - x_{shape}) e^{-\frac{(x-x_{shape})^2}{2\sigma^2}}.$$

The amplitudes and directions of  $f_{image}^{1d}$  and  $f_{shape}^{1d}$  are presented in Fig. 3 for fixed values of  $x_{image}$ ,  $x_{shape}$ , and  $\sigma$ , as well as for different values of  $\beta$ . The amplitude of  $f_{shape}^{1d}$  presents two maxima around  $x_{shape}$ . The values of  $x$ , denoted  $x_{max1}$  and  $x_{max2}$ , for which these maxima occur are  $x_{max1} = x_{shape} - \sigma$  and  $x_{max2} = x_{shape} + \sigma$ . During gradient descent, the variable  $x$  evolves from an initial value of  $x = x_0$  in the direction of the force that has the highest amplitude to converge to the value of  $x = x_{final}$  for which  $f_{image}^{1d}$  and  $f_{shape}^{1d}$  have the same amplitudes and opposite directions. Although the results presented in Fig. 3 are dependent on the particular choice of  $x_{image}$ ,  $x_{shape}$ , and  $\beta$ , valuable insights about the consequences of the nonlinear behavior of  $E_{shape}^F$  may be inferred:

- If  $\beta$  is too large,  $|f_{image}^{1d}|$  (■ line on the graph) is always above  $|f_{shape}^{1d}|$ : The evolution of  $x$  is not influenced by  $f_{shape}^{1d}$  and  $x$  should converge to  $x_{image}$ . Hence, it can be expected that for too large  $\beta$ s, no shape knowledge will contribute at all to the contour evolution and the contour will converge to the image-shape model. This is counterintuitive since, from (19), one expects that both energies contribute to the evolution (for  $\beta \neq 0$ ).
- If the initial value of  $x$  is biased toward  $x_{image}$ , for instance,  $x_0$  located on the right of  $x_{image}$ ,  $f_{shape}^{1d}$  will not contribute to the evolution of  $x$  and convergence again occurs for  $x = x_{image}$ . Thus, it can be expected that the final contour will be very much dependent on initial conditions. For example, if the shape of the initial contour is biased toward the image-shape model the contour could converge toward the image-shape model and shape-knowledge information would not influence the evolution.

- The darker area under the  $|f_{shape}^{1d}|$  curve represents the locus of the possible final values of  $x$  realizing a trade-off between image information and the knowledge of shape. These possible final values are between  $x_{shape}$  and  $x_{max2}$ . Final results for  $x$  will thus be skewed toward the shape-knowledge model and it is not possible to reach any trade-off value located between  $x_{max2}$  and  $x_{image}$  (for any initial value  $x_0$ ). Thus, the parameter  $\sigma$  can be expected to have a limiting influence on the possible trade-offs realizable between image information and shape knowledge, resulting in not meaningfully balanced final contours. Ideally, *only* the parameter  $\beta$  should influence the trade-off between forces, whereas  $\sigma$  should be chosen based on considerations of the training data *only*.

Fig. 3b presents the typical behavior of the  $L_2$  norm of  $f_{shape}^F$  obtained when warping an initial contour very “far from” (that is, dissimilar to) the elements of the training set, until convergence. This empirical result for the norm of  $f_{shape}^F$  exhibits very similar behavior to its one-dimensional theoretical counterpart  $|f_{shape}^{1d}|$ , validating the remarks made above.<sup>8</sup>

Hence, these issues need to be addressed in order to take advantage of the KPCA technique in a robust (for example, less dependent on initial conditions), intuitive, and meaningful manner. We now proceed to explain a methodology to address the aforementioned issues for the exponential kernel (the methodology is general enough to accommodate any invertible kernel of the form  $k(x, y) = k(\|x - y\|)$ ). The nonlinear behavior of  $\nabla E_{shape}^F$  stems from the fact that the energy

8. Also, during our initial experiments using (19) directly, meaningful balances between shape knowledge and image information were difficult to reach (despite investigating wide ranges of  $\beta$ ). Final contours were either biased toward the image-shape model or adopted a familiar shape that did not segment the image in a satisfactory manner. Satisfactory balances were obtained only for large values of  $\sigma$ .

is defined as a distance in the feature space. We would like to redefine the shape energy so that the amplitude of its gradient exhibits a linear behavior (similar to  $\nabla E_{\text{image}}$  for which the energy is defined as a distance in the input space). We note that distances in the feature space and the input space are related as follows, for any SDFs  $\phi_a$  and  $\phi_b$ :

$$\begin{aligned} d_F^2(\phi_a, \phi_b) &= 2 - 2k_{\varphi_\sigma}(H\phi_a, H\phi_b), \\ &= 2 - 2e^{-\frac{\|H\phi_a - H\phi_b\|^2}{2\sigma^2}} = 2 - 2e^{-\frac{d_I^2(H\phi_a, H\phi_b)}{2\sigma^2}}, \end{aligned} \quad (20)$$

where  $d_I^2$  is a squared distance in the input space. Using the invertibility of the kernel in (20), one can write  $d_I^2(H\phi_a, H\phi_b) = -2\sigma^2 \log\left(\frac{2 - d_F^2(\phi_a, \phi_b)}{2}\right)$ . Let us define a new shape energy from  $E_{\text{shape}}^F$ :

$$E_{\text{shape}} = -2\sigma^2 \log\left(\frac{2 - E_{\text{shape}}^F}{2}\right). \quad (21)$$

This energy can be expected to behave in a similar fashion as a squared distance in the input space. The gradient of the energy  $E_{\text{shape}}$  should now exhibit a more stable behavior than  $\nabla E_{\text{shape}}^F$ . By applying the chain rule, one can verify that  $\nabla_\phi E_{\text{shape}} = \frac{2\sigma^2}{2 - E_{\text{shape}}^F} \nabla_\phi E_{\text{shape}}^F$ . Hence,  $\nabla E_{\text{shape}}^F$  and  $\nabla E_{\text{shape}}$  have an identical direction ( $\frac{2\sigma^2}{2 - E_{\text{shape}}^F} \in \mathbf{R}^+$  since  $E_{\text{shape}} \in [0, 2]$ ) and will influence the evolution in similar ways since both gradients point in the direction of the projection in the feature space. However, the extra term in the expression of  $\nabla_\phi E_{\text{shape}}$ , namely,  $\frac{2\sigma^2}{2 - E_{\text{shape}}^F}$ , modifies the amplitude of the gradient and was found to indeed drastically improve balancing performances between image and shape forces (for a rather large range of values of  $\sigma$ ) during our experiments.

It is important to note that, for a certain  $\phi$ ,  $P^l\varphi(\phi)$  does not necessarily have a preimage in the input space  $\mathcal{I}$ : It may not be possible to define  $\phi_P \in \mathcal{I}$  so that  $\phi_P = \varphi^{-1}(P^l\varphi(\phi))$ . Thus, rigorously defining an energy functional as a distance between the contour and a model encoding shape knowledge obtained from KPCA directly in the input space is problematic. A few authors proposed methods to compute an approximate preimage of  $P^l\varphi(\phi)$ . In [19], an iterative scheme is proposed to compute  $\phi_P^{\text{approx}} \simeq \varphi^{-1}(P^l\varphi(\phi))$ , while, in [35], a direct method is proposed. These methods could have been used to define  $E_{\text{shape}}^{\text{approx}} = \|H\phi - H\phi_P^{\text{approx}}\|^2$  directly in  $\mathcal{I}$ . However, defining  $E_{\text{shape}}$  as the scaled log of a distance in the feature space was found to be more stable and computationally efficient while not requiring any approximation of the shape model.

Both the shape and image energies now behave as distances in the input space between shapes that are represented by binary maps. This consistent description of energies allows for meaningful trade-offs between image cues and shape knowledge through the energy functional:

$$E_{\text{total}}(\phi, I) = (1 - \alpha) E_{\text{shape}}(\phi) + \alpha E_{\text{image}}(\phi, I), \quad (22)$$

where  $\alpha$  is a balancing parameter, effectively and intuitively encoding the level of trust accorded to each model (image-shape model or shape-knowledge model). For

example, if the same level of trust should be accorded to each model during segmentation,  $\alpha$  can be chosen to be 0.5.

## 5.2 Pose Invariance

Prior to the construction of the space of shapes, the elements of the training set need to be aligned, e.g., in order to disregard differences due to affine transformations. As a result, the elements of the training set are registered at a certain position in the image domain (usually at the center of the domain). Typically, the object of interest in image  $I$  and the registered training shapes do not have the same pose and differ by some transformation. This transformation needs to be recovered during the segmentation process in order to constrain the shape of the contour properly (see, for example, [1], [2]). In what follows, we detail our approach to dealing with differences in pose within our segmentation framework.

Let us assume that the object of interest in  $I$  differs from the registered elements of the training set by the transformation  $T[\mathbf{p}]$  with parameters  $\mathbf{p} = [p_1, p_2, p_3, p_4] = [t_x, t_y, \theta, \rho]$ , in which  $t_x$  and  $t_y$  correspond to translation according to the  $x$  and  $y$ -axis,  $\theta$  is the rotation angle, and  $\rho$  is the scale parameter. Let us denote by  $\tilde{I}(\tilde{x}, \tilde{y})$  the image obtained by applying the given transformation on  $I(x, y)$ . We have  $\tilde{I}(\tilde{x}, \tilde{y}) = I(x, y)$ , with

$$\begin{aligned} (\tilde{x}, \tilde{y}) &= T[\mathbf{p}](x, y) \\ &= (\rho(x \cos \theta - y \sin \theta) + t_x, \rho(x \sin \theta + y \cos \theta) + t_y). \end{aligned}$$

The transformation  $T[\mathbf{p}]$  may be recovered by minimizing  $E_{\text{total}}(\phi, \tilde{I})$  with respect to the  $p_i$ s via gradient descent. Since  $E_{\text{shape}}(\phi)$  does not depend on  $\mathbf{p}$ , minimizing  $E_{\text{total}}(\phi, \tilde{I})$  is equivalent to minimizing  $E_{\text{image}}(\phi, \tilde{I})$  only with respect to the  $p_i$ s. In particular, we have  $\nabla_{p_i} E_{\text{total}}(\phi, \tilde{I}) = \nabla_{p_i} E_{\text{image}}(\phi, \tilde{I})$  for  $i = 1, \dots, 4$ . Assuming that the image-shape model  $G_{[\tilde{I}, \phi(t)]}$  is negligibly modified by an infinitesimal transformation of  $\tilde{I}$ ,<sup>9</sup> one can compute  $\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I})$  by considering that  $E_{\text{image}}(\phi, \tilde{I})$  is a distance between two determined binary maps, namely,  $H\phi(x, y)$  and  $G_{[\tilde{I}, \phi(t)]}(\tilde{x}, \tilde{y}) = G_{[\tilde{I}, \phi(t)]}(T[\mathbf{p}](x, y))$ . One can thus write

$$\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I}) = 2 \int_{\Omega} (H\phi - G_{[\tilde{I}, \phi(t)]}) \nabla_{p_i} G_{[\tilde{I}, \phi(t)]} dx dy, \quad (23)$$

where

$$\begin{aligned} \nabla_{p_i} G_{[\tilde{I}, \phi(t)]} &= \\ &= \left[ \frac{\partial G_{[\tilde{I}, \phi(t)]}(\tilde{x}, \tilde{y})}{\partial \tilde{x}} \quad \frac{\partial G_{[\tilde{I}, \phi(t)]}(\tilde{x}, \tilde{y})}{\partial \tilde{y}} \right] \left( \frac{\partial T[\mathbf{p}](x, y)}{\partial p_i} \right)^t. \end{aligned} \quad (24)$$

To perform segmentation of images with shape priors, the following gradient descent scheme is undertaken in parallel to the contour evolution in (1) for  $i = 1, \dots, 4$ :

$$\frac{dp_i}{dt} = -\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I}). \quad (25)$$

To initiate the segmentation process, an estimate  $\mathbf{p}_0$  of  $\mathbf{p}$  needs to be computed (the initial curve is drawn around the

9. This amounts to assuming that the pixel statistics inside and outside the curve (computed for  $\tilde{I}$ ) are constant for an infinitesimal transformation of  $\tilde{I}$ .

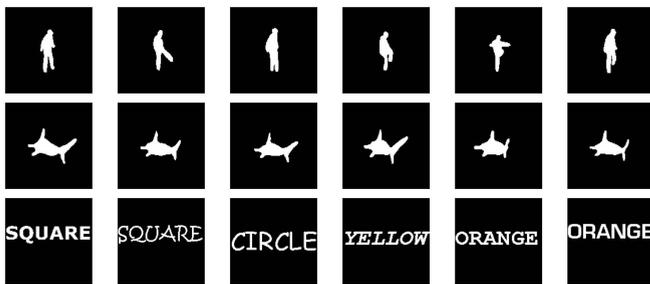


Fig. 4. Three training sets (before alignment—binary images are presented here). First row: “Soccer Player” silhouettes (6 of the 22 used). Second row: “Shark” silhouettes (6 of the 15 used). Third row: “4 Words” (6 of the 80 learned).

object in  $\tilde{I}$ ) so that the pose of the object of interest in  $\tilde{I}$  roughly matches the pose of one of the registered training shapes. Recovering the transformation  $T[\mathbf{p}]$  by working with the image  $I$  only instead of transforming each element of the training set is different with previously proposed approaches (for example, [8], [2]) and is computationally very advantageous when dealing with large training sets.

## 6 EXPERIMENTS

In this section, we present experimental results that compare linear PCA and KPCA as means to introduce shape priors and we test our segmentation framework. In Section 6.1, we describe the training sets used in the experiments. In Section 6.2, we deform an initial shape using only shape knowledge and compare the performances obtained using KPCA and linear PCA in various settings. In Section 6.3, we test our region-based approach to intensity segmentation (as presented in Section 4) on artificial and real-world images. Finally, in Section 6.4, we test our complete segmentation framework ( $E_{\text{total}}$  is minimized, and the transformation  $T[\mathbf{p}]$  is recovered) on challenging images and video sequences.

### 6.1 Training Sets and Learning

In the experiments presented below, three training sets of shapes were used. Shapes were represented by either signed distance functions or (smoothed) binary maps. The first training set of shapes consists of 22 shapes of a man playing soccer. The second training set is composed of 28 shapes of a shark. These shapes were aligned using an appropriate registration scheme [2] to remove differences between them due to translation, rotation, and scale. In order to test for the ability of the proposed framework to learn and deal with multimodal distributions, a third training set was built. This training set consists of four words, “orange,” “yellow,” “square,” and “circle,” each written using 20 different fonts, leading to a training set of 80 shapes in total. The size of the fonts was chosen to lead to words of roughly the same length. The obtained words were then registered according to their centroid. No further effort such as matching the letters of the different words was pursued. The binary maps corresponding to the diverse training shapes are presented in Fig. 4. The shapes depicted are from the original training sets, before alignment. The first row in Fig. 4 presents a few elements of the “Soccer

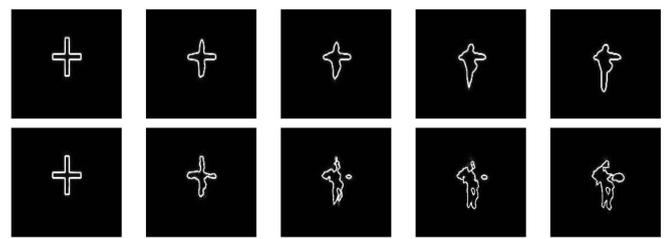


Fig. 5. Warping results obtained for the “Soccer Player” training set, starting with an initial contour representing a “plus” sign. First row: Evolution obtained using kernel PCA. Second row: Evolution obtained using linear PCA. When kernel PCA is used, the final contour resembles the elements of the training set. In contrast, the final contour obtained using linear PCA bears little resemblance to the learned shapes.

Player” training set and the second row of Fig. 4 presents a few shapes from the “Shark” training set. The third row in Fig. 4 presents a few of the words used to build the “4 Words” training set. Shape learning was performed on each training set, as presented in Section 3. The familiar spaces of shapes (KPCA spaces) were built for each of the kernels presented in (5) to (8), whether linear PCA or KPCA was respectively performed, on binary maps or SDFs.

### 6.2 Warping Results: Linear PCA versus KPCA for Shape Priors (No Image Information)

In this section, we compare the performances of KPCA to linear PCA as methods for introducing shape priors in the GAC framework. The fact that energy functional  $E_{\text{shape}}$  is consistently defined for both shape learning methods in (9) ensures that the performances obtained from applying linear PCA or KPCA can be accurately compared. No image information was used in these experiments; instead, the contour evolution was carried out to minimize  $E_{\text{shape}}$  only (“warping”). This further guarantees a meaningful comparison among methods since performances do not stem from the balancing factor between image information and shape knowledge. In what follows, the same initial shapes were warped and the final contours obtained were compared for both methods in terms of their resemblance to the elements of the training sets utilized. Equation (1) was run until convergence, using the expression of the gradients presented in (13) and (14) for linear PCA and (11) and (12) for KPCA.

#### 6.2.1 Warping an Arbitrary Initial Shape

Fig. 5 presents the warping results obtained starting from an arbitrary initial shape representing a “plus” sign. The first row of Fig. 5 shows the results obtained using kernel PCA (with SDFs as a representation of shape). The second row of Fig. 5 presents the results obtained using linear PCA (with shapes represented as SDFs). As can be noticed, the results obtained with linear PCA bear little resemblance to the elements of the training sets. In contrast, the final contours obtained by employing KPCA are more faithful to the learned shapes.

Fig. 6b shows the warping results obtained by applying linear PCA on SDF. Fig. 6d shows the warping results obtained by applying linear PCA on binary maps. Note that the results obtained for the SDF representation bear little resemblance to the elements of the training sets. The results obtained for binary maps are more faithful to the learned

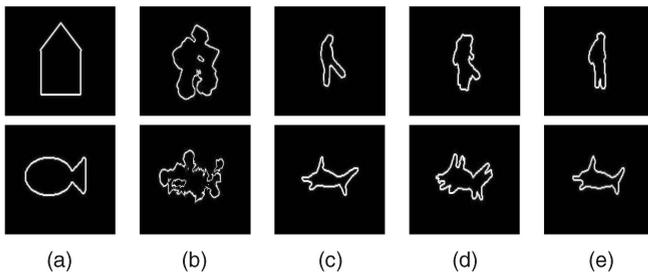


Fig. 6. Warping results of an arbitrary shape, obtained using linear PCA and kernel PCA applied on both SDFs and binary maps. First row: Results for the “Soccer Player” training set. Second row: Results for the “Shark” training set. (a) Initial shape. (b) PCA on SDF. (c) Kernel PCA on SDF. (d) PCA on binary maps. (e) Kernel PCA on binary maps.

shapes (see [36] for a more in-depth study of this fact). Figs. 6c and 6e presents the warping results obtained by applying kernel PCA on SDF and binary maps, respectively. In both cases, the final contour is very close to the training set and the results are better than any of the results obtained with linear PCA.

### 6.2.2 Robustness to Misalignment

In these experiments, the robustness of each method (linear and kernel PCA) to misalignments relative to the registered shapes of a training set is tested. This robustness to misalignment is an interesting property to evaluate since to perform segmentation, differences in pose (for example, translations) between the object of interest in the image and the registered training shapes need to be accounted for. Hence, whether a probabilistic method [1] or a gradient descent scheme [2] is used to compensate for these differences, more or less important misalignments between the contour and the registered training shapes may occur during the evolution process. This can impair the ability of the shape energy to properly constrain the shape of the segmenting contour.

In each of the following three experiments, an initial contour representing a chosen training shape for which pose differs from the registered training shapes is used as the initial contour for warping (SDFs were used to represent shapes). For the first initial contour, a small translation of about 5 pixels in the  $x$  and  $y$ -directions was performed. The second initial contour was a translation of about 10 pixels in the  $x$  and  $y$ -directions. Figs. 7 and 8 present the warping results obtained for the smaller and larger misaligned initial shapes, respectively. The first lines of Figs. 7 and 8 show the results obtained using KPCA; the second lines of Figs. 7 and 8 present the results using linear PCA. Once again, the results obtained with linear PCA bear little resemblance to the elements of the training sets. In contrast, the final contours obtained using kernel PCA are more faithful to the learned shapes. Besides, whether the initial contour is slightly or heavily misaligned, the contour evolution results in a translation of the contour when kernel PCA is used. Interestingly, the shape of the final contour obtained using KPCA is very much like the shape of the initial contour for both experiments. In Fig. 9, the initial contour represents one of the learned shapes scaled up by 10 percent and rotated by 10 degrees. As a result of warping using linear

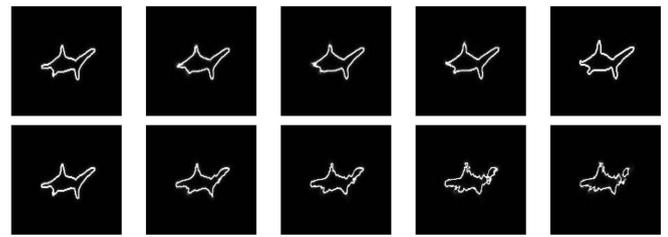


Fig. 7. Warping results obtained for the “Shark” training set. The initial contour (leftmost image) represents one of the learned shapes slightly misaligned (5 pixels) with the corresponding element in the registered training set. First line: Evolution obtained using kernel PCA. Second line: Evolution obtained using linear PCA. The final contour obtained using kernel PCA resembles the elements of the training set and is positioned where the learned shapes were registered. The final contour obtained using linear PCA bears little resemblance to the learned shapes.

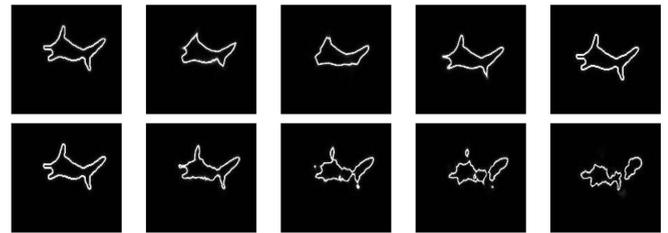


Fig. 8. Warping results obtained for the “Shark” training set. The initial contour (leftmost image) represents one of the learned shapes misaligned by approximately 10 pixels. First line: Evolution obtained using kernel PCA. Second line: Evolution obtained using linear PCA.

PCA, an unrealistic shape is obtained. In contrast, the final contour obtained when using kernel PCA is a realistic shape that is, in effect, rotated back and scaled down compared with the initial contour. Similar satisfying results were obtained with KPCA using binary maps as a representation of shapes. Hence, KPCA appears to offer a higher level of robustness than linear PCA to the misalignment of the initial contour (at least in the experiments we performed).

### 6.2.3 Multimodal Learning

Kernel methods have been used to learn complex multimodal distributions in an unsupervised fashion [9]. The goal of this section is to investigate the ability of kernel PCA to simultaneously learn objects of different shapes and to constrain the contour evolution in a meaningful fashion. Besides, we want to contrast the performances obtained with KPCA to the performances obtained with linear PCA. The “4 Words” training set was used for these experiments. Diverse contours, whose shapes bore some degree of

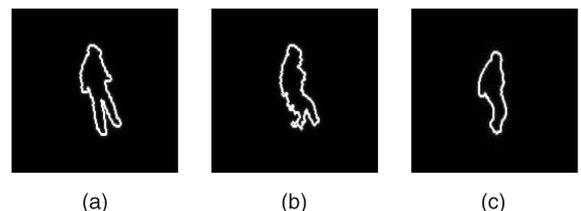


Fig. 9. Warping results obtained for the “Soccer player” training set. (a) Initial contour representing one of the learned shapes scaled up by 10 percent and rotated by 10 degrees. (b) Final result using linear PCA (an unrealistic shape is obtained). (c) Final result using KPCA (the final contour is a realistic shape, effectively rotated and scaled down).

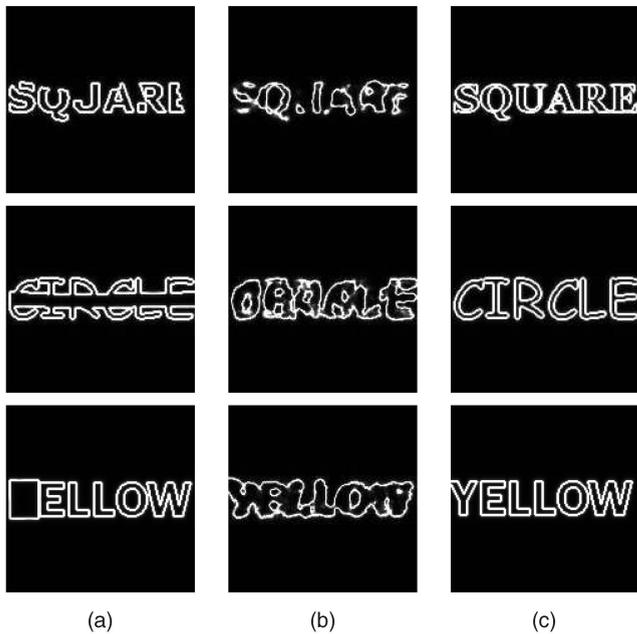


Fig. 10. Warping results for the “4 Words” training set. (a) Initial contours. (b) Warping using linear PCA. (c) Warping using kernel PCA.

resemblance to any of the four words (“orange,” “yellow,” “square,” or “circle”), were then used as initial contours for warping.

Each line in Fig. 10 presents the warping results obtained for the diverse initial contours used, for both linear and kernel PCA, using SDFs as representations of shape. The initial contour presented on the first line in Fig. 10a is the word “square” in which the letters are partially erased. The warping result using linear PCA bears little resemblance to the word “square.” In contrast, using kernel PCA, the word square is accurately reconstructed. In particular, a police close to the original one used in the initial contour is obtained. The initial contour presented on the second line in Fig. 10a is the word “circle” occluded by a line. Again, the warping result using linear PCA bears little resemblance to the word “circle” (a mixing of the different training words is obtained). Using KPCA, not only is the word “circle” accurately reconstructed, but the line is also completely removed. Besides, the original scheme used for the initial contour (which belongs to the training set) is preserved. The initial contour presented on the third line in Fig. 10a is the registered training word “yellow” slightly translated. The letter “y” is also replaced by a rectangle. Employing kernel PCA, the word “yellow” is perfectly reconstructed: The letter “y” is recovered, the contour is translated back, and the original police used for the initial contour is preserved. In comparison, the word “yellow” is barely recognizable from the final contour obtained using linear PCA.

In each of the experiments above, the accurate word is detected and reconstructed using kernel PCA. Similar robust performances were obtained with kernel PCA on binary maps. The shape of the final contour obtained with linear PCA was oftentimes the result of a mixing between words of different classes. This mixing between classes can lead to unrealistic shapes. Thus, KPCA appears to be a strong method for introducing shape priors within the GAC

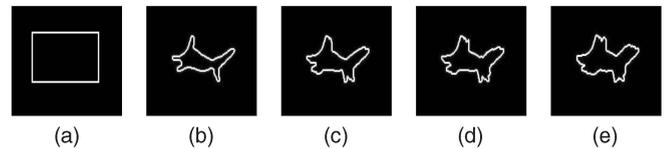


Fig. 11. Influence of  $\sigma$  for the kernel PCA method (exponential kernel) on binary maps. Warping results of an arbitrary shape are presented for the “Shark” training set. (a) Initial shape. (b) Warping result for  $\sigma = 3$ , (c)  $\sigma = 7$ , (d)  $\sigma = 9$ , (e)  $\sigma = 15$ .

framework when training sets involving different types of shapes are used.

#### 6.2.4 Influence of the Parameter $\sigma$ in Exponential Kernels

The goal of this section is to study the influence of the parameter  $\sigma$  when kernel PCA is performed using exponential kernels. Fig. 11 presents warping results of an arbitrary shape using different values of the parameter  $\sigma$  for the kernel defined in (8), which corresponds to performing kernel PCA on binary maps. The Shark training set was used for the shape learning.

As can be seen in Fig. 11, as the value of  $\sigma$  increases, more and more mixing among the shapes of the training set is allowed. Such shape mixing naturally occurs when linear PCA is used for learning (for example, in Fig. 6d, second row, this type of mixing can be observed). Similar results were obtained for the SDF representation and the kernel given in (7). Hence, the parameter  $\sigma$  allows for *controlling* the degree of “mixing” allowed among the learned shapes in the shape prior, which is another advantage of kernel PCA over linear PCA. The choice of  $\sigma$  should typically depend upon how much shape variation occurs within the data set. To emphasize this, larger  $\sigma$ 's allow for more mixing among the shapes, whereas smaller  $\sigma$ 's lead to more “selective” shape priors.

### 6.3 Segmentation Results: Image-Shape Model Only (No Shape Prior)

This section presents segmentation results of real and synthetic images obtained from using the image-shape segmentation technique presented in Section 4. No shape information was used to perform these segmentations. To ensure the smoothness of the contour, a classical curvature term which penalizes longer contours was introduced. Equation (1) was run until convergence using the expression of  $\nabla_{\phi} E_{\text{image}}$  as given in (17).

Fig. 12 presents results obtained for each of the image-shape models (involving different intensity statistics) presented above. Fig. 12a represents a couple (darker intensities) on a lighter background. At each iteration of the contour evolution, an image-shape model was built, using the mean intensities inside and outside the curve. Fig. 12b is a synthetic image representing a square. The mean intensities inside and outside the square are the same; only the variances differ. Fig. 12c represents a zebra on a savannah plain. For both images, image-shape models were built from the probability distributions of the intensities inside and outside the curve at each evolution step. Satisfying segmentation results were obtained in each case,

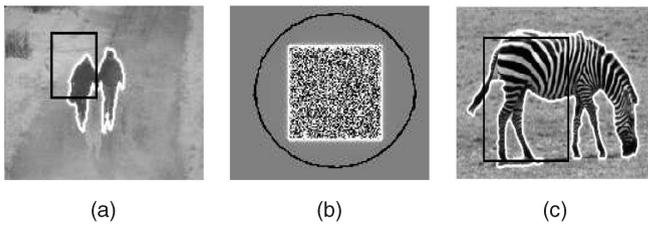


Fig. 12. Segmentation results obtained using  $E_{\text{image}}$  from (16). The initial contour is in black and the final contour is in white. (a) Couple (dark) on lighter background, segmented using first moment only. (b) Two regions of same mean intensity and different variances segmented using global intensity distributions. (c) Zebra segmented using global intensity distributions.

resulting in the accurate separation of the object of interest from the background.

#### 6.4 Segmentation Results: Image-Shape Model + Shape Prior

This section presents segmentation results obtained by introducing shape prior using KPCA on binary maps and using our intensity-based segmentation methodology as presented in Section 5. For all of the following experiments, (1) was run until convergence on diverse images, using (22) as an energy functional (the parameter  $\alpha$  was chosen to be  $\alpha = 50\%$ ). In parallel, the transformation  $T[\mathbf{p}]$  between the object of interest and the registered training set was recovered. No curvature term was used for contour regularization since smoothness information is indeed already present in the binary maps used to build shape priors.

##### 6.4.1 Toy Example: Shape Priors Involving Objects of Different Types

The goal of this section is to investigate the ability of the algorithm described in Section 5 to accurately detect and segment objects of different shapes. For these experiments, the “4 Words” training set was used. We tested our algorithm on images where a corrupted version of the word “orange,” “yellow,” “square,” or “circle” was present (Fig. 13, first row). Word recognition is a challenging task and addressing it using GACs may not be a panacea. However, the ability of the level set representation to naturally handle topological changes was found to be useful for this purpose. During evolution, the contour split and merged a certain number of times to segment the disconnected letters of the words. In all of the following experiments, the *same* initial contour was used.

**Experiment 1.** In this experiment, the word “square” was corrupted: The letter “u” was almost completely erased. The shape thus obtained was filled with Gaussian noise of mean  $\mu_o = 0.5$  and variance  $\sigma_o = 0.05$ . The background was also filled with Gaussian noise of the same mean  $\mu_b = 0.5$  but of variance  $\sigma_b = 0.2$ . The result of applying our method is presented Fig. 13a. The image-shape model was built using the method involving global statistics described in Section 4.2. Despite the noise and the partial deletion, a convincing segmentation and a smooth contour are obtained. In particular, the correct font is detected and the letter “u” is accurately reconstructed.

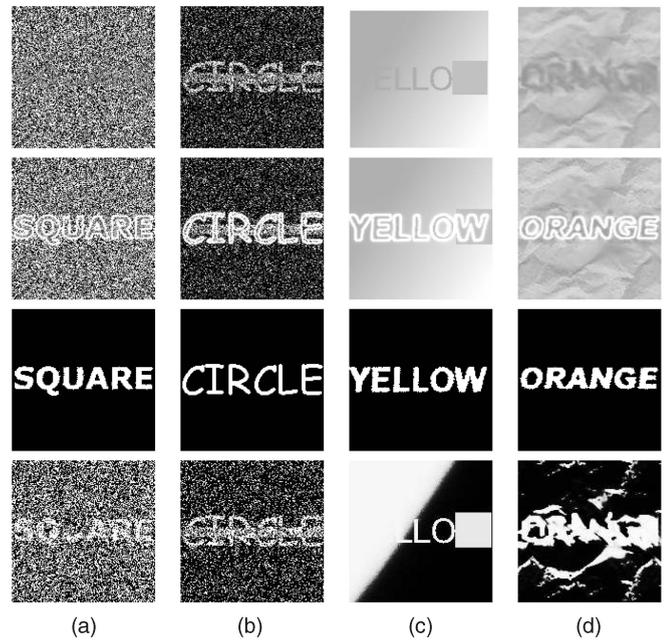


Fig. 13. Segmentation results for the “4 Words” training set. Shape priors were built by applying the KPCA on binary maps. First row: Original images to segment. Second row: Segmentation results. Third row: Shape underlined by the final contour ( $H\phi$ ). Fourth row: Image-shape model ( $G_{[L, \phi(t)]}$ ) obtained when computing  $E_{\text{image}}$  for the final contour.

**Experiment 2.** In this second experiment, one of the elements of the training set was used. A thick line (occlusion) was drawn on the word and a fair amount of Gaussian noise was added to the resulting image. The result of applying our method is presented in Fig. 13b. Despite the noise and the occlusion a very convincing segmentation is obtained. In particular, the correct font is detected and the thick line is completely removed. Once again, the final contour is smooth despite the fairly large amount of noise.

**Experiment 3.** In this third test, the word “yellow” was written using a different font from the ones used to build the training set. In addition, “linear shadowing” was used in the background, making the first letter “y” completely hidden. The letter “w” was also replaced by a gray square. The result of applying our framework is presented in Fig. 13c. The word “yellow” is correctly recognized and segmented. In particular, the letters “y” and “w” were completely reconstructed.

**Experiment 4.** In this experiment, the word “orange” was handwritten in capital letters roughly matching the size of the letters of the words in the training set. The intensity of the letters was chosen to be rather close to some parts of the background. In addition, the word was blurred and smeared in a way that made its letters barely recognizable. This type of blurring effect is often observed in medical images due to patient motion. This image is particularly difficult to segment, even when using shape prior, since the spacing between letters and the letters themselves are very irregular due to the combined effects of handwriting and blurring. Hence, mixing among classes (confusion between any of the four

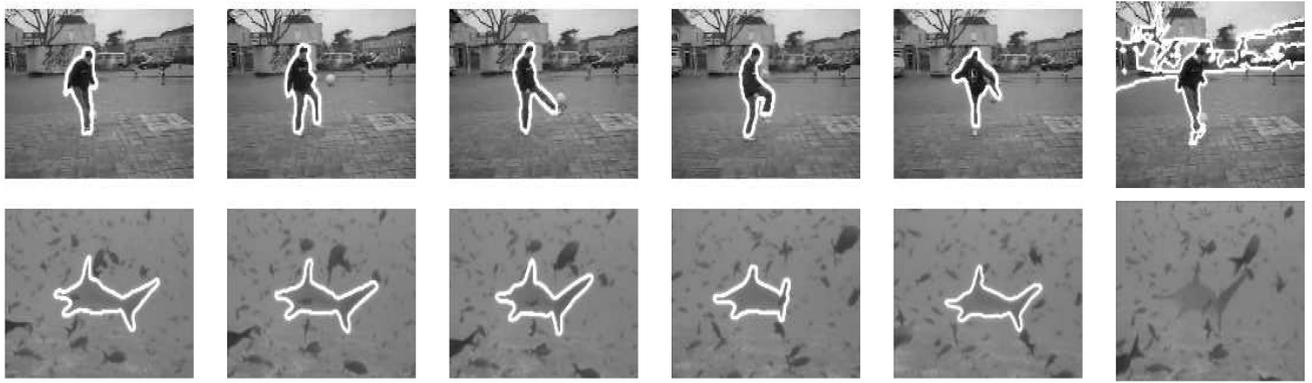


Fig. 14. Tracking results with the proposed method. First row: Soccer Player sequence (the rightmost frame is the result obtained without shape prior;  $\alpha = 1$  in (22)). Second row: “Shark” sequence (the rightmost frame is an original image from the sequence, reproduced here to assess the poor level of contrast).

words) can be expected in the final result. In the final result obtained, the word “orange” is not only recognized but satisfyingly recovered; in particular, a thick font was obtained to model the thick letters of the word (Fig. 13d).

Hence, starting from the same initial contour in each experiment, our algorithm was able to accurately detect which word was present in the image. This highlights the ability of our method not only to gather image information throughout evolution but also to distinguish between objects of different classes (“orange,” “yellow,” “square,” and “circle”). Comparing the final contours obtained in each experiment to the final image-shape model  $G_{[I, \phi(t)]}$  (last row in Fig. 13), one can measure the effect of our shape prior model in constraining the contour evolution. The image information alone would lead to a shape bearing very little resemblance to any of the learned words.

## 6.5 Real-Image Examples: Tracking Temporal Sequences

To test the robustness of the framework, tracking was performed on two sequences. A very simple tracking scheme was used with the *same* initial contour being used for each image in the sequence. This contour was initially positioned wherever the final contour was in the preceding image. Of course, many efficient tracking algorithms have already been proposed. However, convincing results were obtained here, without considering the system dynamics, for instance. This highlights the efficiency of including prior knowledge on shape for the robust tracking of deformable objects. The parameter  $\alpha$  was *fixed* to  $\alpha = 55$  percent in (22) to make up for the influence of clutter in the images. The nice tracking performance observed, despite the fact that parameters are fixed throughout the sequence, further highlights the robustness of the proposed segmentation framework.

### 6.5.1 Soccer Player Sequence

In this sequence (composed of 130 images), a man is playing with a soccer ball. The challenge is to accurately capture the large deformations due to the movement of the person (for example, limbs undergo large changes in aspect) while

sufficiently constraining the contour to discard clutter in the background. The training set of 22 silhouettes, presented in the first row of Fig. 14, was used. The version of  $E_{\text{image}}$  involving the intensity means only was used to capture image information. Despite the small number of shapes used for training and the initialization of the contour evolution with the same (arbitrary) contour, successful tracking was obtained, correctly capturing the very diverse postures of the player.

### 6.5.2 Shark Video

In this sequence (composed of 70 images), a shark is evolving in a highly cluttered environment. Note that the shark is at times partially occluded by other fish and has poorly contrast. To perform tracking, 15 shapes were extracted from the first half of the video (second row of Fig. 14) and used to build the shape prior. The version of  $E_{\text{image}}$  involving global statistics was used to make up for the poor contrast of the shark in the images. Once again, despite the small training set, successful tracking performances were observed. The shark was correctly captured, while clutter and occlusions were rejected.

## 7 CONCLUSIONS

In this paper, we used an unsupervised learning technique to introduce shape priors in the GAC framework. Our approach uses KPCA, a technique borrowed from the machine learning community. We employed the projections on the learned spaces of shapes (using either linear PCA or kernel PCA) as models. The projections on the learned spaces appear to be more representative than the commonly used average of the (mapped) data when dealing with multimodal distributions. Further, using such models allows for comparing learning techniques in a rigorous manner. For each learning technique, the same initial contour can be deformed (using the projection on the corresponding learned space as a model) and the final contours thus obtained can be compared in terms of their similarity with the elements of the training sets. This approach allows for investigating the performance of each learning method without using any image information. Using this technique, better performance of kernel PCA as

compared to linear PCA was demonstrated for two representations of shapes (binary maps and SDFs).

In addition, we proposed a region-based approach aimed at separating an object from the background. This method consists of extracting an image-shape model from the image to direct the evolution of the contour. The technique can successfully deal with diverse image intensity statistics. In the present work, we also highlighted the importance of expressing image and shape energies in a consistent manner. This is especially important when dealing with nonlinear techniques such as KPCA. Valuable insights about the pitfalls of balancing forces extracted directly from the feature space with "linear" (image) forces were obtained from the study of one-dimensional approximations. A method for addressing some of the potential problems was also proposed which drastically improved balancing performances. In the proposed algorithm, image information and shape knowledge are combined in a consistent fashion, both energies are expressed in terms of shapes, and meaningful balances between image information and shape knowledge are realized.

Finally, very reasonable segmentation performances were obtained on complex images, highlighting the power and accuracy of the proposed framework. The method not only allowed us to simultaneously learn shapes of different objects but was also robust to noise, blurring, occlusion, and clutter.

In this paper, the image-shape model  $G_{[I,\phi]}$  was considered to be constant between two evolution steps. Interesting additional segmentation properties can be observed by also considering the variation of  $G_{[I,\phi]}$  in the computation of  $\nabla_{\phi} E_{\text{image}}$ ; these properties will be studied in detail in our future work. In addition, we plan to investigate the performance of other kernels to introduce shape priors within the GAC framework. Our goal is to apply the proposed framework on 3D medical images. The precision and robustness afforded by the proposed technique will hopefully be a valuable asset for automated medical segmentation.

## ACKNOWLEDGMENTS

Allen Tannenbaum is with both the Georgia Institute of Technology and the Technion, Israel, where he is supported by a Marie Curie Grant through the European Union. This work was supported in part by grants from the US National Science Foundation, US Air Force Office of Scientific Research, US Army Research Office, MURI, and MRIHEL, as well as by a grant with the US National Institutes of Health (NIH) (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the NIH through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>.

## REFERENCES

- [1] M. Leventon, E. Grimson, and O. Faugeras, "Statistical Shape Influence in Geodesic Active Contours," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1316-1324, 2000.
- [2] A. Tsai, T. Yezzi, and W. Wells, "A Shape-Based Approach to the Segmentation of Medical Imagery Using Level Sets," *IEEE Trans. Medical Imaging*, vol. 22, no. 2, pp. 137-153, 2003.
- [3] A. Yezzi, S. Kichenassamy, and A. Kumar, "A Geometric Snake Model for Segmentation of Medical Imagery," *IEEE Trans. Medical Imaging*, vol. 16, pp. 199-209, 1997.
- [4] A. Yezzi and S. Soatto, "Deformation: Deforming Motion, Shape Average and the Joint Registration and Approximation of Structures in Images," *Int'l J. Computer Vision*, vol. 53, pp. 153-167, 2003.
- [5] T. Cootes, C. Taylor, and D. Cooper, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, 1995.
- [6] Y. Wang and L. Staib, "Boundary Finding with Correspondence Using Statistical Shape Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 338-345, 1998.
- [7] D. Cremers, T. Kohlberger, and C. Schnoerr, "Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional," *Int'l J. Computer Vision*, vol. 50, pp. 295-313, Feb. 2002.
- [8] D. Cremers, T. Kohlberger, and C. Schnoerr, "Shape Statistics in Kernel Space for Variational Image Segmentation," *Pattern Recognition*, vol. 36, pp. 1292-1943, 2003.
- [9] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and De-Noising in Feature Spaces," *Advances in Neural Information Processing Systems*, vol. 11, MIT Press, 1999.
- [10] *Geometric Partial Differential Equations and Image Analysis*, G. Sapiro, ed. Cambridge Univ. Press, 2001.
- [11] S.J. Osher and J.A. Sethian, "Fronts Propagation with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics*, vol. 79, pp. 12-49, 1988.
- [12] M. Rousson and N. Paragios, "Shape Priors for Level Set Representations," *Proc. Seventh European Conf. Computer Vision*, pp. 78-92, 2002.
- [13] D. Cremers, S. Osher, and S. Soatto, "Kernel Density Estimation and Intrinsic Alignment for Knowledge-Driven Segmentation: Teaching Level Sets to Walk," *Proc. 26th DAGM Pattern Recognition Symp.*, pp. 36-44, 2004.
- [14] S. Dambreville, Y. Rathi, and A. Tannenbaum, "Shape-Based Approach to Robust Image Segmentation Using Kernel PCA," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 977-984, 2006.
- [15] J.A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge Univ. Press, 1999.
- [16] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [17] N. Paragios, Y. Chen, and O. Faugeras, *Handbook of Mathematical Models in Computer Vision*. Springer, 2005.
- [18] B. Schölkopf, S. Mika, and K. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.
- [19] J. Kwok and I. Tsang, "The Pre-Image Problem in Kernel Methods," *IEEE Trans. Neural Networks*, vol. 15, pp. 1517-1525.
- [20] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Comparative Analysis of Kernel Methods for Statistical Shape Learning," *Proc. Second Int'l Workshop Computer Vision Approaches to Medical Image Analysis*, pp. 96-107, 2006.
- [21] J. Mercer, "Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations," *Philosophical Trans. Royal Soc. of London*, vol. 209, pp. 415-446, 1909.
- [22] B. Schölkopf, S. Mika, and K. Müller, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks*, vol. 12, pp. 181-201, 2001.
- [23] A. Yezzi, A. Tsai, and A. Willsky, "A Statistical Approach to Snakes for Bimodal and Trimodal Imagery," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 898-903, 1999.
- [24] T. Chan and L. Vese, "Active Contours without Edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266-277, 2001.
- [25] M. Rousson and R. Deriche, "A Variational Framework for Active and Adaptive Segmentation of Vector Valued Images," *Proc. IEEE Workshop Motion and Video Computing*, p. 56, 2002.
- [26] N. Paragios and R. Deriche, "Geodesic Active Regions for Supervised Texture Segmentation," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 926-932, 1999.
- [27] N. Paragios and R. Deriche, "Geodesic Active Regions: A New Paradigm to Deal with Frame Partition Problems in Computer Vision," *J. Visual Comm. and Image Representation*, vol. 13, pp. 249-268, 2002.

- [28] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int'l J. Computer Vision*, vol. 22, pp. 61-79, 1997.
- [29] S. Kichenassamy, S. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal Curvature Flow: From Phase Transitions to Active Vision," *Archives for Rational Mechanics and Analysis*, vol. 134, pp. 275-301, 1996.
- [30] S.C. Zhu and A.L. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, Sept. 1996.
- [31] J.-M. Morel and S. Solimini, *Variational Methods for Image Segmentation*. Birkhauser, 1994.
- [32] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky, "Nonparametric Methods for Image Segmentation Using Information Theory and Curve Evolution," *Proc. Int'l Conf. Image Processing*, vol. 3, pp. 797-800, 2002.
- [33] Y. Rathi and O. Michailovich, "Seeing the Unseen: Segmenting with Distributions," *Proc. Int'l Conf. Signal and Image Processing*, vol. 534, 2006.
- [34] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience, 2001.
- [35] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Statistical Shape Analysis Using Kernel PCA," *Proc. SPIE*, vol. 6064, pp. 425-432, 2006.
- [36] S. Dambreville, Y. Rathi, and A. Tannenbaum, "A Shape-Based Approach to Robust Image Segmentation," *Proc. Int'l Conf. Image Analysis and Recognition*, vol. 1, pp. 173-183, 2006.



**Samuel Dambreville** received the master of science degree in electrical engineering and the master's degree in business administration from the Georgia Institute of Technology in 2003 and 2006, respectively, and the "Diplome d'ingenieur en Genie Electrique et Automatique" (MEng) from the ENSEEIHT, Toulouse, France, in 2003. He is currently a PhD candidate in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests include the robust tracking of deformable objects and the development of statistical and shape-driven techniques for active contour segmentation. He was the recipient of the Hugo O. Schuck Best Paper Award at the 2007 American Control Conference (along with Yogesh Rathi and Allen Tannenbaum). He is a student member of the IEEE.



**Yogesh Rathi** received the bachelor's degree in electrical engineering and the master's degree in mathematics from the Birla Institute of Technology and Science, Pilani, India, in 1997 and the PhD degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2006. His PhD thesis entitled "Filtering for Closed Curves" was on tracking highly deformable objects in the presence of noise and clutter. He is currently a postdoctoral fellow in the Psychiatry Neuroimaging Laboratory at Brigham and Women's Hospital, Harvard Medical School, Boston. His research interests include image processing, medical imaging, diffusion tensor imaging, computer vision, control, and machine learning. He is a member of the IEEE.



**Allen Tannenbaum** is a faculty member in the Schools of Electrical and Computer and Biomedical Engineering at the Georgia Institute of Technology. His research interests include systems and control, computer vision, and image processing. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).