Available online at www.sciencedirect.com

ScienceDirect

Signal Processing 87 (2007) 3021–3044

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Nonparametric shape priors for active contour-based image segmentation

Junmo Kim[a,*], Müjdat Çetin[b], Alan S. Willsky[c]

[a]Samsung Advanced Institute of Technology, Yongin 446-712, Republic of Korea
[b]Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey
[c]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received 27 September 2006; received in revised form 12 March 2007; accepted 20 May 2007
Available online 6 June 2007

Abstract

When segmenting images of low quality or with missing data, statistical prior information about the shapes of the objects to be segmented can significantly aid the segmentation process. However, defining probability densities in the space of shapes is an open and challenging problem. In this paper, we propose a nonparametric shape prior model for image segmentation problems. In particular, given example training shapes, we estimate the underlying shape distribution by extending a Parzen density estimator to the space of shapes. Such density estimates are expressed in terms of distances between shapes, and we consider the L_2 distance between signed distance functions for shape density estimation, in addition to a distance measure based on the template metric. In particular, we consider the case in which the space of shapes is interpreted as a manifold embedded in a Hilbert space. We then incorporate the learned shape prior distribution into a maximum *a posteriori* (MAP) estimation framework for segmentation. This results in an optimization problem, which we solve using active contours. We demonstrate the effectiveness of the resulting algorithm in segmenting images that involve low-quality data and occlusions. The proposed framework is especially powerful in handling “multimodal” shape densities.

Keywords: Image segmentation; Curve evolution; Level set method; Nonparametric density estimation; Shape prior

1. Introduction

We consider image segmentation problems that involve limited and low-quality data. Such segmentation problems are ill-posed and require the incorporation of prior information about the objects to be segmented. When human experts segment images, they clearly make use of such prior information. For example, a radiologist can usually manually segment an organ (e.g. the prostate) in a magnetic resonance image, although the boundaries are mostly invisible to a layperson. Clearly, radiologists augment the observed data with their expertise, in other words with statistical prior information, about the shape of the organ. Existing automatic segmentation methods (either explicitly or implicitly) enforce only very simple constraints about the underlying shapes. For example, many active contour-based methods (which is the framework

*Corresponding author.
E-mail addresses: junmo@alum.mit.edu (J. Kim), mcetin@sabanciuniv.edu (M. Çetin), willsky@mit.edu (A.S. Willsky).

0165-1684/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.sigpro.2007.05.026

we also use in our work) involve a *curve length penalty* [1–7], which translates to the assumption that shorter curves are statistically more likely than longer ones. However, in many applications, more structured prior information about the shapes is available. Yet the challenge is how to construct probabilistic descriptions in the space of shapes, and then incorporate such statistical information into the segmentation process.

Early work on this problem involves landmark-based representations of shapes, and the construction of typical shapes and typical variability based on a set of training shapes via principal component analysis (PCA) [8]. The use of landmarks has the drawback that the performance of shape analysis depends on the quality of those landmarks. Recently, there has been increasing interest in using level set-based representations for shape priors [9,10], which avoid landmarks. In [9,10], PCA of the signed distance functions of training data is used to capture the variability of shapes. These techniques have been applied to segmentation problems involving low SNR or occluded images successfully, especially when the shape variability is small. However, there are two major shortcomings of such techniques. First, these methods treat the signed distance functions as elements of a linear vector space, and perform operations such as averaging. Yet, the space of signed distance functions is a nonlinear manifold and is not closed under linear operations. For example, the average of signed distance functions, which is commonly used to obtain a mean shape, is not necessarily a signed distance function. Therefore, the use of linear analysis tools such as PCA gives rise to an inconsistent framework for shape modeling [10]. Second, these techniques can handle only unimodal, Gaussian-like shape densities. In particular, these methods cannot deal with "multimodal" shape densities,which involve multiple classes of shapes (e.g. a shape density of handwritten digits, composed of multiple digits).

Besides the work that involves level set methods, there has also been some other interesting work on analysis of shape. Klassen and Srivastava et al. [11] represent shapes by so-called direction functions and define the space of shapes as a sub-manifold embedded in the L_2 space of direction functions. The key element in that work is the numerical computation of a geodesic path on the shape space connecting any two different shapes, where the distance between two shapes is defined as the length of the geodesic path. However, this method cannot be easily extended to deal with 3D shapes. Minchor and Mumford [12] also considered a space of curves and obtained a numerical computation of a geodesic path. Charpiat et al. [13] used an approximation of the Hausdorff metric in order to make it differentiable and used a gradient of the approximate Hausdorff metric to warp one shape into another shape. Soatto and Yezzi [14] proposed a method of extracting both the motion and the deformation of moving deformable objects. In that work, they propose the notion of shape average and motions such that all the example shapes are obtained by rigid transformation (motion) of the shape average followed by diffeomorphism (deformation), where the shape average and motions are defined such that the total amount of deformation is minimized. In that work, the amount of such diffeomorphism is measured by a simple template metric, i.e. the area of set-symmetric difference. There is also recent work by Cootes et al. [15], which constructs a model that obeys such diffeomorphic constraints.

In our work, we propose a framework for constructing nonparametric shape densities from example training shapes. In particular, we assume that the training shapes are drawn from an unknown shape distribution, and we estimate the underlying shape distribution by extending a Parzen density estimator to the space of shapes. Such density estimates are expressed in terms of distances between shapes. We propose two specific distance metrics, namely the L_2 distance between signed distance functions and the template metric (which is the area of set symmetric difference of two shape interiors), to be used for nonparametric density estimation, although other metrics could be used in our framework as well. We then formulate the segmentation problem as maximum a posteriori (MAP) estimation, in which we use the learned nonparametric shape density as the prior. This leads to an optimization problem for the segmenting curve, for which we develop and use an active contour-based iterative algorithm. We present experimental results of segmenting low-quality and occluded images. We also demonstrate how the proposed algorithm can successfully incorporate shape densities involving multiple object classes.

Recently Cremers et al. [16] also proposed a nonparametric density estimation-based technique for shape priors and demonstrated how the level set-based segmentation can benefit from such shape priors. In particular, they considered a kernel density estimate with the square root of the template metric. They also incorporated the alignment with respect to translation and scaling directly into the first variation of their energy functional.

Our work and the work in [16] have been done in parallel independently and share many common aspects. Yet, our work is different from the work in [16] in three major ways. First, we consider another shape distance measure, namely the Euclidean (i.e. $L2$) distance between signed distance functions for shape density estimation, in addition to a distance measure based on the template metric which is similar to the one used in [16]. We compare the two kinds of metrics from both theoretical and practical standpoints. Second, our framework can handle alignment with respect to similarity transforms, which consist of translation, scaling, and rotation. Third, we further analyze the issue of density estimation on the space of shapes. In particular, we consider the case in which the space of shapes is represented as a space of signed distance functions, which we interpret as a manifold embedded in a Hilbert space. We conjecture that the best metric for density estimation in this case is a geodesic distance and suggest a density estimate with $L2$ distance as a good approximation of the density estimate with the geodesic distance. We also provide a theoretical comparison of our framework with that based on PCA.

The remainder of this paper is organized as follows. In Section 2, we motivate the problem of building a shape prior and present nonparametric shape priors based on the two shape distance measures. In Section 3, we present our framework for shape-based image segmentation, where we derive the gradient flows for active contour evolution for maximizing shape priors. We then present experimental results in Section 4 with a variety of low quality images. Finally, we conclude in Section 5 with a summary.

2. Shape priors

Let us consider a segmentation problem. If the image to be segmented is of high quality (defined appropriately based on context), then the observed image data provide a large amount of information about the true boundary. However, in many applications, this may not be the case. For example, if the image is of low contrast, the amount of information provided by the data will be small. Similarly, if there are occlusions or missing data around a portion of the boundary, the data will not tell us much about that part of the boundary. For such low-quality images, data alone will not be sufficient for accurate segmentation. Considering that segmentation is equivalent to extracting the pose and the shape of the boundary of the object, prior information on shapes will be helpful in segmentation, if we have any such information.

Now let us consider the case where we know the category of the object in the image. If there is only one possible fixed shape in that category, then we know the exact shape of the object a priori, and the segmentation problem comes down to estimation of pose. However, in general, there is shape variation even within a single category of objects, so that there are considerably more (possibly a continuum of) "candidate" shapes in the image than those corresponding simply to variations in pose. For example if the category is a particular organ in a medical imaging application, there will be variability in the shape of the organ across patients. Since such candidate shapes may not be equally likely, it is desirable to have a quantitative measure of how good a candidate shape is or how likely such a shape is. In this sense, a probability measure on the set of shapes of a given category is the desirable description of the prior knowledge about shapes of the objects in the category.

Now the question is how to compute such a probability measure on a set of shapes. An intuitive idea is that a shape is more likely if it is similar to the shapes of the same category seen before. This raises the issue of how to define a notion of similarity. Mathematically, this suggests that a measure of distance between two shapes will play an important role in statistical analysis of shapes. In the following section, we state more formally the problem of building shape priors from available example shapes.

2.1. Problem of building a shape prior

In typical active contour-based image segmentation, a curve length penalty term $\alpha\oint_C \mathrm{d}s$ for the segmenting curve C is often used for regularization. The basic idea behind this is that shorter curves are more likely as a boundary of an object than longer ones. Such a regularization term can be considered as a prior term, more accurately, the negative logarithm of a prior density. This interpretation is motivated by the Bayesian interpretation of the energy functional $E(C)$ for image segmentation.

$$E(C) = -\log p(\text{data}|C) - \log p_C(C) \propto -\log p(C|\text{data}). \tag{1}$$

In this respect, the curve length term corresponds to the prior density for the curve $p_C(C) \propto \mathrm{e}^{-\alpha \oint_C \mathrm{d}s}$.

If we have more information about the shape of the object to segment, we can build a more sophisticated shape prior and use it to guide the evolution of the curve C. In particular, we are interested in the case where we have a set of example shapes of the object class. Suppose that the example shapes are given in terms of n curves $C_1, \ldots, C_n$ that delineate the boundaries of the example shapes. The basic idea we use is that a candidate segmenting curve C will be more likely if it is similar to the example curves. In order to measure similarity between curves, we need to compare the candidate curve C with the example curves. However, when the candidate C and the training curves $C_1, \ldots, C_n$ are not aligned, a direct comparison of C with $C_1, \ldots, C_n$ will include not only the shape difference but also artifacts due to pose difference such as translation, rotation, and scaling. In order to deal with this pose issue, we align the curves $C_1, \ldots, C_n$ and C into $\tilde{C}_1, \ldots, \tilde{C}_n$ and $\tilde{C}$, which have the same pose. In this paper, we denote the aligned curves with tildes, whereas we denote the candidate curve C without a tilde. Hence, the procedure of estimating $p_C(C)$ consists of the following steps:

(1) Align $C_1, \ldots, C_n$ into $\tilde{C}_1, \ldots, \tilde{C}_n$. (Section 2.2.1).
(2) Align C w.r.t. $\tilde{C}_1, \ldots, \tilde{C}_n$ into $\tilde{C}$. (Section 2.2.2).
(3) Estimate $p_{\tilde{C}}(\tilde{C})$ the prior probability density of $\tilde{C}$ given $\tilde{C}_1, \ldots, \tilde{C}_n$. (Section 2.3).
(4) Relate $\hat{p}_{\tilde{C}}(\tilde{C})$ to $\hat{p}_C(C)$. (Section 2.3.4).

Our approach can incorporate any available alignment algorithms in steps 1–2. Also we can use a variety of distance metrics in step 3. Hence, this framework can provide several variations of algorithms depending on the choice of the alignment algorithm and the distance metric. We now discuss each of the above steps. Steps 2–4 are used in the segmentation algorithm to be described in Section 3.[1]

2.2. Alignment

2.2.1. Alignment of training curves by similarity transforms

Here we discuss how to align the n training curves $C_1, \ldots, C_n$. In particular, we use the alignment algorithm presented in Tsai et al. [10], in which a similarity transform is applied to each curve such that the transformed curves are well aligned. Let us first define the similarity transform and then provide a criterion for alignment.

The similarity transformation $T[\mathbf{p}]$ with the pose parameter $\mathbf{p} = [a\ b\ \theta\ h]$ consists of translation $M(a,b)$, rotation $R(\theta)$, and scaling $H(h)$, and it maps a point $(x,y) \in \mathscr{R}^2$ to $T[\mathbf{p}]\ (x,y)$ as follows:

$$T[\mathbf{p}]\begin{pmatrix} x \\ y \end{pmatrix} = R(\theta) \circ H(h) \circ M(a,b)\begin{pmatrix} x \\ y \end{pmatrix} \tag{2}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} h(x+a) \\ h(y+b) \end{pmatrix}. \tag{3}$$

We define the transformed curve $T[\mathbf{p}]C$ to be the new curve that is obtained by applying the transformation to every point on the curve. The shape represented by a curve C can also be represented by a binary image $I(x,y)$ whose value is 1 inside C and 0 outside C. The transformation of $I(x,y)$ is defined to be the new image obtained by moving every pixel (x,y) of the image I to a new position $T[\mathbf{p}](x,y)$ making the intensity of $\tilde{I}$ at pixel $T[\mathbf{p}](x,y)$ the same as the intensity of I at pixel (x,y) as illustrated in Fig. 1. Thus the two images I and $\tilde{I} \triangleq T[\mathbf{p}]I$ are related by

$$I(x,y) = \tilde{I}(T[\mathbf{p}](x,y)) \quad \text{for all } (x,y) \in \Omega. \tag{4}$$

Equivalently, $\tilde{I}$ can be written in terms of I as follows:

$$\tilde{I}(x,y) = I(T^{-1}[\mathbf{p}](x,y)). \tag{5}$$

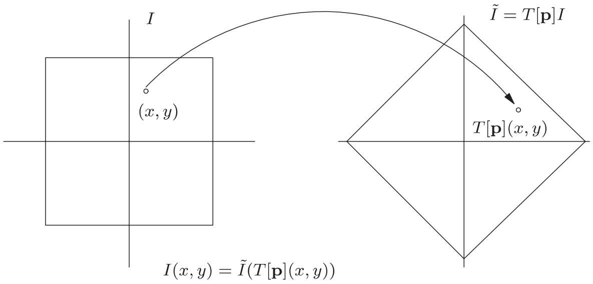
I
(x,y)
$\tilde{I} = T[\mathbf{p}]I$
$T[\mathbf{p}](x,y)$
$I(x,y) = \tilde{I}(T[\mathbf{p}](x,y))$

Fig. 1. Illustration of the similarity transformation $T[\mathbf{p}]I$.

[1] We are focusing on 2D segmentation problems here although our approach can be directly applied to 3D problems.

We now provide a criterion for alignment. Given n training curves, we obtain aligned curves $\tilde{C}_1, \ldots, \tilde{C}_n$ by a similarity transformation $\tilde{C}_i = T[\hat{\mathbf{p}}_i]C_i$ with pose estimate $\hat{\mathbf{p}}_i$ for each i. The pose estimates are chosen such that they minimize an energy functional for alignment. The energy functional we use is given by

$$E_{\text{align}}(\mathbf{p}_1, \ldots, \mathbf{p}_n) = \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left\{ \frac{\int\int_\Omega (T[\mathbf{p}_i]I^i - T[\mathbf{p}_j]I^j)^2 \,\mathrm{d}x\,\mathrm{d}y}{\int\int_\Omega (T[\mathbf{p}_i]I^i + T[\mathbf{p}_j]I^j)^2 \,\mathrm{d}x\,\mathrm{d}y} \right\}, \tag{6}$$

where I^i is a binary map whose value is 1 inside C_i and 0 outside C_i, and $T[\mathbf{p}]I^i$ is a transformed binary map whose value is 1 inside $T[\mathbf{p}_i]C_i$ and 0 outside $T[\mathbf{p}_j]C_j$. As in (5), I^i and $T[\mathbf{p}_i]I_i$ are related by

$$T[\mathbf{p}_i]I^i(x, y) = I^i(T^{-1}[\mathbf{p}_i](x, y)). \tag{7}$$

The numerator in (6), which is the area of set-symmetric difference between two interior regions of $T[\mathbf{p}_i]C_i$ and $T[\mathbf{p}_j]C_j$, basically measures the amount of mismatch between $T[\mathbf{p}_i]I^i$ and $T[\mathbf{p}_j]I^j$, and the denominator is present to prevent all the binary images from shrinking to improve the cost function. The double summation in (6) implies that we compare every pair of the binary maps in the training database.

To estimate the pose parameters, we fix the pose parameter for the first curve as the one for the identity transform and compute $\mathbf{p}_2, \ldots, \mathbf{p}_n$ by

$$\{\hat{\mathbf{p}}_2, \ldots, \hat{\mathbf{p}}_n\} = \arg \min_{\mathbf{p}_2, \ldots, \mathbf{p}_n} E_{\text{align}}(\mathbf{p}_1, \ldots, \mathbf{p}_n)|_{\mathbf{p}_1 = [0\ 0\ 0\ 1]}, \tag{8}$$

where we use gradient descent to compute $\mathbf{p}_2, \ldots, \mathbf{p}_n$.

2.2.2. Alignment of the candidate curve

Now we consider the problem of aligning the candidate curve C w.r.t. the n aligned training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. To this end, we estimate a pose parameter $\hat{\mathbf{p}}$ such that $\tilde{C} = T[\hat{\mathbf{p}}]C$ is well aligned to $\tilde{C}_1, \ldots, \tilde{C}_n$ by minimizing the energy

$$\begin{aligned}\hat{\mathbf{p}} &= \arg \min_{\mathbf{p}} E(\mathbf{p}) \\ &= \arg \min_{\mathbf{p}} \sum_{i=1}^{n} \left\{ \frac{\int_\Omega (T[\mathbf{p}]I - \tilde{I}^i)^2 \,\mathrm{d}x}{\int_\Omega (T[\mathbf{p}]I + \tilde{I}^i)^2 \,\mathrm{d}x} \right\},\end{aligned} \tag{9}$$

where I and $\tilde{I}^i$ are binary maps whose values are 1 inside and 0 outside C and $T[\hat{\mathbf{p}}_i]C_i$, respectively.

2.3. Estimating the shape density

Now the problem is to estimate how likely the curve $\tilde{C}$ is, given the training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. We assume that the n aligned curves are i.i.d. according to a density $p_{\tilde{C}}(\cdot)$ and estimate the density $p_{\tilde{C}}(\cdot)$ from n i.i.d. samples.

2.3.1. Nonparametric density estimation

Let us first consider density estimation for a finite dimensional random vector. Suppose that we have n samples $x_1, x_2, \ldots, x_n \in \mathscr{R}^m$ drawn from an m-dimensional density function $p(x)$. The Parzen density estimate is given by

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{k}(x - x_i, \Sigma), \tag{10}$$

where we use an m-dimensional Gaussian kernel $\mathbf{k}(x, \Sigma) = N(x; 0, \Sigma^{\mathrm{T}}\Sigma)$. If the kernel is spherical, i.e. $\Sigma = \sigma I$, the above density estimate becomes

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} k(d(x, x_i), \sigma), \tag{11}$$

where $d(x, x_i)$ is the Euclidean distance between x and x_i in $\mathscr{R}^m$, and $k(x, \sigma)$ is the 1D Gaussian kernel $k(x, \sigma) = N(x; 0, \sigma^2)$.

Given a distance measure $d_{\mathscr{C}}(\cdot, \cdot)$ in $\mathscr{C}$, the space of curves, we can extend this Parzen density estimator with a spherical Gaussian kernel to the infinite dimensional space $\mathscr{C}$ as follows:

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma). \tag{12}$$

In this density estimate, the composite of the 1D kernel and the distance metric plays the role of an infinite dimensional kernel. For the kernel size σ, we use an ML kernel size with leave-one-out [17]

$$\sigma_{\text{ML}} = \arg \max_{\sigma} \sum_{i} \log \hat{p}_{\tilde{C}}(\tilde{C}_i) \tag{13}$$

$$= \arg \max_{\sigma} \sum_{i} \log \frac{1}{n-1} \sum_{j \neq i} k(d_{\mathscr{C}}(\tilde{C}_i, \tilde{C}_j), \sigma). \tag{14}$$

Our nonparametric shape priors in (12) can be used with a variety of distance metrics. In the following sections, we consider two specific metrics, namely the template metric and the L_2 distance between signed distance functions.

2.3.2. Parzen density estimate with the template metric

We now consider the Parzen density estimate in (12) with a specific metric, namely the template metric $d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) = \mathrm{Area}(R_{\mathrm{inside}\ \tilde{C}} \triangle R_{\mathrm{inside}\ \tilde{C}_i})$ [18], where $\triangle$ denotes set symmetric difference. The density estimate with the template metric is given by

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_{i=1}^{n} k(d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i), \sigma). \tag{15}$$

We can also use square root of the template metric as a distance measure and have the following density estimate:

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_{i=1}^{n} k\left(\sqrt{d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)}, \sigma\right). \tag{16}$$

Cremers et al. [16] use this density estimate for shape-based image segmentation. As we will see in Section 3.2, the template metric can also be expressed in terms of level set functions and the Heaviside function,[2] and as a result it has a practical merit that the gradient flows $\frac{\partial \tilde{C}}{\partial t}$ for the above two density estimates are given in closed form.

One drawback of the template metric and its variants is that they can miss a significant shape difference or difference in topology if the area of difference is small. For instance, the template metric may not be able to distinguish a circle from a circle with a thin blob attached or a circle from a doughnut with a very small hole inside.

From the theoretical standpoint, the property that the template metric is insensitive to this kind of shape differences suggests that the geometry of the shape space defined by the template metric is not very appropriate for defining shape density, which we explain now. When one defines a shape probability density, the shape density at a specific shape $C^{\dagger}$ is related to the probability of the random shape being inside a small neighborhood of the shape $C^{\dagger}$ as follows:

$$p_{\tilde{C}}(C^{\dagger})\mathrm{volume}(N_{\varepsilon}(C^{\dagger})) = \mathrm{Prob}(\tilde{C} \in N_{\varepsilon}(C^{\dagger})), \tag{17}$$

where $N_{\varepsilon}(C^{\dagger}) = \{C | d(C, C^{\dagger}) < \varepsilon\}$ is a neighborhood of $C^{\dagger}$ with radius ε. This 'density' will make sense only if we can make sure that all the shapes inside the small neighborhood $N_{\varepsilon}(C^{\dagger})$ will look similar to human observer. The template metric and its variants do not satisfy this condition.

Despite the shortcomings mentioned above, in practice, the template metric provides a viable solution for the shape density estimation and shape-based segmentation. We discuss the nature of the gradient flow derived from the template metric-based shape prior and the segmentation results in Sections 3.4 and 4.

2.3.3. Parzen density estimate on the space of signed distance functions

In this section, we consider another way of defining a metric between two curves based on level set representation of curves. In particular, we represent each curve $\tilde{C}_i$ by its corresponding signed distance function $\phi_{\tilde{C}_i}$, where we use the sign convention of $\phi < 0$ inside the curve and $\phi > 0$ outside the curve. The magnitude of signed distance function $\phi_{\tilde{C}_i}(x)$ is the distance from the point x to the curve $\tilde{C}_i$, and the magnitude grows as the point is more inside/outside of the boundary curve. In other words, we put more weight on points which we are confident are inside/outside the object. Now we can define the distance between two curves $\tilde{C}$ and $\tilde{C}_i$ as the distance between the two corresponding signed distance functions $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ as follows:

$$d(\tilde{C}, \tilde{C}_i) = d_{\mathscr{D}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \tag{18}$$

where we let $\mathscr{D}$ denote the space of signed distance functions and $d_{\mathscr{D}}(\cdot,\cdot)$ denote the metric in space $\mathscr{D}$. The issue here is how to define a distance metric $d_{\mathscr{D}}(\cdot,\cdot)$ in the space of signed distance functions.

We interpret the space of signed distance functions $\mathscr{D}$ as a subset of an infinite dimensional Hilbert space[3] $\mathscr{L}$, which is defined by $\mathscr{L} \triangleq \{\phi | \phi : \Omega \to \mathscr{R}\}$, with the following inner product and L_2 distance:

$$\langle \phi_1, \phi_2 \rangle = \int_{\Omega} \phi_1(x)\phi_2(x)\,\mathrm{d}x, \tag{19}$$

$$d_{L_2}(\phi_1, \phi_2) = \sqrt{\langle \phi_1 - \phi_2, \phi_1 - \phi_2 \rangle}. \tag{20}$$

As the inner product is an integral over the image domain Ω, the inner product and its induced L_2 norm depend on the image domain Ω. However, this does not cause problems in practice, as we can assume that the image domain Ω is fixed over the

[2] We can also interpret the template metric and the square root of the template metric as L_1 and L_2 norms on differences between binary maps, respectively.

[3] As the space of signed distance functions is embedded in the space of level set functions $\{\phi | \phi : \Omega \to \mathscr{R}\}$, defining the geometry of this bigger space automatically determines the distance for the smaller space $\mathscr{D}$. There are many candidate geometries for this bigger space, and we choose to interpret the space of the level set functions as a Hilbert space.

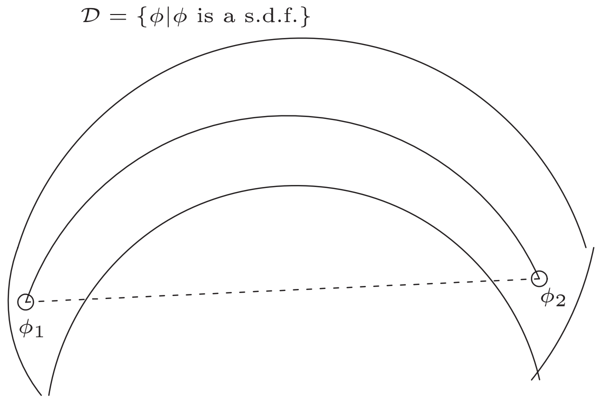

Fig. 2. Illustration of the space of signed distance functions $\mathscr{D}$ and the geodesic path (solid line) between ϕ_1 and ϕ_2 compared with the shortest path in Hilbert space $\mathscr{L}$ (dashed line) which is off the space $\mathscr{D}$.

entire segmentation process. Another issue is that this distance is not invariant with respect to translation and rotation. As we mentioned before, we first remove differences due to pose variation by alignment. Hence, the shape prior we propose does not require such invariance properties.

Since the space $\mathscr{D}$ is embedded in a Hilbert space, a natural metric $d_{\mathscr{D}}(\phi_1, \phi_2)$ for this space will be a minimum geodesic distance, i.e. the distance of the shortest path from ϕ_1 to ϕ_2 lying in the signed distance function space $\mathscr{D}$. Fig. 2 provides a conceptual picture of the space $\mathscr{D}$, and the geodesic path connecting two distance functions ϕ_1 and ϕ_2. The direct line (dashed line) connecting ϕ_1 and ϕ_2 gives a shortest path in the Hilbert space and its length corresponds to the L_2 distance $d_{L_2}(\phi_1, \phi_2)$.

If one could compute the minimum geodesic distances $d_{\text{geodesic}}(\cdot, \cdot)$, the corresponding Parzen density estimate would be

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_i k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma). \tag{21}$$

We conjecture that the geodesic distance is the best metric for the density estimation and that this density estimate is inherently concentrated on the shape manifold even with finite number of samples.[4] The expected property that the density estimate is concentrated on the shape manifold would make the kernel density estimate with the geodesic distance appealing from a theoretical standpoint. However, computing a geodesic distance in an infinite dimensional manifold is a challenging problem. There is some previous work on computing geodesic distances in the space of curves such as Minchor and Mumford [12] and Klassen et al. [11], but there is little work when the shape is represented by signed distance functions.

Instead, we now consider the Parzen density estimate with the L_2 distance in $\mathscr{L}$

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma). \tag{22}$$

Note that we defined $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ to be the signed distance functions representing the curves $\tilde{C}$ and $\tilde{C}_i$ and that this constraint that $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ are signed distance functions is necessary to make this density estimate be uniquely determined, as this equation can give different values when $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ are other level set functions representing the same curves.[5] In addition, the constraint that $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ are signed distance functions enables this density estimate to approximate the density estimate with geodesic distance in (21), which we explain next.

Let us first consider the case where the example shapes are of small variation. Fig. 3 illustrates this situation. In this case, the part of the manifold supporting the example shapes is approximately flat or linear provided that the manifold does not have too much curvature. This is why methods based on PCA of signed distance functions [9,10] (hence based on linear vector space tools), work well when there is small shape variation.

For the Parzen density estimate in such a case, we can take advantage of the same phenomenon, namely that the part of the manifold supporting

[4] Cremers et al. [16] also interpret the shape space as a nonlinear manifold, and mention that asymptotically the kernel density estimate (with the square root of the template metric) becomes concentrated on the manifold. This asymptotic behavior will be true of many distance metrics, but with finite samples the density estimate will not be concentrated on the manifold with most of the metrics.

[5] There are infinitely many level set functions whose zero-level sets give the same curve, and due to this redundancy the distance between two level set functions representing two distinct shapes can be arbitrarily small by scaling the level set functions. For instance, let ϕ_{C_1} and ϕ_{C_2} the two signed distance functions representing two distinct shapes C_1 and C_2, then any scaled versions of the signed distance functions also represent the same shapes, and we have

$$d_{L_2}(\alpha\phi_{C_1}, \alpha\phi_{C_2}) = \alpha d_{L_2}(\phi_{C_1}, \phi_{C_2}),$$

where the right-hand side can approach zero as the scaling factor α approaches zero. Hence, the way we constrain the shape to be represented only by the signed distance functions not only removes the redundancy in shape representation but also makes the distance well defined.

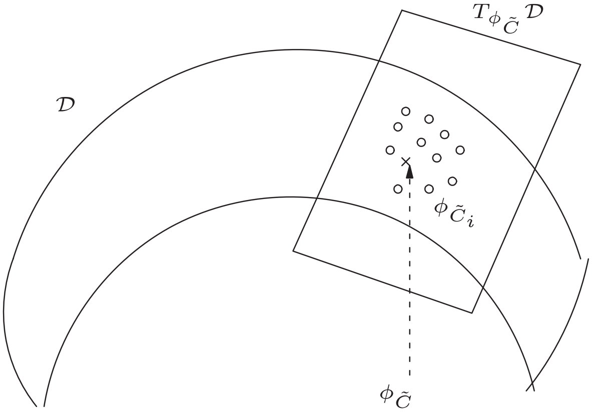

Fig. 3. Illustration of example shapes in $\mathscr{D}$ with small shape variation.

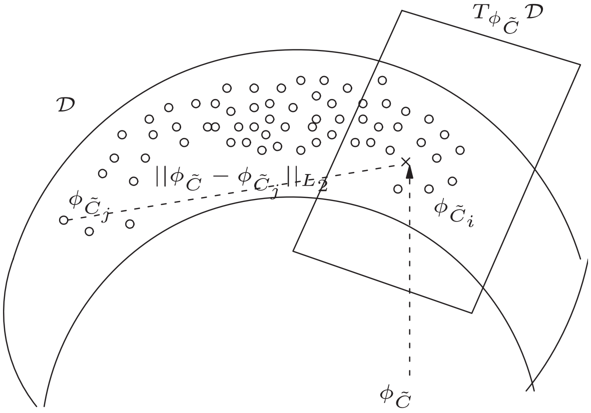
$T_{\phi_{\tilde{C}}}\mathcal{D}$
$\mathcal{D}$
$||\phi_{\tilde{C}} - \phi_{\tilde{C}_j}||_{L_2}$
$\phi_{\tilde{C}_j}$
$\phi_{\tilde{C}_i}$
$\phi_{\tilde{C}}$

Fig. 4. Illustration of example shapes in $\mathscr{D}$ with broad range.

example shapes is approximately flat and that the L_2 distance is close to the geodesic distance. Thus, in this case, the nonparametric density estimate with L_2 distance can be a good approximation of that with the geodesic distance.

Now consider the case where the example shapes have a broad range as illustrated in Fig. 4. In this case, the part of the manifold supporting the samples is no longer flat, and PCA will not be very effective. In contrast, the density estimate with L_2 distance can still be a good approximation of (21) for the following reasons. When $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$ are close enough, the L_2 norm will be a good approximation of the geodesic distance. On the other hand, when $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_j}$ are far from each other, there will be an error in approximation of distance, but the overall error in density estimate will be small as long as the kernel size σ is small compared to the distance $d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_j})$. The kernel size σ will be small provided that we have a sufficiently large number of example shapes.

More precisely, we have the following approximation if there exists some constant M such that $M\sigma$ is small and M is large

$$
\begin{aligned}
&\frac{1}{n}\sum_i k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\
&\quad = \frac{1}{n}\left(\sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) \leqslant M\sigma}} k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) + \sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) > M\sigma}} k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) \right)\\
&\quad \overset{(1)}{\approx} \frac{1}{n} \sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) \leqslant M\sigma}} k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\
&\quad \overset{(2)}{\approx} \frac{1}{n} \sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) \leqslant M\sigma}} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\
&\quad \overset{(3)}{\approx} \frac{1}{n}\left(\sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) \leqslant M\sigma}} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) + \sum_{\substack{i\\ d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) > M\sigma}} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) \right),\\
&\quad = \frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma), \qquad (23)
\end{aligned}
$$

where

- We can make the approximation (2), provided that $M\sigma$ is small enough such that if $d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}) \leqslant M\sigma$, then $d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}) \approx d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i})$.
- We can make the approximation (1) and (3), provided that M is large enough such that if $d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}) > M\sigma$, then $k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) \approx 0$ and $k(d_{\text{geodesic}}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) \approx 0$.

These conditions can be satisfied if the kernel size σ is small enough.

A similar argument will hold for the case where the samples form multiple clusters as illustrated in Fig. 5, and we can make the same approximation as Eq. (23). When the density is multi-modal, if we knew which mode of the density the candidate shape

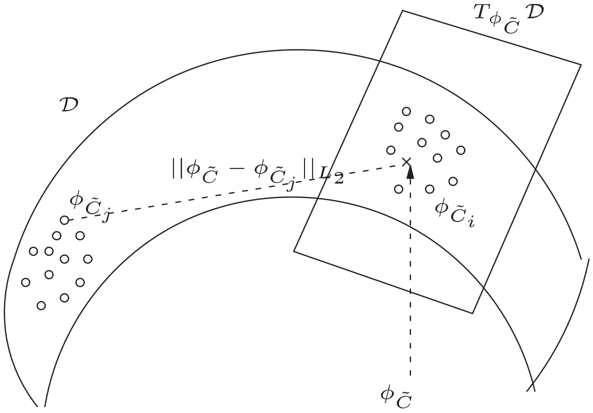
$T_{\phi_{\tilde{C}}}\mathcal{D}$
$\mathcal{D}$
$||\phi_{\tilde{C}} - \phi_{\tilde{C}_j}||_{L_2}$
$\phi_{\tilde{C}_j}$
$\phi_{\tilde{C}_i}$
$\phi_{\tilde{C}}$

Fig. 5. Illustration of two clusters of example shapes in $\mathcal{D}$ and the tangent space.

was in,[6] we could use a simpler approach like [10] for shape-based segmentation. However, in scenarios where that is not the case, our approach is more powerful.

2.3.4. Relating $\hat{p}_{\tilde{C}}(\tilde{C})$ to $\hat{p}_C(C)$

So far we have derived a density estimate $\hat{p}_{\tilde{C}}(\tilde{C})$ for an aligned candidate curve $\tilde{C}$, which is given in terms of aligned training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. We remind the readers that we have all the training curves aligned, but given an arbitrary image, the object in that scene is not necessarily aligned with our training set $\{\tilde{C}_i\}$. Hence we need a density estimate for the unaligned curves we encounter during the curve evolution process in segmenting that image. Here we will relate the density estimate $\hat{p}_C(C)$ for such an unaligned candidate curve C to the density estimate $\hat{p}_{\tilde{C}}(\tilde{C})$ for an aligned candidate curve $\tilde{C}$. To this end, we first consider the relationship between the two densities $p_{\tilde{C}}(\tilde{C})$ and $p_C(C)$.

Conceptually, every candidate curve C can be described by its shape and its pose. For instance, when $\tilde{C} = T[\mathbf{p}]C$ is aligned w.r.t. training curves $\{\tilde{C}_i\}$, we can interpret $\tilde{C}$ as the shape of C and $\mathbf{p}$ as the pose of C. Thus the probability density for the candidate curve can be written in terms of the density for its shape and its pose as follows:

$$p_C(C) \propto p_{\tilde{C},\mathbf{p}}(\tilde{C}, \mathbf{p}) \tag{24}$$

$$= p_{\tilde{C}}(\tilde{C})p(\mathbf{p}|\tilde{C}). \tag{25}$$

If the prior information about the pose $p(\mathbf{p}|\tilde{C})$ is available, one can use that information to evaluate $p_C(C)$. In this work, we assume that $p(\mathbf{p}|\tilde{C})$ is uniform, i.e. all poses $\mathbf{p}$ are equally likely. In this case, all slices of the joint density $p_{\tilde{C},\mathbf{p}}(\tilde{C}, \mathbf{p})$ along a fixed $\mathbf{p}$ are identical and simply proportional to $p_{\tilde{C}}(\tilde{C})$. Hence, we have

$$p_C(C) = \gamma p_{\tilde{C}}(\tilde{C}) \quad \text{for all } \mathbf{p} \tag{26}$$

where γ is a normalizing constant. Therefore, given the density estimate $\hat{p}_{\tilde{C}}(\tilde{C})$, we can estimate the density estimate of any candidate curve C in terms of its shape estimate $T[\hat{\mathbf{p}}]C$ as follows:

$$\hat{p}_C(C) = \gamma \hat{p}_{\tilde{C}}(T[\hat{\mathbf{p}}]C), \tag{27}$$

where the pose estimate $\hat{\mathbf{p}}$ is obtained by Eq. (9).

3. Shape-based segmentation

Now we combine the nonparametric shape prior and a data term within a Bayesian framework to form the energy functional for segmentation. The data term we use is from the piecewise constant version of the Mumford–Shah functional [20], and the shape term comes from the nonparametric shape priors introduced in Section 2. We choose this data term as a representative one, as it has found use in a wide array of previous work [21,5,16]. However, note that the shape priors can be combined with any other data term such as the information theoretic term proposed in [22] as well. The task of segmentation is to minimize the energy functional.[7]

$$E(C) = -\log p(\text{data}|C) - \log p_C(C) \tag{28}$$

$$= \beta\left[\int_{R_{\text{in}}} (I(x) - m_{\text{in}})^2\,\mathrm{d}x + \int_{R_{\text{out}}} (I(x) - m_{\text{out}})^2\,\mathrm{d}x\right] - \log p_C(C), \tag{29}$$

where R_{in} (R_{out}) is the region inside (outside) the curve C,

$$m_{\text{in}} = \frac{\int_{R_{\text{in}}} I(x)\,\mathrm{d}x}{\int_{R_{\text{in}}} \mathrm{d}x}, \quad m_{\text{out}} = \frac{\int_{R_{\text{out}}} I(x)\,\mathrm{d}x}{\int_{R_{\text{out}}} \mathrm{d}x},$$

and β is a hyper-parameter.

We would like to minimize this functional by gradient descent, and the task comes down to computing the gradient flow for the curve C. The overall gradient flow is the sum of the two terms, one based on data term and the other based on the shape

[6] For instance, one can use a classification algorithm like the one in [19]. However, the algorithm in [19] needs to know the number of the modes in advance, whereas our approach does not require such information.

[7] From now on we drop the hat for simplicity in the density estimate $\hat{p}_C(C)$ and the pose estimate $\hat{\mathbf{p}}$.

prior. The gradient flow $\frac{\partial C}{\partial t}$ for the data term is given by

$$\frac{\partial C}{\partial t} = \beta[-(I(x) - m_{\text{inside}})^2 + (I(x) - m_{\text{outside}})^2]\vec{N}, \tag{30}$$

where $\vec{N}$ is an outward normal of a curve. In this section, we focus on describing how to compute the gradient flow $\frac{\partial C}{\partial t}$ for maximizing the shape term $\log p_C(C)$.

However, we cannot compute $\frac{\partial C}{\partial t}$ directly from the shape prior, since as was mentioned in Section 2.1, the shape prior

$$\log p_C(C) = \log(\gamma p_{\tilde{C}}(\tilde{C})) \tag{31}$$

$$= \log \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma) + \log \gamma \tag{32}$$

basically compares the aligned curve $\tilde{C} = T[\mathbf{p}]C$ with the training curves $\{\tilde{C}_i\}$ and is given in terms of those aligned curves $\tilde{C}$ and $\{\tilde{C}_i\}$. Hence, we first compute $\frac{\partial \tilde{C}}{\partial t}$ from the shape prior, and then compute $\frac{\partial C}{\partial t}$ from $\frac{\partial \tilde{C}}{\partial t}$.

To this end, we need a pose parameter $\mathbf{p}$ for curve C at each time, and the pose $\mathbf{p}$ should be updated concurrently as the curve C evolves. The updates of C and $\mathbf{p}$ are performed iteratively according to Algorithm 1.

Algorithm 1: Iterative algorithm for updating the pose estimate $\mathbf{p}$ and the curve C

```
(1) Evolve the curve $C$ without the shape prior for time $t \in [0, t_0]$
(2) For the curve $C|_{t=t_0}$, compute the pose $\mathbf{p}|_{t=t_0}$ by aligning $C|_{t=t_0}$ with respect to $\{\tilde{C}_i\}$
(3) Iterate until convergence:
    (a) fix $\mathbf{p}$ and
        (i) compute $\tilde{C} = T[\mathbf{p}]C$.
        (ii) compute $\frac{\partial \tilde{C}}{\partial t}$ from the shape prior $\log p_{\tilde{C}}(\tilde{C}) = \log \frac{1}{n}\sum_{i=1}^{n} k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma)$
        (iii) compute $\frac{\partial C}{\partial t}$ from $\frac{\partial \tilde{C}}{\partial t}$ by $\frac{\partial C}{\partial t} = T^{-1}[\mathbf{p}]\frac{\partial \tilde{C}}{\partial t}$
    (b) update $C$ by both the data driven force and the shape driven force
    (c) fix $C$ and
        (i) compute $\frac{\partial \mathbf{p}}{\partial t}$ using the alignment energy functional in Eq. (9)
        (ii) update the pose parameter $\mathbf{p}$ by $\frac{\partial \mathbf{p}}{\partial t}$
```

All the steps except step 3(a)(ii) are straightforward, and we discuss step 3(a)(ii) in the following sections.

In following sections, we discuss how to compute the gradient flow $\frac{\partial \tilde{C}}{\partial t}$ for maximizing the logarithm of the shape prior probability. We first start with the Parzen density estimate with a generic distance metric and give a sufficient condition so that the gradient flow is computable in Section 3.1. In particular, as an example for which the gradient flow is computable, we consider Parzen density estimation with the template metric in Section 3.2. Next, in Section 3.3, we discuss the case where the metric is the Euclidean distance between two signed distance functions and describe how to evolve the curve in the direction of increasing the shape prior.

3.1. Gradient flow for the shape prior with a generic distance metric

In this section, we derive a gradient flow for the Parzen window shape prior with a general distance measure. It turns out that the gradient flow is given as a weighted average of several directions, where the ith direction is an optimal (gradient) direction that decreases the distance between the ith training shape and the evolving shape.

Let us begin by considering the shape term

$$\log p_{\tilde{C}}(\tilde{C}) = \log\left(\frac{1}{n}\sum_i k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma)\right), \tag{33}$$

where $d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)$ is a generic distance measure between the shape described by the curve $\tilde{C}$ and the ith training shape described by $\tilde{C}_i$. Now we need to compute a velocity field f in curve evolution $\frac{\partial \tilde{C}}{\partial t} = f\vec{N}$ that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly.

The time derivative of $\log p_{\tilde{C}}(\tilde{C})$ is given by

$$\frac{\partial \log p_{\tilde{C}}(\tilde{C})}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma)\frac{\partial d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)}{\partial t}. \tag{34}$$

Now suppose that the last term $\frac{\partial d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)}{\partial t}$ is given in the form of $\oint_{\tilde{C}} \langle \frac{\partial \tilde{C}}{\partial t}, f_i\vec{N} \rangle \, ds$, i.e. $\frac{\partial \tilde{C}}{\partial t} = -f_i\vec{N}$ decreases $d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)$ most rapidly, then we have

$$\frac{\partial \log p_{\tilde{C}}(\tilde{C})}{\partial t} = \oint_{\tilde{C}} \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma) \times \left\langle \frac{\partial \tilde{C}}{\partial t}, f_i\vec{N} \right\rangle ds \tag{35}$$

$$= \oint_{\tilde{C}} \left\langle \frac{1}{p_{\tilde{C}}(\tilde{C})} \frac{1}{n} \sum_i k'(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma)\, f_i \vec{N}, \frac{\partial \tilde{C}}{\partial t} \right\rangle \mathrm{d}s \tag{36}$$

and we have the following gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly

$$\frac{\partial \tilde{C}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})} \frac{1}{n} \sum_i k'(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma)\, f_i \vec{N}. \tag{37}$$

In our work, we use a Gaussian kernel $k(x,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2})$, and we have $k'(x,\sigma) = k(x,\sigma)(-\frac{x}{\sigma^2})$. Thus the gradient flow is given by

$$\begin{aligned}\frac{\partial \tilde{C}}{\partial t} = {} & \frac{1}{p_{\tilde{C}}(\tilde{C})} \frac{1}{n} \frac{1}{\sigma^2} \sum_i k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma) \\ & \times d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)(-f_i)\vec{N}\end{aligned} \tag{38}$$

which is a linear combination of n terms $\{-f_i \vec{N}\}_{i=1}^n$, where the ith term contributes a force that decreases the distance $d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)$ most rapidly, and the weight for the ith term is given by $k(d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i), \sigma) d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)$. Now the question is whether we can find a gradient flow $\frac{\partial \tilde{C}}{\partial t} = -f_i \vec{N}$ for decreasing a distance $d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)$.

3.2. Gradient flow for the shape prior with the template metric

As we have seen above, if we can write the term $\frac{\partial d_{\mathscr{C}}(\tilde{C}, \tilde{C}_i)}{\partial t}$ in the form of $\oint_{\tilde{C}} \langle \frac{\partial \tilde{C}}{\partial t}, f_i \rangle \mathrm{d}s$, we have the gradient flow for $\log p_{\tilde{C}}(\tilde{C})$ in closed form. The template metric is such an example, and in this section we compute the gradient flow for the shape prior with the template metric introduced in Section 2.3.2.

Consider the template metric $d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) = \mathrm{Area}(R_{\text{inside } \tilde{C}} \triangle R_{\text{inside } \tilde{C}_i})$. This metric can be written in the form of region integrals as follows:

$$\begin{aligned} d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) &= \int_{\Omega} (1 - H(\phi_{\tilde{C}}(x))) H(\phi_{\tilde{C}_i}(x))\, \mathrm{d}x \\ &\quad + \int_{\Omega} H(\phi_{\tilde{C}}(x))(1 - H(\phi_{\tilde{C}_i}(x)))\, \mathrm{d}x \\ &= \int_{R_{\text{inside } \tilde{C}}} H(\phi_{\tilde{C}_i}(x))\, \mathrm{d}x \\ &\quad + \int_{R_{\text{outside } \tilde{C}}} (1 - H(\phi_{\tilde{C}_i}(x)))\, \mathrm{d}x, \end{aligned} \tag{39}$$

where $\phi_{\tilde{C}}$ and $\{\phi_{\tilde{C}_i}\}$ are signed distance functions for $\tilde{C}$ and $\{\tilde{C}_i\}$, respectively, and $H(\cdot)$ is the Heaviside function, i.e. $H(\phi) = 1$ if $\phi \geqslant 0$ and $H(\phi) = 0$ if $\phi < 0$. For the region integrals in (39), the derivative is well known [3], which is given by

$$\frac{\partial d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)}{\partial t} = \oint_{\tilde{C}} \left\langle \frac{\partial \tilde{C}}{\partial t}, (2H(\phi_{\tilde{C}_i}(s)) - 1) \right\rangle \mathrm{d}s. \tag{40}$$

By substituting $f_i = (2H(\phi_{\tilde{C}_i}(s)) - 1)$ into (38), we have the following gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ based on the template metric most rapidly:

$$\begin{aligned}\frac{\partial \tilde{C}}{\partial t} = {} & \frac{1}{p_{\tilde{C}}(\tilde{C})} \frac{1}{n} \frac{1}{\sigma^2} \sum_i k(d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i), \sigma) \\ & \times d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)(1 - 2H(\phi_{\tilde{C}_i}))\vec{N}.\end{aligned} \tag{41}$$

Fig. 6 illustrates the ith component of this shape force. Note that $(1 - 2H(\phi_{\tilde{C}_i}))$ is 1 inside $\tilde{C}_i$ and -1 outside $\tilde{C}_i$.

3.3. Approximation of the gradient flow for the shape prior with the Euclidean distance

Now we deal with the problem of evolving the curve $\tilde{C}$ so that we increase the shape prior with the $L2$ distance in (22). Since the shape prior in this case is given in terms of signed distance functions $\phi_{\tilde{C}}$ and $\phi_{\tilde{C}_i}$, we derive the evolution of the signed distance function $\phi_{\tilde{C}}$, which is equivalent to evolution of the curve $\tilde{C}$. When evolving $\phi_{\tilde{C}}$, it is desirable to keep $\phi_{\tilde{C}}$ to be a signed distance function in order that the shape density estimate in (22) is meaningful density estimate, since the L_2 distance $d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i})$ becomes less accurate as a shape distance measure as $\phi_{\tilde{C}}$ moves away from the manifold of signed distance functions. In addition, when the level set function is off the manifold, the evolution of the zero level set can be less stable than the case where the evolving level set function is constrained to be a signed

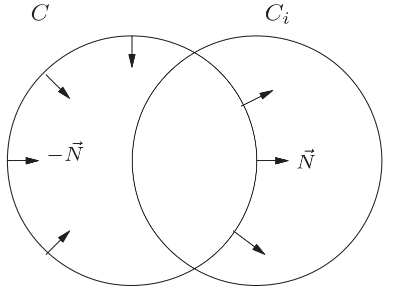

Fig. 6. Illustration of the shape force that decreases the template metric $d_{\mathscr{C}}(C, C_i) = \mathrm{Area}(R_{\text{inside } C} \triangle R_{\text{inside } C_i})$. $\vec{N}$ is the outward unit normal vector.

distance function [23]. With this constraint, there are two ways to compute the evolution equations for $\phi_{\tilde{C}}$. One is to directly compute the gradient flow with the constraint that $\phi_{\tilde{C}}$ remains on the manifold of signed distance functions. The other is to compute gradient flow without the constraint and then modify the evolution equation such that $\phi_{\tilde{C}}$ remains on the manifold of signed distance functions. We choose the second approach here as the first approach may not give a solution in closed form or result in a complicated solution even if there is a solution in closed form.

3.3.1. *Unconstrained gradient flow of level set functions*

Without the constraint that the evolving level set function stays on the manifold $\mathscr{D}$, we compute the gradient flow for $\log p_{\tilde{C}}(\tilde{C})$

$$\log p_{\tilde{C}}(\tilde{C}) = \log \frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma), \tag{42}$$

where $\phi_{\tilde{C}_i}$ is the signed distance function for the ith training shape. Note that $\phi_{\tilde{C}}$ is a function of the time t and $\phi_{\tilde{C}}$ is a shorthand notation for the evolving level set function $\phi_{\tilde{C}}(t)$. Using a Gaussian kernel, we have

$$\begin{aligned} &k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\ &\quad = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\int_\Omega (\phi_{\tilde{C}}(x) - \phi_{\tilde{C}_i}(x))^2 \, \mathrm{d}x\right). \end{aligned} \tag{43}$$

By differentiating the above expression, we have

$$\begin{aligned} &\frac{\partial}{\partial t} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\ &\quad = k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\ &\qquad \times \left[-\frac{1}{2\sigma^2}\int_\Omega 2(\phi_{\tilde{C}}(x) - \phi_{\tilde{C}_i}(x)) \frac{\partial \phi_{\tilde{C}}}{\partial t}(x)\, \mathrm{d}x\right]\\ &\quad = \frac{1}{\sigma^2} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}))\left\langle \phi_{\tilde{C}_i} - \phi_{\tilde{C}}, \frac{\partial \phi_{\tilde{C}}}{\partial t}\right\rangle. \end{aligned} \tag{44}$$

Let us now differentiate $\log p_{\tilde{C}}(\tilde{C})$ in (42).

$$\frac{\partial}{\partial t}\log p_{\tilde{C}}(\tilde{C}) = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i \frac{\partial}{\partial t} k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma) \tag{45}$$

$$\begin{aligned} &= \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\\ &\qquad \times \left\langle \phi_{\tilde{C}_i} - \phi_{\tilde{C}}, \frac{\partial \phi_{\tilde{C}}}{\partial t}\right\rangle \end{aligned} \tag{46}$$

$$\begin{aligned} &= \left\langle \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)\right.\\ &\qquad \left. \times(\phi_{\tilde{C}_i} - \phi_{\tilde{C}}), \frac{\partial \phi_{\tilde{C}}}{\partial t}\right\rangle. \end{aligned} \tag{47}$$

Thus the gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly is

$$\frac{\partial \phi_{\tilde{C}}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)(\phi_{\tilde{C}_i} - \phi_{\tilde{C}}) \tag{48}$$

This velocity field is given by a weighted average of $\{\phi_{\tilde{C}_i} - \phi_{\tilde{C}}\}_{i=1}^n$, where $\phi_{\tilde{C}_i} - \phi_{\tilde{C}}$ is the direction toward the ith training shape $\phi_{\tilde{C}_i}$, and the corresponding weight is $k(d_{L_2}(\phi_{\tilde{C}}, \phi_{\tilde{C}_i}), \sigma)$. Note that the weight for the velocity component $\phi_{\tilde{C}_i} - \phi_{\tilde{C}}$ increases as $\phi_{\tilde{C}}$ gets closer to $\phi_{\tilde{C}_i}$. As a result, an example shape that is closer to the current shape becomes more important during the evolution of the shape.

3.3.2. *Modifying the evolution equation*

Now we describe how we modify the evolution Eq. (48) such that the evolving level set function remains a signed distance function. We start by rewriting the update Eq. (48) and defining the velocity field $v(\cdot)$ as follows:

$$\begin{aligned} &\frac{\partial \phi_{\tilde{C}}(x,t)}{\partial t}\\ &\quad = \frac{1}{p_{\tilde{C}}(\tilde{C}(t))}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}}(\cdot,t), \phi_{\tilde{C}_i}), \sigma)\\ &\qquad \times(\phi_{\tilde{C}_i}(x) - \phi_{\tilde{C}}(x,t)) \triangleq v(x), \end{aligned} \tag{49}$$

where we introduce pixel x and time t explicitly to the velocity field expression.

Now we modify the evolution in (49) and construct a new velocity field $v_{\mathrm{new}}(\cdot)$ which guarantees that the evolving level set function is a signed distance function. The goal here is to extract relevant information for shape evolution from the velocity field $v(\cdot)$ and to construct $v_{\mathrm{new}}(\cdot)$ such that the resulting trajectory of $\phi(\cdot,t)$ is on the space $\mathscr{D}$.

First we observe that the only components of the velocity field $v(\cdot)$ that directly impact the shape evolution are those defined at the points on the boundary $\tilde{C}(t) = \{x|\phi_{\tilde{C}}(x,t) = 0\}$. In this respect, we take $v_{\mathrm{new}}(x) = v(x)$ if $x \in \tilde{C}$. The next key property is that as long as the velocity v_{new} remains constant along the direction normal to the curve $\tilde{C}$, the evolving level set function $\phi_{\tilde{C}}(t)$ remains a signed distance function [24]. Since we have defined values of $v_{\text{new}}(\cdot)$ at all the boundary points, we can extend these values in the direction normal to the boundary. Such a procedure is equivalent to setting the velocity $v_{\text{new}}(x)$ at a point x to be equal to the boundary velocity $v(x_{\tilde{C}})$, where $x_{\tilde{C}}$ is the boundary point closest to the point x.

In summary, we update the level set function $\phi_{\tilde{C}}$ by the modified velocity $v_{\text{new}}(\cdot)$ as follows:

$$\frac{\partial\phi_{\tilde{C}}(x,t)}{\partial t} = v_{\text{new}}(x) = v(x_{\tilde{C}}), \tag{50}$$

where $x_{\tilde{C}}$ is the point on the curve closest to the point x.

This $v_{\text{new}}(\cdot)$ is an approximation of the gradient flow which maximizes the change in the energy.

3.4. Discussion on the gradient flows

In this section, we consider the gradient flow induced by the shape prior with the L_2 distance and compare that with the shape constraint obtained by the PCA-based approach in [10]. Then we add some comments about the gradient flow induced by the shape prior with the template metric.

Let us consider the shape force due to L_2 norm in (48). Such shape force will evolve the curve toward a shape at local maximum of the shape prior, which is approximately a weighted average of the neighboring training shapes. Although the actual shape force is modified version (Section 3.3.2) of Eq. (48), this equation gives a useful interpretation of a shape at a local maximum of the shape prior as follows. At the local maximum of $p_{\tilde{C}}(\tilde{C})$, the gradient flow will be zero.

$$\frac{\partial\phi_{\tilde{C}}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}),\sigma)(\phi_{\tilde{C}_i}-\phi_{\tilde{C}}) \tag{51}$$

$$= \frac{1}{\sigma^2}\sum_i \lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})(\phi_{\tilde{C}_i}-\phi_{\tilde{C}}) \tag{52}$$

$$= 0, \tag{53}$$

where $\lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) = \frac{k(d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}),\sigma)}{np_{\tilde{C}}(\tilde{C})}$ and $\sum_i\lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i}) = 1$. Hence, the shape at the local maximum is given as

$$\phi_{\tilde{C}} = \sum_i \lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})\phi_{\tilde{C}_i} \tag{54}$$

At first sight, this is a linear combination of training shapes, hence one can raise an issue that this approach might have the same problem that PCA-based approaches have, namely, the space of signed distance functions is not linear and such linear combination would be far off the manifold. However, the linear combination has nonlinear weight function $\lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})$ that emphasizes only the training samples within a local neighborhood of the current candidate shape $\phi_{\tilde{C}}$, which we explain below. The weight function is a decreasing function of the distance $d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})$, and in particular, the weight function $\lambda_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})$ will be negligible if $\phi_{\tilde{C}_i}$ is not in the local neighborhood (or within the same mode) of $\phi_{\tilde{C}}$ (i.e. $d_{L_2}(\phi_{\tilde{C}},\phi_{\tilde{C}_i})$ is large enough compared to the kernel size σ). Thus the shape at the local maximum is approximately given as a weighted average of training shapes in local neighborhood. Note that the weight function behaves as a selection function of the local neighbors or samples in the same cluster (or in the same mode). Therefore, the linear combination in (54) is not far away from the manifold. We can also say that only locally relevant shapes are activated by the shape prior and the shape force. In addition, this property gets stronger as the kernel size gets smaller, i.e. the neighboring samples contributing to the shape at the local maximum are more localized, thus the part of the manifold that supports such neighboring samples will be more linear or flat.

In contrast, the PCA-based approach in [10], the candidate shape is constrained to be a linear combination of training samples, yet without a special selection mechanism. In particular, the shape of the segmentation is constrained to be sum of the average shape and a linear combination of eigen-shapes as follows:

$$\phi(\alpha_1,\ldots,\alpha_k) = \phi_{\text{avg}} + \sum_{i=1}^{k}\alpha_i\phi_{\text{eig},i}, \tag{55}$$

where the eigen-shapes $\{\phi_{\text{eig},i}\}$ are obtained by PCA, and according to the algorithm, each of the eigen-shapes is again a linear combination of training shapes. Hence, any candidate shape $\phi(\alpha_1,\ldots,\alpha_k)$ is a linear combination of the training shapes, and the set of possible candidate shapes is contained in the linear subspace spanned by all the training shapes:

$$\{\phi(\alpha_1,\ldots,\alpha_k)\} \subset \text{span}\{\phi_{\tilde{C}_1},\ldots,\phi_{\tilde{C}_n}\}. \tag{56}$$

When the training shapes are localized, the right-hand side of (56) would be also localized around the shape manifold (space of signed distance functions) thereby making the set of candidate shapes localized, too. However, when the training shapes are multi-modal or broadly distributed, a candidate shape $\phi(\alpha_1, \ldots, \alpha_k)$ is likely to be a linear combination of broadly distributed samples. This increases the chance of such a candidate shape being off the manifold and also being a less typical shape. In other words, the shape constraint obtained by PCA analysis can be not restrictive enough when the training samples are broadly distributed.

Finally, we briefly compare the gradient flow for the density estimate with L_2 distance and the one for the density estimate with the template metric. Let us consider the gradient flow equation (41)

$$\begin{aligned}\frac{\partial \tilde{C}}{\partial t} &= \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\frac{1}{\sigma^2}\sum_i k(d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i), \sigma) \\ &\quad \times d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)(1 - 2H(\phi_{\tilde{C}_i}))\vec{N} \qquad (57) \\ &= \frac{1}{\sigma^2}\sum_i \lambda_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)(1 - 2H(\phi_{\tilde{C}_i}))\vec{N}, \qquad (58)\end{aligned}$$

where $\lambda_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) = \frac{k(d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i), \sigma)}{n p_{\tilde{C}}(\tilde{C})}$ and $\sum_i \lambda_{\mathrm{T}}(\tilde{C}, \tilde{C}_i) = 1$. This gradient flow also has a nonlinear weighting function $\lambda_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)$, which selects training shapes in a local neighborhood of the current segmenting shape. The major differences between the two gradient flows are in the flexibility or the degree of the freedom. In particular, the ith component $d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)(1 - 2H(\phi_{\tilde{C}_i}))\vec{N}$ of the gradient flow with the template metric has uniform magnitude $d_{\mathrm{T}}(\tilde{C}, \tilde{C}_i)$ over the entire curve with $(1 - 2H(\phi_{\tilde{C}_i}))$ changing only signs. As each component of the gradient flow has pretty simple structure, the overall gradient flow in (58), which is a linear combination of simple component flows, will still have less degree of the freedom than the gradient flow in the case of L_2 distance. As the shape prior proposed in [16] is also based on the template metric, the gradient flow therein has the same property that it has limited degree of freedom.

4. Experimental results

Now we present experimental results demonstrating our segmentation method based on nonparametric shape priors. We first show results for segmenting occluded objects. Here the shape prior involves a single class of object shapes. Next, we present experimental results on the problem of segmenting hand-written digits (with low quality or missing data), where the prior density involves all digits, and the algorithm does not know which digit the test data is. This problem is more challenging as the prior density is now multimodal.

4.1. Segmentation of occluded objects with various poses

In this section, we demonstrate our shape-based segmentation algorithm with the segmentation of synthetic aircraft images. As example shapes of this class, we have a set of 11 binary images displayed in Fig. 7, whose boundaries provide the training curves

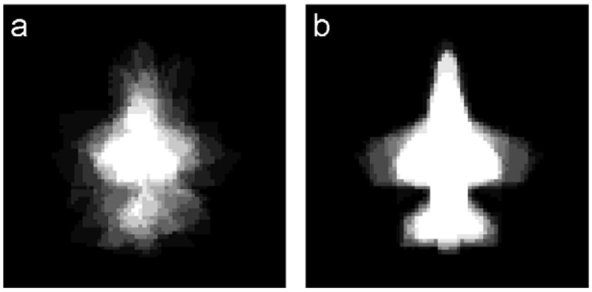

Fig. 9. Overlay of training samples of the aircraft shape: (a) before alignment: (b) after alignment. The images (a) and (b) are generated by taking an average of the binary images in Figs. 7 and 8, respectively.

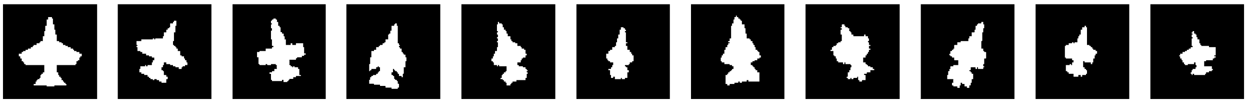

Fig. 7. Training samples of the aircraft shape before alignment.

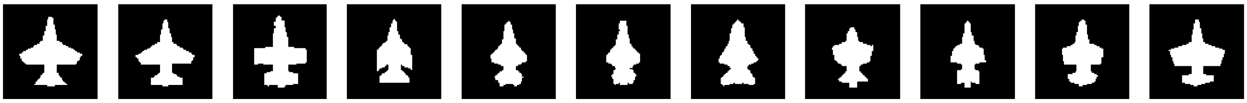

Fig. 8. Aligned training samples of the aircraft shape.

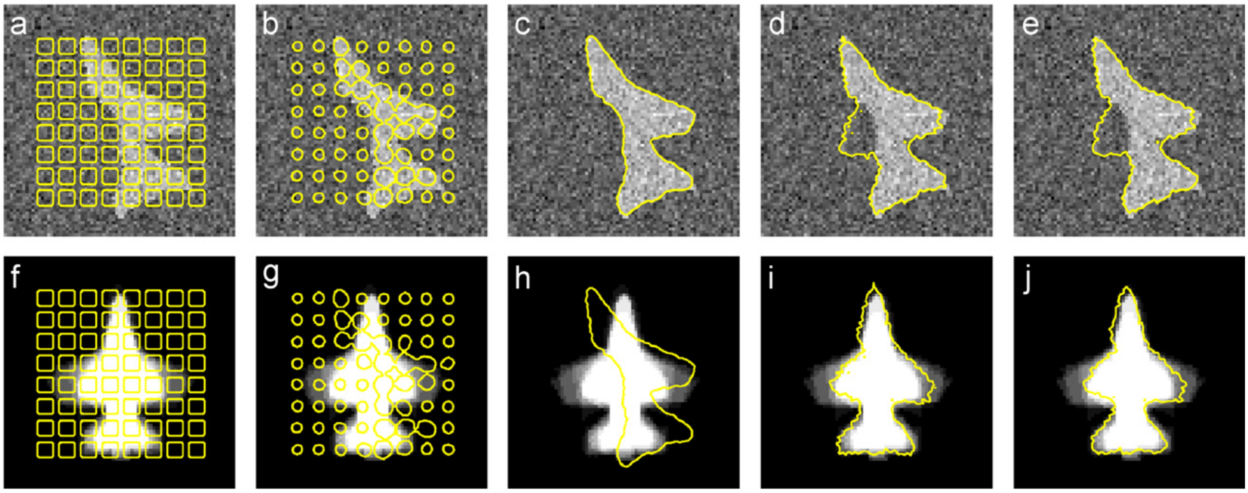

Fig. 10. Segmentation of an occluded aircraft image (rotated) using Parzen shape prior with L_2 distance between signed distance functions. The first row, (a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Fig. 9(b).

$C_1, \ldots, C_n$ $(n = 11)$. Fig. 8 shows the training shapes after alignment, hence the boundaries of these binary images correspond to the aligned training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. Fig. 9(a) and (b) contain overlaid images of the training samples, showing the amount of overlap among training shapes before and after alignment, respectively, and providing a picture of the shape variability.

We now present segmentation results on the image of an aircraft whose shape was not included in the training set. In particular, Fig. 10 shows the noisy aircraft test image with an occluded left wing as well as its segmentation using the Parzen shape prior with L_2 distance between signed distance functions. The first row, (a)–(e), shows the evolving curve C on top of the image to be segmented, and the second row, (f)–(j), shows the transformed curve $\tilde{C} = T[\mathbf{p}]C$ on top of the aligned training shapes shown in Fig. 9(b). In our shape-based segmentation, we evolve the curve as is given in Algorithm 1. First, the curve evolves without the shape prior (using a curve length regularization term instead) as shown in (a)–(c), which corresponds to the step 1 of Algorithm 1. After the curve finds all the portions of the object boundary except those occluded as shown in (c),[8] the shape force is turned on, and both the data force and shape force are applied during the stages (c)–(e). Note that the pose parameter $\mathbf{p}$ is updated as is shown in (i) and (j) while the curve evolves as in (d) and (e). This procedure is more desirable than turning on the shape force from the start, since during the initial stages of the curve evolution, the pose estimate may not be accurate and in that case the shape force might deform the curve with an inaccurate pose estimate. Note that while the shape force is turned off, we need no pose estimates and we have $\tilde{C} = C$. We also note that our algorithm does not have access to any information about which parts of the image contain occlusions. At the final segmentation the shape force and data force are in equilibrium. For instance the data force at the boundary of the left wing will try to shrink the left wing to match the given data, whereas the shape force tries to expand the left wing to increase the fidelity to the shape prior.

In these experiments, we have an issue of how to balance the data force and the shape force. We balance the two forces by subjectively (depending on the SNR of the images) choosing the hyperparameter β in the data driven energy term (29). A rule of thumb is that the higher the noise variance the smaller the data force parameter β.

Fig. 11 shows the results with the same test image with a different shape prior, namely the density estimate with the template metric. As the intermediate steps of the curve evolution are similar to Figs. 10 and 11 for the test images with other poses, in the remainder of this section, we present only the final segmentation results for the aircraft images. Fig. 12(a) is the same image as the one

[8] At stage (c), the curve has converged with the data force and the curve shortening term. Such convergence is detected automatically and then the shape force is turned on.

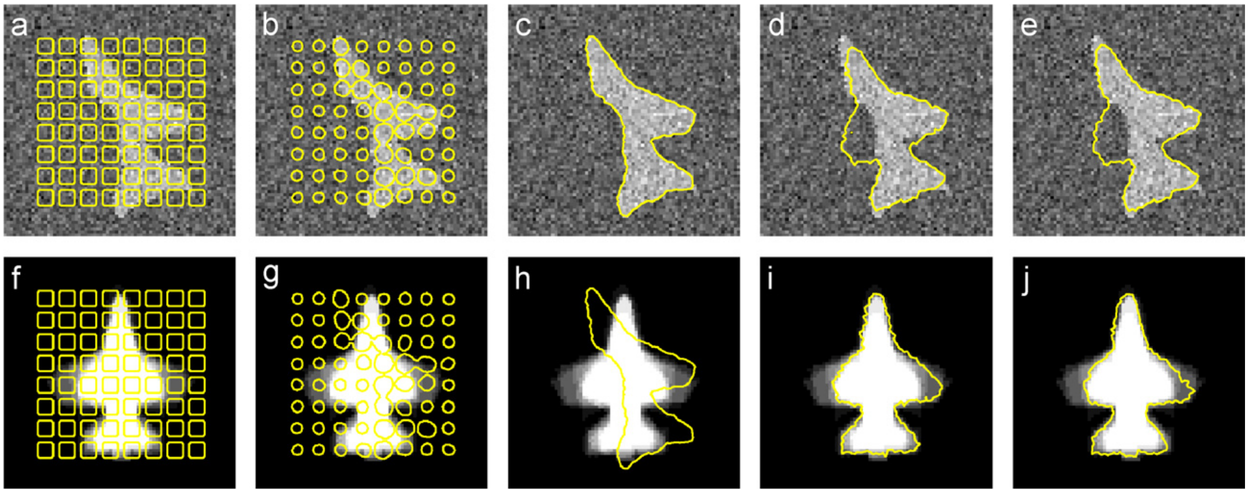

Fig. 11. Segmentation of an occluded aircraft image (rotated) using Parzen shape prior with the template metric. The first row, (a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Fig. 9(b).

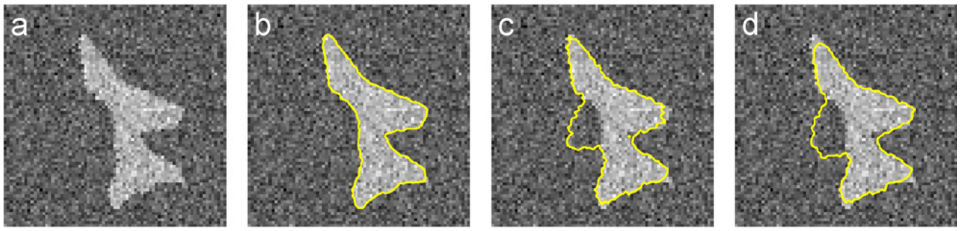

Fig. 12. Segmentation of an occluded aircraft image involving rotation: (a) test image; (b) result without shape prior; (c) nonparametric shape prior with the L_2 distance; (d) nonparametric shape prior with the template metric.

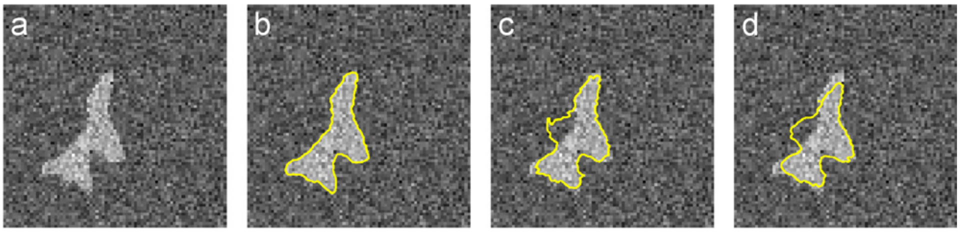

Fig. 13. Segmentation of an occluded aircraft image involving rotation, scaling, and translation: (a) test image; (b) result without shape prior; (c) nonparametric shape prior with the L_2 distance; (d) nonparametric shape prior with the template metric.

shown in Fig. 10, and we present the segmentation result without a shape prior (and with a curve length penalty instead) and the segmentation results obtained by two different shape priors, namely the nonparametric shape prior with the L_2 distance and the nonparametric shape prior with the template metric. Fig. 13 shows a segmentation example involving a rotated, scaled and translated version of the same object.

In these experiments, the nonparametric shape prior with the L_2 distance leads to better segmentation results than the template metric, especially at the front end of the aircraft. This difference seems to come from the nature of gradient flow we discussed in Section 3.4, namely whether the component shape force is variable or constant around the curve, which we explain now. We examine the segmentation result in Fig. 12(c) and the inaccurate segmentation around the front end of the aircraft in Fig. 12(d) by first inspecting the aligned curve $\tilde{C}$ and the aligned training curves in Figs. 10(j) and 11(j). Figs. 10(j) and 11(j) show that the front end portion of the aligned segmenting curve $\tilde{C}$ is outside of the same portion of training shapes. As a result, the shape force will be in a direction to shrink the front end further inside to match the training shapes. The issue is then how large the magnitude of such shape force is relative to the magnitude of the data force. With the L_2 distance, when a portion of the segmenting curve gets near to the same portion of the aligned training curves, the shape force around that portion of the boundary decreases. For instance, in Fig. 10(e), the shape force around the front end will be much smaller than that around the left wing. This variability of shape force around the boundary explains why the front end does not shrink further with the L_2 distance. In contrast, the shape force due to the template metric is controlled by only whether the portion of the boundary is inside or outside the training shape without further adjusting its magnitude. As a result, with the alignment shown in Fig. 11(j), the shape force will try to move the portion of the curve around the front end further inside even though that portion of the boundary is pretty close to the same portion of the training shapes.

In all of these examples, we have reasonable segmentation despite the occlusions. These results demonstrate that our segmentation algorithm can locate an object with an arbitrary pose, when we have prior knowledge about the shape of the object. Since the training shapes are locally distributed, the method based on linear shape prior such as PCA will also work for these images as demonstrated by the work of Tsai et al. [10].

4.2. Segmentation of handwritten digit images

We now consider the problem of segmenting handwritten digits, where there are 10 handwritten

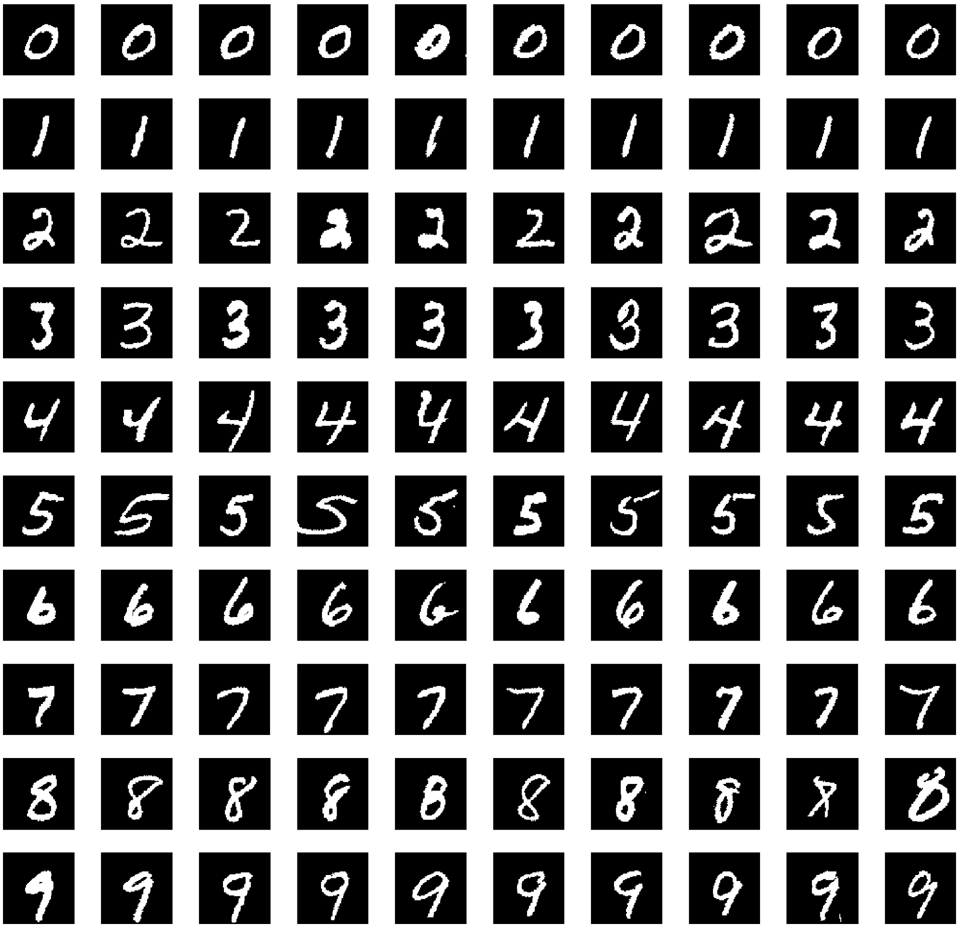

Fig. 14. Training data for handwritten digits; Courtesy of Erik Learned–Miller

digits, i.e. $0, 1, \ldots, 9$. As the prior density is now multimodal, this is a scenario which cannot be readily handled by most existing shape-based segmentation techniques. We use a training set of 100 sample images with 10 segmented images of each digit as shown in Fig. 14. In this experiment, the training shapes and test shapes are already aligned, so we fix the pose parameters $\mathbf{p}_i$ for the training curves and the pose parameter $\mathbf{p}$ for the evolving curve to be [0 0 0 1], the one for the identity transform. Hence $C_i = \tilde{C}_i$ and $C = \tilde{C}$, and we just use C_i and C to denote aligned curves.

Let us consider the low-SNR test images (not included in the training set) in Fig. 15(a). Segmentations without a shape prior (and with a curve length penalty instead) are shown in Fig. 15(b). The results of our shape-based segmentation method together with the results of PCA-based segmentation method of Tsai et al. [10] are shown in Fig. 16. The result of PCA-based segmentation in Fig. 16(a) looks better than the result without a shape prior in Fig. 15(b) as the shape prior constrains the evolving curve to be a linear combination of training shapes, i.e. to remain on the linear subspace spanned by the training shapes. However, as the training shapes are distributed broadly having multiple modes, such a linear subspace is not restrictive enough to obtain very good segmentation results. In contrast, the results of our shape-based segmentation look much better as shown in Figs. 16(b) and 16(c). This

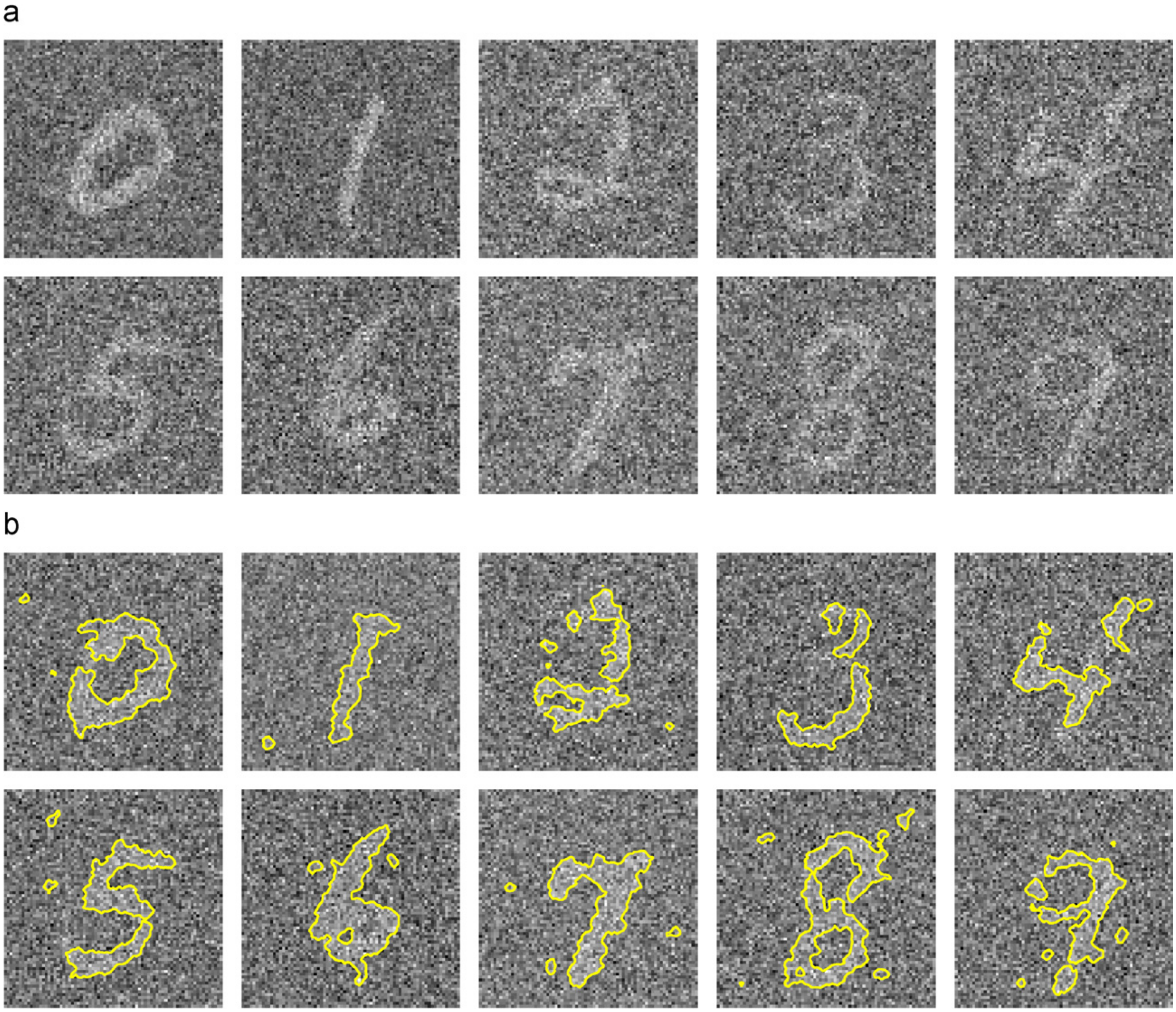

Fig. 15. Segmentation of low SNR digit images: (a) test images; (b) without shape prior.

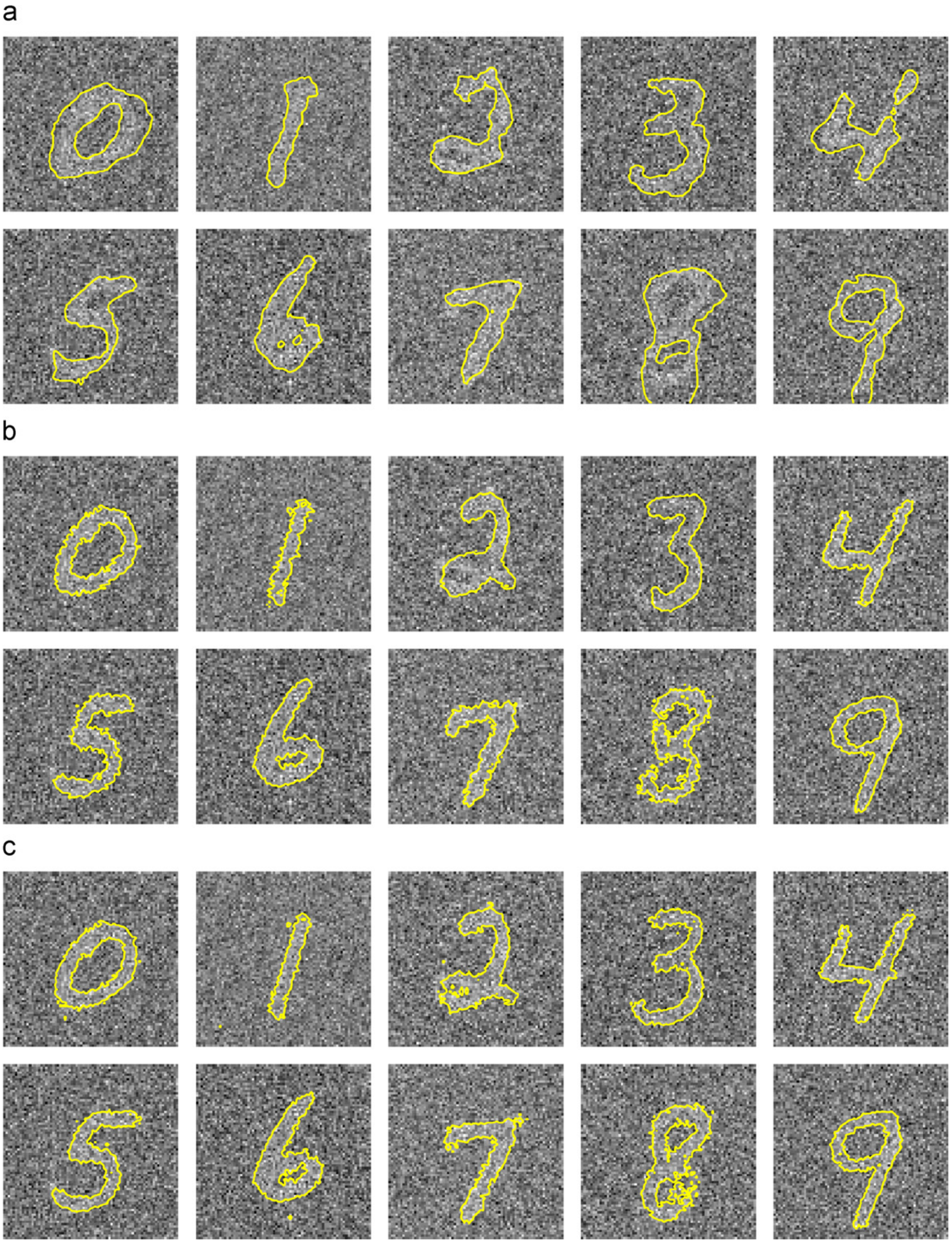

Fig. 16. Segmentation of low SNR digit images: (a) with linear prior (PCA); (b) nonparametric prior with the L_2 distance; (c) nonparametric prior with the template metric.

demonstrates that the nonparametric shape prior can effectively model the shape distribution composed of multiple clusters.

In kernel density estimation, choice of kernel size is an open issue and its choice depends on the application at hand [17]. In general, there is a

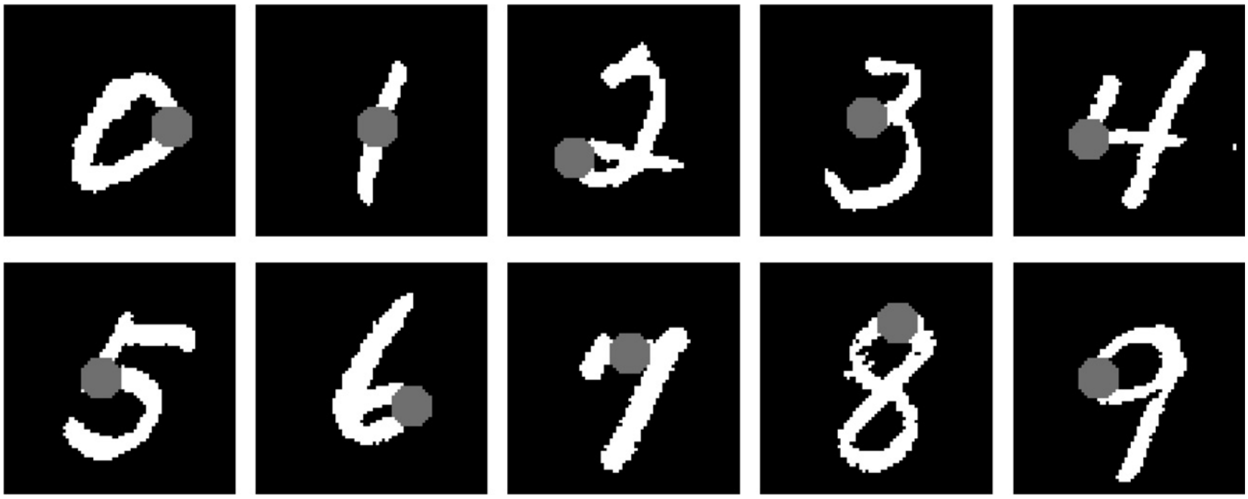

Fig. 17. Handwritten digits with missing data; each of these examples is not included in the training set in Fig. 14. The parts of missing data are displayed in gray.

tradeoff in choosing kernel size, namely if we choose it too small, the density estimate is dominated by the nearest training shape, whereas if we choose it too large, density estimate is over-smoothed across clusters. Our choice of kernel size for this data set is $\sigma = \delta\sigma_{\mathrm{ML}}$, a scaled version of the ML kernel size, which can be automatically estimated from the data, and the scale parameter δ is manually[9] chosen to be 0.2 in this application, in order to prevent over-smoothing across multiple clusters of samples.

Finally, we consider the problem of segmenting a handwritten digit image with missing data as well as additive noise. The gray region in Fig. 17 indicates where we do not have observations, and the test images are shown in Fig. 18(a). In this experiment, we assume that the algorithm knows which pixels are missing, that is the algorithm takes the occlusion mask as an additional input, and disregards the intensities of the pixels that fall under the mask. Since the curve evolution inside the region of missing data will not change the data-based energy term, the data driven force in that region would be zero. Hence, when we evolve the curve, the portion of the curve in the region of missing data will be evolved only by shape force whereas the other portion of the curve will be evolved by both the data force and the shape force.

Segmentations without a shape prior are shown in Fig. 18(b). Again the result of PCA-based segmentation in Fig. 19(a) looks better than the result without a shape prior in Fig. 18(b). However, we can see that the PCA-based shape prior is not restrictive enough as shown in the segmentation results of digits 1, 7, and 9. In contrast, our shape-based segmentation results in Fig. 19(b) and (c) provide fairly accurate segmentation despite the data limitations. We can also compare the segmentation results in Fig. 19 quantitatively by measuring the mismatch between the ground truth boundary C_{true} and the segmentation result C_{result}. We use the template metric $d_{\mathrm{T}}(C_{\mathrm{true}}, C_{\mathrm{result}})$ as such a measure of mismatch and provide the quantitative comparison in Table 1 and Fig. 20. In Table 1, for each digit image, we mark in boldface the minimum $d_{\mathrm{T}}(C_{\mathrm{true}}, C_{\mathrm{result}})$ over the three priors, where we can see that the shape prior with the template metric performed best most frequently and that PCA-based shape prior never performed best. Fig. 20 visualizes Table 1 and shows that most times the nonparametric shape priors give smaller error measures than PCA-based shape prior. In summary, we conclude that segmentation methods based on nonparametric shape priors outperform PCA-based segmentation method both qualitatively and quantitatively.

5. Conclusions

We have considered the problem of estimating shape prior densities from example shapes and proposed a shape-based segmentation method. In

[9] We can fix this scaling parameter and use one-fifth of the ML kernel size as a rule of thumb kernel size. This rule of thumb kernel size is automatically computed from the data.

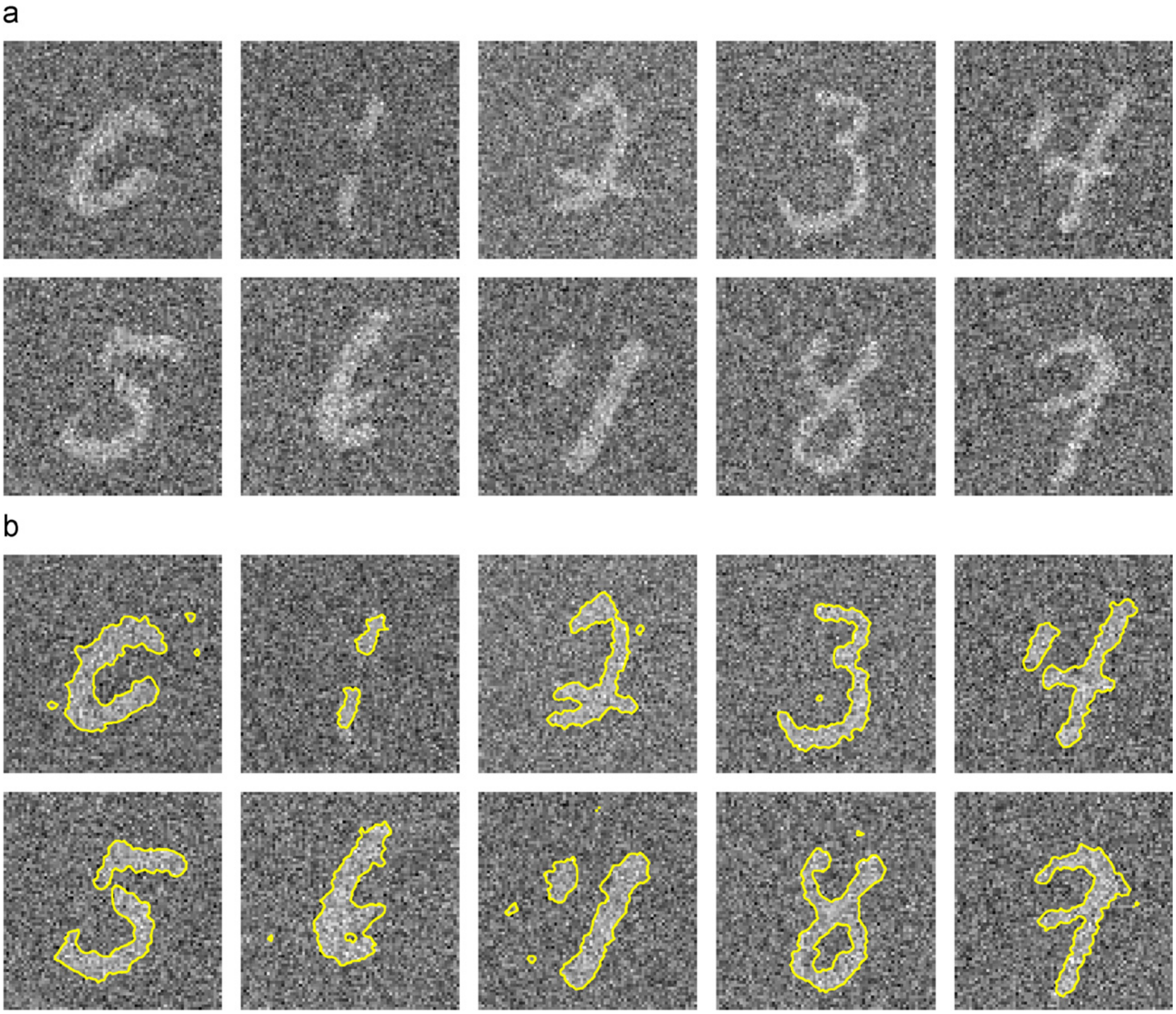

Fig. 18. Segmentation of digit images with missing data: (a) test images; (b) without shape prior.

particular, we have developed a framework for estimating shape priors from training shapes nonparametrically. Based on such nonparametric shape priors, we have formulated the shape-based segmentation problem as a MAP estimation problem. Evaluation of the nonparametric shape prior for a candidate curve for segmentation is given in terms of distances between the candidate curve and the training curves. We consider the L_2 distance between signed distance functions for shape density estimation, in addition to a distance measure based on the template metric. In particular, we consider the case in which the space of shapes is represented as a space of signed distance functions, which we interpret as a manifold embedded in a Hilbert space. We have derived curve evolution equations based on the nonparametric shape priors and provided comparison of the curve evolution equations for the two distance metrics. We have presented experimental results of segmenting partially occluded images, where the similarity transform of the object was handled by alignment. We have considered the case in which the training shapes form multiple clusters, and demonstrated that our nonparametric shape priors model such shape distributions successfully without requiring prior knowledge on the number of clusters. Though we considered the template metric and the L_2 distance between signed

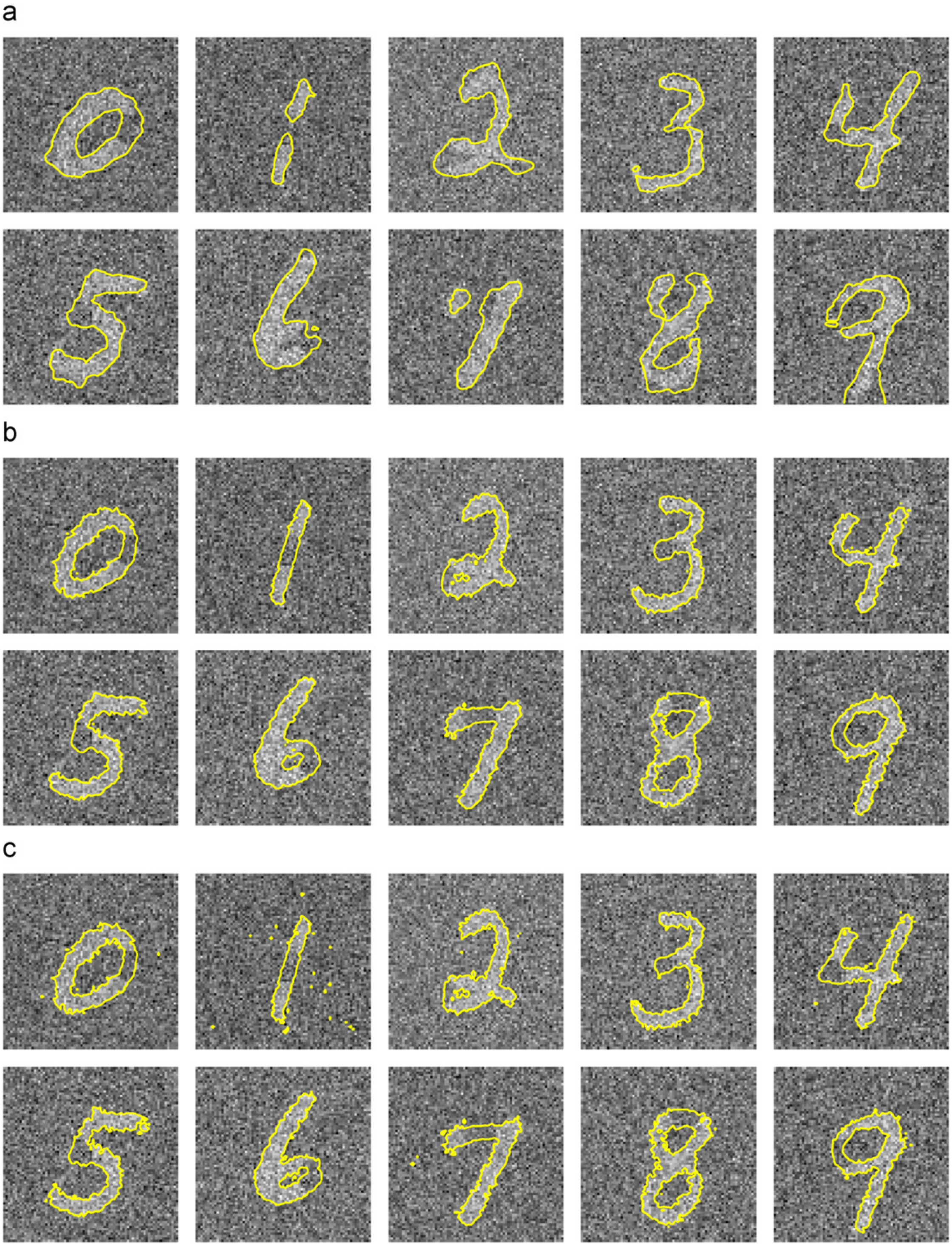

Fig. 19. Segmentation of digit images with missing data: (a) with linear prior (PCA); (b) nonparametric prior with the L_2 distance; (c) nonparametric prior with the template metric.

distance functions, other metrics can also be used for nonparametric shape priors in our framework. One such example is the Hausdorff metric [18] or its differentiable approximation introduced in [13], whose use for shape density estimation deserves some future work.

Table 1
Quantitative comparison of the segmentation results in Fig. 19

Digit	0	1	2	3	4	5	6	7	8	9
PCA	188	76	288	196	153	201	158	200	326	330
Prior with d_{L_2}	**106**	**33**	222	144	191	193	101	**120**	302	108
Prior with d_T	122	55	**205**	**137**	**129**	**165**	**89**	140	**293**	**105**

Each element of the table is given by the template metric $d_T(C_{\text{true}}, C_{\text{result}})$, which is a distance between the ground truth boundary curve C_{true} and the segmentation boundary C_{result} of the corresponding segmentation result.

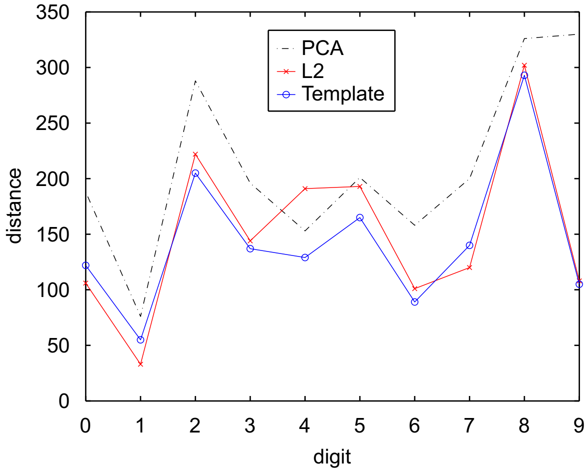

Fig. 20. Quantitative comparison of the segmentation results in Fig. 19. The template metric $d_T(C_{\text{true}}, C_{\text{result}})$ is used as a measure of mismatch between the ground truth and the segmentation result. This figure is obtained from Table 1.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research under Grant No. FA9550-04-1-0351 and Grant No. FA9550-06-1-0324, and in part by the European Commission under Grant MIRG-CT-2006-041919. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Air Force.

References

[1] D. Mumford, J. Shah, Boundary detection by minimizing functionals, in: Proc. IEEE Conf. Comput. Vision Pattern Recognition, 1985, pp. 22–26.

[2] D. Mumford, The problem of robust shape descriptors, in: Proceedings of the IEEE First International Conference on Computer Vision, 1987, pp. 602–606.

[3] S.C. Zhu, A. Yuille, Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, IEEE Trans. Pattern Anal. Machine Intell. 18 (9) (1996) 884–900.

[4] A. Yezzi Jr., A. Tsai, A. Willsky, A statistical approach to snakes for bimodal and trimodal imagery, in: International Conference on Computer Vision, 1999, pp. 898–903.

[5] T.F. Chan, L.A. Vese, An efficient variational multiphase motion for the mumford-shah segmentation model, in: Proceedings of Asilomar Conference on Signals, Systems, and Computers, 2002, pp. 490–494.

[6] A. Tsai, A. Yezzi Jr., A.S. Willsky, Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification, IEEE Trans. Image Process. 10 (8) (2001) 1169–1186.

[7] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the mumford and shah model, Internat. J. Comput. Vision 50 (3) (2002) 271–293.

[8] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models—their training and application, Comput. Vision Image Understanding 61 (1) (1995) 38–59.

[9] M.E. Leventon, W.E.L. Grimson, O. Faugeras, Statistical shape influence in geodesic active contours, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2000, pp. 316–323.

[10] A. Tsai, A. Yezzi Jr., W. Wells, C. Tempany, D. Tucker, A. Fan, W.E. Grimson, A. Willsky, A shape-based approach to the segmentation of medical imagery using level sets, IEEE Trans. Med. Imaging 22 (2) (2003) 137–154.

[11] E. Klassen, A. Srivastava, W. Mio, S.H. Joshi, Analysis of planar shapes using geodesic paths on shape spaces, IEEE Trans. Pattern Anal. Machine Intell. 26 (3) (2004) 372–383.

[12] P.W. Minchor, D. Mumford, Riemannian geometries on spaces of plane curves, J. Eur. Math. Soc. (JEMS).

[13] G. Charpiat, O. Faugeras, R. Keriven, Approximations of shape metrics and application to shape warping and empirical shape statistics, Technical Report, INRIA, May 2003.

[14] S. Soatto, A. Yezzi, Jr., Deformotion, deforming motion, shape average and the joint registration and segmentation of images, in: Proceedings of the Seventh European Conference on Computer Vision, 2002, pp. 32–47.

[15] T.F. Cootes, C.J. Twining, C.J. Taylor, Diffeomorphic statistical shape models, in: Proceedings of British Machine Vision Conference, vol. 1, 2004, pp. 447–456.

[16] D. Cremers, S.J. Osher, S. Soatto, Kernel density estimation and intrinsic alignment for shape priors in level set segmentation, Internat. J. Comput. Vision 69 (3) (2006) 335–351.

[17] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, London, 1986.

[18] D. Mumford, Mathematical theories of shape: do they model perception?, in: SPIE Proceedings, Geometric Methods in Computer Vision, vol. 1570, 1991, pp. 2–10.

[19] A. Tsai, W.M. Wells, S.K. Warfield, A.S. willsky, An EM algorithm for shape classification based on level sets, Med. Image Anal. 9 (5) (2005) 491–502.

[20] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (4) (1989) 577–685.

[21] T. Chan, L. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277.

[22] J. Kim, J.W. Fisher, A. Yezzi Jr., M. Çetin, A.S. Willsky, A nonparametric statistical method for image segmentation using information theory and curve evolution, IEEE Trans. Image Process. 14 (10) (2005) 1486–1502.

[23] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, Berlin, 2003.

[24] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.