Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2007) 3034–3045
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Kernel PCA for similarity invariant shape recognition

Hichem Sahbi

Machine Intelligence Laboratory, Department of Engineering, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, UK

Received 4 October 2005; received in revised form 4 April 2006; accepted 22 June 2006
Communicated by S. Choi
Available online 25 October 2006

Abstract

We present in this paper a novel approach for shape description based on kernel principal component analysis (KPCA). The strength of this method resides in the similarity (rotation, translation and particularly scale) invariance of KPCA when using a family of triangular conditionally positive definite kernels. Beside this invariance, the method provides an effective way to capture non-linearities in shape geometry.

A given two-dimensional curve is described using the eigenvalues of the underlying manifold modeled in a high-dimensional Hilbert space. Using Fourier analysis, we will show that this eigenvalue description captures low to high variations of the shape frequencies. Experiments conducted on standard databases including the SQUID, the Swedish and the Smithsonian leaf databases, show that the method is effective in capturing invariance and generalizes well for shape matching and retrieval.

Keywords: Statistical learning; Kernel principal component analysis; Scale invariance; Triangular kernel; Shape description; Image retrieval

1. Introduction

Closing the *semantic gap* in content based image retrieval basically requires relevant low level characteristics, also referred to as descriptors, including color, texture and shape [36]. Shape descriptors may be efficient in capturing discriminating information for many applications, for instance tracking silhouettes, character recognition, contour matching for medical imaging and 3D reconstruction. Many signatures exist in the literature for the purpose of shape (curve or silhouette) description, among them the well studied edge orientation histogram (EOH) [17] and radon transform [42]. Other methods range from those based on learning shape statistics [1,10,12,35,16], shapelets [13], Fourier transform [46,6,21], geometry and a priori knowledge such as the curvature scale space (CSS) [24,44,48], skeleton and axial representations [26,39,5,8,47,33,19], active appearance models [9], context extraction and alignment [32,23,3,34,45], algebraic description and invariant moments [38,40,41]. A good shape descriptor should be robust to linear transformations and also to local non-linearities, noise and mirror effects.

Let us define a training set as the coordinates of 2D points sampled along a curve. As is known, linear principal component analysis, to this training set provides orthogonal axes and the projection of the training set on the span of these axes will be rotation, translation and scale invariant up to a factor. Hence, these principal axes may be efficient in order to recover the linear (similarity) transformation and possibly to encode each training point using the coefficients of the projection onto these axes. Nevertheless, linear PCA maps each training point in the curve into a space of at most two dimensions, so this will not be sufficient to capture details, mainly the non-linearities in the shape geometry (cf. Table 1); for instance, many concave and convex shapes might have similar principal axes.

Kernel PCA, also known as the non-linear version of PCA, considers a positive definite kernel [31,11] denoted $k(x,x') = \langle \Phi(x), \Phi(x') \rangle$ where $\langle , \rangle$ stands for an inner product and Φ is a mapping from an input space into a higher (possibly infinite) dimensional space, referred to as *the feature space*, where linear PCA can be performed.

E-mail address: hs385@cam.ac.uk.

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.06.007

Table 1
Precision versus recall on the Swedish data set using the eigenvalue description (with linear and triangular kernels, $p = 1$), Hough, EOH and CSS [24]

Recall (over 16)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Precision																
Linear PCA	1.0	.59	.46	.40	.35	.32	.31	.29	.28	.27	.27	.26	.25	.25	.24	.24
Kernel PCA	1.0	.967	.949	.935	.925	.914	.903	.892	.882	.871	.860	.851	.841	.832	.825	.816
Hough	1.0	.899	.848	.814	.785	.765	.749	.737	.724	.711	.702	.693	.684	.677	.671	.664
EOH	1.0	.752	.651	.592	.554	.528	.508	.490	.474	.462	.453	.444	.436	0.431	.424	.418
CSS	1.0	.959	.936	.924	.916	.914	.905	.899	.892	.888	.884	.879	.876	.872	.867	.864

When using the triangular kernel, 100 eigenvalues are considered for our shape description while (at most) two eigenvalues are considered when using the linear kernel.

In contrast to the linear case, this non-linear version is similarity (translation, rotation and scale) invariant in the feature space but not in the input space when using many kernels, for instance the Gaussian. Despite the lack of invariance, it is well-known [30] that the Gaussian kernel achieves good performance when the training data are distributed into clusters with comparable scales. Nevertheless, when training data (actually contours) have large variations in scales, the scale (or the variance) parameter of the Gaussian kernel is difficult to set, so the latter will not be appropriate.

In this paper we will show that our proposed kernel achieves similarity (and in particular scale) invariance both in the feature and the input spaces, also we will discuss later, since any Gram matrix built using the triangular kernel is invertible, the number of eigenvectors of KPCA increases with respect to the size of the training set. In contrast to linear PCA, this increase of dimensionality makes it possible to capture the non-linearities in the shape geometry where each dimension can be regarded as a particular characteristic of a given shape. In the remainder of this paper, we use the following notations: $\mathscr{X} \subset \mathbb{R}^2$ denotes an *input space* and X is a random variable standing for the 2D training examples in $\mathscr{X}$. We denote by $\mathscr{S} = \{x_i, i = 1, \ldots, N\}$ a training set generated i.i.d (independently and identically distributed) according to a particular and possible unknown probability distribution $P(X)$. Other notations will be introduced in different sections of this paper, which is organized as follows: first, we review KPCA in Section 2, and we will show our main result of similarity invariance using the triangular kernel. We present in Section 3 our main application, i.e., shape description, followed by an analysis and an interpretation of the method using Fourier analysis in Section 4 and a discussion in Section 5. We conclude in Section 6 and provide some directions for future work.

2. Kernel principal component analysis and similarity invariance

2.1. A short review of KPCA

Given data $\mathscr{S} \subset \mathscr{X}$, principal component analysis is an unsupervised statistical method which finds a set of orthogonal axes in $\mathscr{X}$ such that the projection of the training set $\mathscr{S}$ using only a few of these axes makes it possible to capture most of the statistical variation of the data in $\mathscr{S}$. In practice, PCA has been successfully used in image processing, feature extraction, dimensionality reduction, reconstruction and classification (see, for instance [31,2,25]).

Let Φ be a mapping from the input space $\mathscr{X}$ into a high-dimensional feature space. Assume the training set $\mathscr{S}$ is centered in the mapping space, i.e., $\sum_{i=1}^{N} \Phi(x_i) = 0$. PCA finds the principal axes by diagonalizing the covariance matrix $M = (1/N)\sum_{j=1}^{N} \Phi(x_j)\Phi(x_j)^{\mathrm{t}}$, where x^{t} stands for the transpose of x. The principal orthogonal axes, denoted $\{V_k, k = 1, \ldots, N\}$, can be found by solving the eigenproblem $MV_k = \lambda_k V_k$, where V_k and λ_k are, respectively, the kth eigenvector and its underlying eigenvalue. It can be shown (see for instance [31]) that the solution of the above eigenproblem lies in the span of the training data, i.e.,

$$\forall k = 1, \ldots, N, \quad \exists \alpha_{k1}, \ldots, \alpha_{kN} \in \mathbb{R} \quad \text{s.t.} \quad V_k = \sum_{j=1}^{N} \alpha_{kj} \Phi(x_j), \tag{1}$$

where $\alpha_k = (\alpha_{k1}, \ldots, \alpha_{kN})$ are found by solving the following eigenproblem [31]:

$$K\alpha_k = \lambda_k \alpha_k. \tag{2}$$

Here K is the Gram matrix on the centered training set $\mathscr{S}$ in the feature space. In case the training data are not centered, this matrix is defined as

$$K_{ij} = \left\langle \Phi(x_i) - \frac{1}{N}\sum_k \Phi(x_k), \Phi(x_j) - \frac{1}{N}\sum_k \Phi(x_k) \right\rangle,$$

where $i, j = 1, \ldots, N$ are, respectively, the row and the column indices of K.

2.2. The triangular kernel

Given x and x', two training samples in $\mathscr{X}$, the global form of the unrectified triangular kernel is [27,28,15]

$$k_{\mathrm{T}}(x, x') = -\|x - x'\|^p, \quad p \in \mathbb{R}. \tag{3}$$

Notice that this kernel is not positive definite [18] but defines a conditionally positive definite kernel [4] for

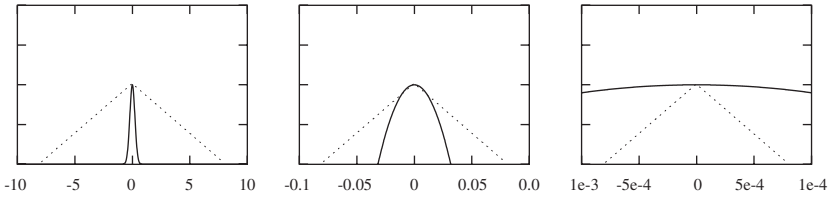

Fig. 1. Gaussian kernel (continuous line) and triangular kernel, $p=1$, (dashed line) at various scales (left to right, respectively $\times 10^0, \times 10^2, \times 10^4$). Intuitively, whereas the triangular kernel has the same shape at all scales, the Gaussian kernel has different shapes, from Dirac-like to a uniform weighting of the neighborhood.

$p \in]0,2[$. This means that for any $x_1,\ldots,x_N \in \mathcal{X}$ and any $c_1,\ldots,c_N \in \mathbb{R}$ such that $\sum_i c_i = 0$, we have $\sum_{i,j} c_i c_j k_{\mathrm{T}}(x_i,x_j) \geqslant 0$. Due to the centering of the data in the feature space, we can show that this kernel can be used for KPCA [29].

Obviously, this kernel is invariant under *translation* and *rotation* but not under scaling. Nevertheless, it still has an interesting weak property of invariance that we could describe as an invariance "in shape". Given a scaling factor $\gamma > 0$, this "weak invariance" can formally be expressed as

$$k_{\mathrm{T}}(\gamma x, \gamma x') = -\gamma^p \|x - x'\|^p = \gamma^p \; k_{\mathrm{T}}(x,x').$$

Thus, when the points are scaled by a certain factor γ, the value of the kernel scales by γ^p. We can also write $k_{\mathrm{T}}(\gamma x, \gamma x') = \langle \sqrt{\gamma^p}\Phi(x), \sqrt{\gamma^p}\Phi(x')\rangle$ which means that any scaling of a population in $\mathcal{X}$ could also be interpreted as a scaling by a factor $\gamma^{p/2}$ in the feature space (see. Fig. 1).

2.3. Similarity invariance

In the following, we consider a situation where we scale the data by a factor $\gamma > 0$. Let us denote by $\mathcal{S}^\gamma = \{\gamma x_1,\ldots,\gamma x_n\}$ a training set for that population. Again, the triangular kernel is translation and rotation invariant, so when performing KPCA using this kernel the underlying eigenvalues and the projections in the principal eigenvectors will be invariant to these two transformations. Now, our main result for KPCA is that the projection of any training example will also be scale invariant, i.e.,

$$\forall\, x \in \mathbb{R}^n \quad \forall\, k = 1,\ldots,N \quad \langle V_k^{(\gamma)}, \gamma x\rangle = \langle V_k^{(1)}, x\rangle, \tag{4}$$

where $\{V_1^{(1)},\ldots,V_N^{(1)}\}$ and $\{V_1^{(\gamma)},\ldots,V_N^{(\gamma)}\}$ denote the eigenvectors of respectively the original and the scaled training sets. Notice that from (1) the eigenvectors are, of course, not scale invariant but the projections on these axes are scale invariant as shown below.

Proof. The proof is straightforward, and comes from the fact that the Gram matrix, denoted $K^{(\gamma)}$ at the scale γ, can be written as

$$K^{(\gamma)} = \gamma^p \; K^{(1)}. \tag{5}$$

Using (2) it follows that

$$K^{(\gamma)}\alpha_k^{(\gamma)} = \gamma^p K^{(1)}\alpha_k^{(1)} \;\Rightarrow\; \lambda_k^{(\gamma)}\alpha_k^{(\gamma)} = \gamma^p \lambda_k^{(1)}\alpha_k^{(1)}. \tag{6}$$

Here $\lambda_k^{(\gamma)}$ denotes the kth eigenvalue of $K^{(\gamma)}$ at the scale γ. The above equality implies: $\forall k = 1,\ldots,N$, $\lambda_k^{(\gamma)} = \gamma^p \lambda_k^{(1)}$ and $\alpha_k^{(\gamma)} = \alpha_k^{(1)}$. Using the latter equations, (4) is not valid but if we consider, instead of (1), a new expansion of $V_k^{(\gamma)}$ as $(1/\lambda_1^{(\gamma)})\sum_{j=1}^N \alpha_{kj}^{(\gamma)}\Phi(\gamma x_j)$,we can show that the projection in the span of these eigenvectors is scale invariant

$$\begin{aligned}\forall x \quad \langle V_k^{(\gamma)}, \gamma x\rangle &= \frac{1}{\lambda_1^{(\gamma)}}\sum_j \alpha_{kj}^{(\gamma)} k_{\mathrm{T}}(\gamma x_j, \gamma x)\\ &= \frac{1}{\lambda_1^{(\gamma)}}\gamma^p \sum_j \alpha_{kj}^{(1)} k_{\mathrm{T}}(x_j, x)\\ &= \frac{1}{\gamma^p \lambda_1^{(1)}}\gamma^p \sum_j \alpha_{kj}^{(1)} k_{\mathrm{T}}(x_j, x) = \langle V_k^{(1)}, x\rangle. \end{aligned} \tag{7}$$

Thus, in the feature space, the eigenvalues and the projection of the training data in the span of the principal eigenvectors can be normalized to be similarity invariant.

Remark. Notice that any Gram matrix built using this kernel is invertible for $p \in]0,2[$ [22], so the VC-dimension [43] related to this kernel is infinite. Besides the similarity invariance of KPCA using this kernel (which is also achieved by the standard linear one), its discriminating power is high as the invertibility of the Gram matrix ensures that any training set can be approximated with an eigenspace with as dimensions as the size of the training set. Given a training set $\mathcal{S}$, the statistics related to the principal eigenvectors and the underlying eigenvalues, considered as linear in the feature space, will capture non-linearities in the input space, as will be shown in the following section.

3. Shape description

We ran our experiments on the SQUID,[1] Smithsonian and Swedish [37] databases consisting respectively of 1100 fish images, 1525 and 1125 images of leaves (cf. Fig. 2). For each curve in the SQUID set, we synthesize four other curves with random orientations (in $[0^\circ, 360^\circ]$), scale factors (in $[0,2]$) and locations (± 20 pixels). In the end, a total of 5500 curves from SQUID were used in our experiments. The SQUID set is only used to show the similarity invariance of our descriptor while the Smithsonian

[1] www.ee.surrey.ac.uk/Research/VSSP/imagedb/squid.htm

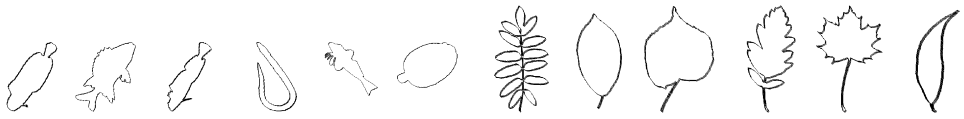

Fig. 2. Some external contours from the SQUID (left) Swedish databases (right).

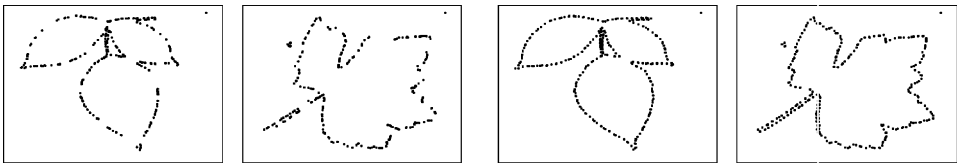

Fig. 3. Comparison between two sampling methods on two contours taken from the Smithsonian set: random sampling (left) uniform adaptive sampling (right). The number of samples in both cases is equal to 256.

and Swedish sets are used to evaluate its precision. The Smithsonian data set has 135 categories with different cardinalities while the Swedish set has 15 classes, each one containing 75 images. Notice that only the external contours are used for shape description even though this is not sufficient to make the "best" predictions as color and internal structures are also prominent.

3.1. Indexing and retrieval

Let us define an external contour (or curve) as: $\mathscr{L} = \{x(s) = (u(s), v(s)) \in \mathbb{R}^2,\ s \in [0,1]\}$ and $\mathscr{S}$ as a sampling of $\mathscr{L}$. Here s is referred to as the curvilinear abscissa. We consider in this work two sampling methods: adaptive uniform, i.e., $s = k\Delta,\ k \in \mathbb{N}^+,\ \Delta = 1/|\mathscr{L}|$, and random sampling, i.e., s is randomly taken from $[0,1]$. The sampling is considered representative enough to capture the whole shape of $\mathscr{L}$ (cf. Fig. 3).

Again, the triangular kernel is rotation and translation invariant, so under these transformations the eigenvalues of KPCA on $\mathscr{S}$ remain unchanged. Scaling the data in $\mathscr{S}$ with γ scales the eigenvalues by γ^p. Hence, the eigenvalues $\{\lambda_i^{(\gamma)},\ i = 1,\ldots,N\}$ can be normalized with respect to the largest value $\lambda_1^{(\gamma)}$ in order to cancel the factor γ^p. Therefore, $\{\lambda_i^{(\gamma)}/\lambda_1^{(\gamma)},\ i = 1,\ldots,d<N\}$ will be scale invariant and can be used as a similarity invariant description of a given curve. Notice that computing this description does not necessitate sampling a curve according to an *ordered* curvilinear abscissa which might be unavailable and difficult to find for complex contours.

Given a set of contours, each one modeled using its d largest eigenvalues, retrieval is based on the nearest-neighbor classifier using the L_2 distance. Fig. 4 illustrates some results on, respectively, SQUID, Swedish and Smithsonian data sets; for each submitted query in SQUID, the system finds first the four most similar shapes which differ only by similarity transformations, then the system finds the other similar curves with an increasing order of the distance.

The performance of our shape description is measured empirically on the Swedish and Smithsonian sets using the usual recall–precision measures defined as

$$\text{precision} = E\left(\frac{\text{Number of relevant returned images}}{\text{Number of returned images}}\right),$$

$$\text{recall} = E\left(\frac{\text{Number of relevant returned images}}{\text{Number of images in the class of the query}}\right),$$

where $E(\cdot)$ denotes the expectation over all the possible queries in the database.

3.2. Sampling and kernel parameters

Sampling: Again, two sampling methods were used in order to extract a sample set $\mathscr{S}$ from a given contour and to apply KPCA: random (ad hoc) sampling and uniform adaptive (cf. Section 3.1). Recall–precision, in (Fig. 5, left), shows that adaptive uniform sampling clearly outperforms random sampling.

The kernel parameter: The effect of the parameter p of the triangular kernel has also been studied. (Fig. 5, right) shows the recall–precision curves for different values of p. The precision reaches its maximum value for $p = 1.9$ and drops dramatically when p is close to its upper bound (i.e., $p = 2$).

The parameter p acts as a regularizer. In order to understand the effect of this regularizer, let us consider a training set (shape contour) $\mathscr{S}$ and let us denote by $\{\lambda_1(p,\mathscr{S})\}$ the underlying 1D ($d = 1$) shape descriptor obtained by applying KPCA to $\mathscr{S}$ and taking only the first (highest) eigenvalue. This eigenvalue is proportional to the maximal distance between two training data taken from $\mathscr{S}$ (see for instance Fastmap [14]). Hence, we can write

$$\begin{aligned}\lambda_1(p,\mathscr{S}) &\simeq \max_{x,x' \in \mathscr{S}} \|\Phi(x) - \Phi(x')\|^2 \\ &= \max_{x,x' \in \mathscr{S}} k_T(x,x) + k_T(x',x') - 2k_T(x,x') \\ &= \max_{x,x' \in \mathscr{S}} 2\|x - x'\|^p. \end{aligned} \quad (8)$$

Consider $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$, three shape contours: $\mathscr{S}_1$, $\mathscr{S}_2$ belong to the same class while the class of $\mathscr{S}_3$ is different. From (8), it is reasonable to say that $\{\lambda_1(p,\mathscr{S}_1)\}$,

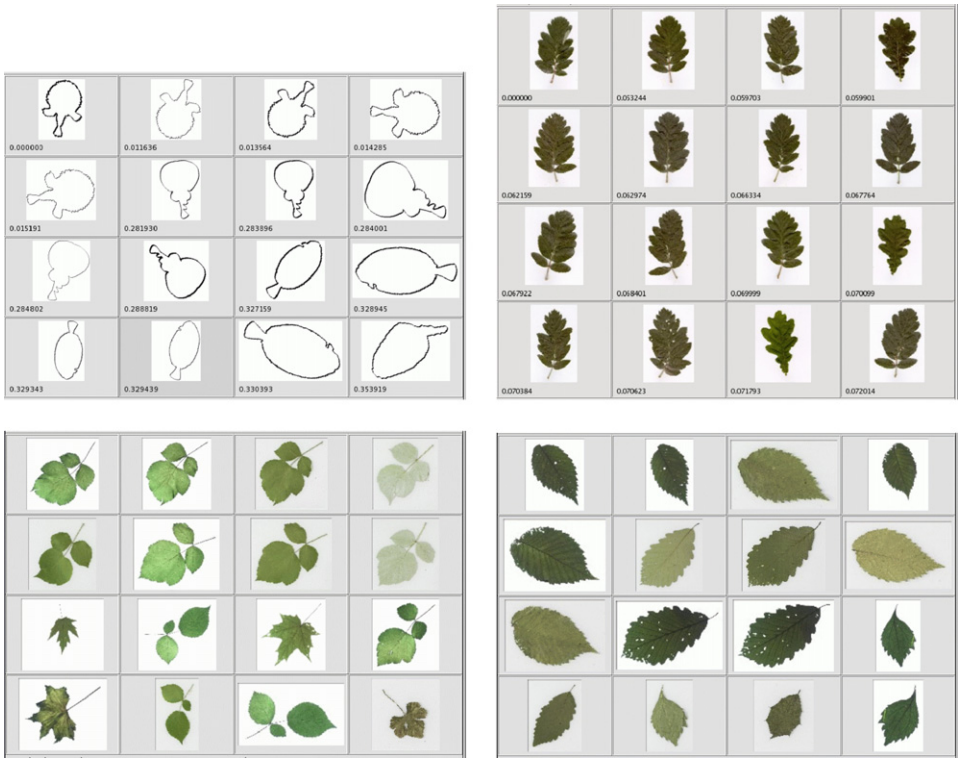

Fig. 4. In all these sessions, the top-left image is the query shape while the others are some results sorted from top-left to bottom-right according to their dissimilarity. These curves are taken from SQUID (top left), Swedish (top-right) and Smithsonian (bottom) data bases. In the case of SQUID the first four curves returned by the system differ only by a similarity transformation. We set $p = 1$ and $d = 100$.

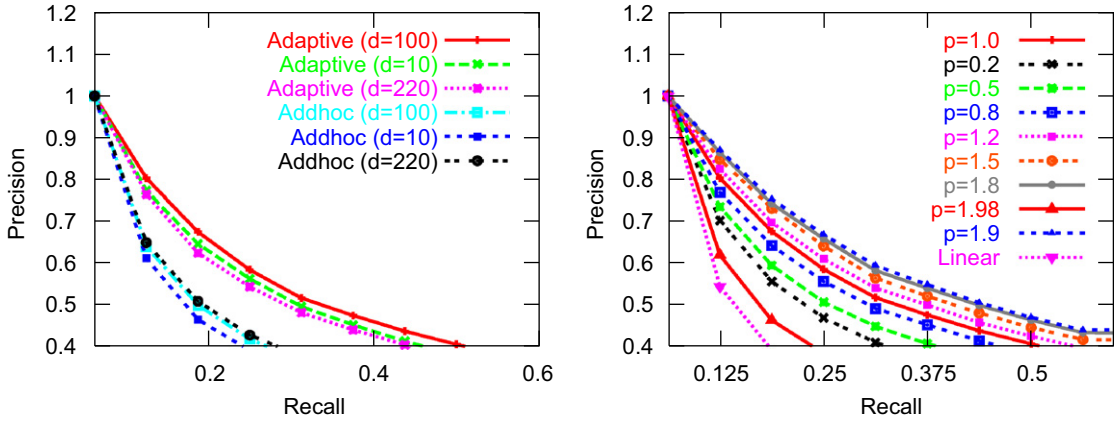

Fig. 5. (Left) Precision–recall on the SmithSonian data set with respect to sampling. Adaptive stands for uniform sampling while ad hoc is random sampling. These results are shown for different values of the dimension d ($p = 1$). (Right) Variation of precision–recall on the Smithsonian data set with respect to the kernel parameter p, ($d = 100$).

$\{\lambda_1(p, \mathscr{S}_2)\}$ remain similar as p changes whereas the descriptions of $\mathscr{S}_1$ and $\mathscr{S}_3$ are more different as p increases (cf. Fig. 6, right). Therefore, the intra-class variance is stable with respect to p while the inter-class variance increases as p increases, so the performance achieved (recall–precision) improves when p increases (cf. Fig. 5, right). This argument can be extended to $d > 1$.

When p reaches it upper bound (i.e., $p = 2$), the triangular kernel can be written as $k_T(x, x') = -(\langle x, x\rangle + \langle x', x'\rangle - 2\langle x, x'\rangle)$. Without loss of generality, we can assume that $\langle x, x\rangle = 1$. Now the triangular kernel is linear, so the recall–precision performance, when p is close to 2, is similar to that of the linear kernel (cf. Fig. 5, right).

3.3. Dimensionality

The issue of selecting d, i.e., the number of eigenvectors sufficient to capture the major statistical variation in the data (actually points in a contour) has been tackled in many works, for instance [25]. The d-principal eigenvectors

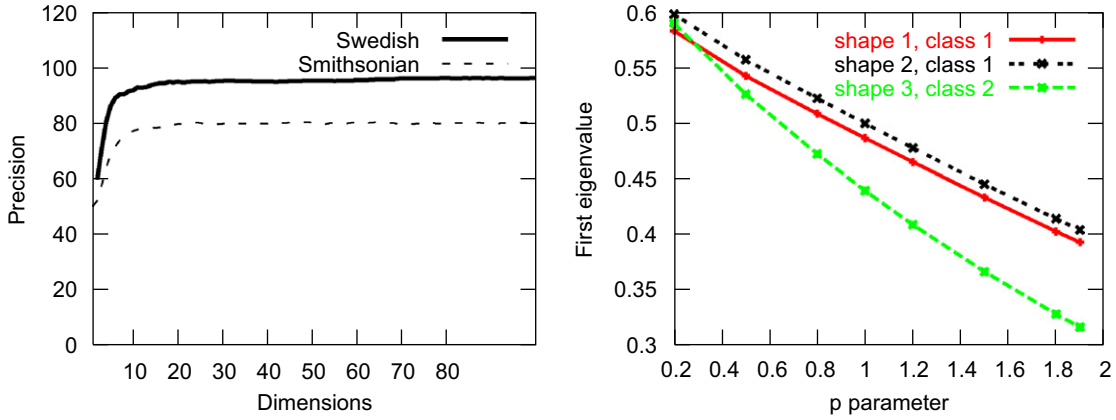

Fig. 6. (Left) This figure shows the probability that a query and the first nearest neighbor, on the LEAF sets, belong to the same class with respect to d, the number of eigenvalues used for shape description. (Right) Variation of the first eigenvalue of three shapes: two of them taken from the class shown in the bottom-left of Fig. 4 and other taken from the class in the bottom-right side. We can clearly see that the first eigenvalues of shape 1 and 2 (from the same class) are close while the first eigenvalue of shape 3 diverges as p increases.

(and eigenvalues) capture the major variations while the $N-d$ remaining eigenvectors correspond to the less significant and useful details, even noise, which are not necessary for discrimination. In our experiments, we select d eigenvectors in order to capture 99% of the statistical variation, i.e.,

$$\sum_{k=1}^{d}\lambda_k \Big/ \sum_{k=1}^{N}\lambda_k \geqslant 0.99. \tag{9}$$

When averaging this criteria through all the training sets (curves) from the SQUID, Swedish and Smithsonian data bases, we found that 68-principal axes capture more than 99% of the statistical variance for each curve in the SQUID and Swedish while 60 are necessary in the case of the Smithsonian set. Fig. 6 (left) shows that for $d=68$ and 60, the precision values on Swedish and Smithsonian sets (resp. 95.95% and 80.26%) are close to the highest values (resp. 96.31% and 80.52%). Therefore, the criteria (9) predicts well the optimal number of dimensions d.

Table 1 shows the precision–recall on the Swedish data set for $d=100$ dimensions using our shape description. Comparisons are reported using linear PCA[2] ($d=2$) and other shape descriptors including the EOH [17], Hough [20] and CSS [24]. When the recall is less than 30%, the precision of KPCA is better than the other descriptors including CSS.

3.4. Matching

Many applications, for instance 3D object reconstruction, require recovering the parameters of a similarity transformation by finding a set of *good matches* between points in two curves. Given two curves $\mathscr{S}_1$ and $\mathscr{S}_2$, we define $(x_i, x'_j) \in \mathscr{S}_1 \times \mathscr{S}_2$ as a good match, if:

$$x'_j = \arg\min_{x'_k \in \mathscr{S}_2} \|y_i - y'_k\|^2. \tag{10}$$

Here y_i, y'_k correspond, respectively, to the projection of x_i and x'_k in the span of the d-selected eigenvectors. We used this criteria in order to find these good matches through the SQUID data set; Fig. 7 shows some results. Notice that we did not consider any smoothness or neighborhood criterion such as requiring that every two neighboring points in a curve should have neighboring matches in the other one, which can reduce the number of false matches.

4. Spectral analysis

The principal eigenvectors of (2D) points sampled from a 2D curve, correspond to orthogonal axes which maximize the variance in the mapping space. However, it is not clear how to interpret these eigenvectors in the input space. For instance, let us consider two training sets of 2D points sampled from two fish contours and let us apply KPCA on these sets (cf. Fig. 8). For each case, a test set consisting of 800×600 (2D) points were projected on the span of the four principal eigenvectors. The level curves in Fig. 8 show, for each principal component, the (2D) points with the same correlation with respect to this component in the feature space. We can see from the left-hand side figure that the intensity value of the level curves decreases in the direction of shape elongation, and this clearly shows that the first principal component characterizes the shape width. Using the same reasoning, we can see that the second eigenvector characterizes the curve height but it is more difficult to interpret the third and the fourth eigenvectors. The latter may relate to details such as the shape of the tail or the fin, or other non-linearities in the contours (cf. Fig. 9).

In the reminder of this section, we will discuss the interpretation of the principal eigenvalues of our shape

[2]As the training examples live in 2D, we have at most two non-null eigenvalues when solving KPCA using the linear kernel.

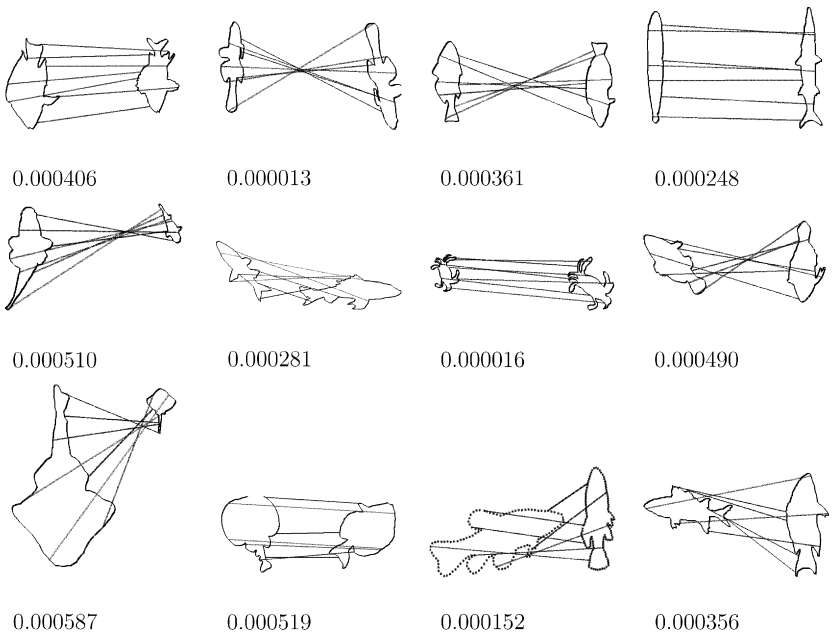

Fig. 7. Some matching results. The matching scores using the Euclidean distance were also reported. For ease of visualization, only a subset of matches is displayed.

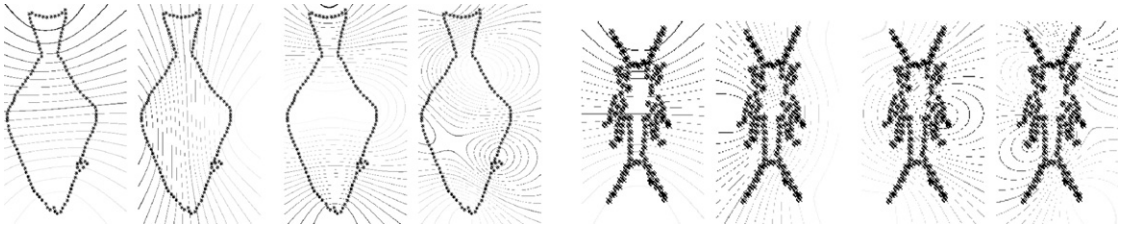

Fig. 8. Interpretation of the first principal components. By definition, these level-curves characterize the 2D points in the input space $\mathscr{X}$ with the same correlation with respect to, respectively, from the left to the right, the first, second, third and the fourth eigenvectors. The first and the second principal axes correspond, respectively, to the variation in width and height while the other axes correspond to variation in details such as the length of the tail, diagonal characteristics, etc.

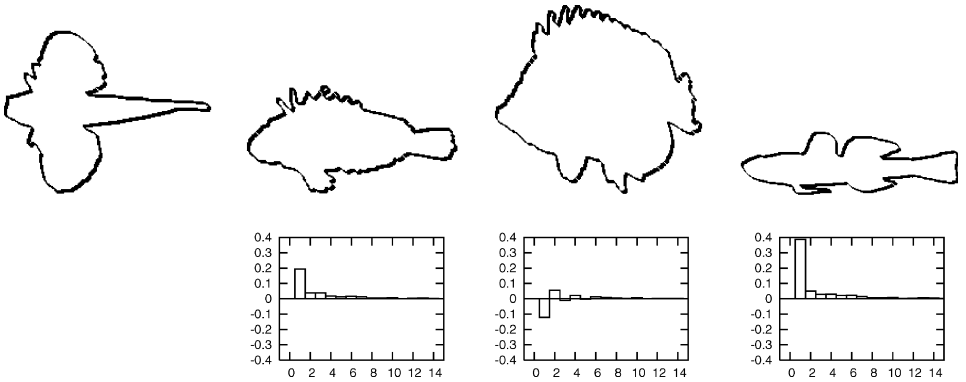

Fig. 9. Top: some fish contours randomly selected from the SQUID data set. Bottom: "eigenvalue spectrum" differences between the first shape and the others. For each dimension, the bottom figures show the differences between the eigenvalue spectrums related to the left-hand side curve denoted C_1 and the others denoted $\{C_2, C_3, C_4\}$, respectively. We can see, through all these diagrams, that the differences between the second eigenvalues characterize variations of the height between C_1 and C_i, $i = 2, 3, 4$. We can also interpret the third "eigenvalue differences" as variation in fluctuations, the fourth one can be some noise, etc. As all the first eigenvalues are normalized to 1, their differences of course vanish.

description using Fourier analysis. We will show that, indeed, these eigenvalues capture a low to high frequency variation of shapes as the dimension of the eigenvalues increases.

4.1. Polar coordinates

Again, let $\mathscr{L} = \{x(s) = (u(s), v(s)) \in \mathbb{R}^2,\ s \in [0, 1]\}$ be a 2D curve where s is a curvilinear abscissa. Denote by C_0

the center of gravity of $\mathscr{L}$ and let $\mathscr{S} = \{x(s_i) = (u(s_i), v(s_i))\}_{i=1}^N$ be a particular sampling of $\mathscr{L}$ according to s. We can easily derive $x(s_i)$ using polar coordinates

$$u(s_i) = d(s_i)\cos(\theta_i),$$
$$v(s_i) = d(s_i)\sin(\theta_i), \tag{11}$$

where $d(s_i)$ is the distance of $x(s_i)$ to the center of gravity C_0 and θ_i is the angle of the segment joining C_0 to $x(s_i)$ with respect to the horizontal axis. Assuming $\theta_i = i\theta_1$, it is clear that $\mathscr{S}$ can be described using only one free parameter $d(s_i)$ (cf. Fig. 10). Unless mentioned explicitly, we will write $x(s_i)$ and $d(s_i)$ simply as x_i and d_i, respectively.

4.2. Prefeature estimation

Let $\Phi(x_i)$ be a mapping of x_i into a high dimensional feature space. Define $\Phi(x_{i,\text{def}}) = \Phi(x_i) + \gamma V_k$ to be a perturbation of a sample $\Phi(x_i)$ in the direction of the eigenvector V_k, which is considered as a non-linear (and unknown) transformation in the input space $\mathscr{X}$. The perturbation of several samples $\mathscr{S} = \{x_1, \ldots, x_N\}$ in the mapping space may be interpreted as a deformation S_{def} of S in $\mathscr{X}$. The new set $S_{\text{def}} = \{x_{1,\text{def}}, \ldots, x_{N,\text{def}}\}$, referred to as the *prefeature* set, is found by

$$x_{i,\text{def}} = \arg\min_x \|\Phi(x) - \Phi(x_i) - \gamma V_k\|^2, \quad i = 1, \ldots, N. \tag{12}$$

Solving this minimization problem makes it possible to understand visually the interpretation of a perturbation along the principal axis V_k. In practice, this minimization problem is not convex and difficult to solve when using many kernels including the triangular and the Gaussian [7]. However, we can approximate S_{def} using (11) by finding the distances $\{d_{i,\text{def}}\}$. Here, $d_{i,\text{def}}$ is taken as the Euclidean distance of $\Phi(x_{i,\text{def}})$ to M_0: the center of gravity of S in the mapping space. This distance is defined as

$$\begin{aligned}\|\Phi(x_{i,\text{def}}) - M_0\|^2 &= \|\Phi(x_i) + \gamma V_k - M_0\|^2 \\ &= \left\| M_0 + \sum_{j=1}^N c_{ji} V_j + \gamma V_k - M_0 \right\|^2 \\ &= \left\| \sum_{j\neq k}^N c_{ji} V_j + (\gamma + c_{ki}) V_k \right\|^2. \end{aligned} \tag{13}$$

Here $\{c_{ji}\}_{j=1}^N$ are the coefficients of projection of $\Phi(x_i)$ on the eigenspace spanned by $\{V_j\}_{j=1}^N$. As the latter form an orthogonal basis, and using the kernel trick, we can easily show that the above distance can be rewritten as

$$\begin{aligned} d_{i,\text{def}}^2 \simeq \|\Phi(x_{i,\text{def}}) - M_0\|^2 &= \sum_{j=1}^N c_{ji}^2 + 2\gamma c_{ki} + \gamma^2 \\ &= d_i^2 + 2\gamma c_{ki} + \gamma^2. \end{aligned} \tag{14}$$

Hence, it is trivial to recover the distances $\{d_{i,\text{def}}\}_{i=1}^N$ and the underlying (2D) coordinates in the input space using (11) (cf. example in Fig. 10).

4.3. Fourier interpretation

Given $\mathscr{S}$, applying the discrete Fourier analysis on the underlying set of distances $\{d_i\}_{i=1}^N$, makes it possible to determine the frequency distribution of the shape. The low frequency Fourier coefficients model the global shape characteristics including the width and the height while the high frequency coefficients are related to the presence of bumps, noise effects and other unknown characteristics.

Again, let us consider a perturbation of training samples in $\mathscr{S}$ defined as $\{\Phi(x_i) + \gamma V_k\}_{i=1}^N$ and let $\{d_{\text{def}}(s_i)\}_{i=1}^N$ be the underlying distances to the center of gravity (cf. Section 4.2). Our objective is to localize the frequency domain in the Fourier transform which is sensitive to this perturbation. Denote $\mathscr{F}$ as the discrete Fourier transform applied on $\{d_{\text{def}}(s_i)\}_{i=1}^N$. Using (14), this transform is

$$\begin{aligned}\mathscr{F}(d_{\text{def}}^2(s_i)) &= \mathscr{F}(d^2(s_i)) + \mathscr{F}(\gamma^2 + 2c_{ki}\gamma) \\ &= \mathscr{F}(d^2(s_i)) + \gamma^2\delta() + 2\gamma\mathscr{F}(c_{ki}). \end{aligned} \tag{15}$$

Here $\delta()$ is the Dirac function. It follows from the above equation that the Fourier response to the perturbation depends mainly on $\{c_{ki}\}$, i.e., the coefficients of the projection of $\Phi(x_i)$ on the kth eigenvector. When the dimension k is low, the variation of these coefficients is smooth according to the curvilinear abscissa, resulting in a Fourier spectrum with high energy in the low frequencies (cf. Fig. 11, top) while a high value of k produces unstable and noisy projection coefficients in V_k, resulting in a Fourier spectrum with more energy in the high frequencies (cf. Fig. 11, bottom). Hence, it is clear (cf. also example in Fig. 11) that the dimension of the eigenvalues is directly related to the frequency of the Fourier spectrum; low dimensional eigenvectors (and eigenvalues) characterize low frequency variation of a curve, such as the width, the height and low frequency bumps, whereas high-dimensional eigenvectors are related to details including noise and fluctuations.

5. Discussion

The property of similarity invariance, shown in this paper, can also be achieved using other kernels, for instance the linear one. As the training data (a shape contour) $\mathscr{S}$ live in 2D, the dimension of the eigenspace, found by applying linear PCA on $\mathscr{S}$, will not exceed 2. The underlying eigenvalues will only capture the height and the width of contours which may be similar for many concave and convex silhouettes. As shown earlier in experiments, linear PCA is not suitable to capture details, concavities and the non-linearities in shapes.

In contrast to the linear case, when using the Gaussian kernel, the 2D points in a contour $\mathscr{S}$ are mapped into a high (possibly infinite) dimensional space [43] and the number of eigenvalues increases as the cardinality of $\mathscr{S}$ increases and this makes it possible to capture more details in $\mathscr{S}$. Nevertheless, the scale parameter of the Gaussian

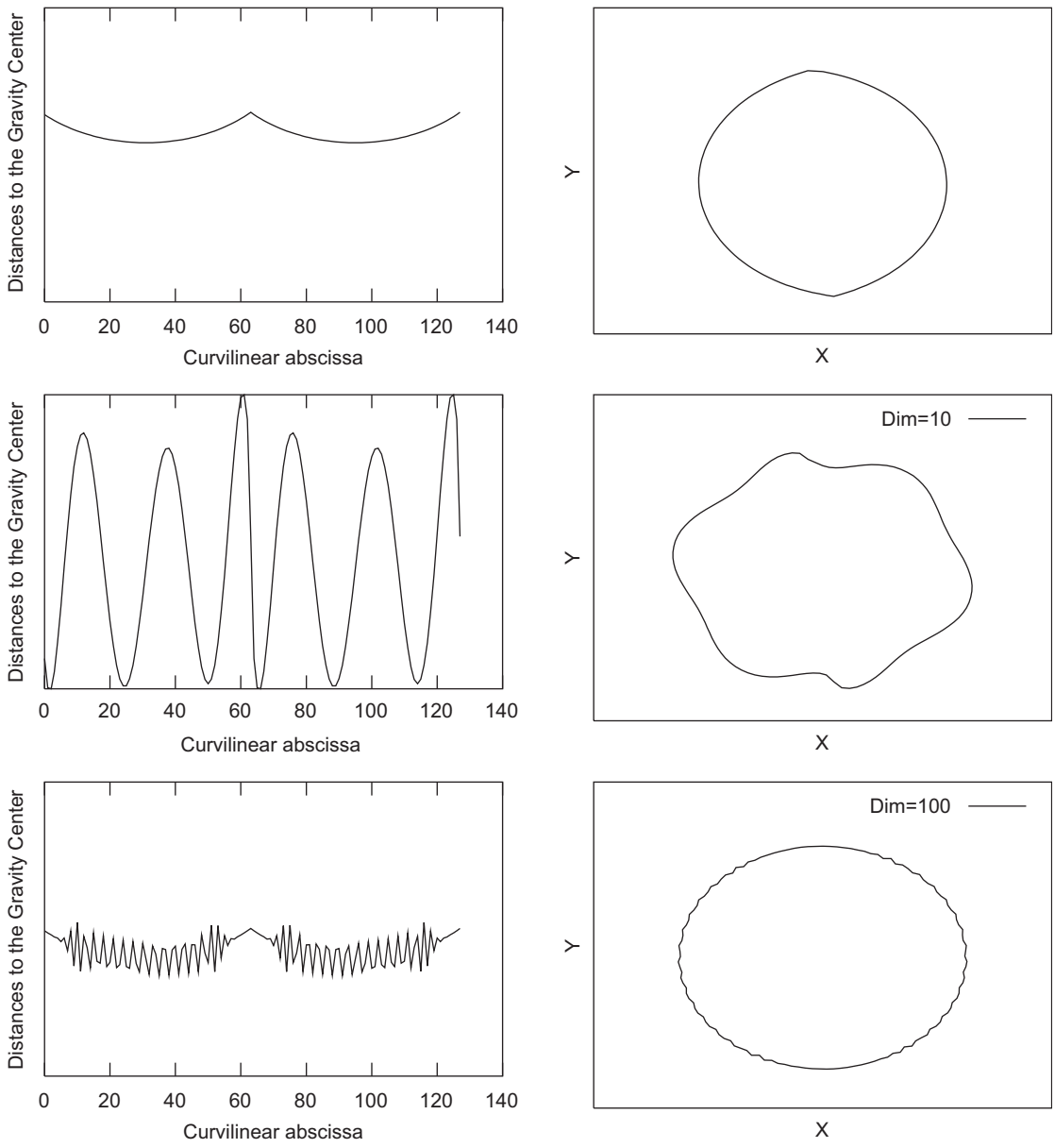

Fig. 10. Distance representations and the underlying silhouettes. The distances were perturbed in the direction of the kth eigenvectors (resp. from the middle to the bottom 10, 100)(left). Reconstruction of the underlying (2D) silhouettes (right). Notice that no perturbation was performed in the top figure.

kernel depends on $\mathscr{S}$, so when processing many contours, the scale should be adapted to each one. When using Gaussians with different scale parameters, each one associated to a given contour, 2D samples from different contours are not mapped in the same high-dimensional space and the underlying eigenvalues are therefore not comparable.

The triangular kernel gathers two properties: it achieves scale (and similarity) invariance while ensuring a "rich" description as the number of eigenvalues increases as the cardinality of a contour $\mathscr{S}$ increases (cf. remark in Section 2.3). Nevertheless, a drawback appears when the size of $\mathscr{S}$ is large enough to make KPCA numerically intractable.

It is also possible to extend the triangular kernel in order to define a shape descriptor invariant to articulations. Consider the example in Fig. 12, where a human body is represented by a skeleton. Clearly, when estimating the Gram matrix using the triangular kernel (based on the Euclidian distance), KPCA (and hence the eigenvalues) will not be invariant to articulations. A new definition of the kernel

$$K(x,x') = -\sum_{i=1, X_1=x, X_M=x'}^{M-1} \|X_i - X_{i+1}\|^p = \sum k_{\mathrm{T}}(X_i, X_{i+1})$$

which is dependent on the geodesic distance linking x and x', makes this kernel "articulation" invariant. Notice also that each term corresponds to the triangular kernel and this makes KPCA also similarity invariant.

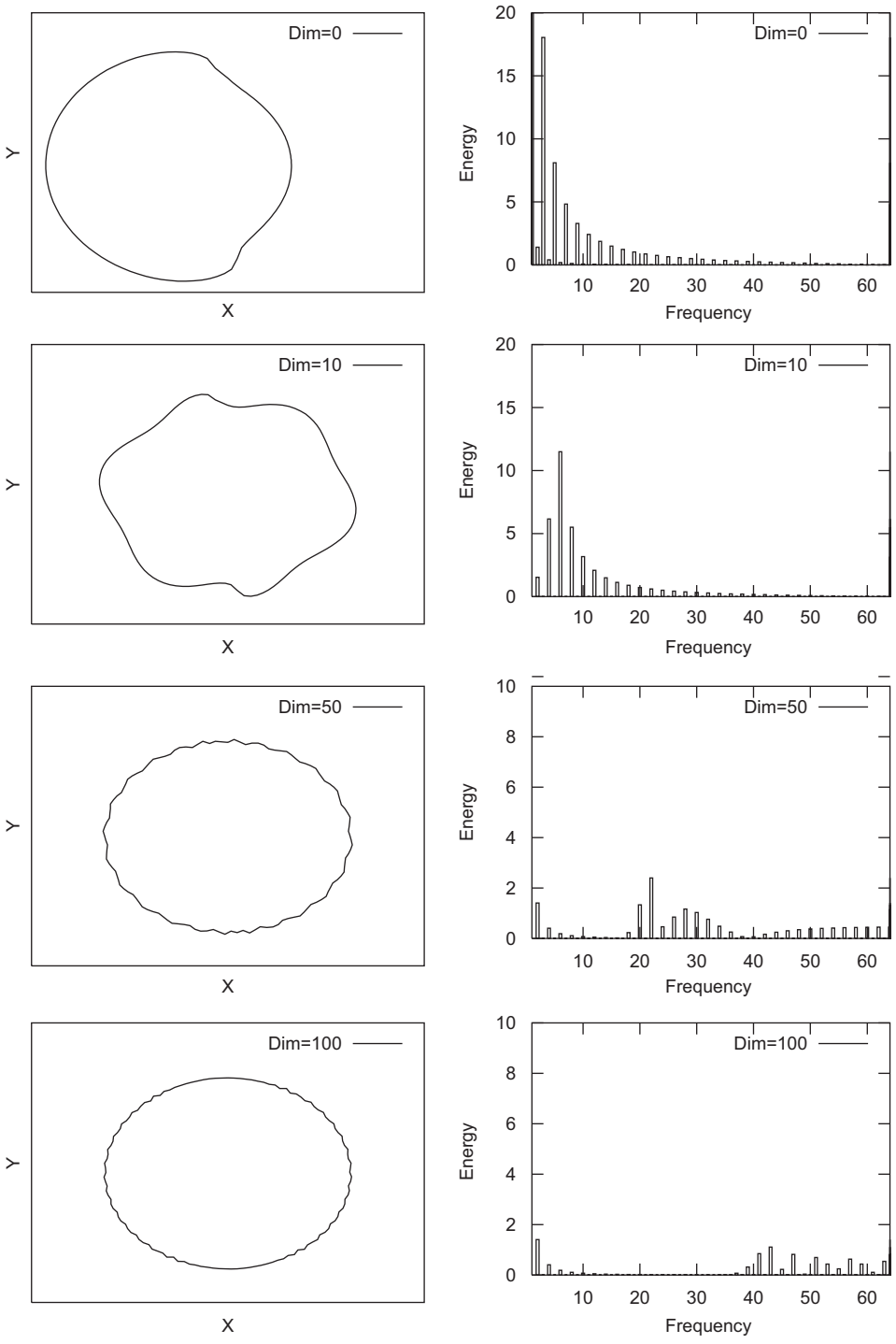

Fig. 11. (left) The left-hand side pictures show the reconstruction of the (2D) silhouettes using their distances after perturbation. As mentioned, the perturbations were performed through the kth eigenvectors (resp. from the top to the bottom $k = 0,10,50,100$). (Right) Fourier amplitude/frequency spectrum on the reconstructed silhouettes.

6. Conclusion and future work

We introduced in this paper an invariant shape description based on the triangular kernel. Our method is also scale invariant when using the linear kernel, nevertheless, the dimension of the underlying eigenspace cannot exceed two, so the eigenvalues of this eigenspace will not be sufficient to discriminate different and complex shapes especially those containing fine details as shown in the experiments. While this problem might be overcome when using the Gaussian kernel, KPCA is not similarity invariant when using this kernel and this necessitates finding different scales for different curves and hence it is meaningless to compare the underlying eigenvalues.

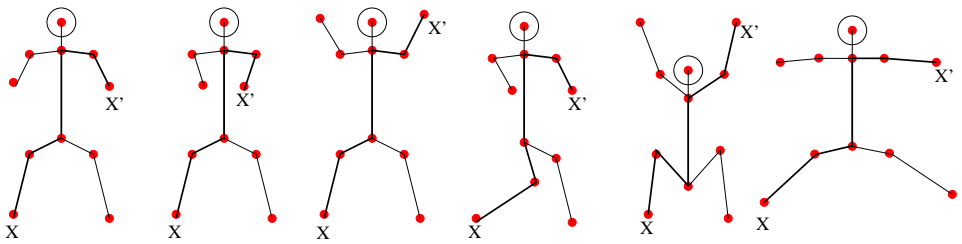

Fig. 12. This picture shows an example of a skeleton and the geodesic distance estimated between two sampled points x and x' which is stable with respect to articulations.

As future work, we aim to address the main limitation of this approach which resides in the sensitivity of the estimated eigenvectors and eigenvalues to partial occlusion. We also envisage the application of this approach for face and mainly expression analysis. Indeed, similarity invariant shape description of mouth contours can be a preliminary step for classification. Another future direction will be the application of the method to 3D shape description.

Acknowledgments

The author would like to thank the anonymous referees for very useful and comprehensive comments. The author also would like to thank Prof. Donald Geman for the useful comments and Prof. Peter Belhumeur for the interesting discussions.

References

[1] A. Baumberg, D. Hogg, An adaptive eigenshape model, in: Proceedings of the Sixth British Machine Vision Conference, vol. 1, 1995, pp. 87–96.
[2] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[3] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. and Mach. Intell. 24 (4) (2002) 509–522.
[4] C. Berg, Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions, Springer, Berlin, 1984.
[5] H. Blum, A transformation for extracting new descriptors of shape, in: W. Wathen-Dunn (Ed.), Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, MA, 1967, pp. 362–380.
[6] R. Bracewell, The Fourier Transform and its Applications, second ed., McGraw-Hill, New York, 1986.
[7] C. Burges, B. Schölkopf, Improving the accuracy and speed of support vector machines, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, The MIT Press, Cambridge, 1997, pp. 375–381.
[8] J.M. Coggins, A statistical approach to multiscale, medial vision, Research report, Department of Computer Science, University of North Carolina at Chapel Hill, 1992.
[9] T.F. Cootes, D. Cooper, C. Taylor, J. Graham, Active shape models—their training and application, Comput. Vision Image Understand. 61 (1) (1995) 38–59.
[10] D. Cremers, T. Kohlberger, C. Schnörr, Nonlinear shape statistics via kernel spaces, in: German National Conference on Pattern Recognition (DAGM), 2001, pp. 269–276.
[11] N. Cristianini, J. Shawe-Taylor, An introduction to Support Vector Machines, Cambridge University Press, Cambridge, 2000.
[12] I.L. Dryden, K.V. Mardia, Statistical Shape Analysis, Wiley Publisher, New York, 1998.
[13] A. Dubinskiy, S. Zhu, A multi-scale generative model for animate shapes and parts, in: International Conference on Computer Vision, 2003, pp. 249–256.
[14] C. Faloutsos, K. Lin, A fast algorithm for indexing, data-mining and visualization, in: Proceedings of ACM SIGMOD, 1995, pp. 163–174.
[15] F. Fleuret, H. Sahbi, Scale-invariance of support vector machines based on the triangular kernel, in: Third International Workshop on Statistical and Computational Theories of Vision (part of ICCV), 2003.
[16] A. François, G. Medioni, Generic shape learning and recognition, in: Proceedings of the International Workshop on Object Representation in Computer Vision, vol. II, 1996, pp. 287–320.
[17] W. Freeman, M. Roth, Orientation histograms for hand gesture recognition, in: International Workshop on Automatic Face and Gesture Recognition, 1995, pp. 296–301.
[18] M.G. Genton, Classes of kernels for machine learning: a statistics perspective, J. Mach. Learn. Res. (Kernel Machines Section) 2 (2001) 299–312.
[19] P. Giblin, B. Kimia, On the local form and transitions of symmetry sets and medial axes, and shocks in 2d, in: Proceedings of IEEE International Conference on Computer Vision, IEEE Computer Society Press, Silver Spring MD, 1999, pp. 385–391.
[20] J. Illingworth, J. Kittler, A survey of the hough transform, Computer Vision, Graphics, and Image Processing 44 (1) (1988) 87–116.
[21] C.C. Lin, R. Chellappa, Classification of partial 2-d shapes using fourier descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 9 (5) (1987) 686–690.
[22] C. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, Constr. Approx. 2 (11) (1986).
[23] E. Milios, E. Petrakis, Shape retrieval based on dynamic programming, IEEE Trans. Image Process. 9 (1) (2000) 141–146.
[24] F. Mokhtarian, S. Abbasi, J. Kittler, Robust and efficient shape indexing through curvature scale space, in: Proceedings of British Machine Vision Conference, 1996, pp. 53–62.
[25] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspace for face recognition, in: Proceedings of the International Conference on Computer Vision, 1994, pp. 84–91.
[26] A. Rosenfeld, Axial representation of shape, in: CVGIP, vol. 33, 1986, pp. 156–173.
[27] H. Sahbi, Invariant shape description using the triangular kernel and its application for leaf recognition, in: Proceedings of the IEEE Workshop on Content Based Image Retrieval, 2005.
[28] H. Sahbi, F. Fleuret, Scale-invariance of support vector machines based on the triangular kernel, INRIA Research Report, N 4601, 2002.
[29] B. Schölkopf, The kernel trick for distances, in: Proceedings of Neural Information Processing Systems, 2000, pp. 301–307.
[30] B. Schölkopf, A. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.

[31] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with gaussian kernels to radial basis function classiers, IEEE Trans. Signal Process. 45 (11) (1997) 2758–2765.
[32] S. Sclaroff, A. Pentland, Modal matching for correspondence and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17 (6) (1995) 545–561.
[33] T. Sebastian, B. Kimia, Curves vs. skeletons in object recognition, Signal, Process. 85 (2) (2005) 247–263 special section on content-based image and video retrieval.
[34] T. Sebastina, P. Klein, B. Kimia, Recognition of shapes by editing shock graphs, in: International Conference on Computer Vision, 2001, pp. 755–762.
[35] Y. Shan, Z. Zhang, New measurements and corner-guidance for curve matching with probabilistic relaxation, Int. J. Comput. Vision 46 (2) (2002) 157–171.
[36] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.
[37] O. Soderkvist, Computer vision classification of leaves from swedish trees, Master's Thesis, Linkoping University, 2001.
[38] J. Subrahmonia, D. Cooper, D. Keren, Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants, IEEE Trans. Patt. Anal. Mach. Intell. 18 (5) (1996) 505–519.
[39] M. Tanase, R. Veltkamp, A straight skeleton approximating the medial axis, in: Proceedings European Symposium on Algorithms, 2004, pp. 809–821.
[40] J.-P. Tarel, D.B. Cooper, The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition, IEEE Trans. Pattern Anal. Mach. Intell. 22 (7) (2000) 663–674.
[41] T. Tasdizen, J.-P. Tarel, D.B. Cooper, Improving the stability of algebraic curves for applications, IEEE Trans. Image Process. 9 (3) (2000) 405–416.
[42] P. Toft, The radon transform—theory and implementation, Ph.D. Thesis, Department of Mathematical Modelling, Technical University of Denmark June 1996.
[43] V. Vapnik, Statistical Learning Theory, A Wiley-Interscience Publication, New York, 1998.
[44] H.J. Wolfson, On curve matching, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 483–489.
[45] L. Younes, Computable elastic distances between shapes, SIAM J. Appl. Math. 58 (2) (1998) 565–586.
[46] C. Zahn, R. Roskics, Fourier descriptors for plane closed curves, IEEE Trans. Comp. 21 (3) (1972) 269–281.
[47] S. Zhu, Stochastic jump-diffusion process for computing medial axes in markov random fields, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1158–1169.
[48] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vision 13 (2) (1994) 119–152.

Hichem Sahbi received his M.S. degree in theoretical computer science from university of Paris Sud (Orsay, France) in 1999 and the Ph.D. degree in applied mathematics from INRIA (France) in 2003. He spent one year at the Fraunhofer (GMD) institute (Darmstadt, Germany) in 2004 as a postdoctoral fellow and one year back at INRIA as a research assistant. Since 2005, he is a research associate at Cambridge university (UK) in the machine intelligence laboratory. His research interests include statistical machine learning, computer vision and machine translation.