

# AFFINE INVARIANT SHAPE DESCRIPTION USING THE TRIANGULAR KERNEL AND ITS APPLICATION TO LEAF RECOGNITION

*Hichem Sahbi*

IMEDIA Research Group  
INRIA Rocquencourt, France  
Hichem.Sahbi@INRIA.fr

## 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 affine and particularly scale invariance of KPCA when using a particular kernel referred to as the triangular kernel. Beside this affine invariance, the method provides an effective way to capture the non-linearities in the shape geometry. Experiments conducted on the SQUID and LEAF databases show this affine invariance and the good performances for shape matching and retrieval.

## 1. INTRODUCTION

Solving the *semantic gap* in content based image retrieval basically requires relevant low level characteristics, also referred to as descriptors, including color, texture and shape [19]. The latter may be efficient to capture the discriminant information for many applications, for instance tracking silhouettes, character recognition, contour matching for medical imaging, 3D reconstruction, etc. Many signatures exist in the literature for the purpose of shape and curve description; among them the well studied edge orientation histogram, radon transform, etc. Other methods range from those based on: learning shape statistics [1, 6], geometry and a priori knowledges such as the curvature scale space (CSS) [11], skeleton and axial representations [13], context extraction and alignment [3, 18], algebraic description and invariant moments [9, 20]. A good shape description should be robust to affine transformations and also to the local non-linearities, the noise and the mirror effects.

Given a training set of  $2D$  points sampled from a curve. As already well known, linear principal component analysis provides a set of orthogonal axes and the projection of training points on the span of these axes will be rotation, translation and also scale invariant up to a factor. Hence these principal axes may be efficient in order to recover the affine transformation and possibly to encode each training point

using the coefficients of projection into 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 and mainly the non-linearities in the shape geometry (cf. Figure 6), 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 [7] denoted  $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$  where  $\langle \cdot, \cdot \rangle$  stands for the inner product and  $\Phi$  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. In contrast to the linear case, this non-linear version is affine invariant in the feature space but not in the input space when using many kernels for instance the well studied Gaussian. In this paper, we will show that our proposed kernel achieves this affine invariance both in the feature and the input spaces and we will discuss later, the fact that any Gram matrix built using the triangular kernel is invertible so 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 (cf. Figure 6).

In the remainder of this paper, we use the following notations:  $\mathcal{X} \subset \mathbb{R}^2$  denotes an *input space* and  $X$  is a random variable standing for the  $2D$  training examples in  $\mathcal{X}$ . We denote by  $\mathcal{S} = \{x_i, i = 1, \dots, N\}$  a training set generated i.i.d (independently and identically distributed) according to a particular and may be unknown probability distribution  $P(X)$ . Other notations will be introduced as we go along through different sections of this paper which is organized as follows: First, we review KPCA in §2, and we will show in §3 our main result of affine invariance using the triangular kernel. We present in §4 our main application, i.e., shape description, then we conclude in §5 and we provide some directions for a future work.

## 2. KERNEL PCA

PCA is an unsupervised statistical analysis which provides a set of orthogonal axes where the projection of a training set using few of these axes, hopefully makes it possible to capture most of the statistical variance of the data. In practice, PCA have been successfully used in image processing, feature extraction, reconstruction, classification, etc [17, 2, 12].

Assuming centered training examples, i.e.,  $\sum_{i=1}^N \Phi(x_i) = 0$ . In the linear case, PCA finds the principal axes by diagonalizing the covariance matrix  $M = \frac{1}{N} \sum_{j=1}^N \Phi(x_j) \Phi(x_j)^t$  where  $x^t$  stands for the transpose of  $x$ . The principal orthogonal axes denoted  $\{V_k, k = 1, \dots, N\}$  can be found by solving the eigenproblem  $M V_k = \lambda_k V_k$  where  $V_k$  and  $\lambda_k$  are respectively the  $k^{th}$  eigenvector and its underlying eigenvalue. It can be shown (see for instance [17]) that the solution of the above eigenproblem lies in the span of the training data, i.e.:

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

where  $\alpha_k = (\alpha_{k1}, \dots, \alpha_{kN})$  are found by solving the following eigenproblem [17]:

$$K \alpha_k = \lambda_k \alpha_k \quad (2)$$

here  $K$  is the Gram matrix on the centered training set  $\mathcal{S}$  in the feature space defined as

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

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

## 3. SCALE INVARIANCE

In this section, we describe the general form of the triangular kernel [15, 8, 14] and the scale invariance of kernel principal component analysis.

### 3.1. The Triangular Kernel

The global form of the unrectified triangular kernel is:

$$k_T(x, x') = -\|x - x'\|^p, \quad p \in \mathbb{R} \quad (3)$$

This defines a conditionally positive definite kernel [4] for  $p \in ]0, 2[$ . This means that for any  $x_1, \dots, x_N$  and any  $c_1, \dots, c_N \in \mathbb{R}$  such that  $\sum_i c_i = 0$ , we have  $\sum_{i,j} c_i c_j k_T(x_i, x_j) \geq 0$ . Due to the centering of the data in the feature space, we

can show that this kernel can be used for KPCA [16].

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:

$$\begin{aligned} k_T(\gamma x, \gamma x') &= -\gamma^p \|x - x'\|^p \\ &= \gamma^p k_T(x, x') \end{aligned}$$

Thus, when the points are scaled by a certain factor  $\gamma$ , the value of the kernel scales by  $\gamma^p$ . We can also rewrite  $k_T(\gamma x, \gamma x') = \langle \sqrt{\gamma^p} \Phi(x), \sqrt{\gamma^p} \Phi(x') \rangle$  which means that any scaling of a population in  $\mathcal{X}$  will also be interpreted as a scaling by a factor  $\gamma^{p/2}$  in the feature space.

### 3.2. Invariance

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

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

here  $\{V_1^{(1)}, \dots, V_N^{(1)}\}$  and  $\{V_1^{(\gamma)}, \dots, V_N^{(\gamma)}\}$  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  $\gamma$ , can be written as:

$$K^{(\gamma)} = \gamma^p K^{(1)} \quad (5)$$

Using (2) it follows that:

$$\begin{aligned} K^{(\gamma)} \alpha_k^{(\gamma)} &= \gamma^p K^{(1)} \alpha_k^{(1)} \\ \Rightarrow \lambda_k^{(\gamma)} \alpha_k^{(\gamma)} &= \gamma^p \lambda_k^{(1)} \alpha_k^{(1)} \end{aligned} \quad (6)$$

here  $\lambda_k^{(\gamma)}$  denotes the  $k^{th}$  eigenvalue of  $K^{(\gamma)}$  at the scale  $\gamma$ . The above equality implies:  $\forall k = 1, \dots, N, \lambda_k^{(\gamma)} = \gamma^p \lambda_k^{(1)}$  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  $\frac{1}{\lambda_1^{(\gamma)}} \sum_{j=1}^N \alpha_{kj}^{(\gamma)} \Phi(\gamma x_j)$ , we can show that the projection in the span of these eigenvectors is scale invariant:

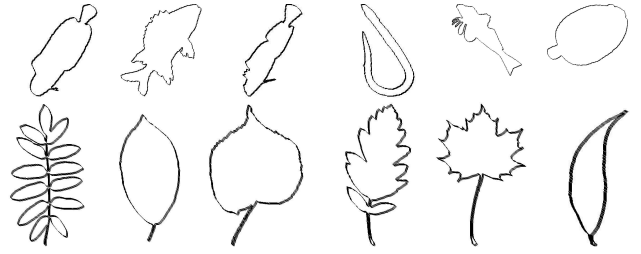
$$\begin{aligned}
\forall x \quad \langle V_k^{(\gamma)}, \gamma x \rangle &= \frac{1}{\lambda_1^{(\gamma)}} \sum_j \alpha_{kj}^{(\gamma)} k_T(\gamma x_j, \gamma x) \\
&= \frac{1}{\lambda_1^{(\gamma)}} \gamma^p \sum_j \alpha_{kj}^{(1)} k_T(x_j, x) \\
&= \frac{1}{\gamma^p \lambda_1^{(1)}} \gamma^p \sum_j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

Thus, this shows that, 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 affine invariant. Notice also that any Gram matrix built using this kernel is invertible for  $p \in ]0, 2[$  [10], so the VC-dimension [21] related to this kernel is infinite. Beside the affine invariance of KPCA using this kernel (which is also achieved by the standard linear one), its discrimination power is high as the invertibility of the Gram matrix ensures that any training set can be approximated with an eigenspace with zero empirical error, i.e., all the data will live in that space. Therefore, 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 shape description.

#### 4. APPLICATION IN 2D SHAPE DESCRIPTION

We ran our experiments both on SQUID<sup>1</sup> and LEAF databases consisting respectively of 1100 fish images and 1125 images of leaves (cf. Figure 1). The SQUID set is only used to show the affine invariance of our description while the LEAF set is used to evaluate its precision. The latter dataset has 15 categories, each one contains 75 images. Notice that for both SQUID and LEAF databases, 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.

Let us define a an external contour (or curve) as  $\mathcal{L} = \{x(s) = (u(s), v(s)) \in \mathbb{R}^2, s \in [0, 1]\}$  and  $\mathcal{S}$  as a uniform sampling of  $\mathcal{L}$ . The former is considered enough representative to capture the whole shape of  $\mathcal{L}$ , so the issues related to adaptive sampling, under-sampling errors will not be considered through this section. In practice, each curve



**Fig. 1.** Some external contours from (top) the SQUID (bottom) LEAF databases.

in the SQUID and LEAF databases containing between 400-1600 points is randomly sampled in order to extract 128 (2D) training points. For each curve in the SQUID set, we synthesize 4 other curves with random orientations (in  $[0^\circ, 360^\circ]$ ), scale factors (in  $[0, 2]$ ) and locations (in  $\pm 20$  pixels). At the end, a total of 5500 curves from SQUID were used in our experiments.

The performance of our shape description is measured empirically on the LEAF set using recall-precision measures. These measures are defined as:

$$\begin{aligned}
\text{precision} &= E \left( \frac{\text{Number of relevant returned images}}{\text{Number of returned images}} \right) \\
\text{Recall} &= E \left( \frac{\text{Number of relevant returned images}}{\text{Number of images in the class of the query}} \right)
\end{aligned}$$

here  $E(\cdot)$  denotes the expectation through all the possible queries in the test set.

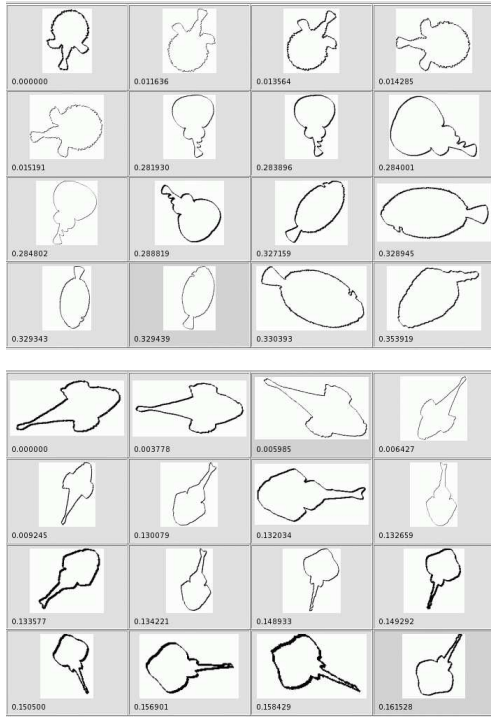
#### 4.1. Indexing

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

For the purpose of shape retrieval, the nearest neighbor classifier based on the Mahalanobis distance is used in the “eigenvalue description” space. Figures (2, 3) illustrates some results on respectively SQUID and LEAF datasets; for each submitted query in SQUID, the system finds first the 4 most similar shapes which differ only by affine transforma-

<sup>1</sup>[www.ee.surrey.ac.uk/Research/VSSP/imagedb/squid.htm](http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/squid.htm)

tions, then the system finds the other similar curves with an increasing order of the Mahalanobis distance.



**Fig. 2.** In both sessions, the top-left images are query shapes while the others are some results sorted from top-left to bottom-right according to their dissimilarity. These curves are taken from SQUID. We can see that the first four curves returned by the system differ only by an affine transformation.

## 4.2. Feature Selection

Let  $\sum_{k=1}^m c_{ik} V_k$  be an expansion of a training sample  $\Phi(x_i)$  in a curve in the span of the  $m$ -principal eigenvectors. The issue of selecting  $m$ , i.e., the number of eigenvectors sufficient to capture the major statistical variance of the data has been tackled in many works for instance [12] and it is interpreted as keeping only few  $m$ -dimensions while minimizing **the Distance From the Feature Space (DFFS)**. This measure corresponds to the deviation of the above expansion from the original one, i.e.,  $\Phi(x_i) = \sum_{k=1}^N c_{ik} V_k$  for  $m < N$ . This error is defined as:

$$\frac{1}{N} \sum_{i=1}^N \left( \sum_{k=m+1}^N c_{ik} V_k \right)^2 \quad (8)$$

Using (8) and  $\langle V_k, V_l \rangle = 1_{\{k=l\}}$ , this error can simply be rewritten as  $\sum_{i=1}^N \sum_{k=m+1}^N c_{ik}^2$ .

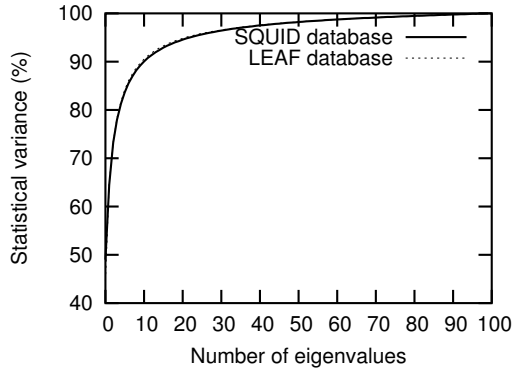


**Fig. 3.** In these sessions, the top-left images are query shapes while the others are some results sorted from top-left to bottom-right according to their dissimilarity. These curves are taken from LEAF database.

Once the  $m$ -principal components are selected, the  $N - m$  remaining eigenvectors correspond to the insignificant and useless details, such as noise, not necessary for discrimination. In our experiments, we select  $m$  eigenvectors in order to capture 99% of the statistical variance, i.e.:

$$\frac{\sum_{k=1}^m \lambda_k}{\sum_{k=1}^N \lambda_k} \times 100 \geq 99\% \quad (9)$$

When averaging this criteria through all the training sets (curves) from SQUID and LEAF databases, we found that among the 128 possible eigenvectors, only the 70-principal axes capture more than 99% of the statistical variance of the data (cf. Figure 4). Using the ground truth in LEAF, we plot in Figure (5) the probability that the first nearest neighbor and the query belong the same class, with respect to  $m$  the number of dimensions. We can see that the precision reaches the highest value (96.31%) when  $m = 84$  dimensions which is close to the precision (95.95%) of the predicted value of  $m = 70$ , i.e., using (9). Figure (6) shows the precision versus recall on the LEAF dataset for  $m = 84$  dimensions using the triangular kernel and  $m = 2$  for the linear one<sup>2</sup>.

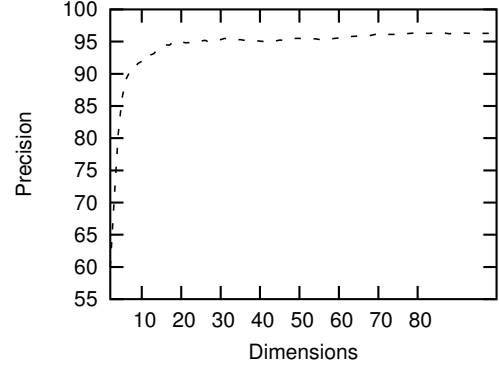


**Fig. 4.** Percentage of the statistical variance with respect to the number of dimensions given by the expectation of  $\sum_{k=1}^m \lambda_k / \sum_{k=1}^N \lambda_k$  over all the curves. We can see that 70 eigenvectors capture more than 99% of the variance. These experiments were conducted on the SQUID and the LEAF datasets. We can see that the two curves are very similar.

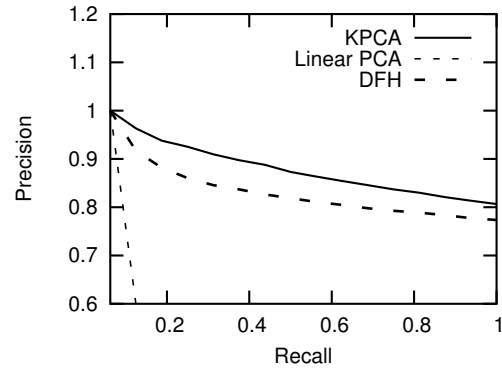
### 4.3. A Possible Interpretation

Define  $\Phi(x_{i,def}) = \Phi(x_i) + c V_k$ , to be a transformation of a curve sample  $x_i$ , which is non-linear in the input space  $\mathcal{X}$ . Moving several samples  $\mathcal{S} = \{x_1, \dots, x_N\}$  according to this transformation may be interpreted as a deformation

<sup>2</sup>As the training examples live in  $2D$ , we have at most two non null eigenvalues when solving KPCA using the linear kernel.



**Fig. 5.** This figure shows the probability that a query and the first nearest neighbor, on the LEAF set, belong to the same class with respect to  $m$  the number of eigenvalues used for shape description. The best precision is reached for  $m = 84$  eigenvalues.

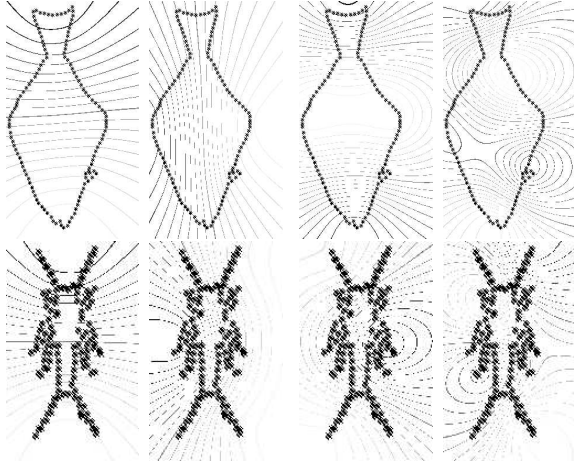


**Fig. 6.** Precision versus recall on the LEAF dataset using the eigenvalue description (with triangular and linear kernels) and DFH “Directional Fragment Histogram” [22]. When using the triangular kernel, 84 eigenvalues are considered for our shape description while (at most) 2 eigenvalues are considered when using the linear kernel.

$S_{def}$  of  $S$  in  $\mathcal{X}$ . The new set  $S_{def} = \{x_{1,def}, \dots, x_{N,def}\}$ , referred to as the *prefeature set* is found by:

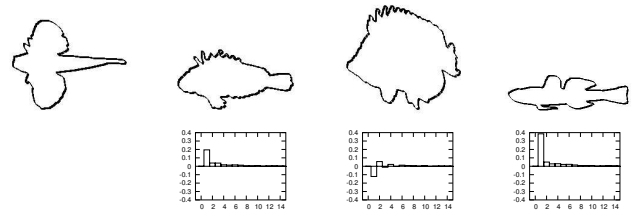
$$x_{i,def} = \arg \min_x \|\Phi(x) - \Phi(x_i) - c V_k\|^2, \quad i = 1, \dots, N \quad (10)$$

Solving this minimization problem makes it possible to understand visually the interpretation of a movement along the principal axis  $V_k$ . In practice, this minimization problem is not convex and difficult when using many kernels including the triangular and the Gaussian[5]. In order to understand the semantic of the principal axes, we proceed as following:



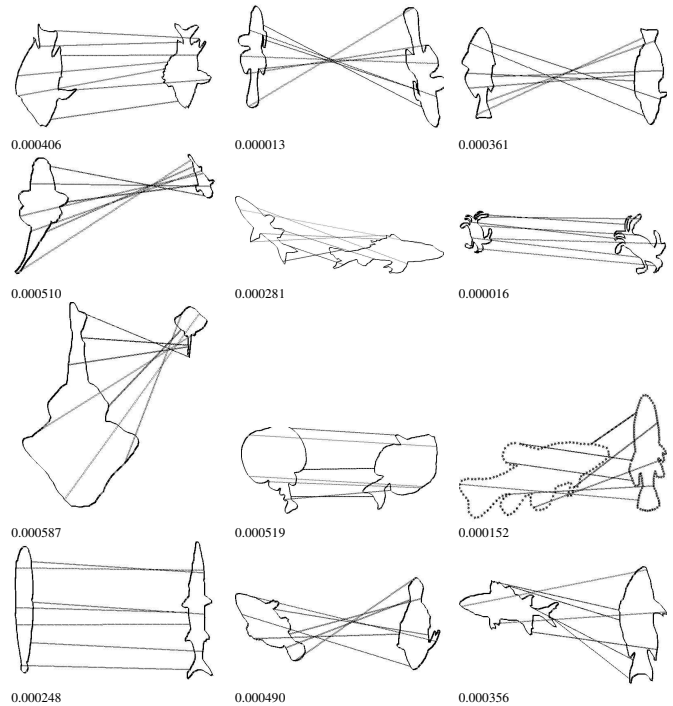
**Fig. 7.** Interpretation of the first principal components. By definition these level-curves characterize the 2D points in the input space  $\mathcal{X}$  with the same correlation with respect to respectively, from the left to the right, the first, second, third and the fourth eigenvectors. The first 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. Usually it is difficult to interpret such dimensions.

First, let us consider two training sets of 2D points, related to two fish contours and let us apply KPCA on these sets (cf. Figure 7). For each case, a test set consisting of  $800 \times 600$  (2D) points were projected on the span of the 4 principal components. The level curves in Figure (7) show for each principal component, the data with the same correlation with respect to this component in the feature space. We can see from the top 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 learn some details, for instance, the shape of the tail or the fin, or other non-linearities in the fish contours.



**Fig. 8.** Top: some fish contours randomly selected from the SQUID dataset. Bottom: “eigenvalue spectrum” differences between the first shape and the others.

In the second experiments, we estimate the principal eigenvectors on different curves shown in Figure (8, top). For each dimension, Figure (8, bottom) shows 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.



**Fig. 9.** Some matching results. The matching scores using the Euclidean distance were also reported. For ease of visualization, only a subset of matches is displayed.

#### 4.4. Matching

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

$$x'_j = \arg \min_{x'_k \in \mathcal{S}_2} \|y_i - y'_k\|^2 \quad (11)$$

Here  $y_i, y'_k$  correspond respectively to the projection of  $x_i$  and  $x'_k$  in the span of the  $m$ -selected eigenvectors. We used this criteria in order to find these good matches through the SQUID dataset, Figure (9) shows some results. Notice that we did not consider any smoothness or neighborhood criteria such as: every two neighboring points in a curve should have neighboring matches in the other one; this can reduce false matches.

#### 5. CONCLUSION AND FUTURE WORK

We introduced in this paper an affine 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 will not 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 affine invariant, when using this kernel and it necessitates finding different scales  $\sigma$  for different curves, so it is meaningless to compare the underlying eigenvalues.

As a 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 envisage also extensive evaluation and comparisons on different databases and also the application of this approach for face and mainly expression analysis. Indeed, affine invariant shape description of mouth contour can be a preliminary step for classification. An other future work will be the application of the method to 3D shape description.

#### Acknowledgment

The author would like to thank the Muscle Network of Excellence for supporting this work, Dr. Itheri Yahiaoui for providing him the contours of the LEAF database and Professor Peter Belhumeur for the useful discussions.

#### 6. REFERENCES

- [1] A. Baumberg and D. Hogg. An adaptive eigenshape model. *In proceedings of the British Machine Vision Conference*, 1995.
- [2] P. Belhumeur, J.P. Hespanha, and D. Kriegman. Eigenfaces vs fisherfaces: Recognition using class specific linear projection. *IEEE transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.
- [3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [4] C. Berg. Harmonic analysis on semigroups: Theory of positive definite and related functions. *Springer Verlag*, 1984.
- [5] C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 375–381. The MIT Press, 1997.
- [6] D. Cremers, T. Kohlberger, and C. Schnrr. Nonlinear shape statistics via kernel spaces. *German National Conference on Pattern Recognition (DAGM)*, 2001.
- [7] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machines. *Cambridge University Press*, 2000.
- [8] F. Fleuret and H. Sahbi. Scale-invariance of support vector machines based on the triangular kernel. *Third International Workshop on Statistical and Computational Theories of Vision (part of ICCV)*, 2003.
- [9] Nishida Hirobumi and Mori Shunji. An algebraic approach to automatic construction of structural models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12), 1993.
- [10] C.A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr Approx*, 2(11), 1986.
- [11] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. *In proceedings of British Machine Vision Conference*, pages 53–62, 1996.
- [12] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspace for face recognition. *In Proceedings of the International Conference on Computer Vision*, pages 84–91, 1994.

- [13] A. Rosenfeld. Axial representation of shape. *CVGIP*, 33(2):156–173, 1986.
- [14] H. Sahbi. Affine invariant shape description using the triangular kernel. *INRIA Research Report, N 5308*, 2004.
- [15] H. Sahbi and F. Fleuret. Scale-invariance of support vector machines based on the triangular kernel. *INRIA Research Report, N 4601*, 2002.
- [16] B. Schölkopf. The kernel trick for distances. *In the proceedings of Neural Information Processing Systems*, pages 301–307, 2000.
- [17] B. Schölkopf, A.J. Smola, and K.-R. Müller. Kernel principal component analysis. *Advances in Kernel Methods - Support Vector Learning (Eds.) B. Schölkopf, C.J.C. Burges and A.J. Smola, MIT Press, Cambridge*, pages 327–352, 1999.
- [18] T.B. Sebastian, P. Klien, and B. Kimia. On aligning curves. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
- [19] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *In Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
- [20] J. Subrahmonia, D.B. Cooper, and D. Keren. Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):505–519, 1996.
- [21] V.N. Vapnik. Statistical learning theory. *A Wiley-Interscience Publication*, 1998.
- [22] I. Yahiaoui and N. Boujemaa. Content-based image retrieval in botanical collections for gene expression studies. *To appear in the Proceedings of the International Conference on Image Processing (ICIP'05)*, 2005.