3D Point Correspondence by Minimum Description Length in Feature Space

Jiun-Hung Chen[1], Ke Colin Zheng[2], and Linda G. Shapiro[1]

[1] Computer Science and Engineering, University of Washington, Seattle, WA 98195
{jhchen,shapiro}@cs.washington.edu
[2] Microsoft Corporation
cozheng@microsoft.com

Abstract. Finding point correspondences plays an important role in automatically building statistical shape models from a training set of 3D surfaces. For the point correspondence problem, Davies *et al.* [1] proposed a minimum-description-length-based objective function to balance the training errors and generalization ability. A recent evaluation study [2] that compares several well-known 3D point correspondence methods for modeling purposes shows that the MDL-based approach [1] is the best method.

We adapt the MDL-based objective function for a feature space that can exploit nonlinear properties in point correspondences, and propose an efficient optimization method to minimize the objective function directly in the feature space, given that the inner product of any vector pair can be computed in the feature space. We further employ a Mercer kernel [3] to define the feature space implicitly. A key aspect of our proposed framework is the generalization of the MDL-based objective function to kernel principal component analysis (KPCA) [4] spaces and the design of a gradient-descent approach to minimize such an objective function. We compare the generalized MDL objective function on KPCA spaces with the original one and evaluate their abilities in terms of reconstruction errors and specificity. From our experimental results on different sets of 3D shapes of human body organs, the proposed method performs significantly better than the original method.

1 Introduction

Statistical shape models show considerable promise as a basis for understanding and interpreting images and have been widely used in model-based image segmentation and tracking [5]. To automatically build statistical shape models [5] from a training set of shapes, finding point correspondence across images becomes an essential task. In this paper, we focus on establishing dense 3D point correspondences between all 3D surfaces of a training set.

There are as many proposed methods and algorithms in automatically computing point correspondences as in statistical shape modeling itself. These approaches vary in terms of the shape representation and registration procedure [6]. Davies *et al.* [1] assumed the projected coefficients of principal component analysis (PCA) of the data have multivariate Gaussian distributions and derived an objective function for point

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 621–634, 2010.

correspondence problems that uses minimum description length (MDL) to balance the training errors and generalization ability. This optimization approach, although slow in convergence, produces high quality matching results. A recent evaluation study [2] compares several well-known 3D point correspondence methods for modeling purposes and shows that the MDL-based approach [1] generates the best results.

Despite all the progress, finding accurate 3D point correspondences has remained a challenging task, largely due to the lack of a well-defined metric for a good correspondence. However, certain properties of a good correspondence can be identified. For example, various nonlinear properties, such as curvature [7] and torsion [8], can not be quantified nor computed by linear combinations of point positions but have been shown not only necessary for modeling shapes but also helpful for finding point correspondences. This suggests that point correspondence algorithms should take nonlinear information into considerations.

Exploiting nonlinear properties in point correspondences to improve results is the main motivation of this paper. Despite being ranked as the state-of-the-art method for finding point correspondences, the MDL-based approach [1] does not capture such knowledge directly, as no local patch information is used. In addition, one key assumption behind the MDL-based approach is that the projected coefficients on principal component analysis have a multivariate Gaussian distribution. Such Gaussian properties are preserved and propagated back via affine transformations (e.g., PCA reconstruction) to all points in the set, which may not reflect reality. In this paper, we propose to overcome this limitation by assuming that the distribution of the projected PCA coefficients of the data in a feature space is a multivariate normal; thus we allow a nonlinear mapping from the input space to the feature space. We further adapt the MDL-based objective function for the feature space, given that the inner product of any vector pair can be computed in the feature space.

Besides presenting a novel objective function, we further propose an efficient optimization method to minimize the objective function directly in the feature space, inspired by the success of applying the gradient descent method proposed by Heimann *et al.* [9][10] on the original MDL-based approach. In order to compute the gradient of the proposed objective function in the feature space, we identify the key condition, which requires the inner product of any vector pair to be computed in the feature space. This requirement is extremely useful for guiding us to a broad set of feature spaces for efficient optimization.

We further employ the Mercer kernel [3] to define the feature space implicitly, given its nice property of supporting pair-wise vector inner product computation. A key aspect of our proposed framework is to generalize the MDL-based objective function to kernel principal component analysis (KPCA) [4] spaces and a gradient descent approach to minimize such an objective function. Although there has been some previous work [11][12] using KPCA in active shape models to model shapes, we are not aware of any previous work that generalizes the MDL-based objective functions to KPCA or shows how to optimize such an objective function. With our generalized framework, the original MDL framework turns out to be a special case where a homogenous polynomial kernel of degree 1 (i.e., an inner product between two vectors) is used.

We compare the generalized MDL objective function on KPCA spaces with the original MDL approach [1] and evaluate their abilities in terms of reconstruction error and specificity. From our experimental results on different sets of 3D shapes of different organs of the body, the proposed method performs significantly better than the original method.

The two main contributions of the paper are summarized below. First, there is a significant theoretical generalization of the MDL-based objective function to feature spaces using gradient descent energy minimization. The original MDL framework, is a special case of this generalization, when an inner product is used. Second, besides the theoretical improvement, the empirical contribution is also substantial. Overcoming the limitation that nonlinear properties are not included in the original MDL framework directly is significant as our proposed KPCA approach yields much better results.

2 Previous Work

The objective functions automatic methods used to quantize the quality of point correspondences can be partitioned into three classes: shape-based, model-based and information-theoretic objective functions [5]. Shape-based objective functions are based on similarity between shapes and the representative examples use Euclidean distances, bending energy, curvatures, shape contexts [13] and SPHARM [14] to measure shape similarity. In contrast model-based objective functions consider the statistics of the dissimilarity among shapes; the determinant of the model covariance is a representative example. Information-theoretic objective functions uses information theoretic measures, such as MDL and mutual information [1][15]. A recent evaluation study [2] that compares several well-known 3D point correspondence methods for modeling purposes shows that an information theoretic objective function, the MDL-based approach [1], is the best method. Because of its superior performance, this class of information theoretic objective functions is the main focus in this paper.

In the following, we first review the MDL-based approach [1] in detail. Then, PCA and KPCA, which play important roles in both MDL-based objective functions and understanding the proposed framework, are reviewed. Assume that we have a training set of N 3D shapes, $\Gamma = \{x_1, x_2, \ldots, x_N\}$, and each shape is represented by M 3D landmarks points. Conventionally, we can represent each such shape by a vector with dimension $3M \times 1$. Note that 3D shapes are used for illustration only and all the methods we will review can be applied to both 2D curves and 3D shapes.

2.1 Correspondence by Minimizing Description Length

Davies *et al.* [1] proposed a MDL-based objective function to quantize the quality of the point correspondences. In this paper, we use the commonly-used version F proposed by Thodberg [16] as defined below.

$$F = \sum_{k=1}^{N} L_k \text{ with } L_k = \begin{cases} 1 + \log(\lambda_k/\lambda_{cut}), & \text{if } \lambda_k \geq \lambda_{cut} \\ \lambda_k/\lambda_{cut}, & \text{otherwise} \end{cases} \tag{1}$$

Given a set of shapes and a set of known point correspondences, PCA is computed on the set of shapes, and the computed eigenvalues, $\{\lambda_k | k = 1, \ldots, N\}$, are used to calculate F in (1). λ_{cut} is a parameter that determines the point where we effectively switch between the determinant-type term (i.e., the if-part in (1)) and the trace-type term (i.e., the otherwise-part in (1)). The determinant-type terms jointly measure the volume of the training set after correspondence in shape space, which favors compactness. The trace-type terms jointly measure similarity of each pair of the training shapes after correspondence via Euclidean distance.

Given the above MDL-based objective function, an efficient method for manipulating point correspondences and an optimization algorithm that minimizes the objective function are required in order to find optimal point correspondences [5][9]. Typically, manipulating point correspondences is treated as parameterizing and then re-parameterizing the surfaces. A parameterization assigns every point on the surface of the mesh to a unique point on the unit sphere, although parameterizations may not exist for arbitrary surfaces. In this paper, we assume that the 3D shapes are closed two-manifolds of genus 0. We use a conformal mapping as a parameterization and a reparameterization that modifies the parameterization based on kernels with strictly local effects, as developed in [9].

We assume that the parameterization of the ith shape is controlled by some parameter vector α_i, for which the individual parameters are given by $\{\alpha_{i,a} | a = 1, \ldots, A\}$. The gradient descent approach is used to minimize F with respect to a parameter vector α_i. The Jacobian matrix for the gradient of the objective function is defined as

$$\frac{\partial F}{\partial \alpha_{i,a}} = \sum_{k=1}^{N} \frac{\partial L_k}{\partial \lambda_k} \frac{\partial \lambda_k}{\partial \alpha_{i,a}} \tag{2}$$

It is easy to compute $\frac{\partial L_k}{\partial \lambda_k}$ (see (1)) and so we focus on $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ in the following discussions. $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ can be computed by using the following chain rule for derivatives.

$$\frac{\partial \lambda_k}{\partial \alpha_{i,a}} = \frac{\partial \lambda_k}{\partial x_i} \cdot \frac{\partial x_i}{\partial \alpha_{i,a}} \tag{3}$$

While $\frac{\partial x_i}{\partial \alpha_{i,a}}$ is typically computed by using finite differences, the following analytic form for $\frac{\partial \lambda_k}{\partial x_i}$ exists:

$$\frac{\partial \lambda_k}{\partial x_i} = 2(1 - \frac{1}{N}) c_{i,k} b_k. \tag{4}$$

where $c_{i,k}$ is the projection coefficient of the i-th shape vector x_i onto the k-th eigenvector b_k.

2.2 PCA and KPCA

PCA. PCA is a common approach to model the shape variations of a given training set of 3D shapes. The total scatter matrix $\mathbf{S}$ is defined as

$$\mathbf{S} = \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^t \tag{5}$$

where $\bar{x}$ is the mean shape vector as defined below.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{6}$$

PCA finds a projection axis b that maximizes $b^t S b$. Intuitively, the total scatter of the projected samples is maximized after the projection of the samples onto b. The optimal Q projection axes $b_q, q = 1, \ldots, Q$ that maximize the above criterion are the eigenvectors of S corresponding to the largest Q eigenvalues, $\{\lambda_q | q = 1, \ldots, Q\}$. The reconstruction $\tilde{x}$ of shape vector x can be used to approximate it.

$$\tilde{x} = \bar{x} + \sum_{q=1}^{Q} c_q b_q \tag{7}$$

where $c_q = (x - \bar{x})^t b_q$.

KPCA. Assume that we have an input space of shapes $\Psi = R^{3M \times 1}$, a feature space Ω, and a mapping $\phi : \Psi \rightarrow \Omega$. Instead of performing PCA in the input space Ψ, KPCA performs PCA in the feature space Ω.

The mean of the data points in the feature space, $\hat{x}$, is defined as follows.

$$\hat{x} = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \tag{8}$$

The covariance matrix $\mathbf{C}$ can be defined as follows.

$$\mathbf{C} = \sum_{i=1}^{N} (\phi(x_i) - \hat{x})(\phi(x_i) - \hat{x})^t \tag{9}$$

Let β denoting the column vector with entries, $\beta_1, \beta_2, \ldots, \beta_N$, which can be computed by solving the following eigenvalue problem.

$$N \lambda \beta = \tilde{\mathbf{K}} \beta \tag{10}$$

where $\tilde{\mathbf{K}}_{ij} = (\mathbf{K} - 1_N \mathbf{K} - \mathbf{K} 1_N + 1_N \mathbf{K} 1_N)_{ij}$, $\mathbf{K} = [\mathbf{K}_{ij}]$ is a $N \times N$ Gram matrix, and $\mathbf{K}_{ij} = \phi(x_i) \cdot \phi(x_j)$.

To require e, an eigenvector, to be a unit vector, an additional constraint on β must be posed.

$$1 = \lambda \beta \cdot \beta \tag{11}$$

Let $\{e_q, q = 1, \ldots, Q\}$ be the eigenvectors of $\mathbf{C}$ with the largest Q eigenvalues $\{\lambda_q | q = 1, \ldots, Q\}$. Any eigenvector e_i of $\mathbf{C}$ can be expressed as

$$e_i = \sum_{j}^{N} \beta_{ij} \phi(x_j) \tag{12}$$

The reconstruction $\widetilde{\phi(x)}$ of $\phi(x)$ can be used to approximate it.

$$\widetilde{\phi(x)} = \hat{x} + \sum_{q=1}^{Q} c_q e_q \tag{13}$$

where $c_q = (\phi(x) - \hat{x}) \cdot e_q$.

Instead of using an explicitly defined mapping ϕ, we can use a Mercer kernel [3] that satisfies the following constraint:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{14}$$

Commonly used Mercer kernels include Gaussian radial basis functions (RBFs), inhomogeneous polynomial functions, and sigmoidal functions. Gaussian RBFs are defined as

$$K(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{15}$$

where $\sigma \in R$ is a kernel parameter, and $||x||$ is the Euclidean norm of x. Inhomogeneous polynomial kernels of degree $d \in R$ are defined as

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{16}$$

In contrast with inhomogeneous polynomial kernels where the constant one is added in the definition, homogeneous polynomial kernels of degree d are defined as

$$K(x_i, x_j) = (x_i \cdot x_j)^d \tag{17}$$

The common inner product between two vectors x_i and x_j is a special case of a homogenous kernel of degree 1. If such a kernel is used in KPCA, KPCA degenerates to PCA.

3 The Proposed Framework

In the following, we first focus on general feature spaces and then on special feature spaces called Mercer-kernels-induced feature spaces.

3.1 General Feature Spaces

In contrast with [1][16][9] that perform all the work in the input space Ψ, we generalize and perform our work in the feature space Ω. In other words, instead of using the eigenvalues computed by PCA in (1), we propose to use those computed by PCA in the feature space Ω. We propose a gradient descent approach to minimize the objective function based on the ideas in Section 2.2. to compute the Jacobian matrix for the gradient of the objective function.

The Jacobian matrix for the gradient of the objective function is defined as

$$\frac{\partial F}{\partial \alpha_{i,a}} = \sum_{k=1}^{N} \frac{\partial L_k}{\partial \lambda_k} \frac{\partial \lambda_k}{\partial \alpha_{i,a}} \tag{18}$$

As in Section 2.1, it is easy to compute $\frac{\partial L_k}{\partial \lambda_k}$, and so we focus on $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ here. $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ can be computed by using the following chain rule for derivatives.

$$\frac{\partial \lambda_k}{\partial \alpha_{i,a}} = \frac{\partial \lambda_k}{\partial \phi(x_i)} \cdot \frac{\partial \phi(x_i)}{\partial \alpha_{i,a}} \tag{19}$$

The term, $\frac{\partial \phi(x_i)}{\partial \alpha_{i,a}}$, is typically approximated by using finite differences. For example, we can use a forward difference to approximate $\frac{\partial \phi(x_i)}{\partial \alpha_{i,a}}$ as follows.

$$\frac{\partial \phi(x_i)}{\partial \alpha_{i,a}} \approx \phi(x_i + \triangle\alpha_{i,a}) - \phi(x_i) \tag{20}$$

where $\triangle\alpha_{i,a}$ is a predefined small quantity. In addition to the above forward difference method, it is also possible to use other finite difference methods, such as backward and central differences and high-order difference methods.

In this paper, we focus on a general class of finite difference methods whose calculations can be represented by a weighted linear combination, $\sum_{p=1}^{P} w_p \phi(y_p)$, where $\{w_p | p = 1, \ldots, P\}$ is a given set of weights and $\{y_p | p = 1, \ldots, P\}$ is a given set of shape vectors as shown below.

$$\frac{\partial \phi(x_i)}{\partial \alpha_{i,a}} \approx \sum_{p=1}^{P} w_p \phi(y_p) \tag{21}$$

Note that forward, backward and central differences, as well as high order difference methods, are representative examples in this class.

In contrast with using finite differences to approximate $\frac{\partial \phi(x_i)}{\partial \alpha_{i,a}}$, the following analytic form for $\frac{\partial \lambda_k}{\partial \phi(x_i)}$ exists[1].

$$\frac{\partial \lambda_k}{\partial \phi(x_i)} = 2(1 - \frac{1}{N}) c_{i,k} e_k \tag{22}$$

where $c_{i,k}$ is the projection coefficient of the feature vector $\phi(x_i)$ of i-th shape vector x_i onto the k-th eigenvector e_k.

By plugging (21), (22) and (12) into (19),

$$\begin{aligned}
\frac{\partial \lambda_k}{\partial \alpha_{i,a}} &= \frac{\partial \lambda_k}{\partial \phi(x_i)} \cdot \frac{\partial \phi(x_i)}{\partial \alpha_{i,a}} \\
&\approx 2(1 - \frac{1}{N}) c_{i,k} e_k \cdot (\sum_{l}^{L} w_l \phi(y_l)) \\
&= 2(1 - \frac{1}{N}) c_{i,k} (\sum_{j=1}^{N} \beta_{k,j} \phi(x_j)) \cdot (\sum_{p=1}^{P} w_p \phi(y_p)) \\
&= 2(1 - \frac{1}{N}) c_{i,k} \sum_{j=1}^{N} \beta_{k,j} \sum_{p=1}^{P} w_p \phi(x_j) \cdot \phi(y_p)
\end{aligned} \tag{23}$$

[1] The full derivations can be found in appendix A.

From (23), a key insight is that the calculation of $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ depends on the inner product of two vectors in the feature space $(\phi(x_j) \cdot \phi(y_p))$. If an explicitly defined mapping $\phi(x)$ from Ψ to Ω is used, $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ can be computed by (23).

It is easy to see that the previous methods [1][16][9] are special cases of our work when $\phi(x) = x$. In other words, when $\phi(x) = x$ is used in the proposed framework, the objective function degenerates to (1)[1][16], and the above gradient descent optimization approach degenerates to the one in Section 2.2 [9]. In addition, our framework allows nonlinear information easily if we choose $\phi(x)$ as a nonlinear mapping from Ψ to Ω.

3.2 Mercer-Kernel-Induced Feature Spaces

Instead of using an explicitly defined mapping $\phi(x)$ from Ψ to Ω , we can in (23) use a Mercer kernel (14) that implicitly induces a mapping. In other words, (23) can be further simplified as follows by plugging (14) into the right-hand side of the last equation in (23).

$$\frac{\partial \lambda_k}{\partial \alpha_{i,a}} \approx 2(1 - \frac{1}{N})c_{i,k} \sum_{j=1}^{N} \beta_{k,j} \sum_{p=1}^{P} w_p K(x_j, y_p) \tag{24}$$

Although nonlinear mappings are allowed in both (23) and (24), their time complexities can be very different. In contrast with the time complexity of using (23) to compute $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ depending on the dimensionality of the feature space, the time complexity of using (24) to compute $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ depends on the dimensionality of the input space. If a Mercer kernel is used, our framework can deal with nonlinear mapping functions whose feature spaces with infinite dimensionality (the dimensionality of $\phi(x)$ is infinite) and still keep its time complexity dependent on the dimensionality of the input space and not on the dimensionality of the feature space. Note that although we focus on using a Mercer kernel in the above discussions, the proposed framework naturally allows using multiple Mercer kernels without any modifications.

4 Experiments

We have 3D triangular mesh models of 17 left kidneys, 15 right kidneys, and 18 spleens as shown in Figure 1. All 3D meshes are constructed from CT scans of different patients[2]. After correspondences are found, all the mesh models of the same organ have the same number of vertices (2563) and the same number of faces (5120), and all vertices are used as landmarks to represent the shapes. Two methods, the proposed method and MDL, are compared. The code [10][9] that implements the ideas described in Section 2.1 is used as an implementation of MDL, and the implementation of the proposed method is built on top of it. The same heuristic used in [10][9] is used to select λ_{cut} values for the organ dataset on which the two methods are compared. A weighted forward difference (e.g., a weighted form of (20)) is used in (21).

[2] We constructs the shape of an organ from manual segmentation of CT scans of a patient by using marching cubes in ITK-SNAP.

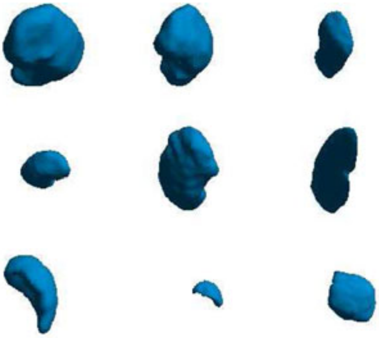
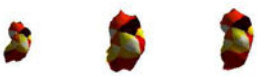
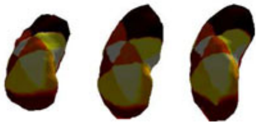
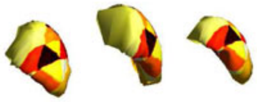
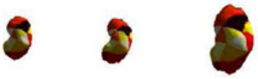
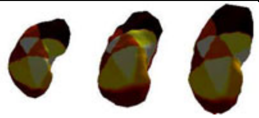
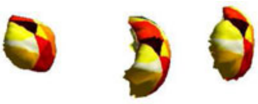
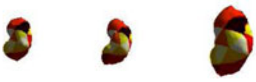
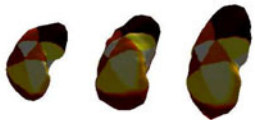
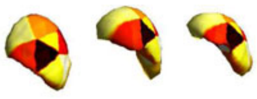
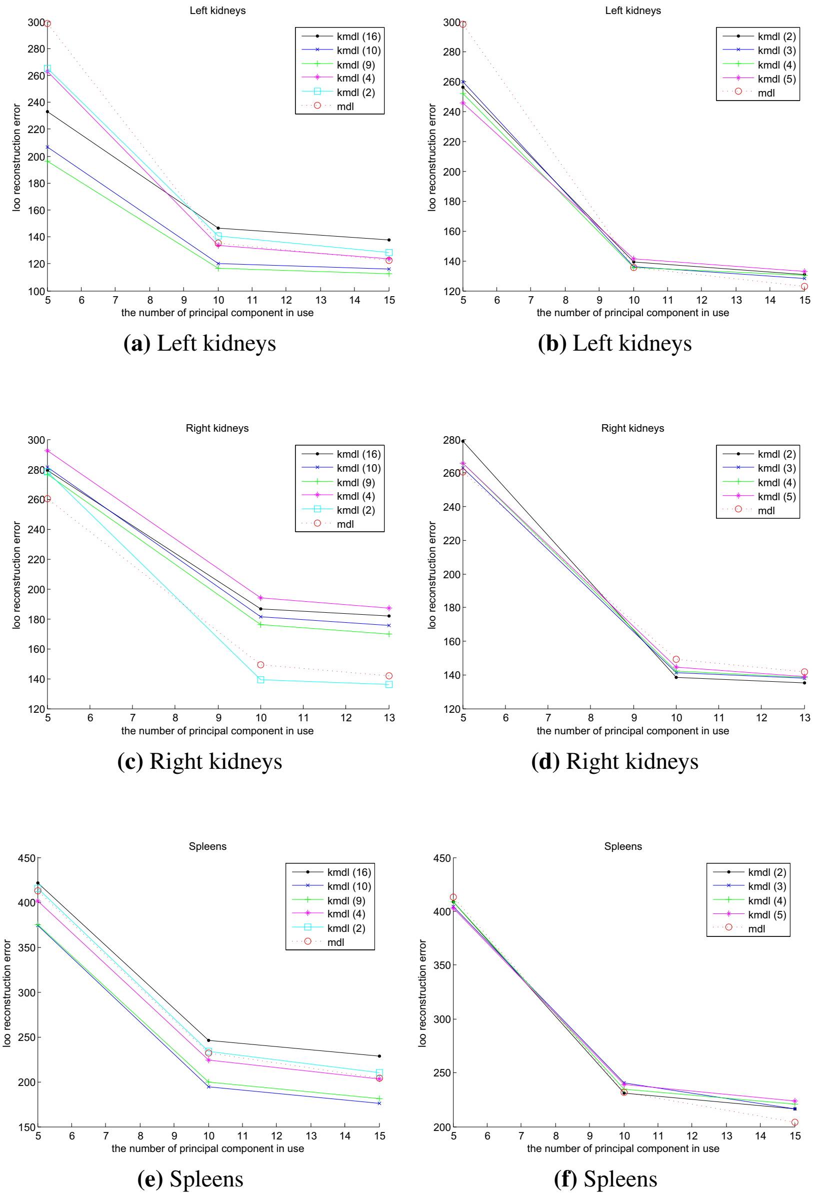


Fig. 1. Some examples of the 3D triangular meshes of different organs used in the experiments. From the top row to the bottom row are left kidneys, right kidneys and spleens, respectively.

We follow a standard procedure extensively used in [1][5][2][9] to compare different point correspondence methods when the ground truth correspondences among different shapes are not available, and two standard evaluation measures, leave-one-out cross validation and specificity, are used. Leave-one-out cross validation is used to determine how accurately an algorithm will be able to predict data that it was not trained on. The evaluation measure for this method is the difference between an unknown shape and its reconstruction. In contrast, given a set of shapes sampled from the probability density function of the training set, the specificity measure computes the average distance from each sampled shape to the nearest element of the training set. In both measures, the Euclidean distance (i.e, the sum of the distances between all pairs of corresponding landmarks) is used to measure the difference between two shapes.

Table 1. The point correspondences found with the compared methods. The columns show different organs. The rows show the results with the proposed method with Gaussian RBF kernels, the results with the proposed method with inhomogeneous polynomial kernels and the results with MDL, respectively. Points that correspond are shown in same colors.

	Left Kidney	Right Kidney	Spleens
MDL+K(G)			
MDL+K(P)			
MDL			



(a) Left kidneys

(b) Left kidneys

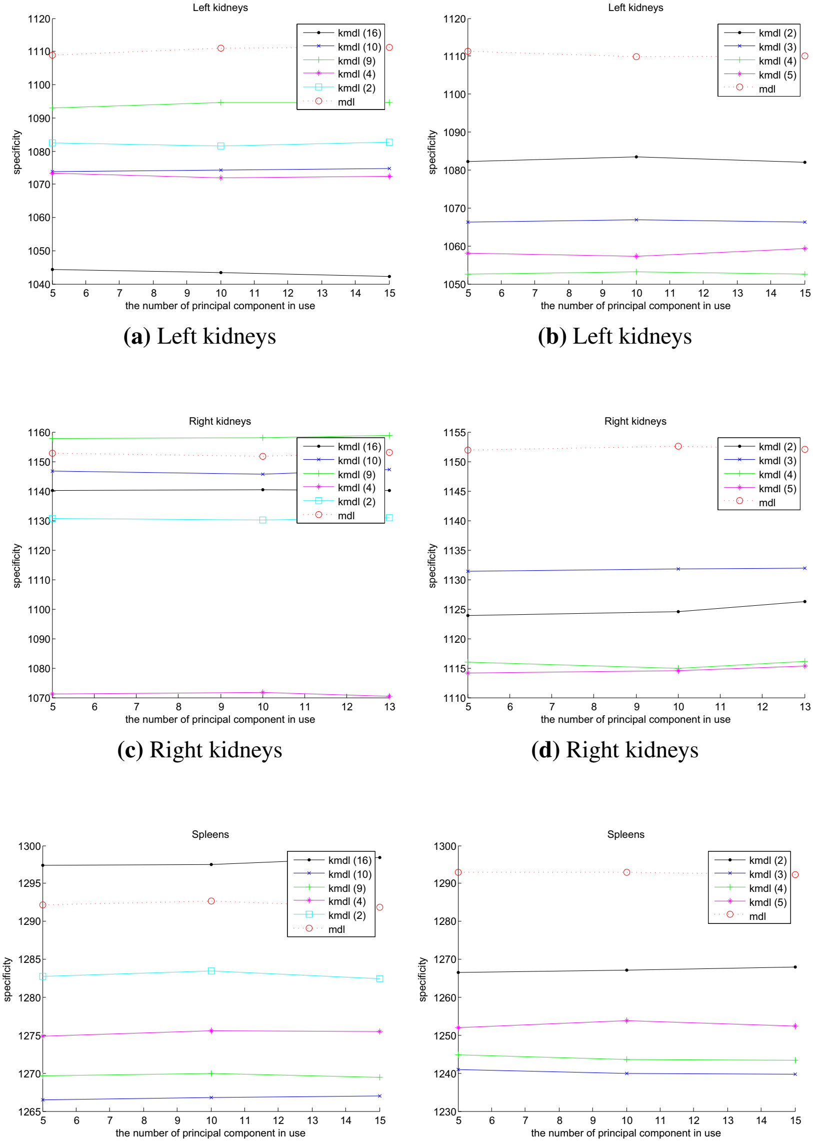
(c) Right kidneys

(d) Right kidneys

(e) Spleens

(f) Spleens

Fig. 2. How the leave-one-out reconstruction errors for different organs change with different kernel parameters and the numbers of principal components in use. The rows show different organs. The first column shows the results with the Gaussian RBF kernels while the second column shows the results with inhomogeneous polynomial kernels.



(a) Left kidneys

(b) Left kidneys

(c) Right kidneys

(d) Right kidneys

(e) Spleens

(f) Spleens

Fig. 3. How the average specificity for different organs change with different kernel parameters and the numbers of principal components in use. The rows show different organs. The first column shows the results with the Gaussian RBF kernels while the second column shows the results with inhomogeneous polynomial kernels.

Figure 2 shows the changes in leave-one-out reconstruction errors for different organs with different kernel parameters and the numbers of principal components in use. The kernel parameters can greatly affect the reconstruction errors; for example, the parameters in Gaussian RBF kernels, 9 and 10, gave significantly lower errors than 2 and 16 in Figure 2(a). In addition, some feature spaces induced by using different parameters in Mercer kernels failed to capture the nonlinear properties in the point correspondences and performed worse than MDL. The proposed method with Gaussian RBF kernels, MDL+K(G), and the best kernel parameters, is better than MDL for left kidneys and spleens and comparable to MDL for right kidneys. In contrast,the proposed method with inhomogeneous polynomial kernels, MDL+K(P), and the best kernel parameters, is comparable to MDL in all the datasets. Figure 3 shows that the specificity measures for different organs change with different kernel parameters and with the numbers of principal components in use. MDL+K(G) and MDL+K(P) have better performances than MDL, which has either the worst or the second worst performance in all datasets. From these two figures, it can be concluded that MDL+K(G) is the best among the compared methods. Table 1 shows point correspondences resulting from the models with the lowest reconstruction errors in Figure 2 in for visual comparisons.

The better performance of MDL+K(G) is mainly attributed to the fact that KPCA with Gaussian RBF kernels can model nonlinear properties in the point correspondences, while PCA can not. The comparisons between MDL+K(G) and MDL+K(P) show that for the test datasets, Gaussian RBF kernels are more suitable than MDL+K(P) in capturing nonlinear properties in the point correspondences.

5 Conclusions and Future Work

In this paper, we generalize the MDL-based objective function to feature spaces and propose a gradient descent approach to minimize the objective function. The original MDL framework is a special case of this theory when an inner product is used in the proposed framework. We empirically compare the generalized MDL objective function on KPCA spaces with the original one. From our experimental results on different sets of 3D shapes of different organs, the proposed method is better than the MDL in terms of the reconstruction errors and specificity.

Instead of using the reconstruction errors and specificity, we plan to use some datasets whose ground truth correspondences are known to directly compare the proposed method with other existing methods. We currently use a brute-force approach to test all possible kernel parameter values and select the best one. Because the effect of kernel parameters can affect the reconstruction errors and specificity greatly, a future study is to investigate how to choose the kernel parameters that perform best and under what conditions on input shapes the proposed framework is guaranteed to perform better than the original one. In the experiments, we only focus on Mercer kernels that can implicitly induce nonlinear feature spaces. However, the induced nonlinear mappings may not be anatomically meaningful. Hence, an interesting future direction is to incorporate priori knowledge into the Mercer kernel, so that an anatomically meaningful feature space can be induced.

Acknowledgment

This work was supported by NSF Grant DBI-0543631 and NIH-NIDCR grant 1U01DE 020050-01.

References

1. Davies, R., Twining, C., Cootes, T., Waterton, J., Taylor, C.: A minimum description length approach to statistical shape modeling. IEEE TMI 21(5), 525–537 (2002)
2. Styner, M., Rajamani, K., Nolte, L.P., Zsemlye, G., Szekely, G., Taylor, C., Davies, R.H.: Evaluation of 3d correspondence methods for model building. In: IPMI, pp. 63–75 (2003)
3. Vapnik, V.: The Nature of Statisticl Learning Theory. Springer, New York (1995)
4. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10(5), 1299–1319 (1998)
5. Davies, R., Twining, C., Taylor, C.: Statistical Models of Shape: Optimisation and Evaluation. Springer Publishing Company, Heidelberg (2008)
6. Heimann, T., Meinzer, H.P.: Statistical shape models for 3d medical image segmentation: A review. Medical Image Analysis 13(4), 543–563 (2009)
7. Styner, M., Oguz, I., Heimann, T., Gerig, G.: Minimum description length with local geometry. In: ISBI 2008, pp. 1283–1286 (2008)
8. Corouge, I., Gouttard, S., Gerig, G.: Towards a shape model of white matter fiber bundles using diffusion tensor mri. In: ISBI 2004, vol. 1, pp. 344–347 (2004)
9. Heimann, T., Wolf, I., Williams, T.G., Meinzer, H.P.: 3d active shape models using gradient descent optimization of description length. In: IPMI, pp. 566–577 (2005)
10. Heimann, T., Oguz, I., Wolf, I., Styner, M., Meinzer, H.: Implementing the automatic generation of 3d statistical shape models with itk. In: Open Science Workshop at MICCAII (2006)
11. Romdhani, S., Gong, S., Psarrou, R.: A multi-view nonlinear active shape model using kernel pca. In: BMVC, pp. 483–492 (1999)
12. Twining, C.J., Taylor, C.J.: Kernel principal component analysis and the construction of non-linear active shape models. In: BMVC, pp. 23–32 (2001)
13. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24, 509–522 (2002)
14. Brechbühler, C., Gerig, G., Kübler, O.: Parametrization of closed surfaces for 3-d shape description. CVIU 61(2), 154–170 (1995)
15. Wang, Y., Chiang, M., Thompson, P.: Mutual informationbased 3d surface matching with applications to face recognition and brain mapping. In: ICCV 2005, vol. 1, pp. 527–534 (2005)
16. Thodberg, H.H.: Minimum description length shape and appearance models. In: IPMI, pp. 51–62 (2003)

Appendix A

In the following, we will derive an analytic form for $\frac{\partial \lambda_k}{\partial \phi(x_i)}$. Assume the k-th eigenvector and eigenvalue of $\mathbf{C}$ defined in (9) are e_k and λ_k, respectively. By definition of eigenvalues and eigenvectors of a matrix, we have $\mathbf{C}e_k = \lambda_k e_k$. The inner product of e_k

and $\mathbf{C}e_k$, $e_k^t\mathbf{C}e_k$, is λ_k because $e_k^t\lambda_k e_k = \lambda_k$ and the above relation can be expressed by using the following equation.

$$\begin{aligned}\lambda_k &= \sum_{i=1}^{N} e_k^t(\phi(x_i) - \hat{x})(\phi(x_i) - \hat{x})^t e_k \\ &= \sum_{i=1}^{N}((\phi(x_i) - \hat{x})^t e_k)^2 \end{aligned} \tag{25}$$

where the second equality is obtained by substituting $\mathbf{C}$ by its definition in (9).

$\frac{\partial\lambda_k}{\partial\phi(x_i)}$ can be obtained by the following chain rules.

$$\begin{aligned}\frac{\partial\lambda_k}{\partial\phi(x_i)} &= \frac{\partial\lambda_k}{\partial(\phi(x_i) - \hat{x})^t e_k}\frac{\partial(\phi(x_i) - \hat{x})^t e_k}{\partial(\phi(x_i) - \hat{x})}\frac{\partial(\phi(x_i) - \hat{x})}{\partial\phi(x_i)} \\ &= 2(\phi(x_i) - \hat{x})^t e_k e_k^t(1 - \frac{1}{N})I \\ &= 2(1 - \frac{1}{N})(\phi(x_i) - \hat{x})^t e_k e_k^t \end{aligned} \tag{26}$$

I is an identity matrix both of whose dimensions are the same as e_k. For some nonlinear mapping functions ϕ, the dimensionality of $\phi(x)$ can be infinite. Note that $\frac{\partial\lambda_k}{\partial\phi(x_i)}$ in (26) is treated as a row vector but it is defined as a column vector in the main paper.