Hierarchical Organization of Shapes for Efficient Retrieval

Shantanu Joshi[1], Anuj Srivastava[2], Washington Mio[3], and Xiuwen Liu[4]

[1] Department of Electrical Engineering, joshi@eng.fsu.edu,
[2] Department of Statistics, anuj@stat.fsu.edu,
[3] Department of Mathematics, mio@math.fsu.edu
[4] Department of Computer Science, liux@cs.fsu.edu
Florida State University, Tallahassee, FL 32310

Abstract. This paper presents a geometric approach to perform: (i) hierarchical clustering of imaged objects according to the shapes of their boundaries, and (ii) testing of observed shapes for classification. An intrinsic metric on nonlinear, infinite-dimensional shape space, obtained using geodesic lengths, is used for clustering. This analysis is landmark free, does not require embedding shapes in $\mathbb{R}^2$, and uses ordinary differential equations for flows (as opposed to partial differential equations). Intrinsic analysis also leads to well defined shape statistics such as means and covariances, and is computationally efficient. Clustering is performed in a hierarchical fashion. At any level of hierarchy clusters are generated using a minimum dispersion criterion and an MCMC-type search algorithm. Cluster means become elements to be clustered at the next level. Gaussian models on tangent spaces are used to pose binary or multiple hypothesis tests for classifying observed shapes. Hierarchical clustering and shape testing combine to form an efficient tool for shape retrieval from a large database of shapes. For databases with n shapes, the searches are performed using $\log(n)$ tests on average. Examples are presented for demonstrating these tools using shapes from Kimia shape database and the Surrey fish database.

1 Introduction

An important goal in image analysis is to classify and recognize objects of interest present in the observed images. Imaged objects can be characterized in many ways: according to their colors, textures, shapes, movements, and locations. The past decade has seen large efforts in modeling and analysis of pixel values or textures in images to attain these goals albeit with limited success. An emerging opinion in the scientific community is that global features such as shapes be taken into account. *Characterization of complex objects using their global shapes is fast becoming a major tool in computer vision and image understanding.* Analysis of shapes, especially those of complex objects, is a challenging task and requires sophisticated mathematical tools. Applications of shape analysis include biomedical image analysis, fisheries, surveillance, biometrics, military target recognition and general computer vision.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3023, pp. 570–581, 2004.

Shape is a characteristic that is invariant to rigid motion and uniform scaling of objects, and it becomes natural to analyze shape in a quotient space where these shape preserving transformations have been removed. Shapes have been an important topic of research over the past decade. A majority of this research has been restricted to "landmark-based" analysis where shapes are represented by a coarse, discrete sampling of the object contours [3]. One establishes equivalences of samples (or landmarks) with respect to shape preserving transformations, i.e. rigid rotation and translation, and non-rigid uniform scaling, and then compares shapes in the resulting quotient spaces. This approach is limited in that automatic detection of landmarks is not straightforward and the ensuing shape analysis depends heavily on the choice of landmarks. Another approach is to study the whole object: boundary + interior, in modeling shapes [7]. One limitation of this approach is the need to find computationally expensive diffeomorphisms of $\mathbb{R}^n$ that match shapes. In case the interest lies only in the shapes of boundaries, more efficient techniques can be derived. Another active area of research in image analysis has been the use of level sets and active contours in characterizing object boundaries. However, the focus here is mainly on solving partial differential equations (PDEs) for shape extraction, driven by image features under smoothness constraints, and statistical analysis is more recent [6, 2].

In a recent paper [10], Klassen et al. consider the shapes of continuous, closed curves in $\mathbb{R}^2$, without the need for defining landmarks, diffeomorphisms of $\mathbb{R}^2$, or nonlinear PDEs. The basic idea is to identify a space of allowable shapes, impose a Riemannian structure on it and utilize its geometry to solve optimization and inference problems. They consider the space of simple closed curves (extended somewhat to simplify analysis), and remove invariances such as rotation, translation, scaling, and re-parametrization to form a shape space. For planar curves in $\mathbb{R}^2$, of length 2π and parameterized by the arc length, the coordinate function $\alpha(s)$ relates to the direction function $\theta(s)$ according to $\dot{\alpha}(s) = e^{j\,\theta(s)}$, $j = \sqrt{-1}$. Direction functions are used to represent curves. Considering closed curves, and making them invariant under rigid motions (rotations, translations), and uniform scaling, one obtains:

$$\mathcal{C} = \{\theta \in \mathbb{L}^2 | \frac{1}{2\pi}\int_0^{2\pi} \theta(s)ds = \pi,\ \int_0^{2\pi} e^{i\theta(s)}ds = 0\}\ . \tag{1}$$

Here $\mathbb{L}^2$ is the set of square integrable functions on $[0, 2\pi)$. After removing the re-parametrization group (relating to different placements of $s = 0$ on the same curve), the quotient space $\mathcal{S} \equiv \mathcal{C}/S^1$ is considered as the shape space. Analysis on shapes as elements of $\mathcal{S}$ requires computation of geodesic paths on $\mathcal{S}$. Let $\Psi(\theta, f, t)$ denote a geodesic path starting from $\theta \in \mathcal{S}$, in the direction $f \in T_\theta(\mathcal{S})$, as a function of time t. In practice, f is represented using an orthogonal expansion according to $f(s) = \sum_{i=1}^{\infty} x_i e_i(s)$, where $\{e_i,\ i = 1, \ldots,\}$ form an orthonormal basis of $T_\theta(\mathcal{S})$, and the search for f (to go from one shape to another) can be performed via a search for corresponding $\mathbf{x} = \{x_1, x_2, \ldots,\}$.

In this paper we advance this idea by studying the following problems.

1. **Problem 1: Hierarchical Clustering**: We will consider the problem of clustering planar objects according to the shapes of their boundaries. To improve efficiency, we will investigate a hierarchy of clusters in which the shapes are recursively clustered in form of a tree. Such an organization can significantly improve database searches and systems with shape-based queries. While retrieving shapes, one can test against a representative shape of each cluster at each level, instead of the whole shape database, and then search for the closest shape in that cluster.
2. **Problem 2: Testing Shape Hypotheses**: Once a shape model is established, it allows for application of decision theory. For example, the question *given an observed shape and two competing shape classes, which class does this shape belong to?* is essentially a binary hypothesis test. Hypothesis tests for landmark-based shape analysis have already been derived [3]. Hypothesis testing, together with hierarchical organization can provide an efficient tool for shape retrieval from a large database.

Beyond these stated goals, these tools can contribute in robust algorithms for computer vision by incorporating shape-based recognition of objects. In this paper we will assume that the shapes have already been extracted from the images and the data is available in form of contours. Of course, in many applications extraction of contours itself is a difficult problem but our focus here is on analyzing shapes once the contours are extracted.

2 Problem 1: Shape Clustering

An important need in shape studies is to classify and cluster previously observed shapes. In this section, we develop an algorithm for clustering of objects according to shapes of their boundaries.

Classical clustering algorithms on Euclidean spaces are well researched and generally fall into two main categories: partitional and hierarchical [5]. Assuming that the desired number k of clusters is known, partitional algorithms typically seek to minimize a cost function Q_k associated with a given partition of the data set into k clusters. The total variance of a clustering is a widely used cost function. Hierarchical algorithms, in turn, take a bottom-up approach. If the data set contains n points, the clustering process is initialized with n clusters, each consisting of a single point. Then, clusters are merged successively according to some criterion until the number of clusters is reduced to k. Commonly used metrics include the distance of the means of the clusters, the minimum distance between elements of clusters, and the average distance between elements of the clusters. If the number k of clusters is not known, the problem is harder. Advances in this direction were made in [4].

Clustering algorithms purely based on distances between data points readily generalize to other metric spaces. However, extensions of algorithms that involve finding means of clusters are only meaningful in metric spaces where means can be defined and computed. For the shape space considered here, the notion of

Karcher means for shapes is studied in [10]. However, in some cases the computational cost of computing means may prove to be high. Therefore, it is desirable to replace quantities involving the calculation of means by approximations that can be derived directly from distances between data points as described next.

2.1 Minimum-Variance Clustering

Consider the problem of clustering n shapes (in $\mathcal{S}$) into k clusters. To motivate our algorithm, we begin with a discussion of a classical clustering procedure for points in Euclidean spaces, which uses the minimization of the total variance of clusters as a clustering criterion. More precisely, consider a data set with n points $\{y_1, y_2, \ldots, y_n\}$ with each $y_i \in \mathbb{R}^d$. If a collection $C = \{C_i, 1 \leq i \leq k\}$ of subsets of $\mathbb{R}^d$ partitions the data into k clusters, the total variance of C is defined by $Q(C) = \sum_{i=1}^{k} \sum_{y \in C_i} \|y - \mu_i\|^2$, where μ_i is the mean of data points in C_i. The term $\sum_{y \in C_i} \|y - \mu_i\|^2$ can be interpreted as the total variance of the cluster C_i. The total variance is used instead of the (average) variance to avoid placing a bias on large clusters, but when the data is fairly uniformly scattered, the difference is not significant and either term can be used. The widely used *k-Means Clustering Algorithm* is based on a similar clustering criterion (see e.g. [5]). The *soft k-Means Algorithm* is a variant that uses ideas of simulated annealing to improve convergence [1,9].

These ideas can be extended to shape clustering using $d(\theta, \mu_i)^2$ instead of $\|y - \mu_i\|^2$, where $d(\cdot, \cdot)$ is the geodesic length and μ_i is the Karcher mean of a cluster C_i on the shape space. However, the computation of Karcher means of large shape clusters is a computationally demanding operation. Thus, we propose a variation that replaces $d(\theta, \mu_i)^2$ with the average distance-square $V_i(\theta)$ from θ to elements of C_i. If n_i is the size of C_i, then $V_i(\theta) = \frac{1}{n_i} \sum_{\theta' \in C_i} d(\theta, \theta')^2$. The cost Q associated with a partition C can be expressed as

$$Q(C) = \sum_{i=1}^{k} \frac{2}{n_i} \left(\sum_{\theta_a \in C_i} \sum_{b < a, \theta_b \in C_i} d(\theta_a, \theta_b)^2 \right). \tag{2}$$

If the average distance-square within the clusters is used, the scale factor in each term is modified to $\frac{2}{n_i(n_i-1)}$. In either case, we seek configurations that minimize Q, i.e., $C^* = \operatorname{argmin} Q(C)$. In this paper we have used the latter cost function.

2.2 Clustering Algorithm

We will minimize the clustering cost using a Markov chain Monte Carlo (MCMC) search process on the configuration space. The basic idea is to start with a configuration of k clusters and keep on reducing Q by re-arranging shapes amongst the clusters. The re-arrangement is performed in a stochastic fashion using two kinds of moves. These moves are performed with probability proportional to negative exponential of the Q value of the resulting configuration.

1. **Move a shape**: Here we select a shape randomly and re-assign it to another cluster. Let $Q_j^{(i)}$ be the clustering cost when a shape θ_j is re-assigned to the cluster C_i keeping all other clusters fixed. If θ_j is not a singleton, i.e. not the only element in its cluster, then the transfer of θ_j to cluster C_i is performed with the probability:

$$P_M(j,i;T) = \frac{\exp(-Q_j^{(i)}/T)}{\sum_{i=1}^{k} \exp(-Q_j^{(i)}/T)} \ , \quad i = 1, 2, \ldots, k \ .$$

   Here T plays the role of temperature as in simulated annealing. Note that moving θ_j to any other cluster is disallowed if it is a singleton in order to fix the number of clusters at k.
2. **Swap two shapes**: Here we select two shapes from two different clusters and swap them. Let $Q^{(1)}$ and $Q^{(2)}$ be the Q-values of the original configuration (before swapping) and the new configuration (after swapping), respectively. Then, swapping is performed with the probability:

$$P_S(T) = \frac{\exp(-Q^{(2)}/T)}{\sum_{i=1}^{2} \exp(-Q^{(i)}/T)} \ .$$

Additional types of moves can also be used to improve the search over the configuration space although their computational cost becomes a factor too. In view of the computational simplicity of moving a shape and swapping two shapes, we have restricted the algorithm to these two simple moves.

In order to seek global optimization, we have adopted a simulated annealing approach. That is, we start with a high value of T and reduce it slowly as the algorithm search for configurations with smaller dispersions. Additionally, the moves are performed according to a Metropolis-Hastings algorithm (see [8] for reference), i.e. candidates are proposed randomly and accepted according to certain probabilities (P_M and P_S above). Although simulated annealing and the random nature of the search help in getting out of local minima, the convergence to a global minimum is difficult to establish. As described in [8], the output of this algorithm is a Markov chain but is neither homogeneous nor convergent to a stationary chain. If the temperature T is decreased slowly, then the chain is guaranteed to converge to a global minimum. However, it is difficult to make an explicit choice of the required rate of decrease in T and instead we rely on empirical studies to justify this algorithm. First, we state the algorithm and then describe some experimental results.

Algorithm 1 *For n shapes and k clusters initialize by randomly distributing n shapes among k clusters. Set a high initial temperature T.*

1. *Compute pairwise geodesic distances between all n shapes. This requires $n(n-1)/2$ geodesic computations.*
2. *With equal probabilities pick one of two moves:*

a) **Move a shape***:*
   i. Pick a shape θ_j randomly. If it is not a singleton in its cluster then compute $Q_j^{(i)}$ for all $i = 1, 2, \ldots, k$.
   ii. Compute the probability $P_M(j, i; T)$ for all $i = 1, \ldots, k$ and re-assign θ_j to a cluster chosen according to the probability P_M.

b) **Swap two shapes***:*
   i. Select two clusters randomly, and select a shape from each of them.
   ii. Compute the probability $P_S(T)$ and swap the two shapes according to that probability.

3. Update temperature using $T = T/\beta$ and return to Step 2. We have used $\beta = 1.0001$ in our experiments.

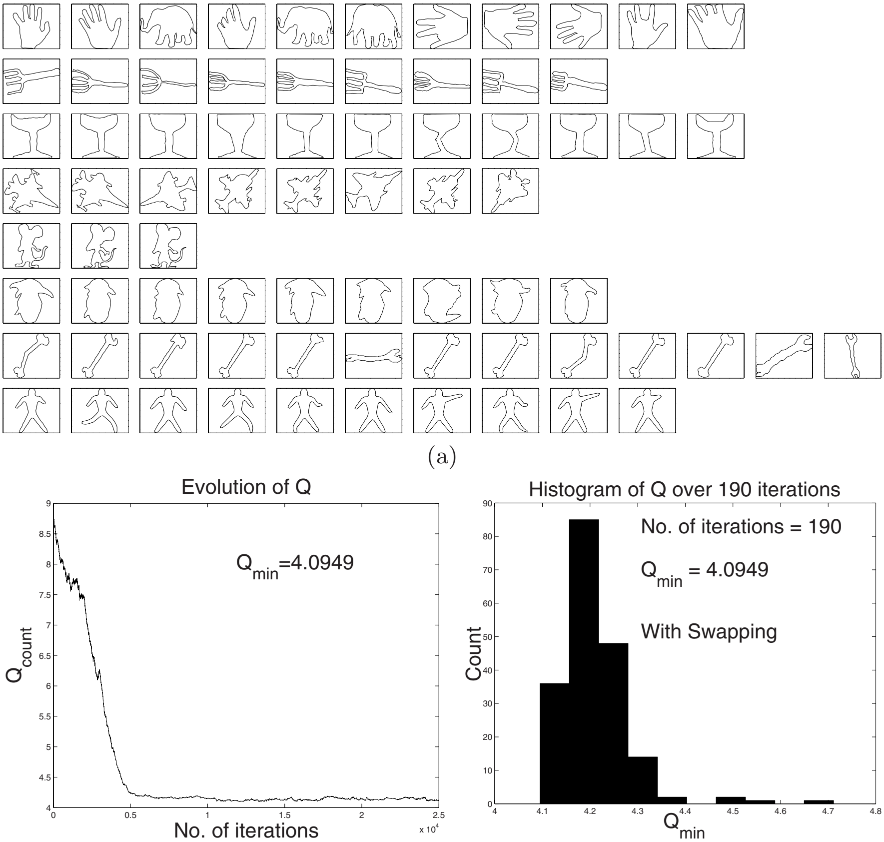
It is important to note that once the pairwise distances are computed, they are not computed again in the iterations. Secondly, unlike k-mean clustering mean shapes are not used here. These factors make Algorithm 1 efficient and effective in clustering diverse shapes.

Now we present some experimental results generated using Algorithm 1. We have applied Algorithm 1 to organize a collection of $n = 300$ shapes (not shown) from the Kimia shape database [12] into 26 clusters. Shown in Figure 1(a) are a few samples from the 26 clusters. In each run of Algorithm 1, we keep the configuration with minimum Q value. Figure 1(b) shows an evolution of the search process where the Q values are plotted against the iteration index. Figure 1(c) shows a histogram of the best Q values obtained in 190 such runs, each starting from a random initial configuration. It must be noted that 90% of these runs result in configurations that are quite close to the optimal. Once pairwise distances are computed, it takes approximately 10 seconds to perform 25,000 steps of Algorithm 1 in the matlab environment. The success of Algorithm 1 in clustering these diverse shapes is visible in these results as similar shapes have been clustered together.

2.3 Hierarchical Organization of Shapes

An important goal of this paper is to organize large databases of shapes in a fashion that allows for efficient searches. One way of accomplishing this is to organize shapes in a tree structure, such that shapes are refined regularly as we move down the tree. In other words, objects are organized (clustered) according to coarser differences (in their shapes) at top levels and finer differences at lower levels. This is accomplished in a bottom up construction as follows: start with all the shapes at the bottom level and cluster them according to Algorithm 1 for a pre-determined k. Then, compute a mean shape for each cluster and at the next level cluster these mean shapes according to Algorithm 1. Applying this idea repeatedly, one obtains a tree organization of shapes in which shapes change from coarse to fine as we move down the tree. Critical to this organization is the notion of the mean of shapes for which we utilize Karcher means.

We follow the procedure above to generate an example of a tree structure obtained for 300 shapes selected from the Kimia database. The Figure 2 shows



(b) (c)

Fig. 1. (a) Samples from 26 clusters of the kimia dataset. Each row is a cluster. (b) Sample evolution of Algorithm 1 for the configuration in (a). (c) Histogram of $Q(C^*)$ for 190 runs.

a hierarchical organization of 300 shapes all the way up to the top. At the bottom level, these 300 shapes are clustered in $k = 26$ clusters, with the clusters denoted by the indices of their element shapes. Computing the means of each these clusters, we obtain shapes to be clustered at the next level. Repeating the clustering for $k = 9$ clusters we obtain the next level and their mean shapes. In this example, we have chosen to organize shapes in five levels with a single shape at the top. The choice of parameters such as the number of levels, and the number of clusters at each level, depends on the required search speed and performance. It is interesting to study the variations in shapes as we follow a path from top to bottom in this tree. Two such paths from the right tree are shown in Figure 3. This multi-resolution representation of shapes has important implications. One is that very different shapes can be efficiently compared at a low resolution while only similar shapes require high-resolution comparison.

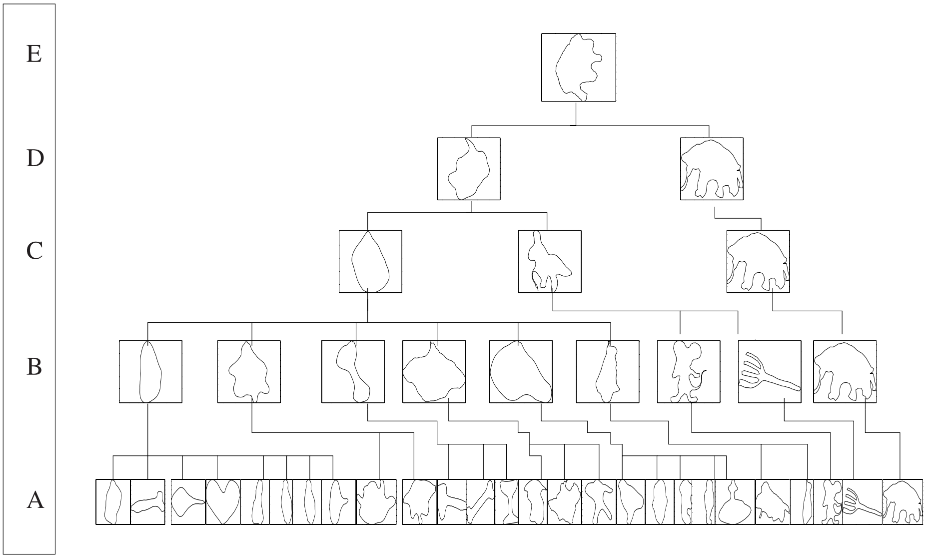


Fig. 2. Hierarchical Organization of 300 shapes from the Kimia database.

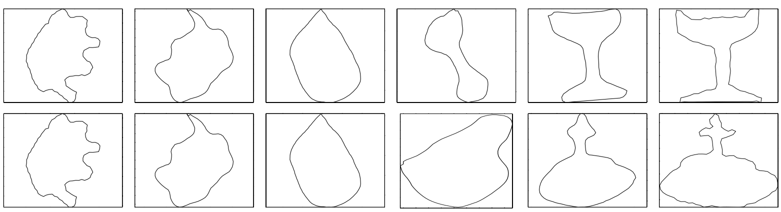


Fig. 3. Increasing resolution of shape as we go from top to bottom of tree in Figure 2.

3 Problem 2: Shape Testing

An important step in statistical analysis of shapes is to impose probability models on shape spaces. Once the shapes are clustered, we are interested in learning models to capture shape variations. Since the shape space $\mathcal{S}$ is curved and infinite-dimensional, we need some approximations to proceed further. We will impose probability models on $\mathcal{S}$ implicitly by imposing models on $T_\mu(\mathcal{S})$, where μ is the mean shape, and inheriting it on $\mathcal{S}$ using the geodesic flows. Secondly, we will approximate the element $f \in T_\mu(\mathcal{S})$ by its coefficients $\mathbf{x}$ truncated to obtain a finite-dimensional representation.

We still have to decide what form does the resulting probability distribution takes. One common approach is to assume a parametric form so that learning is now simply an estimation of the relevant parameters. As an example, a popular idea is to assume a Gaussian distribution on the underlying space. Let $\mathbf{x}$ be multivariate normal with mean $\mathbf{0}$ and variance $K \in \mathbb{R}^{m \times m}$. Estimation of μ and K from the observed shapes follows the usual procedures. Using μ and an observed shape θ_j, we find the tangent vector $g_j \in T_\mu(\mathcal{S})$ such that the geodesic from μ in the direction g_j passes through θ_j in unit time. This tangent vector is actually computed implicitly through $\mathbf{x}_j$. From the observed values of $\mathbf{x}_j$s, one can estimate the covariance using sample covariance. Depending on the number and the nature of the shape observations, the rank of estimated covariance is generally much smaller than its size. Extracting the dominant eigen vectors of the estimated covariance matrix, one can capture the dominant modes of variations. The density function associated with this family of shapes is given by:

$$h(\theta; \mu, K) \equiv \frac{1}{(2\pi)^{m/2} \det(K^\epsilon)^{1/2}} \exp(-\mathbf{x}^T (K^\epsilon)^{-1} \mathbf{x}/2) \ , \tag{3}$$

where $K^\epsilon = K + \epsilon I$, $\Psi(\mu, g, 1) = \theta$ and $g = \sum_{i=1}^m x_i e_i(s)$.

This framework of shape representations, and statistical models on shape spaces, has important applications in decision theory. One is to recognize an imaged object according to the shape of its boundary. Statistical analysis on shape spaces can be used to make a variety of decisions such as: Does this shape belong to a given family of shapes? Does these two families of shapes have similar means or variances? Given a test shape and two competing probability models, which one explains the test shape better?

We restrict to the case of binary hypothesis testing since for multiple hypotheses, one can find the best hypothesis using a sequence of binary hypothesis tests. Consider two shape families specified by their probability models: h_1 and h_2. For an observed shape $\theta \in \mathcal{S}$, we are interested in selecting one of two following hypotheses: $H_0 : \theta \sim h_1$ or $H_1 : \theta \sim h_2$. We will select a hypothesis according to the likelihood ratio test: $l(\theta) \equiv \log(\frac{h_1(\theta)}{h_2(\theta)}) \underset{H_0}{\overset{H_1}{\gtrless}} 0$. Substituting for the normal distributions (Eqn. 3) for $h_1 \equiv h(\theta; \mu_1, \Sigma_1)$ and $h_2 \equiv h(\theta; \mu_2, \Sigma_2)$, we can obtain sufficient statistics for this test. Let $\mathbf{x}_1$ be the vector of Fourier coefficients that encode the tangent direction from μ_1 to θ, and $\mathbf{x}_2$ be the same for direction from μ_2 to θ. In other words, if we let $g_1 = \sum_{i=1}^m x_{1,i} e_i$ and $g_2 = \sum_{i=1}^m x_{2,i} e_i$, then we have $\theta = \Psi(\mu_1, g_1, 1) = \Psi(\mu_2, g_2, 1)$. It follows that

$$l(\theta) = (\mathbf{x}_1^T (\Sigma_1^\epsilon)^{-1} \mathbf{x}_1 - \mathbf{x}_2 (\Sigma_2^\epsilon)^{-1} \mathbf{x}_2) - \frac{1}{2} (\log(\det(\Sigma_2^\epsilon)) - \log(\det(\Sigma_1^\epsilon))) \tag{4}$$

In case the two covariances are equal to Σ, the hypothesis test reduces to

$$l(\theta) = (\mathbf{x}_1^T (\Sigma^\epsilon)^{-1} \mathbf{x}_1 - \mathbf{x}_2 \Sigma^\epsilon \mathbf{x}_2) \underset{H_0}{\overset{H_1}{\gtrless}} 0 \ ,$$

and when Σ is identity, and $\epsilon = 0$, the log-likelihood ratio is given by $l(\theta) = \|\mathbf{x}_1\|^2 - \|\mathbf{x}_2\|^2$. The curved nature of the shape space $\mathcal{S}$ makes the analysis of this test difficult. For instance, one may be interested in probability of type one error but that calculation requires a probability model on $\mathbf{x}_2$ when H_0 is true.

As a first order approximation, one can write $\mathbf{x}_2 \sim N(\bar{\mathbf{x}}, \Sigma_1)$, where $\bar{\mathbf{x}}$ is the coefficient vector of tangent direction in $T_{\mu_2}(\mathcal{S})$ that corresponds to the geodesic from μ_2 to μ_1. However, the validity of this approximation remains to be tested under experimental conditions.

Shape Retrieval: We want to use the idea of hypothesis testing in retrieving shapes from a database that has been organized hierarchically. In view of its organization, a natural way is to start at the top, compare the query with the shapes at each level, and proceed down the branch that leads to the best match. At any level of the tree, there are some number, say p, of possible shapes, and our goal is to find the shape that matches the query θ best. This is performed using $(p-1)$ binary tests leading to the selection of the best hypothesis. Then, we proceed down the tree following that selected hypothesis and repeat the testing involving shapes at the next level. This continues till be reach the last level and have found the best overall match to the given query. For demonstration of retrieval, we hierarchically organize a collection of 100 shapes from the Surrey fish database [11] as shown in Figure 4.

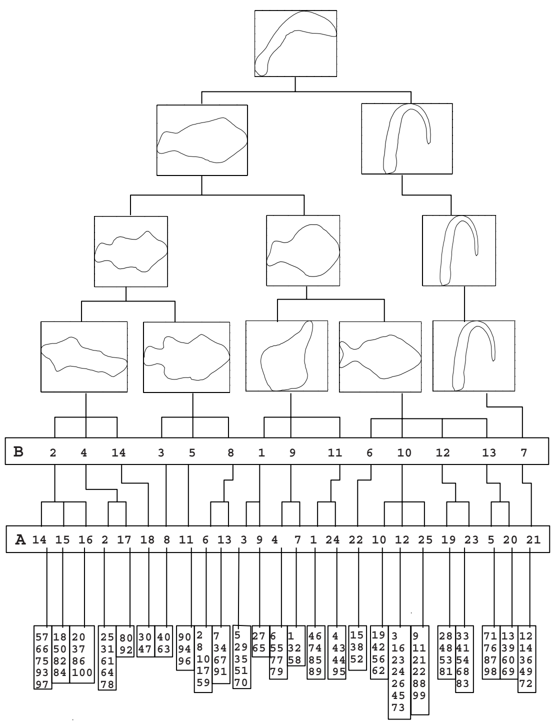


Fig. 4. Hierarchical Organization of 100 shapes from the Surrey Fish Database.

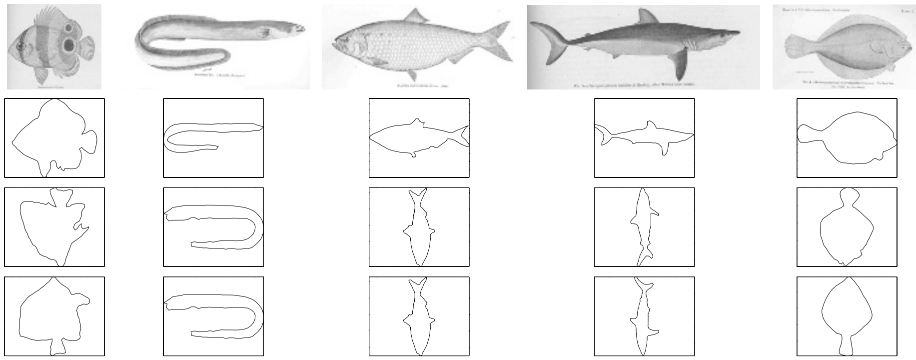


Fig. 5. Image retrieval. Top panels show the test images, the second row shows object boundaries, and the third row shows the closest shape in database obtained using a hierarchical search. Bottom row shows results of an exhaustive search.

Figure 5 shows some examples of the retrieval procedure. Shown in the top panels are images of test objects whose shapes provide queries for shape retrieval. These shapes are not included in the training database. First, we hand extract the contours of test objects (second row), and then use the shapes of these contours to search the database given in Figure 4. Third row shows the best matches from the database obtained using the tree search. For comparison, bottom row shows results from an exhaustive search over the full database. The resulting retrieval times are as follows. For exhaustive search, one needs 100 comparisons while for the organization shown in Figure 4, the required number of comparisons lies between seven and twenty. For example, in above figure, when the query is eel, it takes seven comparisons to reach the answer while for shad it takes seventeen comparisons. In our current implementation, each comparison takes 0.02 seconds approximately, providing a worst case retrieval time of 0.4 seconds for this database.

4 Conclusion

Using Riemannian structure of a space of planar shapes and geodesic length on it as a shape metric, we have presented statistical approaches to clustering and testing shapes. Clustering is performed by minimizing average variance within the clusters, and is used in hierarchically organizing large databases of objects according to their shapes. Using Gaussian distributions model on tangent spaces, we have derived a technique for shape testing, and have applied it to retrieval of shapes from a large database.

Acknowledgements. This material is based upon work supported by NSF DMS-0101429 and NMA 201-01-2010, and by NSF and the Intelligence Technology Innovation Center through the joint "Approaches to Combat Terrorism" Program Solicitation NSF 03-569 (DMS-0345242).

References

1. D. E. Brown and C. L. Huntley. A practical application of simulated annealing to clustering. Technical Report IPC-TR-91-003, Institute for Parallel Computing, University of Virginia, Charlotesville, VA, 1991.
2. D. Cremers and S. Soatto. A pseudo distance for shape priors in level set segmentation. In *2nd IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*, France, 2003.
3. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* John Wiley & Son, 1998.
4. M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 2002.
5. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, 1988.
6. P. Maurel and G. Sapiro. Dynamic shapes average. In *2nd IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*, France, 2003.
7. M. I. Miller and L. Younes. Group actions, homeomorphisms, and matching: A general framework. *International Journal of Computer Vision*, 41(1/2):61–84, 2002.
8. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer Text in Stat., 1999.
9. K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, November 1998.
10. E. Klassen, A. Srivastava, W. Mio and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machiner Intelligence*, to appear in March 2004.
11. F. Mokhtarian, S. Abbasi and J. Kittler. Efficient and robust shape retrieval by shape content through curvature scale space. *Proceedings of First International Conference on Image Database and MultiSearch, 1996*
12. Sharvit, D. and Chan, J. and Tek, H. and Kimia, B.B. Symmetry-based indexing of image databases. *Content-Based Access of Image and Video Libraries, 1998*