**Statistical Shape Analysis: Clustering, Learning, and Testing**

Anuj Srivastava † Shantanu Joshi ‡ Washington Mio § Xiwen Liu ¶

**Abstract**

Using a differential-geometric treatment of planar shapes, we present tools for: (i) hierarchical clustering of imaged objects according to the shapes of their boundaries, (ii) learning of probability models from clustered shapes, and (iii) testing of newly observed shapes under competing probability models. Clustering at any level of hierarchy is performed using a minimum dispersion criterion and a Markov search process. Statistical means of clusters provide shapes to be clustered at the next higher level, thus building a hierarchy of shapes. Using finite-dimensional approximations of spaces tangent to the shape space at sample means, we (implicitly) impose probability models on the shape space; results are illustrated via random sampling and classification (hypothesis testing). Together, hierarchical clustering and hypothesis testing provide an efficient framework for shape retrieval. Examples are presented using shapes and images from ETH, Surrey, and AMCOM databases.

**Keywords**: shape analysis, shape statistics, shape learning, shape testing, shape retrieval, shape clustering

1 **Introduction**

An important goal in image analysis is to classify and recognize objects of interest in given images. Imaged objects can be characterized in several ways, using their colors, textures, shapes, movements, and locations. The past decade has seen large efforts in modeling and analysis of pixel statistics in images to attain these goals albeit with limited success. An emerging opinion 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, morphometry, database retrieval, surveillance, biometrics, military target recognition and general computer vision.

In order to perform statistical shape analysis, one needs a shape space and probability models that can be used for future inferences. Since past observations can lead to future shape models, we are interested in learning probability models from a given set of observations on a shape space. Towards that goal, we will study the following topics.

**Problem 1: Hierarchical Clustering**: We will consider the problem of clustering planar objects, or images of objects, according to the shapes of their boundaries. To improve efficiency, we will investigate a hierarchy in which the mean shapes are recursively clustered. This can significantly
improve database searches in systems with shape-based queries. For instance, testing a shape against prototypes of different clusters, to select a cluster, and then testing against shapes only in that cluster, is much efficient than the exhaustive testing.

**Problem 2: Learning Shape Models:** Given a cluster of similar shapes, we want to “learn” a probability model that captures observed variability. Examples of this problem using landmark-based shape analysis are presented in [4]. For the representation introduced in [15], the problem of model building is more difficult for two reasons: the shape space is (a quotient space of) a nonlinear manifold, and it is infinite dimensional. First is handled by using tangent spaces at the mean shapes, as suggested in [3], and second is handled via finite-dimensional approximations of tangent functions.

**Problem 3: Testing Shape Hypotheses:** Probability models on shape spaces can be used to perform statistical shape analysis. For example, one can study the question: Given an observed shape and two competing probability models, which model does this shape belong to? We are interested in analyzing binary hypothesis tests on shape spaces. More general \( m \)-ary testing is approached similarly. Hypothesis tests for landmark-based shape analysis have previously been studied in [3]. Hypothesis testing, together with hierarchical organization forms an efficient tool for shape retrieval.

Furthermore, these tools can also contribute in developing robust algorithms for computer vision, by incorporating shape information in image models for recognition of objects. We will assume that either the shapes have already been extracted from training images (e.g. Surrey database) or can be extracted using a standard edge detector (e.g. ETH image database). Of course, in many applications, extraction of contours itself is a difficult problem but our focus here is on analyzing shapes. However, we remark that this framework for analyzing shapes is also useful in extracting shapes from images using informative priors [15, 19].

1.1 *Past Research in Shape Analysis*

Shapes have been an important topic of research over the past decade. A significant part has been restricted to “landmark-based” analysis, where shapes are represented by a coarse, discrete sampling of the object contours [3, 23]. One establishes equivalences 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. In addition, shape interpolation with geodesics in this framework lacks a physical interpretation, as exemplified later. Despite these limitations, landmark-based representations have been successful in many applications, especially in physician-assisted medical image analysis, where landmarks are readily available, and have led to advanced tools for statistical analysis of shapes [3, 14, 10]. A similar approach, called *active shape models*, uses principal component analysis (PCA) of coordinates of landmarks to model shape variability [1]. Despite its simplicity and efficiency, its scope is rather limited because it ignores the nonlinear geometry of shape space. Grenander’s formulation [6] considers shapes as points on infinite-dimensional manifolds, and the variations between the shapes are modeled by the action of Lie groups (diffeomorphisms) on these manifolds [7]. A major limitation here is the high computational cost. Although level set methods have recently been applied to shape analysis, an intrinsic formulation of shape analysis using level sets, that is invariant to all shape-preserving transformations, is yet to be presented. A large number of studies on shape metrics have been published with a more limited goal of fast shape retrieval from large databases. One example is the use of scale space representations of shapes, as described in [20]. In summary, the majority of previous work on analyzing shapes of planar curves involves either
(i) use a discrete collection of points (landmarks or active shape models), or (ii) use of mappings (diffeomorphism) or functions (level sets) on $\mathbb{R}^2$, seldom have they been studied as curves!

### 1.2 A Framework for Planar Shape Analysis

Klassen et al. [15] consider the shapes of **continuous**, closed curves in $\mathbb{R}^2$, without any need for landmarks, diffeomorphisms, or level sets to model shape variations. Elements of the shape space here are actually shapes of closed planar curves. The basic idea is to identify a space of closed curves, remove shape-preserving transformations, impose a Riemannian structure on it, and utilize its geometry to solve optimization and inference problems. Using the Riemannian structure, they have developed algorithms for computing geodesic paths on these shape spaces. We advance this idea here by developing several tools that can prove to be important in shape analysis. A pictorial outline of the full framework is presented in the left panel of Figure 1. As depicted, shapes are extracted from observed images either manually or automatically, and then organized using clustering algorithms. Probability models are learnt from this clustered data for use in future statistical inferences involving object retrieval, identification, detection, and tracking. Next we reproduce and summarize main ideas from [15] and refer to the paper for details.

#### 1. Geometric Representation of Shapes

Consider the boundaries or silhouettes of the imaged objects as closed, planar curves in $\mathbb{R}^2$ parameterized by the arc length. Coordinate function $\alpha(s)$ relates to the direction function $\theta(s)$ according to $\dot{\alpha}(s) = e^{j \theta(s)}$, $j = \sqrt{-1}$. For the unit circle, a direction function is $\theta_0(s) = s$. For any other closed curve of rotation index 1, the direction function takes the form $\theta = \theta_0 + h$, where $h \in \mathbb{L}^2$, and $\mathbb{L}^2$ denotes the space of all real-valued functions with period $2\pi$ and square integrable on $[0, 2\pi]$. To make shapes invariant to rotation, restrict to $\theta \in \theta_0 + \mathbb{L}^2$ such that, $\frac{1}{2\pi} \int_0^{2\pi} \theta(s)ds = \pi$. Also, for a closed curve, $\theta$ must satisfy the closure condition: $\frac{1}{2\pi} \int_0^{2\pi} \exp(j \theta(s))ds = 0$. Summarizing, one restricts to the set $C = \{\theta \in \theta_0 + \mathbb{L}^2 | \frac{1}{2\pi} \int_0^{2\pi} \theta(s)ds = \pi, \int_0^{2\pi} e^{j\theta(s)}ds = 0\}$. To remove the re-parametrization group (relating to different placements of $s = 0$ on the same curve), define the quotient space $S \equiv C/S^1$ as the space of continuous, planar shapes.

![Figure 1: Left: An overview of a statistical modeling approach to object detection, identification, and tracking, with a focus on shape analysis. Right: A cartoon diagram of a shooting method to find geodesics in shape space.](image)
1.3 Comparison with Previous Approaches

For an observed contour, denoted by a set of non-uniformly sampled points in $\mathbb{R}^2$, one can generate a representative element $\theta \in \mathcal{S}$ as follows. For each neighboring pair of points, compute the chord angle $\theta_i$ and the Euclidean distance $s_i$ between them. Then, fit a smooth $\theta$ function, e.g., using splines, to the graph formed by $\{(s_i, \theta_i)\}$. Finally, resample $\theta$ uniformly (using arc-length parametrization) and project onto $\mathcal{S}$.

2. Geodesic Paths Between Shapes: An important tool in a Riemannian analysis of shapes is to construct geodesic paths between arbitrary shapes. Klassen et al. [15] approximate geodesics on $\mathcal{S}$ by successively drawing infinitesimal line segments in $\mathbb{L}^2$ and projecting them onto $\mathcal{S}$, as depicted in the right panel of Figure 1. For any two shapes $\theta_1, \theta_2 \in \mathcal{S}$, they use a shooting method to construct the geodesic between them. The basic idea is search for a tangent direction $g$ at the first shape $\theta_1$, such that a geodesic in that direction reaches the second shape $\theta_2$, called target shape, in unit time. This search is performed by minimizing a “miss function”, defined as a $\mathbb{L}^2$ distance between the shape reached and $\theta_2$, using a gradient process. The geodesic is with respect to the $\mathbb{L}^2$ metric $\langle g_1, g_2 \rangle = \int_0^{2\pi} g_1(s)g_2(s)ds$ on the tangent space of $\mathcal{S}$. This choice implies that a geodesic between two shapes is the path that uses minimum energy to bend one shape into the other.

We will use the notation $\Psi(\theta, g, t)$ for a geodesic path starting from $\theta \in \mathcal{S}$, in the direction $g \in T_\theta(\mathcal{S})$, as a function of time $t$. Here $T_\theta(\mathcal{S})$ is the space of tangents to $\mathcal{S}$ at the point $\theta$. In practice, the function $g$ is represented using an orthogonal expansion according to $g(s) = \sum_{i=1}^{\infty} x_i e_i(s)$, where $\{e_i, i = 1, \ldots, \}$ forms an orthonormal basis of $T_\theta(\mathcal{S})$ and the search for $g$ is performed via a search for corresponding $x = \{x_1, x_2, \ldots, \}$. An additional simplification is to let $\{e_i\}$ be a basis for $\mathbb{L}^2$, represent $g$ using this basis, and subtract its projection onto the normal space at $\theta$ to obtain a tangent vector $g$. On a desktop PC with Pentium IV 2.6GHz processor, it takes on average 0.065 seconds to compute a geodesic between any two shapes. In this experiment, the direction functions were sampled at 100 points each and $g$ is approximated using 100 Fourier terms.

Empirical evidence motivates us to consider shapes of two curves that are mirror reflections of each other as equivalent. For a $\theta \in \mathcal{S}$, its reflection is given by: $\theta_R(s) = 2\pi - \theta(2\pi - s)$. To find a geodesic between two shapes $\theta_1, \theta_2 \in \mathcal{S}$, we construct two geodesics: one between $\theta_1$ and $\theta_2$, and the other between $\theta_1$ and $\theta_2\!$R. The shorter of these two paths denotes a geodesic in the quotient space. With a slight abuse of notation, from here on, we will call the resulting quotient space as $\mathcal{S}$.

3. Mean Shape in $\mathcal{S}$: [15] suggests the use of Karcher mean to define mean shapes. For $\theta_1, \ldots, \theta_n$ in $\mathcal{S}$, and $\text{d}(\theta_i, \theta_j)$ the geodesic length between $\theta_i$ and $\theta_j$, the Karcher mean is defined as the element $\mu \in \mathcal{S}$ that minimizes the quantity $\sum_{i=1}^n \text{d}(\theta, \theta_i)^2$. A gradient-based, iterative algorithm for computing the Karcher mean is presented in [16, 13] and is particularized to $\mathcal{S}$ in [15]. Statistical properties, such as bias and efficiency, of this estimator remain to be investigated.

1.3 Comparison with Previous Approaches

Some highlights of the above-mentioned approach are that it: (i) analyzes full curves and not a coarse collections of landmarks, i.e. there is no need to determine landmarks a-priori, (ii) completely removes shape-preserving transformations from the representation and explicitly specifies a shape space, (iii) utilizes nonlinear geometry of the shape space (of curves) to define and compute statistics, (iv) seeks full statistical frameworks and develops priors for future Bayesian infer-
ences, and (v) can be applied to real-time applications. Existing approaches seldom incorporate all of these features: Kendall’s representations are equipped with (ii)-(v) but using landmark-based representations; active shape models are fast and efficient but do not satisfy (ii) and (iii). Diffeomorphism-based models satisfy (i)-(iv) and can additionally take into account image intensities, but are currently slow for real-time applications. Curvature scale space methods are not equipped with either (iii) or (iv). It must be noted that for certain specific applications, existing approaches may suffice or maybe more efficient than the proposed approach. For example, for retrieving shapes from a database, a simple metric using either Fourier descriptors or a PCA of coordinate vectors, or scale-space shape representations, may prove sufficient. However, the proposed approach is intended as a comprehensive framework to handle a wide variety of applications.

An interesting point relating to non self-intersecting closed curves deserves a separate mention. Although our interest lies in shapes of simple (i.e. non self-intersecting), closed curves, no explicit constraint has been imposed on the elements of $S$ to avoid self-intersection. Experimental results show that the representation given in Section 1.2 is “stable” with respect to this property. That is, a geodesic path between two simple, closed curves seldom passes through a self-intersecting closed curve. As an illustration, left column of Figure 2 shows a few examples of geodesic paths in $S$ between some intricate shapes. For comparison, the right column shows corresponding geodesic paths using Kendall’s Procrustean analysis. All the paths in the right column have shapes that self intersect, while none in the left column does so. We attribute this behavior to the direction-function representation chosen in $S$, and the resulting bending energy criterion for finding geodesic paths.

Additionally, representation of shapes by their direction functions allows for using tools from functional analysis in compression, denoising, or multi-resolution studies of shapes. These tools are not available in point-based representations of shapes.

<table>
<thead>
<tr>
<th>Our Approach</th>
<th>Procrustes Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Geodesics" /></td>
<td><img src="image2" alt="Procrustean Geodesics" /></td>
</tr>
</tbody>
</table>

Figure 2: Left shows geodesics in $S$ while right shows Procrustean geodesics using landmarks.

2 Analysis of Noisy Shapes

Often in practical situations, observed shapes are corrupted by noise or clutter present in images. For instance, shapes extracted from low-resolution images may exhibit pixellated boundaries. The noise mainly leads to rough boundaries and a smoothing procedure may be helpful. Similar to
considerations in image denoising and restoration, a balance between smoothing and preservation of true edges is required, and the required amount of smoothing is difficult to predict in practice. In this section, we briefly discuss three ways to avoid certain noise effects:

1. **Robust Extraction**: Use a smoothness prior on shapes during Bayesian extraction of shapes. For a shape denoted by \( \theta \), one can use elastic energy, define as \( \int \dot{\theta}(s)^2 ds \), as a roughness prior for Bayesian extraction of shapes from images.

2. **Shape Denoising**: Since shapes are represented as parameterized direction functions, one can use tools from functional analysis for denoising. In case the extracted shapes have rough parts, one can replace them with their smooth approximations. A classical approach is to use wavelet denoising or Fourier smoothing. For example, shown in Figure 3 are examples of denoising the direction function of a noisy shape using (i) wavelets (Daubechies, at level 4 in matlab), and (ii) Fourier expansion using first 20 harmonics.

3. **Tangent Denoising**: Another possibility is to denoise the tangent function, rather than the original direction functions (denoting shapes), as follows. A geodesic path between two given shapes is found using a shooting method. By restricting the search for tangent direction \( g \) to a set of smooth functions, one can avoid certain noise effects. Shown in Figure 4 is an example of this idea, where the target shape is corrupted by noise. In top left panel, each curve shows the gradient-based evolution of “miss function” versus the iteration index. Different curves correspond to different number of terms used in Fourier expansion for the tangent direction; increasing the number of Fourier terms results in a better approximation of optimal \( g \), and a smaller value of “miss” function. Top middle panel shows resulting geodesic lengths plotted against the Fourier resolution. A tangent function with smaller Fourier terms can help avoid the problem of over-fitting that often occurs with the noisy data. Bottom figures show the resulting geodesics for two different tangent resolutions: the left path results from restricting the tangent direction to 10 terms, and the right path shows over fitting to noise in the target shape using 60 terms. Top right shows the direction functions reached in unit time for these two cases.

3 Problem 1: **Shape Clustering**

An important need in statistical shape studies is to classify and cluster observed shapes. In this section, we develop an algorithm for clustering 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 [12]. 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. Hierarchical algorithms, in turn, take a bottom-
Figure 4: Tangent denoising. Top: Left shows evolutions of “miss function” versus iteration index, for increasing number of Fourier components of the tangent vector, and middle shows resulting geodesic lengths plotted against Fourier resolution. Top right shows the direction functions reached for 10 and 60 terms. Bottom: left shows a geodesic path using 10 Fourier terms while right shows a path using 60 terms.

up approach where points are clustered successively until the number of clusters is reduced to \( k \). Commonly used metrics include distances between means of clusters, minimum distances between elements of clusters, and average distances between elements of the clusters.

### 3.1 Minimum-Variance Clustering

Consider the problem of clustering \( n \) shapes (in \( S \)) into \( k \) clusters. A general approach is to form clusters in such a way that they minimize total “within-cluster” variance. Let a configuration \( C \) consists of clusters denoted by \( C_1, C_2, \ldots, C_k \), and let \( \mu_i \)'s be the mean shapes in \( C_i \)'s and \( n_i \)'s be the sizes of \( C_i \)'s. There are several cost functions used for clustering, e.g., the sum of traces of covariances within clusters. However, the computation of means \( \mu_i \)'s of large shape clusters, and therefore their variances, is computationally expensive, especially when they are updated at every iteration. As a solution, one often uses a variation, called pairwise clustering [9], where the variance of a cluster is replaced by a scaled sum of distances (squared) between its elements:

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

We seek configurations that minimize \( Q \), i.e., \( C^* = \arg\min Q(C) \).

An important question in any clustering problem is: How to choose \( k \)? In case of shape retrieval applications, the answer is easier. Here, \( k \) determines a balance between the retrieval speed and performance, and a wide range of \( k \) can be tried. In the worst case, one can set \( k = n/2 \) at every level of hierarchy (described later) and still obtain \( O(\log(n)) \) retrieval speeds. (This assumes that the shapes are uniformly distributed in the clusters.) However, the choice of \( k \) is much more important in the case of learning. Probability models estimated from the clustered shapes are sensitive to the clustering performance. To obtain a possible \( k \) automatically, one option is to study the variation of \( Q(C^*) \) for different values of \( k \) and select a \( k \) that provides the largest decrease in \( Q(C^*) \) from its value at \( k - 1 \). Another possibility is to use human supervision in selecting \( k \). Ideas presented in [5] can also used for unsupervised clustering.
3.2 Clustering Algorithm

We will take a stochastic simulated annealing approach to solve for $C^*$. Several authors have explored the use of annealing in clustering problems, including soft clustering [22], and deterministic clustering [9]. An interesting idea presented in [9] is to solve an approximate problem, termed mean-field approximation, where $Q$ is replaced by a function in which the roles of elements $\theta_i$s are decoupled. The advantage is the resulting efficiency although it comes at the cost of error in approximation.

We will minimize the clustering cost using a Markov chain search process on the configuration space. The basic idea is to start with a configuration of $k$ clusters and to reduce $Q$ 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 the negative exponential of the $Q$-value of the resulting configuration. The two types of moves are:

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

   $$P_M(j, i; T) = \frac{\exp(-Q^{(i)}_j/T)}{\sum_{i=1}^k \exp(-Q^{(i)}_j/T)}$$

   Here $T$ plays a role similar to temperature in simulated annealing. If $\theta_j$ is a singleton, then moving it is not allowed in order to fix the number of clusters at $k$.

2. **Swap two shapes**: Here we select two shapes randomly 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 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 our algorithm to these two 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 searches for configurations with smaller dispersions. Additionally, the moves are performed according to an acceptance-rejection procedure that is a variant of more conventional simulated annealing (see for example, Algorithm A.20, pg. 200 [21] for a conventional procedure). Here, the candidates are proposed randomly and accepted according to certain probabilities ($P_M$ and $P_S$ are defined above). Although simulated annealing and the random nature of the search help in avoiding local minima, the convergence to a global minimum is difficult to establish. As described in [21], the output of this algorithm is a Markov chain that 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 explicit 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 the two moves:
• Move a shape: Pick a shape \( \theta_j \) randomly. If it is not a singleton in its cluster, then compute \( Q_i^{(j)} \) for all \( i = 1, 2, \ldots, k \). Compute the probability \( P_{M}(j, i; T) \) for all \( i = 1, \ldots, k \) and re-assign \( \theta_j \) to a cluster chosen according to the probability \( P_{M} \).

• Swap two shapes: Select two clusters randomly, and select a shape from each. Compute the probability \( P_{S}(T) \) and swap the two shapes according to that probability.

3. Update the temperature using \( T = T/\beta \) and return to Step 2. We have used \( \beta = 1.0001 \).

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, the mean shapes are never calculated in this clustering. These factors make Algorithm 1 efficient and effective in clustering diverse shapes.

Now we present some experimental results generated using Algorithm 1. We start with a small example to illustrate the basic idea. We have clustered \( n = 25 \) shapes taken from the Surrey fish database, shown in Figure 6, to \( k = 9 \) clusters. In each run of Algorithm 1, we keep the configuration with minimum \( Q \) value. To demonstrate effectiveness of the swap move we also compare results obtained with and without that move. In Figure 5(a), we show an example evolution of the search process, without the swap move, where the \( Q \) values are plotted against the iteration index. In Figure 5(b), we show a histogram of the best \( Q \) values obtained in each of 200 such runs, each starting from a random initial configuration. In Figure 5(c) and (d), we present corresponding results with the swap move as stated in Algorithm 1. It must be noted that 90% of these runs result in configurations that are quite close to the optimal. Computational cost of this clustering algorithm is small; once pairwise distances between shapes are computed, it takes approximately five seconds to perform 250,000 steps of Algorithm 1 in matlab, for \( n = 25 \) and \( k = 9 \). Figure 6 displays the configuration with smallest \( Q \) value; each column shows a cluster with two, three, or five shapes in it. The success of Algorithm 1 in clustering these diverse shapes is visible in these results, similar shapes have been clustered together. As a comparison, the dendrogram clustering results are shown in the lower panel Figure 5. It is easy to see that a dendrogram for \( k = 9 \) clusters will not give a satisfactory configuration.

3.3 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 display increasing resolution, 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 second level cluster these means 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 mean shapes for which we utilize Karcher means mentioned earlier.

We present some experimental results from an application this idea using the ETH object database. This database contains eight different classes of objects – apples, tomatoes, cars, pears, cows, cups, dogs, and horses – each class contains 400 images of 3D objects imaged from different viewing angles. We have used an automated procedure, using standard edge-detection techniques, to extract 2D contours of imaged objects, resulting in 3200 observations of planar shapes. Shown in Figure 7 are some examples of these shapes. For these 3200 shapes, shown in Figure 10 is a hierarchical organization into seven levels. At the very bottom, the 3200 shapes are clustered
Figure 5: Top row: (a) A sample evolution of $Q$ under Algorithm 1 without using the swap move. (b) A histogram of the minimum $Q$ values obtained in 200 runs of Algorithm 1 (without swap move) each starting from a random initial condition. (c) and (d): sample evolution and a histogram of 200 runs with the swap move. Lower row: A clustering result using the dendrogram function in matlab.

into $k = 25$ clusters. It currently takes approximately 100 hours on one computer to compute all pairwise geodesics for these 3200 shapes; one can use multiple desktop computers, or parallel computing, to accomplish this task more efficiently. Shown in Figure 8 left panel is an evolution of Algorithm 1 for this data, and in Figure 9 are some examples shapes in some of these clusters. Figure 8 (right panel) shows a histogram of optimal $Q$-values obtained in 100 runs of Algorithm 1. The clustering in general agrees well with the known classification of these shapes. For example, apples and tomatoes are clustered together while cups and animals are clustered separately. It is interesting to note that, with a few exceptions, shapes corresponding to different postures of animals are also clustered separately. For example, shapes of cows sitting (cluster 25), animal shapes from side views (cluster 6), and animal shapes from frontal views (cluster 23), are all clustered separately. It takes approximately 40 minutes in matlab to run 50K steps of Algorithm 1 with $n = 3200$ and $k = 25$. At the next level of hierarchy statistical means of elements in each cluster are computed and are clustered with $n = 25$ and $k = 7$. These 25 means are shown at level $F$ in Figure 10. This process is repeated till we reach the top of the tree.

It is interesting to study the variations in shapes as we follow a path from top to bottom in this tree. Three such paths from the tree are displayed in Figure 11, showing an increase in shape features as we follow the path (drawn left to right here). This multi-resolution representation of shapes has important implications. One is that very different shapes can be effectively compared at a low resolution and high speed, while only similar shapes require high-resolution comparisons.
Figure 6: A clustering of 25 fish shapes into nine clusters with $Q = 0.933$. Each column shows the shapes that are clustered together.

Figure 7: Some examples of shapes contained in ETH object database.

4 Shape Learning

Another important problem in statistical analysis of shapes is to “learn” probability models from observed shapes. Once the shapes are clustered, we assume that elements in the same cluster are samples from the same probability model, and try to learn this model. These models can then be used for future Bayesian discoveries of shapes or for classification of new shapes. To learn a probability model amounts to estimating a probability density function on the shape space, a task that is rather difficult to perform precisely. The two main difficulties are: nonlinearity and infinite-dimensionality of $\mathcal{S}$, and they are handled here as follows.

1. **Nonlinearity:** $\mathcal{S}$ is a nonlinear manifold, so we impose a probability density on a tangent space instead. For a mean shape $\mu \in \mathcal{S}$, $T_{\mu}(\mathcal{S}) \subset L^2$, is a **vector space** and more conventional statistics applies.

2. **Dimensionality:** We approximate a tangent function $g$ by a finite-dimensional vector, e.g. a vector of Fourier coefficients, and thus characterize a probability distribution on $T_{\mu}(\mathcal{S})$ as that on a finite-dimensional vector space.
Let a tangent element \( g \in T_\mu(S) \) be represented by its Fourier approximation: 
\[
g(s) = \sum_{i=1}^{m} x_i e_i(s),
\]
for a large positive integer \( m \). Using the identification \( g \equiv x = \{x_i\} \in \mathbb{R}^m \), one can define a probability distribution on elements of \( T_\mu(S) \) via one on \( \mathbb{R}^m \).

We still have to decide what form does the resulting probability distribution takes. Following are three possibilities:

1. **Multivariate normal model for principal coefficients**: A common approach is to assume a multivariate normal model on tangent vector \( x \). Assume that variations of \( x \) are mostly restricted to an \( m_1 \)-dimensional subspace of \( \mathbb{R}^m \), called the \textit{principal subspace}, with an orthogonal basis \( \{v_1, \ldots, v_{m_1}\} \) for some \( m_1 << m \). We denote the linear projection of \( x \) to the principal subspace by \( \tilde{x} \) and let \( \tilde{x}^\perp \in \mathbb{R}^{m-m_1} \) be such that \( x = \tilde{x} \oplus \tilde{x}^\perp \), \( \oplus \) denotes the direct sum. Now, model \( \tilde{x} \sim \mathcal{N}(0, K) \), for a \( K \in \mathbb{R}^{m_1 \times m_1} \), and model \( \tilde{x}^\perp \sim \mathcal{N}(0, \epsilon I_{m-m_1}) \), for a small \( \epsilon > 0 \). Note that this procedure uses a \textit{local PCA} in the tangent space \( T_\mu(S) \) and should be distinguished from a \textit{global PCA} of points in a shape space, as is done in active shape models [1]. \( T_\mu(S) \) is a linear space, and its principal elements are mapped to \( S \) via a nonlinear map \( \psi \), as opposed to a direct linear (PCA) approximation of a shape space.

Estimation of \( \mu \) and \( K \) from observed shapes is straightforward. Computation of a mean shape \( \mu \) is described in [15]. Using \( \mu \) and an observed shape \( \theta_j \), find the tangent vector \( g_j \in T_\mu(S) \) such that the geodesic from \( \mu \) in the direction \( g_j \) reaches \( \theta_j \) in unit time. This tangent vector is actually computed via a finite-dimensional representation and results in the corresponding vector of coefficients \( x_j \). From the observed values of \( x_j \in \mathbb{R}^m \), one can estimate the principal subspace and the covariance matrix. Extracting the dominant eigenvectors 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:

\[
h(\theta; \mu, K) \propto \exp\left(-\tilde{x}^T K^{-1} \tilde{x} / 2 - \|\tilde{x}^\perp\|^2 / (2\epsilon)\right) / (\det(K)e^{(m-m_1)}) , \quad \text{for} \quad \Psi(\mu; \sum_{i=1}^{m} \tilde{x}_i e_i, 1) = \theta . \quad (2)
\]

2. **Nonparametric models on principal coefficients**: Instead of assuming a parametric model for \( \tilde{x} \), here we estimate the probability density function of each component using kernel density estimation (with a Gaussian kernel). Shown in Figure 12 are some examples of estimated density functions. The first three plots show density functions of the four dominant components for shapes in cluster 18. These histograms seem typical of the larger database in the sense that both uni-modal and multi-modal densities occur frequently. In principle, one should form joint density estimates from the observed coefficients, as they maybe dependent. However, in this paper we ignore their

---

Figure 8: Left: An evolution of \( Q \) versus iteration index under Algorithm 1 for \( n = 3200 \) and \( k = 25 \). Right: Histogram of minimum \( Q \)-values obtained in 100 different runs.
Figure 9: Examples of shapes in clusters 6, 16, 18, 21, 23, 24 and 25.
Figure 10: Hierarchical Clustering of 3200 shapes from the ETH-80 database

Figure 11: Paths from top to bottom in the tree show increasing shape resolutions.
We will select a hypothesis using the likelihood ratio: \( H_0 : \theta \sim h_1 \) or \( H_1 : \theta \sim h_2 \). We will select a hypothesis using the likelihood ratio: \( l(\theta) = \log(h_2(\theta)/h_1(\theta)) \). Substituting for normal distributions (Eqn. 2) \( h_1 \equiv h(\theta; \mu_1, K_1) \) and \( h_2 \equiv h(\theta; \mu_2, K_2) \), we can obtain sufficient statistics for this test. Let \( x_i \) be the vector of Fourier coefficients that encode the tangent direction from \( \mu_i \) to \( \theta \) such that \( x_i = \tilde{x}_i \oplus \check{x}_i \), \( i = 1, 2 \). It follows that \( l(\theta) = -\tilde{x}_1^T K_1^{-1} \tilde{x}_1 + \tilde{x}_2^T K_2^{-1} \tilde{x}_2 - (\|\check{x}_1\|^2 - \|\check{x}_2\|^2)/(2\epsilon) - \log(\det(K_1)) + \log(\det(K_2)). \) In the special case, when \( m = m_1 \) and \( K_1 = K_2 = I_m \)

dependence and coarsely approximate their joint probability density by a product of the estimated marginals.

3. **Nonparametric models on independent coefficients**: In this case we use independent components, instead of the principal components, to estimate marginal density functions, and can form a joint density by multiplying the marginals. However, the main limitation of independent components is the lack a natural technique to choose \( m_1 \). For computing ICA, we have used the FastICA algorithm presented in [11]. Shown in the last three plots of Figure 12 are some examples of these estimated density functions for independent coefficients, for elements in cluster 18.

We have studied these shape models using ideas from random sampling and shape testing. Sampling results are presented here while testing results are presented in the next section.

**Sampling**: Shown in Figures 13 is an example of random sampling from estimated models. Eighteen observed shapes \( \theta_1, \ldots , \theta_{18} \) of M60 tanks are shown in Figure 13(a) and are analyzed for learning a probability model. The Karcher mean shape and estimated eigenvalues of the sample covariance matrix are shown in Figure 13(b). Generating random samples from \( \tilde{x} \sim N(0, \hat{K}) \) and computing \( \Psi(\mu, \sum_{i=1}^{m_1} \tilde{x}_i v_i, 1) \), we have synthesized new shapes from the model learned shown in Figure 13(c). Four principal modes of shape variations for this dataset are shown in Figure 14. As a comparison, shown in Figure 13(d) are samples from a nonparametric model (on principal coefficients), where the marginal densities of principal coefficients have been estimated from the observed data using a Gaussian kernel.

## 5 Hypothesis Testing and Shape Retrieval

This framework for 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? Do the given two families of shapes have similar means and/or variances? Given a test shape and two competing probability models, which one explains the test shape better?

**Shape Testing**: We start with binary hypothesis testing under the proposed shape models. 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 using the likelihood ratio: \( l(\theta) = \log(h_2(\theta)/h_1(\theta)) \). Substituting for normal distributions (Eqn. 2) \( h_1 \equiv h(\theta; \mu_1, K_1) \) and \( h_2 \equiv h(\theta; \mu_2, K_2) \), we can obtain sufficient statistics for this test. Let \( x_i \) be the vector of Fourier coefficients that encode the tangent direction from \( \mu_i \) to \( \theta \) such that \( x_i = \tilde{x}_i \oplus \check{x}_i \), \( i = 1, 2 \). It follows that \( l(\theta) = -\tilde{x}_1^T K_1^{-1} \tilde{x}_1 + \tilde{x}_2^T K_2^{-1} \tilde{x}_2 - (\|\check{x}_1\|^2 - \|\check{x}_2\|^2)/(2\epsilon) - \log(\det(K_1)) + \log(\det(K_2)). \) In the special case, when \( m = m_1 \) and \( K_1 = K_2 = I_m \)
Figure 13: (a) 18 observed M60 tank shapes, (b) top shows the mean shape and bottom plots the principal eigenvalues of tangent covariance, (c) shows random samples from multivariate normal model, and (d) shows samples from a nonparametric model for principal coefficients.

Figure 14: Principal modes of variations in the 18 observed shapes shown in Figure 13.

(identity), the log-likelihood ratio is given by $l(\theta) = \|x_1\|^2 - \|x_2\|^2$. Multiple hypothesis testing can be accomplished similarly using the most likely hypothesis. The curved nature of the shape space $S$ makes an analytical study of this test difficult. For instance, one may be interested in probability of type one error but that calculation requires a probability model on $\tilde{x}_2$ when $H_0$ is true.

We now present results from an experiment on hypothesis testing using the ETH database. In this experiment, we selected twelve out of 25 clusters shown at the bottom level of Figure 10. Shown in Figure 15 left side are mean shapes of these twelve clusters. In each class (or cluster), we used 50 shapes as training data for learning the shape models. A disjoint set of 2500 shapes (total), drawn randomly from these clusters, was used to test the classification performance; a correct classification implies that the test shape was assigned to its own cluster. Shown in Figure 15 right panel is a plot of classification performance versus $m_1$, the number of components used. This plot shows classification performances using: (i) multivariate normal model on tangent vectors, and (ii) nonparametric models on principal coefficients. We remark that, in this particular instance, nearest-mean classifier also performs well since that metric matches well with the cost function (Eqn. 1) used in clustering.

Hierarchical 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 this structure, 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 is a number, say $k$, of possible shapes,
and our goal is to find the shape that matches the query $\theta$ best. This can be performed using $k - 1$ binary tests leading to the selection of the best hypothesis. In the current implementation, we have assumed a simplification that the covariance matrices for all hypotheses at all levels are identity, and only the mean shapes are needed to organize the database. For identity covariances, the task of finding the best match at any level reduces to finding the nearest mean shape at that level. Let $\mu_i$ be the given shapes at a level, and let $x_i$ be the Fourier vector that encode tangent direction from $\theta$ to $\mu_i$. Then, the nearest shape is indexed by $\hat{i} = \arg\min_i \|x_i\|$. Proceed down the tree following the nearest shape $\mu_{\hat{i}}$ at each level. This continues till we reach the last level and have found the best overall match to the given query.

We have implemented this idea using test images from the ETH database. For each test image, we first extract the contour, compute its shape representation as $\theta \in S$, and follow the tree, shown in Figure 10, for retrieving similar shapes from the database. Figure 16 presents some pictorial examples from this experiment. Shown in the left panels are the original images and in the second left panels their automatically extracted contours. Third column shows six nearest shapes retrieved in response to the query. Finally, the last panel states the time taken for the hierarchical search. A summary of retrieval times is as follows:

<table>
<thead>
<tr>
<th>Type of search</th>
<th>time(sec)</th>
<th>Type of search</th>
<th>time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>exhaustive</td>
<td>126.64</td>
<td>worst case (hierarchical)</td>
<td>70.57</td>
</tr>
<tr>
<td>best case (hierarchical)</td>
<td>0.92</td>
<td>average (hierarchical)</td>
<td>24.84</td>
</tr>
</tbody>
</table>

The time for exhaustive search is computed by averaging the search time for all 3200 shapes, while the search times for hierarchical technique are for approximately 50 query images used in this experiment. These results show a significant improvement in the retrieval times while maintaining a good performance.

In this experiment, retrieval performance is defined with respect to the original labels, e.g. apple, cup, cow, pear, etc. Shown in Figure 17 are plots of retrieval performances, measured using two different quantities. The first quantity is the precision rate, defined as the ratio of number of relevant shapes retrieved, i.e. shapes from the correct class, to the total number of shapes retrieved. Ideally, this quantity should be one, or quite close to one. The second quantity, called the recall rate, is the ratio of number of relevant shapes retrieved to the total number of shapes in that class in the database. Left panel in Figure 17 shows average variation of precision rate plotted against the number of shapes retrieved, for four different classes – apple, car, cup, and pear. As these
curves indicate, the retrieval performance of apple falls quickly while that for the other classes remains high. The reason for a low retrieval performance of apple shapes is their close resemblance in shape to tomatoes. The middle panel shows plots of recall rate plotted against the number of shapes retrieved, and the last panel plots precision rate against the recall rate, for the same four classes. These results for retrieving images using shapes of objects are very encouraging. Of course, a more general retrieval system using additional features, such as colors, should perform better.

It must be noted that this search will potentially produce incorrect results if the clustering process puts very different shapes in the same cluster. In this case, the mean shape of such a cluster will not be a good indicator of the cluster elements, and the search procedure can fail. One can guard against this situation by keeping a high number of clusters at each level, thus ensuring that different shapes are indeed grouped in different clusters.

6 Summary & Extensions

Building on a differential geometric representation of shapes, and geodesic lengths as shape metrics, we have presented tools that enable a statistical analysis of shapes. We have presented methods and algorithms for: shape clustering, learning shape models, and shape testing. We have presented a clustering algorithm, followed an evaluation of cluster means to perform hierarchical clustering. Using these tangent space probability models, we have defined a technique for performing hypothesis testing in the shape space, and have applied it to the problem of shape retrieval. Future research can be directed along these lines:

1. **Statistical Properties of Mean Estimator on $\mathcal{S}$**: Bias, consistency, efficiency, etc of the Karcher mean estimator should be investigated either analytically or empirically.
2. **Probability Models on Tangent Spaces**: It is desirable to have parametric models that

---

**Table: Retrieved shapes from hierarchy**

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Test Shape</th>
<th>Retrieved shapes from hierarchy</th>
<th>time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Car" /></td>
<td><img src="image" alt="Car" /></td>
<td><img src="image" alt="Car" /> <img src="image" alt="Car" /> <img src="image" alt="Car" /> <img src="image" alt="Car" /> <img src="image" alt="Car" /></td>
<td>27.39</td>
</tr>
<tr>
<td><img src="image" alt="Cat" /></td>
<td><img src="image" alt="Cat" /></td>
<td><img src="image" alt="Cat" /> <img src="image" alt="Cat" /> <img src="image" alt="Cat" /> <img src="image" alt="Cat" /> <img src="image" alt="Cat" /></td>
<td>17.23</td>
</tr>
<tr>
<td><img src="image" alt="Mug" /></td>
<td><img src="image" alt="Mug" /></td>
<td><img src="image" alt="Mug" /> <img src="image" alt="Mug" /> <img src="image" alt="Mug" /> <img src="image" alt="Mug" /> <img src="image" alt="Mug" /></td>
<td>15.93</td>
</tr>
<tr>
<td><img src="image" alt="Dog" /></td>
<td><img src="image" alt="Dog" /></td>
<td><img src="image" alt="Dog" /> <img src="image" alt="Dog" /> <img src="image" alt="Dog" /> <img src="image" alt="Dog" /> <img src="image" alt="Dog" /></td>
<td>14.06</td>
</tr>
<tr>
<td><img src="image" alt="Pear" /></td>
<td><img src="image" alt="Pear" /></td>
<td><img src="image" alt="Pear" /> <img src="image" alt="Pear" /> <img src="image" alt="Pear" /> <img src="image" alt="Pear" /> <img src="image" alt="Pear" /></td>
<td>43.26</td>
</tr>
</tbody>
</table>

Figure 16: Some examples of shape retrieval using hierarchical organization.
capture the non-Gaussian behavior of observed tangents. Further investigations are needed to
determine if certain heavy-tailed models, such as Bessel K forms [8] or generalized Laplacian [17],
or some other families, may better explain the observed data.

3. Elastic Shapes: One limitation of the proposed approach is that arc-length parametrization
results in shape comparisons solely on the basis of bending energies, without allowing for stretching
or compression. In some cases matching via stretching of shapes is more natural [2]. An extension
that incorporates stretch elasticity by allowing reparameterizations of curves by arbitrary diffeomor-
phisms is presented in [18]. A curve α is represented by a pair (φ, θ) such that \( \dot{\alpha}(s) = e^{\phi(s)}e^{j\theta(s)} \). Appropriate constraints on (φ, θ) define a pre-shape manifold \( \mathcal{C} \), and the shape space is given by \( \mathcal{C}/\mathcal{D} \), where \( \mathcal{D} \) is the group of diffeomorphisms from [0,1] to itself. The action of a diffeomorphism
γ on a shape representation is given by: \( (\phi, \theta) = (\phi \circ \gamma + \log(\dot{\gamma}), \theta \circ \gamma) \). Computational details are
presented in [18].

Acknowledgement

This research was supported the grants NSF (FRG) DMS-0101429, NMA 201-01-2010, NSF (ACT)
DMS-0345242, and ARO W911NF-04-01-0268. We would like to thank the three reviewers for their
helpful comments. We are also grateful to the producers of ETH, Surrey, and AMCOM databases.

References


