

Statistical Shape Analysis: Clustering, Learning, and Testing

Anuj Srivastava, *Member, IEEE*, Shantanu H. Joshi, Washington Mio, and Xiuwen Liu, *Member, IEEE*

Abstract—Using a differential-geometric treatment of planar shapes, we present tools for: 1) hierarchical clustering of imaged objects according to the shapes of their boundaries, 2) learning of probability models for clusters of shapes, and 3) testing of newly observed shapes under competing probability models. Clustering at any level of hierarchy is performed using a minimum variance type criterion and a Markov 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, and 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.

Index Terms—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. We are interested in learning probability models from a given set of observations on a shape space. Toward 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 query against prototypes of different clusters*

to select a cluster, and then testing against shapes only in that cluster, is much more efficient than testing exhaustively.

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. The first is handled by using tangent spaces at the mean shapes, as suggested in [3], and the 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 clustering 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, the extraction of contours itself is a difficult problem but we do not focus on that problem here. 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

- A. Srivastava is with the Department of Statistics, Florida State University, Tallahassee, FL 32306. E-mail: anuj@stat.fsu.edu.
- S.H. Joshi is with the Department of Electrical Engineering, Florida State University, Tallahassee, FL 32306. E-mail: joshi@eng.fsu.edu.
- W. Mio is with the Department of Mathematics, Florida State University, Tallahassee, FL 32306. E-mail: mio@math.fsu.edu.
- X. Liu is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306. E-mail: liux@cs.fsu.edu.

Manuscript received 12 Jan. 2004; revised 21 June 2004; accepted 28 Sept. 2004; published online 10 Feb. 2005.

Recommended for acceptance by Y. Amit.

For information on obtaining reprints of this article, please send e-mail to: tpani@computer.org, and reference IEEECS Log Number TPAMI-0033-0104.

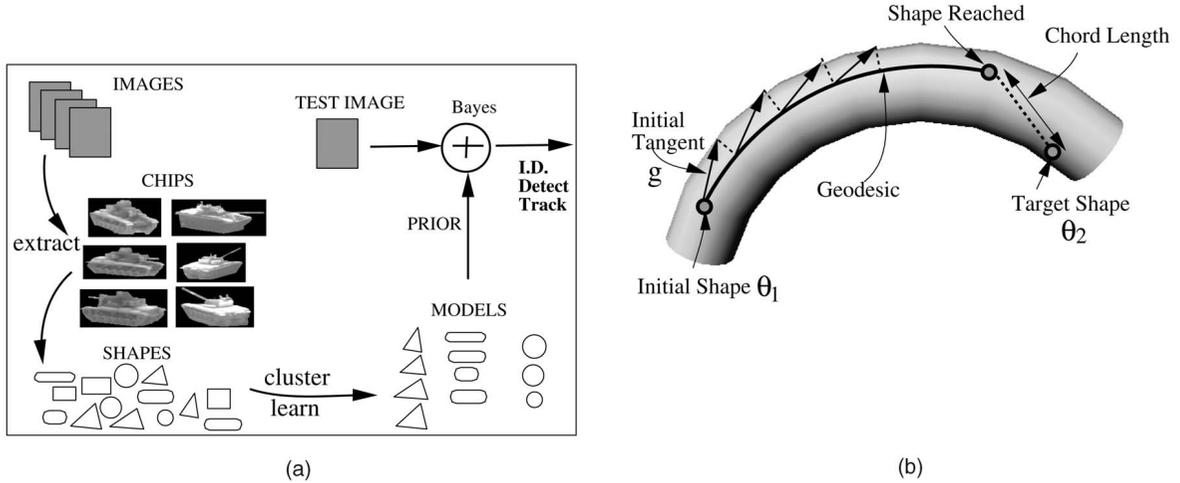


Fig. 1. (a) An overview of a statistical modeling approach to object detection, identification, and tracking, with a focus on shape analysis. (b) A cartoon diagram of a shooting method to find geodesics in shape space.

establishes equivalences with respect to shape preserving transformations, i.e., rigid rotation and translation, and nonrigid 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 1) use of a discrete collection of points (landmarks or active shape models) or 2) 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 Fig. 1a. As depicted, shapes are extracted from observed images either manually or automatically, and then organized using clustering algorithms. Probability models are learned 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 that 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 2π 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, θ must satisfy the *closure condition*: $\int_0^{2\pi} \exp(j\theta(s)) ds = 0$. Summarizing, one restricts to the set $\mathcal{C} = \{\theta \in \theta_0 + \mathbb{L}^2 \mid \frac{1}{2\pi} \int_0^{2\pi} \theta(s) ds = \pi, \int_0^{2\pi} e^{j\theta(s)} ds = 0\}$. To remove the reparametrization group (relating to different placements of $s = 0$ on the same curve), define the quotient space $\mathcal{S} \equiv \mathcal{C}/S^1$ as the space of continuous, planar shapes.

For an observed contour, denoted by a set of nonuniformly 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 θ_i and the Euclidean distance s_i between them. Then, fit a smooth θ function, e.g., using splines, to the graph formed by $\{(\sum_{j=1}^i s_j, \theta_i)\}$. Finally, resample θ 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 Fig. 1b. 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 θ_1 , such that a geodesic in that direction reaches the second shape θ_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 θ_2 , using a gradient process. This path is a geodesic 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 θ . 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 $\{e_i, i = 1, \dots\}$ forms an orthonormal basis of $T_\theta(\mathcal{S})$ and the search for g is performed via a search for corresponding $\mathbf{x} = \{x_1, x_2, \dots\}$. 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 θ 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 θ_1 and θ_2 and the other between θ_1 and θ_{2R} . 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} :** Klassen et al. [15] suggest the use of *Karcher mean* to define mean shapes. For $\theta_1, \dots, \theta_n$ in \mathcal{S} , and $d(\theta_i, \theta_j)$ the geodesic length between θ_i and θ_j , the Karcher mean is defined as the element $\mu \in \mathcal{S}$ that minimizes the quantity $\sum_{i=1}^n d(\theta, \theta_i)^2$. 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 abovementioned approach are that it:

1. analyzes full curves and not a coarse collections of landmarks, i.e., there is no need to determine landmarks a priori,
2. completely removes shape-preserving transformations from the representation and explicitly specifies a shape space,

3. utilizes nonlinear geometry of the shape space (of curves) to define and compute statistics,
4. seeks full statistical frameworks and develops priors for future Bayesian inferences, and
5. can be applied to real-time applications.

Existing approaches seldom incorporate all of these features: Kendall’s representations are equipped with 2-5 but using landmark-based representations; active shape models are fast and efficient but do not satisfy 2 and 3. Diffeomorphism-based models satisfy 1-4 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 3 or 4. 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 nonself-intersecting closed curves deserves a separate mention. Although our interest lies in shapes of simple (i.e., nonself-intersecting), closed curves, no explicit constraint has been imposed on the elements of \mathcal{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, Fig. 2a shows a few examples of geodesic paths in \mathcal{S} between some intricate shapes. For comparison, Fig. 2b shows corresponding geodesic paths using Kendall’s Procrustean analysis. All the paths in Fig. 2b have shapes that self intersect, while none in Fig. 2a does so. We attribute this behavior to the direction-function representation chosen in \mathcal{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 multiresolution studies of shapes. These tools are not available in point-based representations of shapes.

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 pixelated 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 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 θ , one can use elastic energy, define as $\int \theta(s)^2 ds$, as a roughness prior for Bayesian extraction of shapes from images.

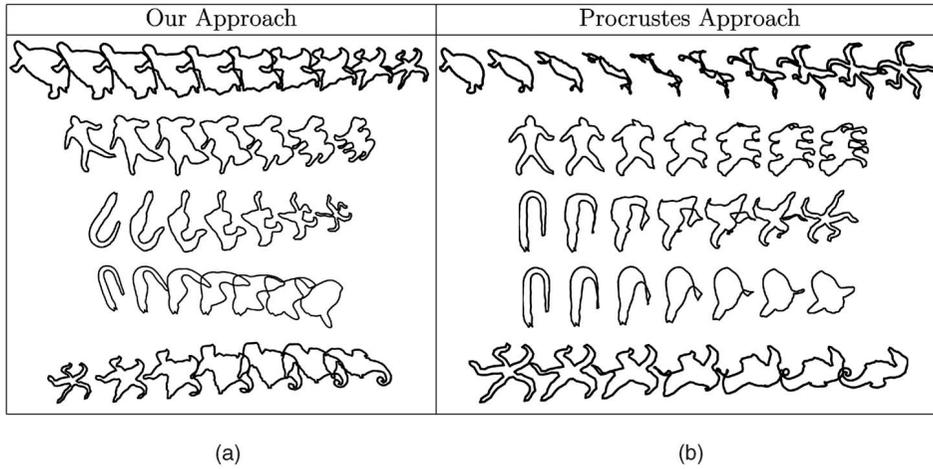


Fig. 2. (a) Shows geodesics in \mathcal{S} , while (b) shows Procrustean geodesics using landmarks.

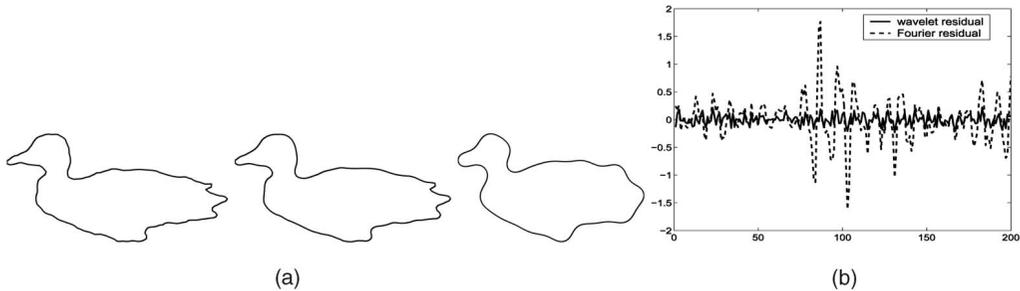


Fig. 3. (a) Noisy shape, (b) wavelet denoised shape, and (c) Fourier denoised shape. (d) Difference (residual) between noisy and denoised direction functions.

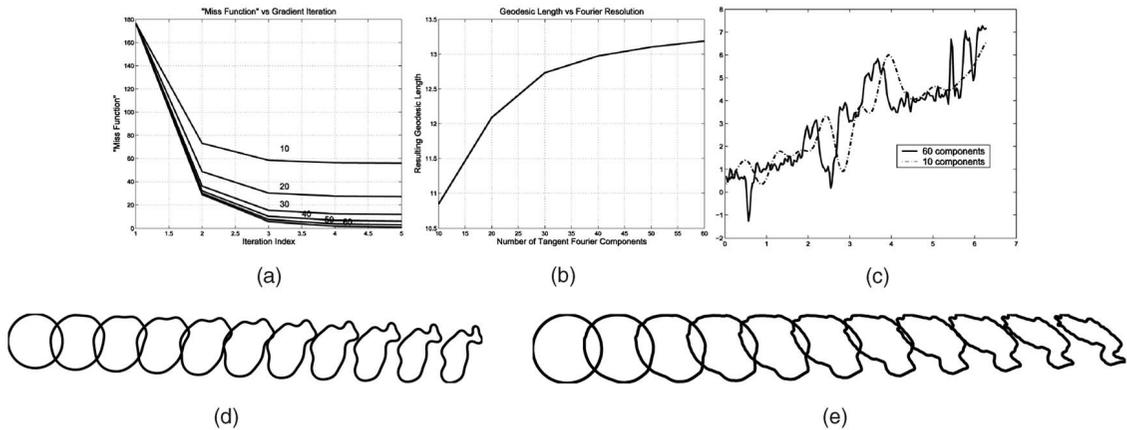


Fig. 4. Tangent denoising. (a) Shows evolutions of “miss function” versus iteration index for an increasing number of Fourier components of the tangent vector and (b) shows resulting geodesic lengths plotted against Fourier resolution. (c) Shows the direction functions reached for 10 and 60 terms. (d) Shows a geodesic path using 10 Fourier terms, while (e) shows a path using 60 terms.

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 Fig. 3 are examples of denoising the direction function of a noisy shape using 1) wavelets (Daubechies, at level 4 in Matlab) and 2) 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 Fig. 4 is an example of this idea, where the target shape is corrupted by noise. In Fig. 4a, 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. Fig. 4b shows resulting geodesic lengths plotted against the Fourier resolution. A tangent function with smaller Fourier terms can help avoid the problem of *overfitting* that often occurs with the noisy data. Figs. 4d and 4e show the resulting geodesics for two different tangent resolutions: The left path results from restricting the tangent direction to the first 10 terms, and the right path shows over fitting to noise in the target shape using 60 terms. Fig. 4c 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-up approach where clustered are merged 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 \mathcal{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, \dots, C_k , and let μ_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 μ_i s of large shape clusters and, therefore, their variances, is computationally expensive, especially when they are to be 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:

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

We seek configurations that minimize Q , i.e., $C^* = \operatorname{argmin} Q(C)$.

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 be 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 θ_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 rearranging shapes amongst the clusters. The rearrangement is performed in a stochastic fashion using two kinds of moves. These moves are performed with probabilities proportional to the negative exponential of the Q -values of the resulting configurations. The two types of moves are:

1. **Move a shape.** Here, we select a shape randomly and reassign it to another cluster. Let $Q_j^{(i)}$ be the clustering cost when a shape θ_j is reassigned 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 probability:

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

$i = 1, 2, \dots, k$. Here, T plays a role similar to temperature in simulated annealing. If θ_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 variances. 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, p. 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

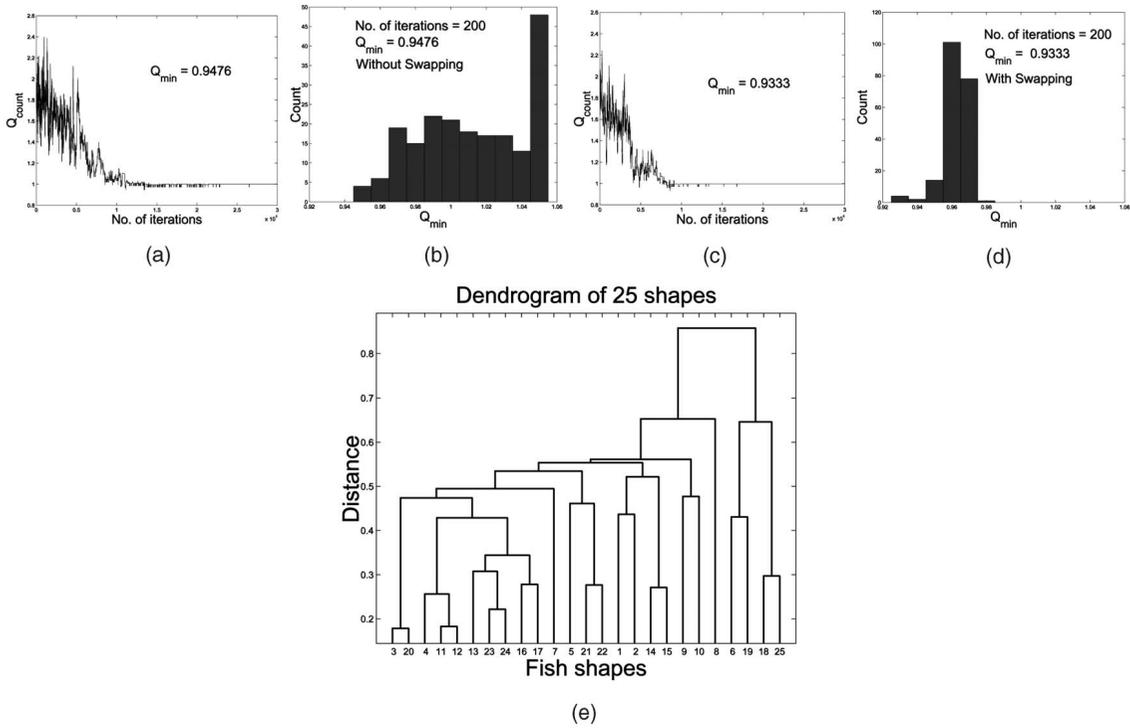


Fig. 5. (a) A sample evolution of Q under Algorithm 1 without using the swap move. (b) A histogram of the minimum Q values obtained in 200 runs of Algorithm 1 (without swap move) each starting from a random initial condition. (c) and (d) Sample evolution and a histogram of 200 runs with the swap move. (e) A clustering result using the dendrogram function in Matlab.

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 θ_j randomly. If it is not a singleton in its cluster, then compute $Q_j^{(i)}$ for all $i = 1, 2, \dots, k$. Compute the probability $P_M(j, i; T)$ for all $i = 1, \dots, k$ and reassign θ_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. Second, 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 Fig. 5, in 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 Fig. 5a, we show an example evolution of the search process, without the swap move, where the Q values are plotted against the iteration index. In Fig. 5b, we show a histogram of the best Q values obtained in each of 200 such runs, each starting from a random initial configuration. In Figs. 5c and 5d, we present corresponding results with the swap move as stated in Algorithm 1. It must be noted that 90 percent 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 $k = 9$. Fig. 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 Fig. 5e. 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

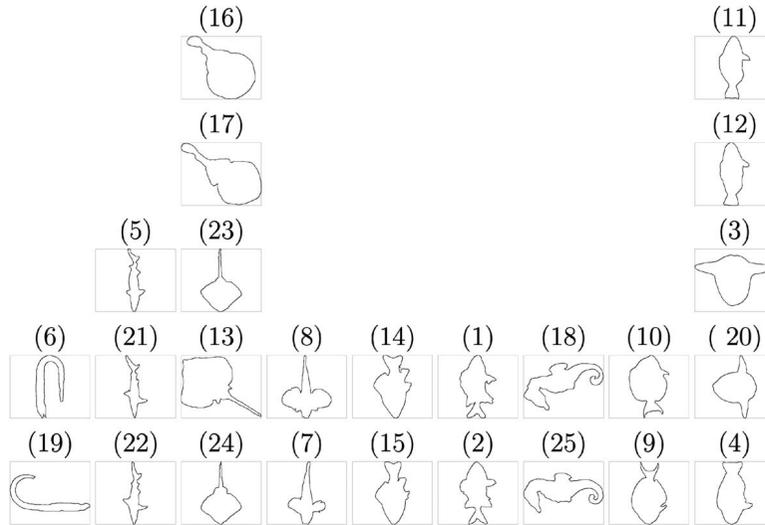


Fig. 6. A clustering of 25 fish shapes into nine clusters with $Q = 0.933$. Each column shows the shapes that are clustered together.

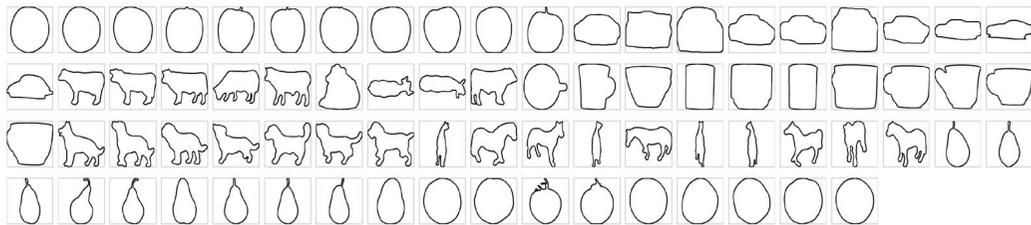


Fig. 7. Some examples of shapes contained in ETH object database.

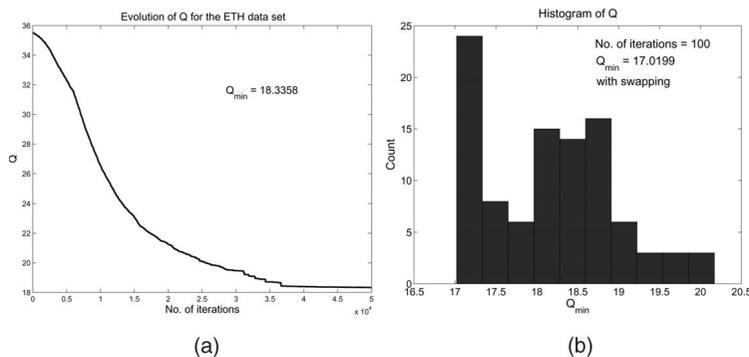


Fig. 8. (a) An evolution of Q versus iteration index under Algorithm 1 for $n = 3,200$ and $k = 25$. (b) Histogram of minimum Q -values obtained in 100 different runs.

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 predetermined 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 of mean shapes for which we utilize Karcher means mentioned earlier.

We present some experimental results 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 3,200 observations of planar shapes. Shown in Fig. 7 are some examples of these shapes. For these 3,200 shapes, shown in Fig. 10 is a hierarchical organization into seven levels. At the very bottom, the 3,200 shapes are clustered into $k = 25$ clusters. It currently takes approximately 100 hours on one computer to compute all pairwise geodesics for these 3,200 shapes; one can use multiple desktop computers, or parallel computing, to accomplish this task more efficiently. Shown in Fig. 8a is an evolution of Algorithm 1 for this data, and in Fig. 9 are some examples shapes in some of these clusters. Fig. 8b shows a histogram of optimal Q -values

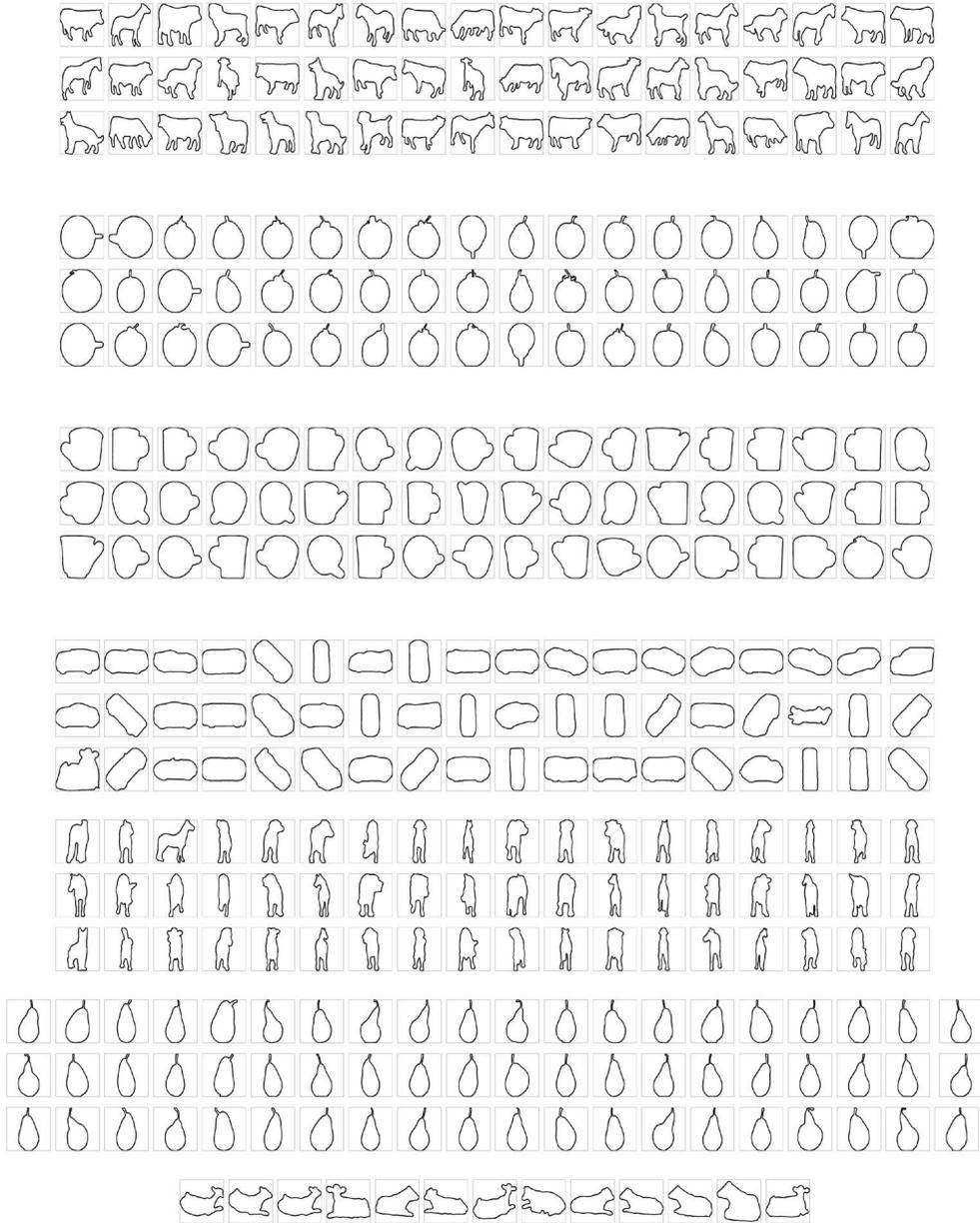


Fig. 9. Examples of shapes in clusters 6, 16, 18, 21, 23, 24, and 25.

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 $n = 3,200$ 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 Fig. 10. This process is repeated until 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 are displayed in Fig. 11, showing an increase in shape features as we follow the path (drawn left to right here). This multiresolution representation of shapes has important implications. One is that very different shapes can be effectively compared at a low resolution and at high speed, while only similar shapes require high-resolution comparisons.

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

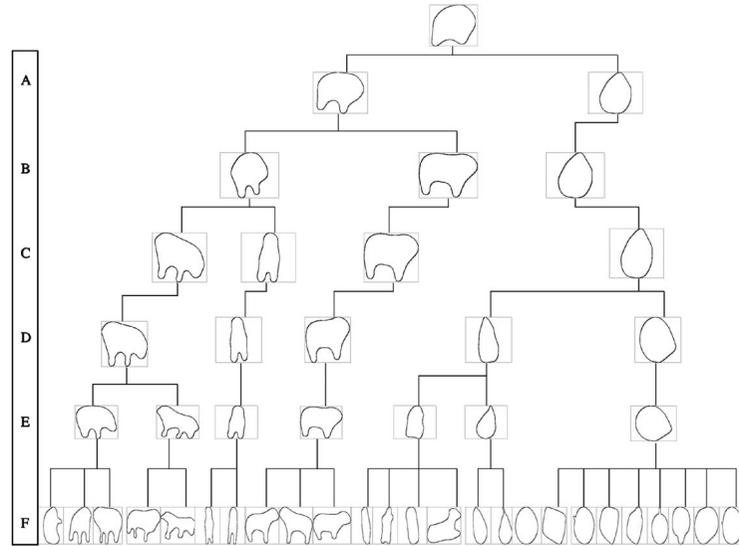


Fig. 10. Hierarchical Clustering of 3,200 shapes from the ETH-80 database.

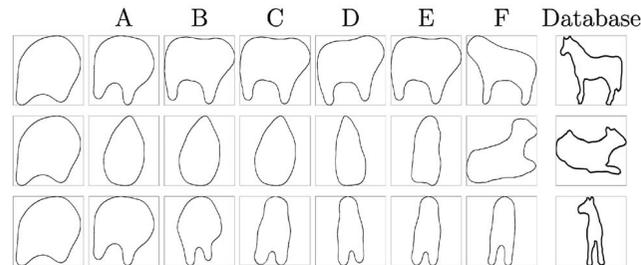


Fig. 11. Paths from top to bottom in the tree show increasing shape resolutions.

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 $\mu \in \mathcal{S}$, $T_\mu(\mathcal{S}) \subset \mathbb{L}^2$, is a **vector space** and more conventional statistics apply.
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 $g \in T_\mu(\mathcal{S})$ 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 $g \equiv \mathbf{x} = \{x_i\} \in \mathbb{R}^m$, one can define a probability distribution on elements of $T_\mu(\mathcal{S})$ via one on \mathbb{R}^m .

We still have to decide what form does this probability distribution take. Following are three possibilities:

1. **Multivariate normal model for principal coefficients.** A common approach is to assume a

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

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

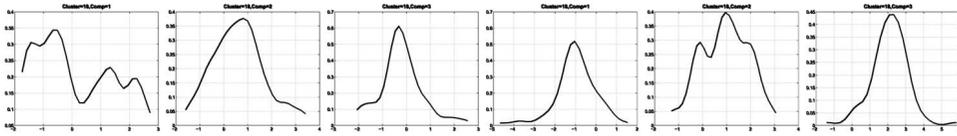


Fig. 12. Estimated densities of four dominant components: 1) PCA (first three) and 2) ICA (last three), for shapes in cluster 18.

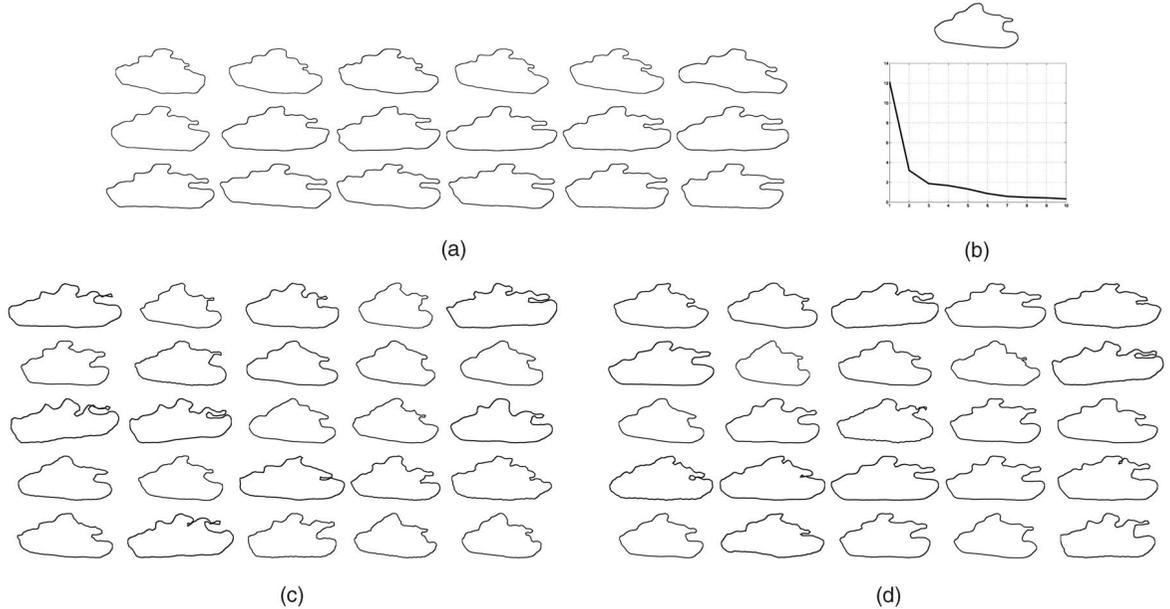


Fig. 13. (a) 18 observed M60 tank shapes, (b) top shows the mean shape and bottom plots the principal eigenvalues of tangent covariance, (c) shows random samples from multivariate normal model, and (d) shows samples from a nonparametric model for principal coefficients.

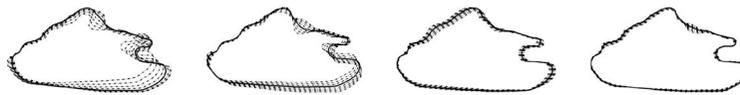


Fig. 14. Principal modes of variations in the 18 observed shapes shown in Fig. 13.

modes of variations. The density function associated with this family of shapes is given by:

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

2. **Nonparametric models on principal coefficients.**

Instead of assuming a parametric model for $\tilde{\mathbf{x}}$, here we estimate the probability density function of each component using kernel density estimation (with a Gaussian kernel). Shown in Fig. 12 are some examples of estimated density functions. The first three plots show density functions of the three dominant components for shapes in cluster 18. These histograms seem typical of the larger database in the sense that both unimodal and multimodal densities occur frequently. In principle, one should form joint density estimates from the observed coefficients, as they may be dependent. However, in this paper, we ignore their 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 Fig. 12 are 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 Fig. 13 is an example of random sampling from estimated models. Eighteen observed shapes $\theta_1, \dots, \theta_{18}$ of M60 tanks are shown in Fig. 13a and are analyzed for learning a probability model. The Karcher mean shape and estimated eigenvalues of the sample covariance matrix are shown in Fig. 13b. Generating random samples from $\tilde{\mathbf{x}} \sim N(0, \hat{K})$ and computing $\Psi(\mu, \sum_{i=1}^{m_1} \tilde{x}_i v_i, 1)$, we have synthesized new shapes from

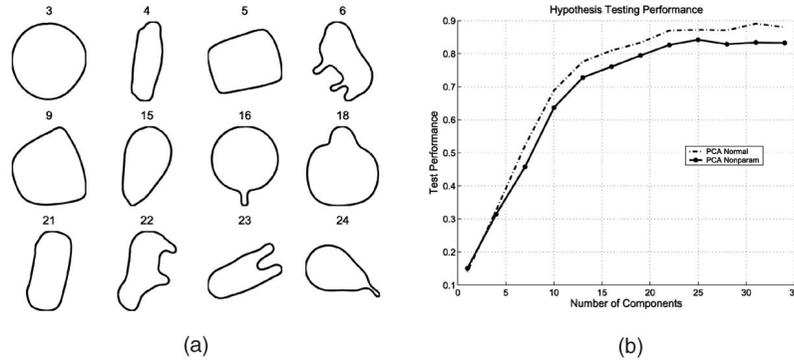


Fig. 15. (a) Mean shapes of 12 clusters. (b) Plots of classification performance versus the number of components used for: 1) multivariate normal modal on principal components and 2) independent nonparametric models on principal components.

the model learned shown in Fig. 13c. Four principal modes of shape variations for this data set are shown in Fig. 14. As a comparison, shown in Fig. 13d 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 log-likelihood ratio: $l(\theta) \equiv \log(\frac{h_1(\theta)}{h_2(\theta)})$. Substituting for normal distributions (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 \mathbf{x}_i be the vector of Fourier coefficients that encode the tangent direction from μ_i to θ such that $\mathbf{x}_i = \tilde{\mathbf{x}}_i \oplus \tilde{\mathbf{x}}_i^\perp$, $i = 1, 2$. It follows that

$$l(\theta) = -\tilde{\mathbf{x}}_1^T K_1^{-1} \tilde{\mathbf{x}}_1 + \tilde{\mathbf{x}}_2^T K_2^{-1} \tilde{\mathbf{x}}_2 - (\|\tilde{\mathbf{x}}_1^\perp\|^2 - \|\tilde{\mathbf{x}}_2^\perp\|^2)/(2\epsilon) - \log(\det(K_1)) + \log(\det(K_2)).$$

In the special case, when $m = m_1$ and $K_1 = K_2 = I_m$ (identity), the log-likelihood ratio is given by $l(\theta) = \|\mathbf{x}_1\|^2 - \|\mathbf{x}_2\|^2$. Multiple hypothesis testing can be accomplished similarly using the most likely hypothesis. The curved nature of the shape space \mathcal{S} makes an analytical study of this test difficult. For instance, one may be interested in the probability of type one error but that calculation requires a probability model on $\tilde{\mathbf{x}}_2$ when H_0 is true.

We now present results from an experiment on hypothesis testing using the ETH database. In this experiment, we selected 12 out of 25 clusters shown at the bottom level of Fig. 10. Shown in Fig. 15a are mean shapes of these 12 clusters. In each class (or cluster), we used 50 shapes as training data for learning the shape models. A disjoint set of 2,500 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 Fig. 15b is a plot of classification performance versus m_1 , the number of components used. This plot shows classification performances using: 1) multivariate normal model on tangent vectors, and 2) 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 (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 θ 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 μ_i be the given shapes at a level and let x_i be the Fourier vector that encode tangent direction from θ to μ_i . Then, the nearest shape is indexed by $\hat{i} = \operatorname{argmin}_i \|\mathbf{x}_i\|$. Proceed down the tree following the nearest shape $\mu_{\hat{i}}$ at each level. This continues until we reach the last level and have found an 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 $\theta \in \mathcal{S}$, and follow the tree, shown in Fig. 10, for retrieving similar shapes from the database. Fig. 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. The 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:

Type of search	time(sec)	Type of search	time(sec)
exhaustive	126.64	worst case (hierarchical)	70.57
best case (hierarchical)	0.92	average (hierarchical)	24.84

Test Image	Test Shape	Retrieved shapes from hierarchy	time(sec)
			27.39
			17.23
			15.93
			14.06
			43.26

Fig. 16. Some examples of shape retrieval using hierarchical organization.

The time for exhaustive search is computed by considering all 3,200 comparisons, 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, etc. Shown in Fig. 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. Fig. 17a 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. Fig. 17b shows plots of recall rate plotted against the number of shapes retrieved, and Fig. 17c 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 even 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 AND 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 by an evaluation of cluster means to perform hierarchical clustering. Using 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 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 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 diffeomorphisms 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 preshape manifold \mathcal{C} , and the shape space is given by \mathcal{C}/\mathcal{D} , where \mathcal{D} is the group of

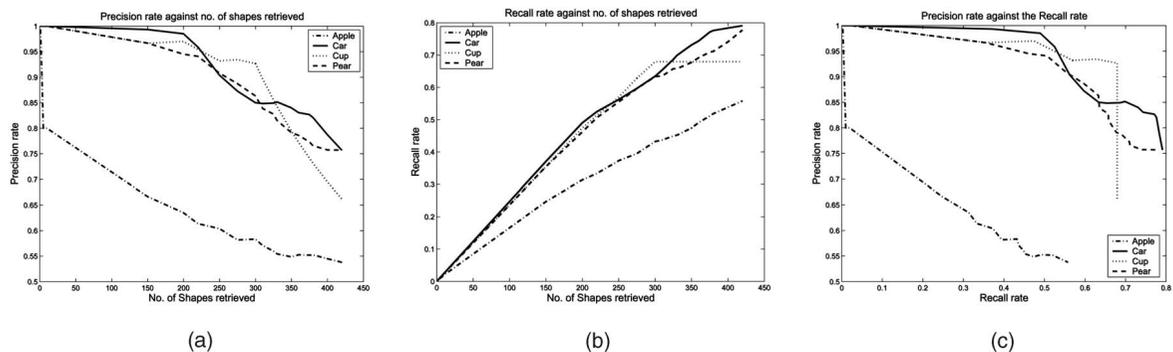


Fig. 17. (a) Precision rate versus number of shapes retrieved, (b) recall rate versus number retrieved, and (c) precision rate versus recall rate.

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].

ACKNOWLEDGMENTS

This research was supported the US National Science Foundation grants (FRG) DMS-0101429, NMA 201-01-2010, NSF (ACT) DMS-0345242, and ARO W911NF-04-01-0268. The authors would like to thank the three reviewers for their helpful comments. They are also grateful to the producers of ETH, Surrey, and AMCOM databases. This paper was presented in part at ECCV 2004, Prague, Czech Republic.

REFERENCES

- [1] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models: Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, 1995.
- [2] R.H. Davies, C.J. Twining, P.D. Allen, T.F. Cootes, and C.J. Taylor, "Building Optimal 2D Statistical Shape Models," *Image and Vision Computing*, vol. 21, pp. 1171-1182, 2003.
- [3] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*. John Wiley & Sons, 1998.
- [4] N. Duta, A.K. Jain, and M.-P. Dubuisson-Jolly, "Automatic Construction of 2D Shape Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, May 2001.
- [5] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, Mar. 2002.
- [6] U. Grenander, *General Pattern Theory*. Oxford Univ. Press, 1993.
- [7] U. Grenander and M.I. Miller, "Computational Anatomy: An Emerging Discipline," *Quarterly of Applied Math.*, vol. LVI, no. 4, pp. 617-694, 1998.
- [8] U. Grenander and A. Srivastava, "Probability Models for Clutter in Natural Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 424-429, Apr. 2001.
- [9] T. Hofmann and J.M. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 1-14, Jan. 1997.
- [10] A. Holboth, J.T. Kent, and I.L. Dryden, "On the Relation between Edge and Vertex Modeling in Shape Analysis," *Scandinavian J. Statistics*, vol. 29, pp. 355-374, 2002.
- [11] A. Hyvarinen, "Fast and Robust Fixed-Point Algorithm for Independent Component Analysis," *IEEE Trans. Neural Networks*, vol. 10, pp. 626-634, 1999.
- [12] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*. Prentice-Hall, 1988.
- [13] H. Karcher, "Riemann Center of Mass and Mollifier Smoothing," *Comm. Pure and Applied Math.*, vol. 30, pp. 509-541, 1977.
- [14] J.T. Kent and K.V. Mardia, "Procrustes Tangent Projections and Bilateral Symmetry," *Biometrika*, vol. 88, pp. 469-485, 2001.
- [15] E. Klassen, A. Srivastava, W. Mio, and S. Joshi, "Analysis of Planar Shapes Using Geodesic Paths on Shape Spaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 372-383, Mar. 2004.
- [16] H.L. Le and D.G. Kendall, "The Riemannian Structure of Euclidean Shape Spaces: A Novel Environment for Statistics," *Annals of Statistics*, vol. 21, no. 3, pp. 1225-1271, 1993.
- [17] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, 1989.
- [18] W. Mio and A. Srivastava, "Elastic String Models for Representation and Analysis of Planar Shapes," *Proc. IEEE Computer Vision and Pattern Recognition*, 2004.
- [19] W. Mio, A. Srivastava, and X. Liu, "Learning and Bayesian Shape Extraction for Object Recognition," *Proc. European Conf. Computer Vision*, T. Pajdla and J. Matas, eds., part IV, pp. 62-73, 2004.
- [20] F. Mokhtarian, "Silhouette-Based Isolated Object Recognition through Curvature Scale Space," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 539-544, May 1995.
- [21] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 1999.
- [22] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210-2239, 1998.
- [23] C.G. Small, *The Statistical Theory of Shape*. Springer, 1996.



Anuj Srivastava received the BTech degree in electronics engineering from Institute of Technology, Banaras Hindu University, Varanasi, India, in 1990, and the MS and PhD degrees in electrical engineering from Washington University, St. Louis, Missouri, in 1993 and 1996, respectively. He was a visiting research associate at the Division of Applied Mathematics, Brown University, from 1996 to 1997, and then joined the Department of Statistics at Florida State University (FSU) in Tallahassee, as an assistant professor in 1997. Currently, he is an associate professor there. He is also a member of the Center for Applied Vision and Imaging Sciences (CAVIS) at FSU. Dr. Srivastava is a member of American Statistical Association. He has been member of several conference technical committees, including IEEE SSP Workshop (2003), ICPR (2004), CVPR (2004), and Biometrics Symposium (2004). He is an associate editor of *IEEE Signal Processing* and the *Journal of Statistical Planning and Inference*. His research interests include statistical computer vision, computational statistics, and the use of differential geometry in statistical inferences. He is a member of the IEEE.



Shantanu H. Joshi received the BE degree in electronics and telecommunication from the University of Pune, India in 1998, and the MS degree in electrical engineering from the Florida State University. He is currently pursuing the PhD degree in the Department of Electrical Engineering at Florida State University and working as a research assistant in the Center for Applied Vision and Imaging Sciences at Florida State University. His current research interests include image analysis and computational differential geometry with applications to computer vision.



Washington Mio received the PhD degree in mathematics in 1984 from New York University for a thesis on geometric topology of manifolds and is currently an associate professor at Florida State University. He was an assistant professor at the Instituto de Matemática Pura e Aplicada in Rio de Janeiro until 1987 and has held visiting positions at the Courant Institute of Mathematical Sciences, Cornell University and the University of Pennsylvania. His main research interests are in the areas of shape analysis, image understanding, and topology of higher dimensional manifolds.



Xiuwen Liu received the BEng degree in Computer Science in 1989 from Tsinghua University, Beijing, China, the MS degrees in geodetic science and Surveying in 1995 and Computer and Information Science in 1996, and the PhD degree in computer and information science in 1999 from The Ohio State University, Columbus. From August 1989 to February 1993, he was with the Department of Computer Science and Technology at Tsinghua University. Since 2000, he has been with the Department of Computer Science of the Florida State University, Tallahassee. His current research interests include low-dimensional representations of images, statistical pattern recognition, manifold-based optimization and inference algorithms, computational modeling of visual perception and inference, and real-time vision algorithms and implementations using FPGAs. He is a member of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.