

Building 3-D Statistical Shape Models by Direct Optimization

Rhodri H. Davies, Carole J. Twining*, Timothy F. Cootes, and Chris J. Taylor

Abstract—Statistical shape models are powerful tools for image interpretation and shape analysis. A simple, yet effective, way of building such models is to capture the statistics of sampled point coordinates over a training set of example shapes. However, a major drawback of this approach is the need to establish a correspondence across the training set. In 2-D, a correspondence is often defined using a set of manually placed “landmarks” and linear interpolation to sample the shape in between. Such annotation is, however, time-consuming and subjective, particularly when extended to 3-D. In this paper, we show that it is possible to establish a dense correspondence across the whole training set automatically by treating correspondence as an optimization problem. The objective function we use for the optimization is based on the minimum description length principle, which we argue is a criterion that leads to models with good compactness, specificity, and generalization ability. We manipulate correspondence by reparameterizing each training shape. We describe an explicit representation of reparameterization for surfaces in 3-D that makes it impossible to generate an illegal (i.e., not one-to-one) correspondence. We also describe several large-scale optimization strategies for model building, and perform a detailed analysis of each approach. Finally, we derive quantitative measures of model quality, allowing meaningful comparison between models built using different methods. Results are given for several different training sets of 3-D shapes, which show that the minimum description length models perform significantly better than other approaches.

Index Terms—Active shape models, correspondence, landmarks, minimum description length, point distribution models, statistical shape models.

I. INTRODUCTION

STATISTICAL models of shape have proven to be an effective basis for image interpretation [1]–[9] and shape analysis [10], [11]. The basic idea is that, given a set of shapes (in 2-D or 3-D), each shape can be represented as a point in a high-dimensional vector space, allowing a statistical analysis of shape variation over the set. By studying the distribution of

shapes in this *shape space*, it is possible to identify systematic differences in shape between subpopulations (e.g., normal versus disease) or within a population over time [10], [11]. It is also possible to use knowledge of the subspace of “legal” shapes for a given class of objects to apply shape constraints during image segmentation [3]. The key to all of this is the ability to construct a shape space that is appropriate for a given set of shapes.

A simple approach to build a shape model, and that followed in this paper, is to sample a fixed number of points on the boundary/surface of each shape, then concatenate the coordinates to create a shape vector [3]. Clearly, the way these points are chosen will affect the distribution of shapes in shape space, and some choices will lead to more useful representations than others. For example, in the extreme, we might have a set of identical shapes but, by sampling the set of points differently from each, induce apparent shape variation over the set.

The underlying problem is this: given a point on one shape, how do we identify the corresponding point on each of the set of shapes? In 2-D, meaningful correspondence can be established using manually defined “landmarks” (e.g., [12] and [13]), but this is time-consuming and subjective. In 3-D, manual landmarking becomes impractical. Various attempts have been described to define correspondence automatically, but these existing methods have significant limitations (see Section II). We have shown previously [14] that establishing an appropriate correspondence for a set of 2-D shapes can be posed as an optimization problem, thus minimizing the description length of the set of shapes in shape space, with respect to the correspondence. In this paper, we show how this minimum description length (MDL [15]) strategy can be extended to 3-D. Our approach is to define a continuous parameterization of each shape, defining corresponding points as those with the same parameter values. This formulation is relevant to any surface/boundary-based approach to shape representation (e.g., [16] and [17]), not just the point-sampling approach.

The first contribution of the paper is an explicit representation of 3-D shape parameterization, which can be used to manipulate correspondence so as to minimize description length. The approach is based on re-parameterizing surfaces using kernel functions, and builds on our previous work on reparameterizing curves in 2-D [18]. The extension to surfaces in 3-D is non-trivial. A curve in 2-D, whatever its intrinsic topology [i.e., a line with endpoints or a closed line (circle)] is a simple 1-D manifold that admits an ordering, so that for a correspondence to be legal (i.e., a homeomorphic mapping), the reparameterization needs only to be a monotonic function. No such ordering exists for surfaces in 3-D, which makes the generation of homeomorphic

Manuscript received August 02, 2009; accepted October 11, 2009. First published November 03, 2009; current version published April 02, 2010. This work was supported in part by the Medical Images and Signals Interdisciplinary Research Consortium/Collaboration, in part by the Engineering and Physical Sciences Research Council under Grant GR/N14248/01, and in part by the U.K. Medical Research Council under Grant D2025/31. The work of R. Davies was supported by the Biotechnology and Biological Sciences Research Council and AstraZeneca, Alderley Park, Macclesfield, U.K. *Asterisk indicates corresponding author.*

R. H. Davies, T. F. Cootes, and C. J. Taylor are with the Division of Imaging Science and Biomedical Engineering (ISBE), University of Manchester, M13 9PT Manchester, U.K..

*C. J. Twining is with the Imaging Science and Biomedical Engineering, University of Manchester, M13 9PT Manchester, U.K. (e-mail: carole.twining@manchester.ac.uk).

Digital Object Identifier 10.1109/TMI.2009.2035048

mappings more difficult. We describe how this can be achieved using a family of parameterized homeomorphic mappings of the sphere, which can be concatenated to produce general homeomorphic mappings. An initial description of this approach appeared previously in [19]. In what follows, we refer exclusively to surfaces in 3-D, but the method can be applied straightforwardly to curves in 2-D.

The second important contribution is an extensive experimental evaluation of different strategies for solving the large-scale optimization problem on which our approach is based. We describe an approach that scales approximately linearly with the number of training shapes, and is sufficiently fast that it can be used routinely in practical applications on a modern desktop computer.

The final contribution is to introduce a set of quantitative measures of model quality, allowing meaningful comparisons between models built using different methods. We define indexes for the important model properties of compactness, specificity, and the ability to generalize. We also derive error bounds for these indexes, allowing the significance of differences between models to be established. We use these measures of model quality to compare models built using our MDL approach and a baseline method, for four different sets of 3-D biological shapes, demonstrating the benefit of the optimization approach. For one set of shapes, we compare a model built using our approach with one constructed using the spherical harmonic description (SPHARM) method [16], [20], which has been used extensively by other researchers. We show that the MDL model is significantly better than the SPHARM model.

II. BACKGROUND

A. Representing Shape

There are many methods of representing shape of varying degrees of complexity. The simplest representation uses a set of discrete points on the shape, with an interpolant between these points producing the continuous surface. A linear interpolant between the points is the simplest choice, producing a polyhedral surface; the density of the point set can be increased so that the polyhedral surface fits the original surface to any required degree of accuracy. However, using this representation directly (for example, by concatenating the coordinates of all the points to give a single vector representing each shape) is problematic, since the number of points can vary between different examples.

A shape can also be represented by using a set of basis functions; for example, Staib and Duncan use Fourier descriptors [21] and Brechbühler *et al.* use spherical harmonics [22]. Statistical analysis can then be performed on the coefficients of the basis functions. However, it should be noted that constructing such a basis function representation of shape necessitates a parameterization of that shape, via the parameterized basis functions. This hence generates a common parameterization across the shapes, which implies a correspondence. This correspondence is, however, implicit.

There are alternative approaches, which rather than working directly with the surface, use medial representations generated from the surface instead—examples include the medial axis

transform (MAT) [23] and M-REPs [17]. These representations do give a compact and intuitive representation of shape, but the correspondence between shapes is again implicit.

Finally, we can also consider volumetric representations of shape that work with the volumes of the interior and the exterior of the shape, rather than just the surface that separates these volumes. The simplest of such representation is a binary image, which labels interior and exterior. Then a correspondence between the surfaces of the shapes is induced by a correspondence of the entire volume. This then links to volumetric image registration techniques for finding correspondence, where the additional information provided by the entire volumetric image is used to generate a correspondence for shapes embedded in/derived from the images (for example, [24]–[27]). A related approach is to represent shapes as “distance maps:” a distance transform is applied to a binary image of the segmentation so that the values in the resulting image give the distance of each pixel/voxel to the nearest boundary/surface point [28]–[31] (this is also related to the medial representations considered before).

We choose to use a point-based representation of shape. The advantage of this approach is that this makes the representation of correspondence explicit, rather than implicit, and means that it is relatively simple to manipulate correspondence independently of shape. The problem with manipulating an implicit shape correspondence is that, as the name suggests, manipulating the correspondence also manipulates the shape representation. The correspondence for such representations can be decoupled (e.g., [32]), but this usually incurs a considerable cost in terms of increased computational complexity.

B. Statistical Shape Models

A 3-D statistical shape model is built from a training set of example surfaces, aligned to a common coordinate frame [3], [33]. Each entire shape surface, $S_i (i = 1, \dots, n_s)$, can—without loss of generality—be represented by a set of n_p points regularly sampled on the shape, as defined by some parameterization ϕ_i . This allows each shape S_i to be represented by an $3n_p$ -dimensional shape vector \mathbf{x}_i , formed by concatenating the coordinates of its sample points. Using principal component analysis, each shape vector can be expressed using a linear model of the form

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i = \bar{\mathbf{x}} + \sum_m \mathbf{p}^m b_i^m \quad (1)$$

where $\bar{\mathbf{x}}$ is the mean shape vector, $\mathbf{P} = \{\mathbf{p}^m\}$ are the eigenvectors of the covariance matrix (with corresponding eigenvalues $\{\lambda^m\}$) that describe a set of orthogonal modes of shape variation, and $\mathbf{b} = \{b^m\}$ are shape parameters that control the modes of variation.

Since our training shapes are continuous, we are interested in the limit $n_p \rightarrow \infty$. This leads to an infinitely large covariance matrix, but we note that there can only be, at most, $n_s - 1$ nonzero eigenvalues. We have shown elsewhere [18] that the eigenvectors $\{\mathbf{p}^m\}$ can be obtained directly from an $n_s \times n_s$ covariance matrix \mathbf{D}' . The ij th element of \mathbf{D}' can be calculated by

$$D'_{ij} = \int d\mu(\mathbf{u}) (\mathbf{S}_i(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})) \cdot (\mathbf{S}_j(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})) \quad (2)$$

where \mathbf{u} is a parameterization of the surface [34]. We see that $[\mathbf{S}_i(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})]$ is a vector representing the difference between a corresponding point (referenced by the particular value of the parameter \mathbf{u}) on the mean shape and on the i th shape. Then D'_{ij} can be considered as the integral over the pointwise inner products $[\mathbf{S}_i(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})] \cdot [\mathbf{S}_j(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u})]$ between shapes i and j relative to the mean $\bar{\mathbf{S}} = (1/(n_s)) \sum_{i=1}^{n_s} \mathbf{S}_i$. This defines an inner product between entire shapes, where for two difference shapes

$$\begin{aligned} \Omega : \Omega(\mathbf{u}) &\doteq \mathbf{S}_\Omega(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u}) \\ \varphi : \varphi(\mathbf{u}) &\doteq \mathbf{S}_\varphi(\mathbf{u}) - \bar{\mathbf{S}}(\mathbf{u}) \end{aligned} \quad (3)$$

with Cartesian components $\{\Omega^x(\mathbf{u}), \Omega^y(\mathbf{u}), \Omega^z(\mathbf{u})\}$ and $\{\varphi^x(\mathbf{u}), \varphi^y(\mathbf{u}), \varphi^z(\mathbf{u})\}$, respectively, the shape inner product is defined as

$$\Omega \cdot \varphi = \int d\mu(\mathbf{u}) \sum_{a=x,y,z} \Omega^a(\mathbf{u}) \varphi^a(\mathbf{u}) = D'_{\Omega\varphi} \quad (4)$$

where $d\mu(\mathbf{u})$ is the area measure for integration. The exact definition of the integral, and hence the shape inner product, depends on the measure used. We choose to take a measure that corresponds to the area on the mean shape $\bar{\mathbf{S}}(\mathbf{u})$. For a finite-point model, this corresponds to a model with equally spaced points on the mean shape.

In practice, the integrals can be calculated numerically—see Section III-C for details.

C. Illustration of the Correspondence Problem

New examples of a class of shapes can be generated by choosing values of $\{b^m\}$ (1) within the range found in the training set. The utility of this linear model of shape depends on the appropriateness of the set of parameterizations $\{\phi_i\}$ that are chosen, and hence the correspondence established between members of the training set. An inappropriate choice can result in the need for a large set of modes of variation (and corresponding shape parameters) to approximate the training shapes to a given accuracy, and may lead to “legal” values of $\{b^m\}$ generating “illegal” shape instances. As an example, we constructed two different 2-D models from a set of 17 hand boundaries. Model *A* uses a set of parameterizations of the shapes that cause “natural” landmarks such as the tips of the fingers to correspond. Model *B* uses a single corresponding point on the tip of the thumb, but then uses a simple path length parameterization to position the other sample points. The variances of the three most significant modes of models *A* and *B* are (1.06, 0.58, 0.30) and (2.19, 0.78, 0.54), respectively. This suggests that model *A* is more compact than model *B*. All the example shapes (see Fig. 1) generated by model *A* using values of $\{b^m\}$ within the range found in the training set are “legal” examples of hands, while model *B* generates implausible examples and is thus of limited utility for imposing shape constraints when the model is used for image search.

D. Previous Work on Defining Correspondence

In this section, we present an overview of previous work on shape correspondence. Note that we do not discuss here recent work that builds on our preliminary publication [19], but we do

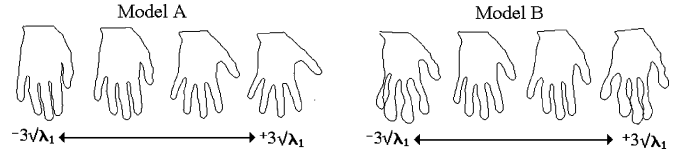


Fig. 1. First mode of variation ($\pm 3\lambda_1$) of two shape models built from the training set of hand shapes but parameterized differently. Model *A* was parameterized using manual “landmarks” and model *B* was parameterized using arc-length parameterization. The figure demonstrates that model *B* can represent invalid shape instances.

so in Section VI, where it can be set appropriately in context. We also confine our attention to approaches that use a point-based representation of shape, rather than the other representations discussed in Section II-A.

Whatever representation of shape is used, it is difficult to get away from the issue of correspondence when building a statistical shape model. This is because the first stage of building a statistical modeling is projecting the training shapes onto points into a continuous, connected shape space (the shape representation stage). The model probability density function (pdf) is then defined on this space. Any two shapes can then be connected by a continuous, smooth path, where the particular path can then be defined, for example, as the most probable or the shortest path connecting these shapes, given the model pdf. Such a path on its own is not sufficient to generate a dense, point-to-point mathematical correspondence between the shapes (that would require, for example, a rule for assigning a point-to-point correspondence between two infinitesimally close shapes). Nevertheless, any human viewer of the sequence of interpolating shapes will tend to assign an intuitive correspondence between shapes along the sequence, and reject methods where the interpolating sequence does not accord with their intuition.

Cootes *et al.* [13] define correspondence with manual landmarks placed on points of anatomical significance. Although manual landmarking often leads to acceptable results, it is a subjective, error-prone, and time-consuming process that cannot be guaranteed to produce good models. For many applications, specialist knowledge is required to place the landmarks in the correct place, which might be argued to be an advantage, since the specialist knowledge is captured by the model but, in practice, further complicates the process of model building. In 3-D, these problems are exacerbated and visualization difficulties make manual landmarking an impractical approach for selecting correspondences.

A simple approach to generate correspondence is to select a starting point on each example and equally space a fixed number of points on each surface/boundary, effectively establishing a correspondence through an arc-length parameterization. A similar scheme is advocated by Baumberg and Hogg [35], who equally space spline control points around 2-D contours. Lameckera *et al.* [36] describe a related approach in 3-D, where a correspondence is established through a shape-preserving parameterization of each shape. As illustrated in Fig. 1, however, such approaches can lead to poor results.

Kelemen *et al.* describe a method of building shape models from a set of closed 3-D surfaces by defining correspondence through a spherical harmonic parameterization of each shape

[16], [37]. We will describe the method in some detail since it is used for comparison purposes later in the paper. An initial parameterization of each shape is found using the method of Brechbühler *et al.* [22], which poses surface parameterization as an optimization problem by finding the mapping, from the surface to a sphere, which minimizes area distortion. Using this parameterization, each shape is represented by its expansion into a sum of spherical harmonics. The shapes are aligned so that the axes of their first spherical harmonics (which are ellipsoidal) coincide and principal component analysis is performed on the coefficients of the expansion. Since the expansion to spherical harmonics is a linear process involving integration over the surface, the net effect is the continuous equivalent of equally spacing points over each surface (according to its initial parameterization).

Meier and Fisher [38] also use spherical harmonic expansions to represent sets of shapes. A pairwise strategy is employed to optimize the parameterization of the shapes in order to maximize some similarity measures.

We next consider a group of methods that deal with varying and refining correspondence in an elastic manner. Kaus *et al.* [39] define a correspondence by fitting an elastically deformable mesh to a set of binary images. Shelton [40] define correspondence by minimizing the distance between sets of points while distorting the objects as little as possible. Sclaroff and Pentland [41] also define correspondences based on the “physical” properties of an object. Similarly, Bookstein [42] describes a method of optimizing the positions of corresponding points by allowing them to slide along a 2-D shape boundary in order to minimize the bending energy required to place one point set on top of another. Although these approaches converge to well-defined optima, the resulting correspondences are not the “correct” ones in any obvious sense. As with elastic methods of image registration, the problem is that such elastic methods penalize large or nonuniform deformations of the correspondence, whereas it may indeed be such a large deformation that is required to define the best correspondence.

Brett and Taylor [2], [43], Wang *et al.* [44], and Subsol *et al.* [45], all use various shape “features” (e.g., regions of high curvature) to establish point correspondences. Although this has an intuitive appeal, there is no particular reason to believe that the correspondences established are in any sense optimal. Garcin *et al.* [46] match points using a clustering algorithm; geodesic splines are then used to ensure that the deformation field needed to overlay the two point sets is diffeomorphic. Vaillant *et al.* [47] represent surfaces as “currents,” which are then used as a basis for matching pairs of shapes. Paulsen and Hilger use a Markov random-field-based approach to define a correspondence between sets of surfaces [48]. A sparse set of manually annotated landmarks give an initial correspondence using a thin-plate spline transformation. The Markov random field acts as a regularizer, ensuring that the correspondence is “legal” (i.e., a homeomorphism). Pitiot *et al.* [49] use a physically based method, in that correspondence between pairs of shape is defined using the “observed transport measure.”

A more principled approach to establishing correspondence is to treat the task as a (groupwise) model optimization problem. Hill and Taylor [50] attempt this by minimizing the total vari-

ance of a shape model, as measured by the sum of the eigenvalues of the covariance matrix. A similar approach is followed by Baumberg and Hogg [51], who use the model in a data-driven image search to optimize an image-based objective function. During the optimization stage of both methods, points are allowed to move normal to the curve so as to better match the image evidence, but there is no mechanism that will lead to their redistribution around the boundary—the key to modifying correspondences. Rangarajan *et al.* [52] also describe a method of shape correspondence that minimizes the total model variance by simultaneously determining a set of correspondences and the similarity transformation required to register pairs of contours.

Kotcheff and Taylor [34] describe an approach based on optimizing an objective function based on the determinant of the model covariance. This favors compact models with a small number of significant modes of variation, though no rigorous theoretical justification for this formulation is offered. They use an explicit representation of the set of shape parameterizations $\{\phi_i\}$ and optimize the model directly with respect to $\{\phi_i\}$ using genetic algorithm search. Their representation of $\{\phi_i\}$ is, however, problematic and does not guarantee a one-to-one mapping. They correct the problem when it arises by reordering correspondences, which is workable for 2-D shapes, but does not extend obviously to 3-D. Although some of the results produced by their method were better than hand-generated models, the genetic algorithm search did not scale well to large training sets, and did not always converge.

Davies *et al.* also pose model building as an explicit optimization task [14]. An objective function based on the MDL principle is explicitly optimized using genetic algorithm search, with respect to a piecewise linear representation of the training shape parameterizations $\{\phi_i\}$. Although in theory the approach extends to 3-D, it was found difficult to realize in practice. The genetic algorithm also takes a long time to converge and does not scale well to large training sets. The optimization method was improved by Davies *et al.* [53] by using a multiresolution algorithm; this scaled much better with the number of training examples, and the convergence time was reduced by an order of magnitude. Davies *et al.* [18] improved the representation of $\{\phi_i\}$ using a set of kernel functions; an improved optimization method was also developed, which led to models that were better than the best models that could be built using manual landmarks.

Thodberg [54] described an iterative method of optimizing correspondence for 2-D shapes, using an MDL-based objective function. However, it should be noted that this objective function was only an *ad hoc* approximation of the full MDL objective function. Thodberg and Olafsdottir later [55] added a curvature-based term to the approximate objective function, but as noted before, regions of similar curvature should not necessarily be corresponded. Most importantly, neither their shape representation nor their method of manipulating correspondence has a simple extension to the case of 3-D shapes.

III. AUTOMATIC MODEL-BUILDING APPROACH

The approach we adopt builds on that of Kotcheff and Taylor [34] and our previous work [14], [18] in attempting to select the correspondence between members of the training set that

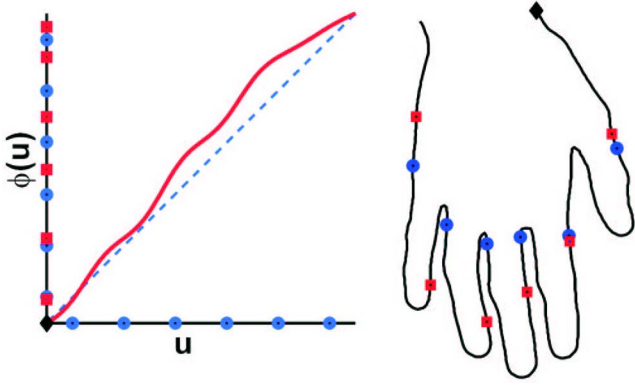


Fig. 2. (Left) Reparameterization function. The dashed line represents the original (path length) parameterization (\mathbf{u}) and the solid line represents the reparameterization function $[\phi(\mathbf{u})]$. (Right) Shape outline. The circles are sampled according to the original parameterization and the squares are sampled according to the reparameterization. The diamond represents the origin of the shape/parameterization.

builds the “best” model, but avoids all the significant limitations of their approach. This requires a framework involving:

- 1) a method of *manipulating correspondence*;
- 2) an *objective function* to assess the “quality” of the model built from a given correspondence;
- 3) a method of *optimizing* the objective function with respect to the correspondence.

We give a brief overview of each of these components in the following sections.

A. Manipulating Correspondence in 3-D

The proposed approach requires an efficient method of manipulating correspondence, which guarantees that only valid correspondences will be generated. One possible approach is to place n points on each of the training shapes and manipulate them directly. It is, however, difficult to ensure the legality of the resulting correspondences using this approach, especially in 3-D.

Instead, we treat the problem of corresponding continuous surfaces as one of explicit reparameterization. A different reparameterization function $\phi_i(\mathbf{u})$ is defined for *each* shape $\mathbf{S}_i(\mathbf{u})$, allowing points to be moved around on the boundary/surface

$$\mathbf{S}_i \mapsto \mathbf{S}'_i, \quad \mathbf{u} \mapsto \mathbf{u}'$$

where $\mathbf{S}_i(\mathbf{u}) = \mathbf{S}'_i(\mathbf{u}')$ and $\mathbf{u}' = \phi_i(\mathbf{u})$ (5) and

with $\mathbf{S}_i(\mathbf{u})$ being some initial parameterization of the shape. The idea is illustrated in 2-D in Fig. 2, where a set of points are sampled from both a path-length parameterization and a reparameterization of a hand shape. For a correspondence to be legal, $\phi_i(\mathbf{u})$ must be a homeomorphic mapping, i.e., ϕ_i must not cause folds or tears. To follow the reparameterization approach, we need: an initial parameterization, $\mathbf{S}_i(\mathbf{u})$, for each shape, and a way of representing the reparameterization function ϕ . We will now describe both of these components in turn.

We can achieve an initial parameterization by projecting each shape onto the relevant topological primitive. For a curve

in 2-D, its topology is either a line (with endpoints) or a circle (i.e., a closed line).¹ Both cases are simple 1-D manifolds that have a natural ordering, which makes it straightforward to project a shape onto a line/circle using path-length parameterization.

For surfaces in 3-D, there are, however, many more intrinsic topologies; for example, for shapes that are closed surfaces, possible topologies correspond to the sphere, the torus, and handle bodies (i.e., a sphere with handles). In this paper, we only consider objects with a spherical topology, thus we first map each shape onto the unit sphere. Since there is no natural ordering on a surface in 3-D, finding an initial parameterization is much more difficult than the 2-D case. Various surface parameterization methods have been proposed (e.g., [22] and [56]–[58]), but these usually introduce some distortion. Since we intend to optimize the reparameterization for each example, the final result should not depend significantly on this initial parameterization. We used the method of Brechbühler *et al.* [22] to produce the results reported in this paper. Each 3-D training example can now be represented parametrically as

$$\mathbf{S}(\mathbf{u}) = [S^x(\mathbf{u}), S^y(\mathbf{u}), S^z(\mathbf{u})] \quad (6)$$

where $\mathbf{S}(\mathbf{u})$ is the vector position in \mathbb{R}^3 of a point on the shape, with parameter \mathbf{u} , and with Cartesian components as given. For a spherical polar parameterization, $\mathbf{u} = (\theta, \psi)$, where θ, ψ are the usual polar angles, and we consider just a sphere of unit radius.

Given such an initial parameterization, we can re-parameterize the shape by applying homeomorphic transformations to the sphere. As noted before, any shape in 2-D has a natural ordering—in this case, a homeomorphic mapping corresponds to a transformation that preserves this ordering. A family of such transformations can be achieved by generating a family of monotonic functions. We have described elsewhere how this can be achieved in 2-D using Cauchy basis functions [18].

For surfaces in 3-D with spherical topology, (5) becomes

$$\mathbf{S}_i \mapsto \mathbf{S}'_i, \quad \theta \mapsto \theta', \quad \psi \mapsto \psi'$$

where

$$\mathbf{S}_i(\theta, \psi) = \mathbf{S}'_i(\theta', \psi')$$

and

$$\theta' = \phi_i^\theta(\theta, \psi), \quad \psi' = \phi_i^\psi(\theta, \psi). \quad (7)$$

Valid reparameterization functions ϕ_i correspond to exact homeomorphic mappings of the sphere. That is, mappings that

¹Note that we are referring here only to the intrinsic topology of the shape representation, not to how that shape is then embedded in real space. For instance, consider a *rose*, which is formed by a collection of circles, intersecting at a single point, the simplest of such case being a figure of eight. If we use a circle to parameterize this shape, then we in effect view the self-intersection involved in the figure of eight as accidental, with infinitesimal variations of the embedding in real space failing to give a self-intersection, and forming simple closed loops, rather than the figure of eight.

are continuous, one-to-one, onto, and invertible.² But since there is no natural ordering on 3-D surfaces as there is on 2-D shapes, the generation of such mappings is much more difficult. We have developed a number of such mappings, but for the objects considered in this paper, we have found experimentally that the symmetric theta transformation, described shortly, provides sufficient representational power.³ A more general set of sphere transformations is described in the Appendix.

1) *Symmetric Theta Transformations*: Consider an arbitrary point \mathbf{P} on the unit sphere. For simplicity, assume that the spherical polar coordinates (θ, ψ) on the sphere have been redefined so that \mathbf{P} corresponds to the point $\theta = 0$. The simplest transformation is a rotationally symmetric mapping that reparameterizes the θ coordinate

$$\theta \mapsto f(\theta). \quad (8)$$

For the mapping to be homeomorphic and continuous with the identity, f must be a differentiable nondecreasing monotonic function over the range $0 < \theta < \pi$, with $f(0) = 0$ and $f(\pi) = \pi$. Any such monotonic function f can be rewritten in terms of the cumulative distribution function of some density function $\rho(\theta)$, defined over the range $0 \leq \theta \leq \pi$.

As our normalized density function, we take a constant term plus a wrapped Cauchy distribution. The wrapped Cauchy [59] is a normalizable, unimodal distribution for circular data, of variable width, with two free parameters α and A . It has an analytical indefinite integral

$$\rho(\theta) = \frac{1}{N} \left[1 + A \left(\frac{1 - \alpha^2}{1 + \alpha^2 - 2\alpha \cos \theta} \right) \right] \quad (9)$$

where $N = \pi[1 + A]$. Hence

$$\begin{aligned} f(\theta) &= \pi \int_0^\theta ds \rho(s) \\ &= \frac{1}{1 + A} \left[\theta + A \arccos \left(\frac{(1 + \alpha^2) \cos \theta - 2\alpha}{1 + \alpha^2 - 2\alpha \cos \theta} \right) \right]. \end{aligned} \quad (10)$$

The parameter α can be interpreted as a width since small values of α correspond to highly peaked distributions and large values to broad distributions. We will define a subsidiary width parameter ω , where

$$\omega \equiv -\log \alpha, \quad \Rightarrow \alpha \equiv \exp(-\omega), \quad \omega \in \mathbb{R}. \quad (11)$$

The parameter $A (A \geq 0)$ is the amplitude of the Cauchy.

²We use the term *homeomorphic* rather than *diffeomorphic* since the mappings we will construct, although continuous (i.e., C^0), are not necessarily differentiable at all points (e.g., at the poles of the sphere).

³The diffeomorphism group of the sphere (a more restricted set of transformations than homeomorphisms) or the homeomorphism group of the sphere are of course infinite dimensional. However, either group modulo the movement of the set of sample points, is a finite-dimensional group. So, from a theoretical point of view, any parameterized set of transformations that can move just a single point, in any arbitrary direction, and can be adjusted to do so when the density of sampled points increases, does enable us to search the whole space of transformations in principle, when we take the limit where the number of points on the shape tends to infinity, $n_p \mapsto \infty$. From a computational point of view, the option we employ is to start from some, perhaps overlarge, set of transformations, and then discard transformations so long as the quality of the found optimum does not decrease. This is what we mean when we state that our set of transformations is sufficient.

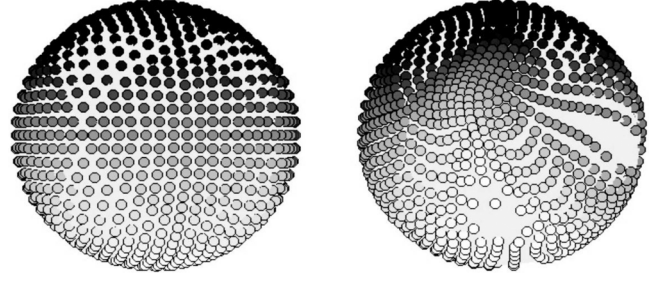


Fig. 3. (Left) Original sphere. (Right) Sphere after a series of concatenated transformations.

This describes a kernel positioned at the north pole of the sphere; for a kernel applied at an arbitrary position \mathbf{P} , the north pole of polar coordinates is taken at \mathbf{P} . The constant term is included so that $f(\theta) = \theta$ when $A = 0$, i.e., the parameterization is unchanged when the Cauchy has zero magnitude.

More general transformations of a sphere can be generated using a composition of several symmetric theta transformations with parameters $\{A_k, \omega_k, \mathbf{P}_k \mid k = 1, \dots, m\}$, which represent the amplitude, width, and center of each of the m Cauchy distributions, respectively

$$\begin{aligned} \phi_i &= \phi(\{A_k, \omega_k, \mathbf{P}_k\}) \\ &= \phi(A_1, \omega_1, \mathbf{P}_1) \circ \phi(A_2, \omega_2, \mathbf{P}_2) \circ \dots \circ \phi(A_m, \omega_m, \mathbf{P}_m). \end{aligned} \quad (12)$$

An example of such a transformation is shown in Fig. 3.

B. Information Theoretic Objective Function

In this section, we give a brief overview of the derivation and form of the MDL objective function; for a full derivation and discussion of the properties of the MDL objective function, see [14].

We wish to define a criterion for selecting the correspondence to be used in constructing a statistical shape model from a set of training surfaces $\{S_i\}$. Our aim is to choose the correspondence so as to obtain the “best possible” model. In Section II-D, we reviewed some possible objective functions, but none of these guarantee a shape model with ideal properties, namely the following.

- 1) *Generalization ability*: The model can describe any instance of the class of shapes—not just those seen in the training set;
- 2) *Specificity*: The model can only represent valid instances of the class of shapes;
- 3) *Compactness*: The variation is explained using a small set of parameters.

The ability of a model to generalize *while being specific* depends on its ability to interpolate and, to some extent, extrapolate the training set. In order to achieve these properties, we apply the principle of Occam’s razor, which can be paraphrased as: “the simplest description of the training set will interpolate/extrapolate best.”

The notion of the “simplest description” can be formalized using ideas from information theory—in particular, by applying the MDL principle [60], [61]. The basic idea is to minimize the

TABLE I
SUMMARY OF FREE PARAMETERS INVOLVED IN THE OBJECTIVE FUNCTION AND THE VARIOUS OPTIMIZATION STRATEGIES

Parameter	Description	Range Tested	Value used
Δ_{\max}	Upper limit of integration range of the data accuracy, (14) for the MDL objective function. Maximum possible value determined by knowledge of the data.	N/A	0.5 voxels
Δ_{\min}	Lower limit of integration range. Value chosen should be several orders of magnitude smaller than Δ_{\max}	N/A	0.0001 voxels
σ_G	Distribution width for the stochastic multi-scale approach	$\frac{1}{1000} \leq \sigma_G \leq \frac{1}{25}$	$\frac{1}{32}$
ω_1	Initial kernel width for multi-resolution approach	$\frac{1}{20} \leq \omega_1 \leq \frac{1}{2}$	$\frac{1}{4}$

length of a message required to transmit a full description of the training set, using the model to encode the data. As the message is encoded, the receiver must know the encoding model in order to fully reconstruct the original data, making it necessary to measure the description length of the encoding model as well as the encoded data.

In what follows, we use a multivariate Gaussian model to encode the data. Although any coding model could be used, a linear Gaussian model is computationally convenient and able to represent the data that we wish to model sufficiently. The MDL approach described here is not, however, restricted to a linear coding model as we describe in Section VI.

Under appropriate conditions, the objective function for a linear model can be *approximated* by

$$F_{\text{MDL}}(\Delta) \approx \sum_{p=1}^{n_g} [f(n_s, R, \Delta) + (n_s - 2) \log(\sigma^p)] + \frac{n_s}{2} + \sum_{q=n_g+1}^{n_g+1+n_{\min}} \left[f(n_s, R, \Delta) + (n_s - 2) \log(\sigma_{\min}) + \frac{n_s}{2} \left(\frac{\sigma^q}{\sigma_{\min}} \right)^2 \right]. \quad (13)$$

The terms in this expression are defined as follows.

Dataset terms:

- 1) n_s : The number of training shapes.
- 2) R : The spatial range of the data, a constant for a given training set.
- 3) Δ : The accuracy to which the data are coded (measured in the same units as R).

This means that the function $f(n_s, R, \Delta)$ is constant for a given training set and coding accuracy.

Model terms:

- 1) $(\sigma^m)^2$: Data variance in the m th principal direction.
- 2) $(\sigma_{\min})^2$: Lower bound on *explicitly* modeled variance.
- 3) n_g : No of directions for which $\sigma > \sigma_{\min}$.
- 4) n_{\min} : No of directions for which $\sigma \leq \sigma_{\min}$.

It is important to note that both this approximate form and the full form of the objective function have the property that they are finite, even in the case where there are modes with small or vanishing variance/eigenvalue. This should be contrasted with other objective functions, such as the determinant of the covariance matrix used in [50], which require the introduction of *ad hoc* regularization terms.

It would appear that the only free variables are σ_{\min} and Δ —however, as explained in [14], σ_{\min} can be directly related to Δ , leaving Δ as the only free variable. Previously, the value

of Δ was chosen as an estimate of the expected uncertainty in the training data, which is rarely known. Here, we overcome this by averaging F_{MDL} over a distribution of Δ values, resulting in an objective function with more continuous behavior close to convergence. In the experiments reported shortly a uniform distribution for Δ over the range Δ_{\min} to Δ_{\max} is assumed

$$F = \int_{\Delta_{\min}}^{\Delta_{\max}} d\Delta F_{\text{MDL}}(\Delta). \quad (14)$$

The integral can be solved by numerical integration [62]. Note that the choice of the maximum allowable value for Δ_{\max} depends on a knowledge of the shape data, since it cannot be made too large without degrading the actual training shapes. Δ_{\min} was chosen to be arbitrarily small, but nonzero (see Table I). We used $\Delta_{\min} = 0.0001$ pixels and $\Delta_{\max} = 0.5$ pixels for the results reported in this paper, a range of three orders of magnitude. We have found that these values are suitable for all datasets that we have tried.

Due to the high computational cost of calculating (14), we use the approximation in (13) (using a fixed value of $\Delta = 0.1$) to obtain an initial estimate of the optimal solution. This is then refined using the full objective function in [14].

C. Practical Method of Calculating the Covariance Matrix

In Section II-B, an integration must be performed over each pair of training shapes to calculate the reduced covariance matrix D' (4). The integration ensures that each shape is represented sufficiently and the true covariance of the data calculated. If the integral is not performed, the optimization can “cheat” by moving sample points away from “difficult” areas on the shapes. As an extreme example, the sample points on each shape can be collapsed to a small region on each shape, which gives a small value of the objective function, but the shapes are not represented properly. This behavior only occurs in the later stages of optimization when the models are almost optimal. In the early stages of the optimization, it makes almost no difference whether the integral is computed or not.

It is difficult to find an analytical solution to the integral and it must therefore be solved numerically, which is usually achieved using computationally expensive quadrature methods [62]. To ease the computational burden, alternative methods of ensuring a faithful sampling of the shapes have been proposed, such as repeatedly resampling the shapes so that points are uniformly distributed on the mean [63], and adding an arbitrary penalty for “runaway” points [54].

However, in practice, we have found that good results can be obtained if the shape inner product integral in (4) is calculated

using a simple finite approximation. If $\psi(\mathbf{u})$ and $\varphi(\mathbf{u})$ are the difference from the mean shape for two parameterized shapes (refeq:shapediff) and $d\mu(\mathbf{u})$ is the area measure on the mean shape, then

$$\int d\mu(\mathbf{u}) [\psi(\mathbf{u}) \cdot \varphi(\mathbf{u})] \approx \sum_{k=1}^n \Delta\mu(\mathbf{u}_k) [\psi(\mathbf{u}_k) \cdot \varphi(\mathbf{u}_k)] \quad (15)$$

where $\{\mathbf{u}_k\}$ are the parameters of some set of sample points.

For 1-D shapes (curves)

$$\Delta\mu(\mathbf{u}_k) = (d_{k,k-1} + d_{k,k+1}) \quad (16)$$

where $d_{k,k-1}$ is the Euclidian distance between the adjacent points k and $k-1$ as measured on the mean shape. For 2-D shapes (surfaces), this becomes

$$\Delta\mu(\mathbf{u}_k) = \sum_{\alpha} A_{k\alpha} \quad (17)$$

where $\{A_{k\alpha}\}$ is the area on the mean shape of all triangles that have point k as a vertex.

D. Optimizing Correspondence

We now wish to optimize the reparameterization of our training shapes so as to minimize the value of the MDL objective function F (14). We require an optimization method that converges robustly and reliably, scales well to deal with large training sets, and converges in a reasonable amount of time on a desktop computer.

We have previously described a scheme where all parameters are optimized simultaneously using genetic algorithm search [14]. However, the genetic algorithm took many hours to converge, even for small sets of shapes, and did not scale well to large training sets. In this paper, we introduce several new optimization strategies that lead to faster and more robust convergence. These are all based on the use of local optimization algorithms instead of genetic algorithm search, since local optimizers are more efficient. Any off-the-shelf optimization algorithm could be used for this purpose; we have used the downhill simplex method [62] to produce the results reported shortly.

As noted before, we represent each reparameterization using a concatenation of symmetric theta transformations. The symmetric theta transformation is a function of the parameters $\{A_k, \omega_k, \mathbf{P}_k\}$, which represent the magnitude, width, and position of the kernel, respectively. For all experiments reported later, the values of the kernel widths and positions $\{\omega_k, \mathbf{P}_k\}$ are chosen and fixed at the start of the optimization, and the magnitudes $\{A_k\}$ are used as the parameters of the optimization.

Four optimization strategies are explored: a multiresolution strategy, a stochastic multiscale strategy, optimizing one example at a time, and explicitly optimizing the pose of the training shapes. Each strategy is described in detail later. The different optimization strategies are compared in Section V by analyzing their convergence properties and evaluating the “quality” of the final models, using the set of objective measures of model quality, which are defined in Section IV.

1) *Multiresolution Approach*: The Cauchy-based representation of reparameterization described before allows the construction of continuous, smooth reparameterization functions with parameters $\{A_k, \omega_k, \mathbf{P}_k\}$. Because kernels of variable width

are used, the method naturally supports a coarse-to-fine representation of reparameterization, which allows a multiresolution approach to optimize the objective function. We described such a scheme in [19]. We begin with a coarse representation of reparameterization using a set of wide kernels. The magnitudes of these kernels are then optimized (simultaneously) using the downhill simplex algorithm—once the optimal values are found, they are fixed for the remainder of the model-building process. The reparameterization is then refined by recursively introducing narrower kernels in between those added at previous levels of recursion.

The optimization algorithm is as follows.

Repeat

- 1) For each reparameterization function ϕ_i , begin with a small set of kernels of width ω_1 , spread uniformly over the shape. The magnitude A_i of the kernels in each example is initialized to zero (equivalent to the original parameterization). The downhill simplex method is used to find the set of magnitudes $\{A_i\}$ that minimize F .
- 2) At each level of recursion k , add additional kernels of width $(\omega_1/2)^k$ halfway between the kernels added at the previous level of recursion. The downhill simplex method is again used to find the best magnitudes for the set of new kernels.

until convergence.

The convergence of the algorithm is quite robust to the selected value of ω_1 : values in the range $0.05 \leq \omega_1 \leq 0.5$ produce very similar models. A value of $\omega_1 = 0.25$ was chosen to obtain the results reported later (see Table I).

2) *Stochastic Multiscale Approach*: In the multiresolution optimization scheme described before, only a limited number of kernel widths and positions is used. Nevertheless, the number of parameters to be optimized simultaneously grows exponentially with the number of recursion levels and linearly with the number of shapes. After a few recursion levels, the number of parameters is too high for most optimization algorithms (such as the downhill simplex method) to converge reliably. Thus, application of the method is limited to small training sets of relatively simple objects. Another disadvantage is that the scheme requires an optimization schedule (number of recursion levels, iterations for each level, etc.). The optimization may also fail to find the best solution, since the magnitudes of the wider kernels are fixed at an early stage in the process.

We have developed a stochastic multiscale scheme to overcome these problems. The idea behind the approach is to select values for $\{\mathbf{P}_k\}$ and $\{\omega_k\}$ stochastically, thus optimizing the magnitude A_k for a single kernel for each example simultaneously, giving n_s parameters to optimize at each iteration. Once the optimal values are found, they are fixed for the remaining iterations, thus reducing the number of optimization parameters at each iteration. However, flexibility is still maintained since another kernel can be added at the same position and optimized at a later iteration.

The values for $\{\mathbf{P}_k\}$ are selected from a uniform distribution over the shape. The widths of the kernels $\{\omega_k\}$ are chosen from

the positive half of a Gaussian distribution with zero mean and standard deviation σ_G .

The algorithm is as follows.

Repeat

- a) randomly assign values $\{\mathbf{P}_i\}$ and $\{\omega_i\}$ for all examples i ;
- b) use the downhill simplex method to find the values of $\{A_i\}$ that minimize F .

Until convergence.

The only free parameter in this algorithm is σ_G . The convergence of the algorithm is relatively insensitive to the actual value chosen, and values in the range $(1/1000) \leq \sigma_G \leq (1/25)$ produce very similar models. A value of $\sigma_G = (1/32)$ was used in the experiments reported later.

3) *Optimizing One Example at a Time*: For all the optimization methods proposed so far, the number of parameters to be optimized simultaneously prevents the local optimization algorithm from converging reliably for relatively large training sets ($\gtrsim 100$ examples). It is also not well suited to an iterative model-building scheme where examples are segmented and added one by one.

The number of variables optimized simultaneously can be reduced by optimizing the reparameterization of only one shape at a time. This is achieved by cycling through the training set, optimizing the current reparameterization of each example before moving on to the next iteration. Note that we are still considering the *entire* training set (i.e., the model is built using the current parameterizations of *all* examples), but the parameterization of each example is optimized independently. To remove any bias, the ordering of the training set is permuted at random before each iteration. The algorithm is a direct extension of that described in the previous section and can be summarized as follows.

Repeat

- a) randomize the ordering of the training set;
- b) **for** each example, i :
 - i) randomly assign values to \mathbf{P}_i and ω_i ;
 - ii) use the downhill simplex method to find the value of A_i that minimizes F .

Until convergence.

As only one free parameter is optimized at a time, we could use a simpler optimization algorithm (such as golden section search [62]). For the sake of consistency, however, we used the downhill simplex method to produce the results reported shortly.

4) *Including Pose Parameters*: The positions of corresponding points depend on the pose parameters of each example as well as the shape reparameterization. This is easily illustrated if we view the full expansion of the inner product defined earlier (4)

$$D'_{ij} = [s_i \mathbf{R}_i \mathbf{S}_i(\phi_i(\mathbf{u})) - \mathbf{d}_i - \bar{\mathbf{S}}(\mathbf{u})] \cdot [s_j \mathbf{R}_j \mathbf{S}_j(\phi_j(\mathbf{u})) - \mathbf{d}_j - \bar{\mathbf{S}}(\mathbf{u})] \quad (18)$$

where \mathbf{S}_i is a continuous, parameterized representation of the i th shape, $\bar{\mathbf{S}} = (1/n_s) \sum_{i=1}^{n_s} (s_i \mathbf{R}_i \mathbf{S}_i - \mathbf{d}_i)$ is the mean of the aligned shapes, and $\{\mathbf{d}_i, s_i, \mathbf{R}_i\}$ is the transformation applied to the i th shape, consisting of a translation, a scaling, and a rotation, respectively.

All the strategies presented so far perform a Procrustes analysis [64] to align the training set, which is equivalent to minimizing the squared distance between each shape and the mean, then leaving the alignment fixed during optimization. We have shown elsewhere [65] that the mean squared error is not a good objective function, so it seems worthwhile to explore the effect of allowing the pose of each training example to vary during optimization. It is important to note that the optimal pose parameters depend on the *current* parameterization of the shape and must therefore be included in the iterative process.

The algorithm is as follows.

Repeat

- a) randomize the ordering of the training set;
- b) **for** each example, i :
 - i) randomly assign values to \mathbf{P}_i and ω_i ;
 - ii) use the Downhill Simplex method to find the value of A_i that minimizes F .
 - iii) use the downhill simplex method to find the values of s_i , \vec{d}_i , and \mathbf{R}_i that minimizes F .

Until convergence.

E. Extension to Multipart Objects

The model-building method described earlier extends straightforwardly to deal with multipart objects (i.e., where each training example consists of multiple simply connected subparts). The basic idea is to construct a parameterization for each subpart (by mapping its surface to the surface of the sphere) and then reparameterize each subpart using the symmetric theta transformations described in Section III-A1. The combined description length of all subparts is minimized with respect to the whole set of reparameterizations. With minor modifications, any of the strategies described before could be used for optimization. For example, here is the extension of the strategy described in Section III-D4.

Repeat

- 1) randomize the ordering of the training set;
- 2) **for** each example, i :
 - a) randomize the ordering of the subparts;
 - b) **for** each subpart, j :
 - i) randomly assign values to \mathbf{P}_{ij} and ω_{ij} ;
 - ii) use the downhill simplex method to find the value of A_{ij} that minimizes F .
 - c) use the downhill simplex method to find the values of s_i , \vec{d}_i , and \mathbf{R}_i that minimizes F . Note that the same pose transformation is applied to all subparts.

Until convergence

IV. EVALUATION APPROACH

In the previous section, we introduced four different optimization strategies. In this section, we begin by describing a quantitative approach to compare these strategies. We also establish objective criteria that can be used to compare models built using different approaches. These measures are described in detail, along with the datasets used to compare the methods.

A. Experimental Evaluation of Optimization Strategies

By performing very long trial runs, we found that each of the optimization strategies described before generally converge to the same approximate minimum, but at dramatically different rates. We now perform a quantitative assessment of the convergence rates for the various strategies, and also investigate how the convergence time scales with the size of the training set.

A qualitative comparison of the convergence of each optimization strategy can be made by plotting the value of the objective function against time (the convergence curve). We will use two measures of “time:” the CPU time and the number of objective function evaluations, which gives a less hardware- and implementation-specific measure of convergence time.

For each optimization strategy, we perform repeated trials. This is achieved by using a large pool of training examples, drawing a different set of n_s samples from the pool for each trial. This removes any bias caused by a particular choice of training set. We also select a different seed for the random number generator (which generates a different set of random numbers) for each trial.

We also wish to *quantify* the convergence properties of each strategy. It was found that the convergence curves did not fit any simple model (e.g., exponential convergence), but curves for different methods tended to possess a similar shape. We therefore developed a method of comparison based on comparing these convergence curve shapes.

Suppose we have two convergence curves, $f_A(t)$ and $f_B(t)$, generated by optimization strategies \mathcal{A} and \mathcal{B} ; the simplest model relating them is a linear model of the form

$$f_A(t) = a + bf_B(\alpha t) \quad (19)$$

where α represents a simple temporal scaling between the two curves, while b represents the corresponding scaling of the values of the objective function, with an associated offset a .

We find that this simple three-parameter model describes the data well, enabling good matching of the shape of the convergence curves. It implies a shape correspondence between the convergence curves, where time t for strategy \mathcal{A} corresponds to time αt for strategy \mathcal{B} . So, we see that if $\alpha > 1$, then strategy \mathcal{A} has a higher convergence rate, and conversely, if $\alpha < 1$, then strategy \mathcal{B} converges at a quicker rate. The value of α thus gives us a convenient way of comparing convergence rates of different strategies. In what follows, we will use α as measuring the convergence rate of strategy \mathcal{B} relative to the reference strategy \mathcal{A} , so that higher values of α will denote poorer convergence.

The optimal value of α for a pair of curves is obtained by numerical optimization of $\sum_{\tau} |f_A(\tau) - a - bf_B(\alpha\tau)|^2$ using the golden section search algorithm [62], where τ is a set of

time values where both curves are defined. The values of a and b are determined by the nonlinear least-squares approach (using the Gauss–Newton method [66]) for each candidate value of α before evaluating the objective function.

We used two sets of shapes to compare the performance of these strategies: a set of 100 hand outlines in 2-D and a set of 82 surfaces of the human hippocampi in 3-D (this dataset is described in more detail in Section IV-C). An in-depth analysis was performed on the 2-D training set for a number of training set sizes: $n_s = \{15, 20, 25, 30, 35\}$. Guided by the results of the 2-D experiments, we performed the same set of experiments on 3-D data, using training set sizes of $n_s = \{15, 25, 35\}$.

The value of α was calculated for all strategies, using a range of training set sizes, relative to a “reference” strategy. As a reference, we used the strategy where one example is optimized at a time with explicit optimization of pose (Section III-D4) using $n_s = 15$ training examples. The standard error of the α values were calculated using the Bootstrap method [67], which works by creating many bootstrap samples (we used 100) by repeatedly sampling (with replacement) a pair of trials (one from each strategy being compared) and calculating the value of α for each bootstrap sample. The standard deviation of the α values for all bootstrap samples gives the standard error.

B. Objective Metrics for Evaluating Shape Models

In order to compare our approach with other methods, it is important that we establish a consistent and objective basis for comparing models. Here, we use criteria that are based on model properties:

- 1) generalization ability;
- 2) specificity;
- 3) compactness.

We have defined quantitative measures based on these criteria elsewhere [11], [65], [68]. Each measure is evaluated for models containing a varying number of modes of variation (1), ordered by variance. The standard error for each evaluation can also be calculated, allowing meaningful comparisons between different approaches.

The objective measure evaluations define distances between shapes with the aid of a finite set of shape points, which have been sampled from the training shapes, and which define the common finite-dimensional shape space. It is therefore important that, when comparing models built using different methods, the points are sampled in a consistent manner. This can be achieved either by explicitly sampling the training shapes in such a way that the points on the mean shape are uniformly distributed (using e.g., [63]) or by performing an integration on the mean shape using the area measure defined in Section II-B.

1) *Generalization Ability:* The generalization ability of a model measures its ability to represent unseen instances of the class of object modeled. This is a fundamental property: if a model is overfitted to the training set, it will be unable to generalize to unseen examples. Generalization ability can be measured using leave-one-out reconstruction. A model is built using all but one member of the training set, and then fitted to the excluded example. The accuracy to which the model can describe the unseen example is measured and the

process is repeated excluding each example in turn. We define generalization as

$$G(M) = \frac{1}{n_s} \sum_{i=1}^{n_s} |\mathbf{x}'_i(M) - \mathbf{x}_i|^2 \quad (20)$$

where n_s is the number of shapes and $\mathbf{x}'_i(M)$ is the model reconstruction of shape \mathbf{x}_i using the model built excluding \mathbf{x}_i with M modes retained. The standard error of $G(M)$, derived from the sampling distribution for a mean, is defined as

$$\sigma_{G(M)} = \frac{\sigma}{\sqrt{n_s - 1}} \quad (21)$$

where σ is the sample standard deviation of $G(M)$.

Note it is also possible to define an alternative generalization measure (e.g., [69]), but we will use the aforementioned definition throughout.

2) *Specificity*: Specificity is central to the usefulness of a model. A model with high specificity should only generate instances of the object class that are similar to those in the training set. To define specificity, we generate a random population of N model instances, using the probability density defined by the model, and compare them to the members of the training set. We define a quantitative measure of specificity

$$S(M) = \frac{1}{N} \sum_{j=1}^N |\mathbf{x}_j(M) - \mathbf{x}'_j|^2 \quad (22)$$

where $\mathbf{x}_j(M)$ are shape examples generated by the model using M shape modes and \mathbf{x}'_j is the nearest member of the training set to \mathbf{x}_j . The standard error of $S(M)$ is given by

$$\sigma_{S(M)} = \frac{\sigma}{\sqrt{N - 1}} \quad (23)$$

where σ is the sample standard deviation of $S(M)$ and N is the number of samples. The value of N is not critical as long as it is large compared to n_s ; in the experiments reported later, we used $N = 10000$.

It can be shown that this specificity measure can be viewed as an estimator of the cross entropy between the model distribution and the data distribution [70].

3) *Compactness*: A compact model is one that has as little variance as possible and requires as few parameters as possible to define a shape instance. We measure compactness using the cumulative variance for the first M modes of the model

$$C(M) = \sum_{m=1}^M \{\lambda^m\}. \quad (24)$$

Using error propagation, the standard error of $C(M)$ can be shown to be

$$\sigma_{C(M)} = \sum_{m=1}^M \sqrt{\frac{2}{n_s}} \lambda^m. \quad (25)$$

Note that this measure of compactness is equivalent to the trace of covariance matrix, which has been used as an objective function elsewhere (e.g., [50] and [71]).

C. 3-D Shape Model Evaluation

Five training sets of 3-D objects were used to evaluate the 3-D model-building method: 23 induced stroke shapes, 8 brain ventricles, 16 rat kidneys, 19 human knees, and 82 hippocampi.

- 1) *Stroke Model*: Permanent focal cerebral ischaemia was induced in rats, and multislice T2-weighted MRI (TE = 60, TR = 2100, matrix $256 \times 256 \times 24$, field of view (FOV) $50 \times 50 \times 24$ mm) was performed *in vivo*, as described previously [72]. For the 23 examples used in this study, the shape of the infarct represents the area of the brain supplied by the middle cerebral artery.
- 2) *Brain Ventricle*: T1-weighted MRI (with voxel size $1 \times 1 \times 1.5$ mm) were acquired for eight normal volunteers on a 0.5-T GE Vectra scanner. For each image, the anterior horn of the right ventricle was segmented by a neuroradiologist.
- 3) *Rat Kidney*: Wistar, Sprague-Dawley, and Fisher rats were imaged using a 4.7-T Varian Inova MRI system (TE = 20, TR = 2000, image matrix $256 \times 256 \times 41$, FOV 64×64 , 41 slices). Multislice T2-weighted MRI was performed in the transverse plane. Sixteen segmentations of the right kidney were used in this study.
- 4) *Human Knee*: T2-weighted MRI of the bones in human knee joint (the femur, tibia, and patella) were acquired (TE = 11, TR = 58, matrix $256 \times 256 \times 60$, FOV 160×160 mm, slice thickness 1.6 mm), as described previously [73]. Segmentation of the bones was performed by a nonexpert segmenter under the supervision of a radiologist using EndPoint software (Imorphics Ltd., Manchester, U.K.). Nineteen images of different subjects were included in the model.
- 5) *Hippocampus*: MRI of the brain (TE = 60, TR = 15, matrix $256 \times 256 \times 124$, FOV 240×240 , slice thickness 1.5 mm) were acquired on a GE Signa Advantage 1.5 T scanner, as described previously [74], [75]. The hippocampus (hippocampus proper, subiculum, fimbria, and subsplenal gyrus) was segmented using IRIS, a 3-D segmentation tool,⁴ and parameterized using the method described in [22]. The data comprised 56 hippocampi of schizophrenic patients and 26 of age-matched control subjects. The two groups were combined to form a training set of 82 examples.

For each dataset, an MDL model was constructed using the algorithm described in Section III-D4 (optimizing one example at a time, with explicit optimization of pose parameters), and compared to a model built using the SPHARM (spherical harmonic) method of Kelemen *et al.* [16], which is arguably the best previously published approach to define correspondence between sets of closed surfaces. As discussed in Section II-D, the method essentially spaces points equally over the surface according to the parameterization. So, for three of the datasets, we obtain a good approximation to the method by equally spacing points over the training surfaces. For the final dataset, (the hippocampus), a benchmark model of the hippocampus was obtained using the SPHARM method of Kelemen *et al.* [16] (see Section II), which has been used extensively in previous studies

⁴Free download: midag.cs.unc.edu.

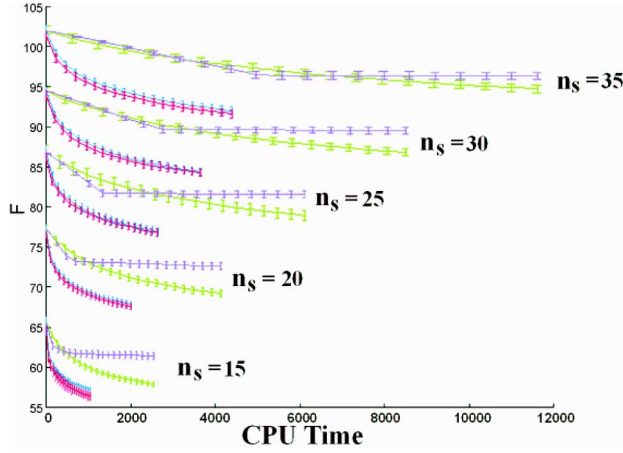


Fig. 4. Plot of the objective function value F against the CPU time for different strategies in 2-D using several training set sizes. The error bars represent one standard error. Magenta: multiresolution strategy. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

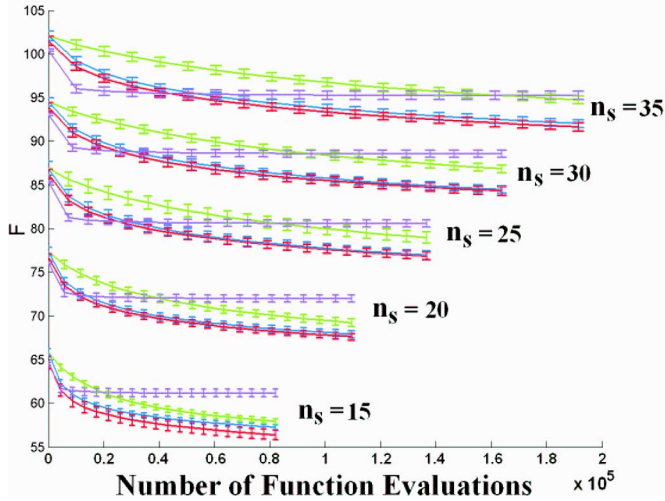


Fig. 5. Objective function value F plotted against the number of function evaluations for different strategies in 2-D using different training set sizes. The error bars represent one standard error. Magenta: multiresolution strategy. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

(e.g., [74]). To distinguish between the two implementations, we refer to them as the uniform and SPHARM models.

Quantitative comparisons of the models are reported by evaluating the models using the criteria described before (generalization, specificity, and compactness) and calculating the objective function value of each model. Qualitative results are presented by showing the first three modes of variation, independently varying the shape parameters by ± 2 (standard deviations found over the training shapes).

V. RESULTS

A. Comparison of Optimization Strategies

A qualitative comparison of the convergence rates of the optimization strategies for different numbers of training examples is shown in Figs. 4 and 6, which shows a plot of the objective function value F against the CPU time. In Figs. 5 and 7, F is

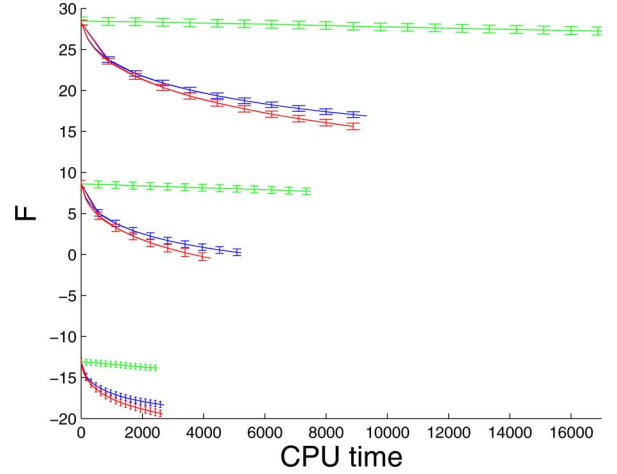


Fig. 6. Plot of the objective function value F against the CPU time for different strategies in 3-D using several training set sizes. The error bars represent one standard error. Magenta: multiresolution strategy; Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

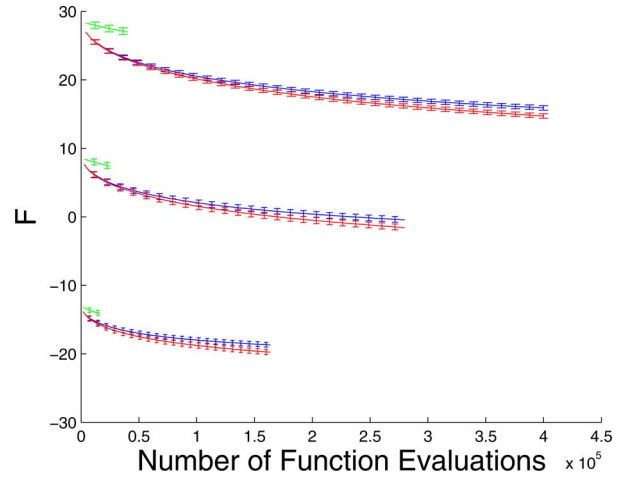


Fig. 7. Objective function value F plotted against the number of function evaluations for different strategies in 3-D using different training set sizes. The error bars represent one standard error. Magenta: multiresolution strategy. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

also plotted as a function of the number of objective function evaluations.

Note that in the 2-D results (Figs. 4 and 5), the multiresolution strategy does not perform as well as the other strategies. This is because number of optimization parameters in the multiresolution strategy doubles at each level of recursion (the number of parameters remains constant for the other strategies), which soon overwhelms the downhill simplex method. This is evident in the figures where the gradient of the multiresolution convergence curves suddenly decreases significantly and the value of the objective function stays largely constant after this point. As a result, we excluded the multiresolution strategy from the 3-D experiments. Because the shape of the convergence curves differ considerably from the other strategies, they could not be related to other strategies using the simple model in (19), so the multiresolution strategy had to be excluded from the quantitative comparison.

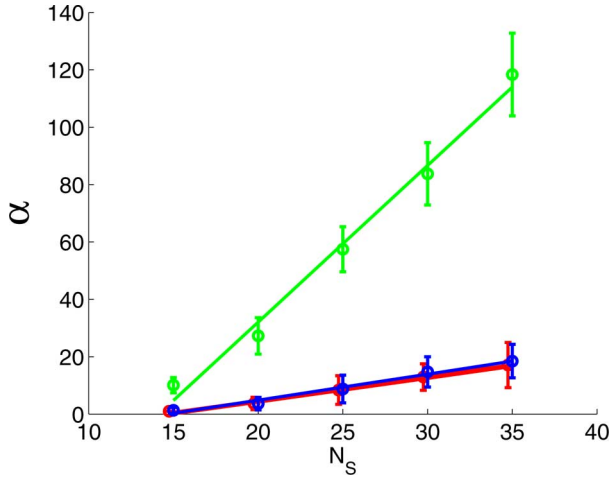


Fig. 8. Relative values of α for each optimization strategy using CPU time to represent t . The lines correspond to a least-squares fit to the points and the error bars represent \pm two standard errors. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

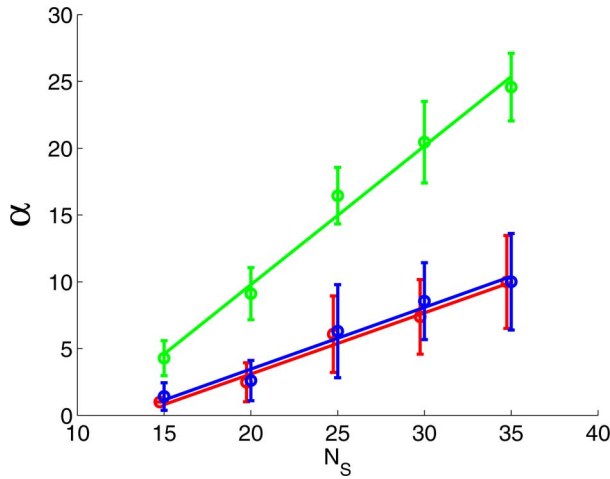


Fig. 9. Relative values of α for each optimization strategy using the number of objective function evaluations to represent t . The lines correspond to a least-squares fit to the points and the error bars represent \pm two standard errors. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

In Figs. 8 and 9, we show the corresponding results for the α values (19), which show the rates of convergence of the objective function F as a function of the number of training shapes n_s , for both measures of “time.” The gradient of the lines shown in Figs. 8 and 9 gives an estimate of how each optimization strategy scales with the number of training examples. Tables II and III shows the gradients for the various optimization strategies and for both definitions of “time.” We can see that multiscale optimization strategy (where all shapes are optimized simultaneously) does not deal well with larger numbers of training examples (higher value of the gradient). We can also see that including the pose parameters in the optimization makes the method scale better with the number of training examples, but the difference is not significant. It should be noted that optimizing the pose allows transformations that other strategies cannot perform, so that this method could potentially produce different optimum values of

TABLE II
GRADIENT OF THE GRAPH OF α VERSUS NUMBER OF TRAINING EXAMPLES FOR 2-D EXPERIMENTS (SEE FIGS. 8 AND 9)

Optimisation Strategy	CPU time	No. of objective function evaluations
Multiscale (§ III-D.2)	5.4613	1.0384
One-at-a-time (§ III-D.3)	0.9082	0.4624
Including Pose (§ III-D.4)	0.8282	0.4576

TABLE III
GRADIENT OF THE GRAPH OF α VERSUS NUMBER OF TRAINING EXAMPLES FOR 3-D EXPERIMENTS (SEE FIGS. 10 AND 11)

Optimisation Strategy	CPU time	No. of objective function evaluations
Multiscale (§ III-D.2)	47.5163	2.4037
One-at-a-time (§ III-D.3)	0.3276	0.2722
Including Pose (§ III-D.4)	0.3106	0.2758

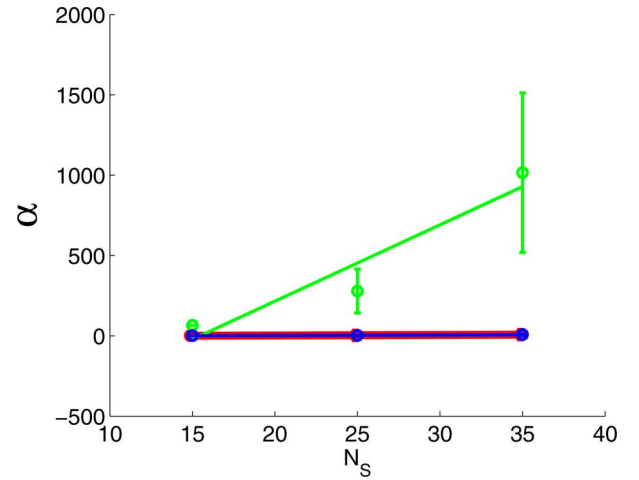


Fig. 10. The relative values of α for each optimization strategy using CPU time to represent t . The lines correspond to a least-squares fit to the points and the error bars represent \pm two standard errors. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

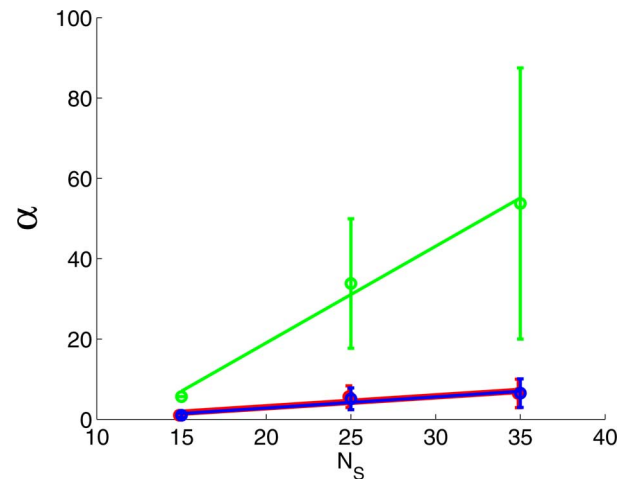


Fig. 11. Relative values of α for each optimization strategy using the number of objective function evaluations to represent t . The lines correspond to a least-squares fit to the points and the error bars represent \pm two standard errors. Green: multiscale strategy (optimizing all examples simultaneously). Blue: optimizing one example at a time. Red: explicitly optimizing the pose of the training shapes.

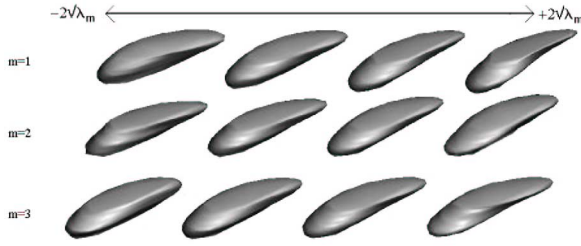


Fig. 12. First three modes of variation of the MDL Stroke model.

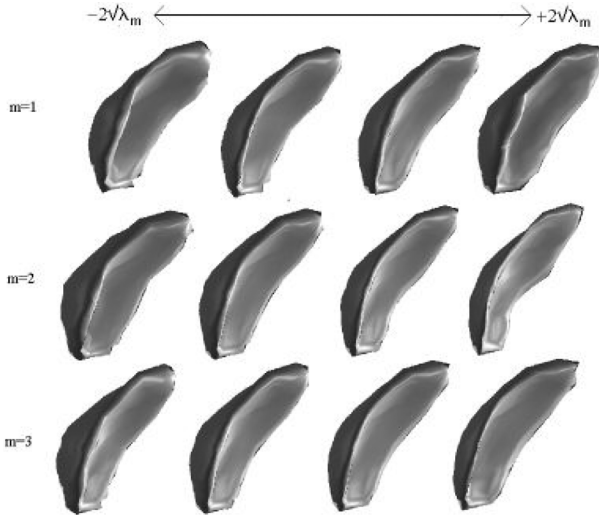


Fig. 13. First three modes of variation of the MDL brain ventricle model.

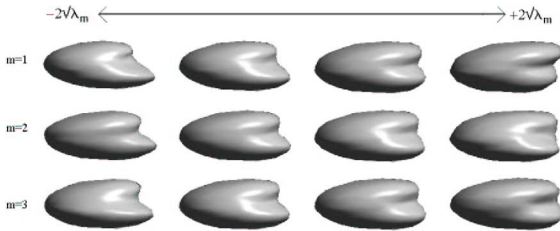


Fig. 14. First three modes of variation of the MDL rat kidney model.

the objective function, although this behavior was not seen in the data we considered.

B. 3-D Model-Building Results

The first few modes of variation for the final MDL models are shown in the appropriate figures (stroke Fig. 12, ventricle Fig. 13, kidney Fig. 14, hippocampus Fig. 15, knee Figs. 21 and 24). We performed a visual comparison of these models with the corresponding uniform/SPHARM models, and found obvious differences between the MDL and uniform/SPHARM models for all shapes except the hippocampi. The results shown in Figs. 17–20 give a quantitative comparison of different modeling approaches; they show that the MDL models have significantly better generalization and specificity properties than the uniform/SPHARM models for all datasets. The MDL model is also more compact for all datasets, but the difference is not statistically significant for the kidney (see Fig. 18) and stroke data

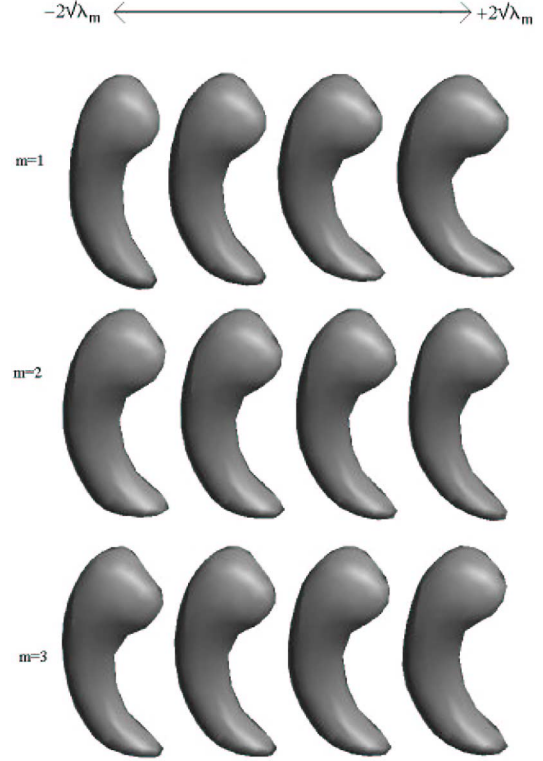


Fig. 15. First three modes of the MDL hippocampus model.

(see Fig. 16). The lack of significance is probably due to the small size of the datasets.

C. Correspondences Found

To complete this section, we will look at some correspondences found by the MDL method. Fig. 22 shows the correspondence established for the hand data described earlier. The figure shows that areas of high curvature (e.g., the tips of the fingers) are in correspondence. We have observed that our method tends to introduce a correspondence between areas of high curvature on many datasets. This is because an incorrect correspondence on an area of high curvature causes much spurious variation, leading to a nonoptimal description length.

For the hippocampus data, the MDL model was initialized using the SPHARM correspondence. This allows a direct comparison of the change in the correspondence generated by the MDL approach; see Fig. 23. It can be seen that the most substantial differences are on the tail and the head; this is not surprising as these are the regions of highest curvature and the regions of greatest variation between examples in the training set. Another interesting effect is the “skewing” that occurs around the middle of the hippocampus body.

VI. DISCUSSION AND CONCLUSION

We have described a principled and practical approach to automatically construct 3-D statistical shape models. We have shown that the method produces models that have better compactness, generalization, and specificity properties than previous methods.

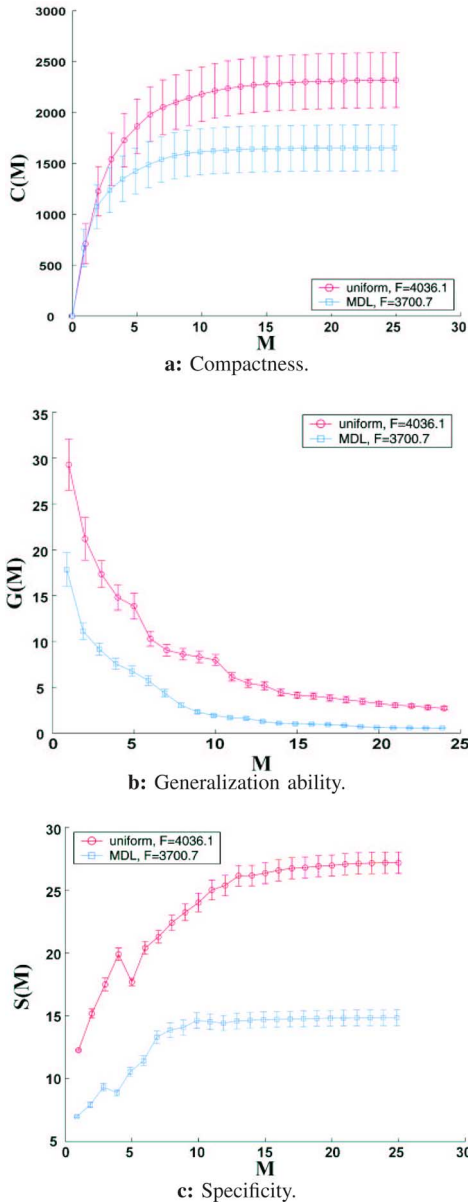


Fig. 16. Quantitative comparison of the stroke models. (a) Compactness. (b) Generalization ability. (c) Specificity.

As noted earlier, other researchers have also reported interesting results in this area.

Ericsson and Åström [76] showed how to calculate the Jacobian for MDL and used it within the steepest descent method to optimize correspondence in 2-D. Hladůvka and Bühler [77] use Bernstein polynomials to represent parameterization and show how the Jacobian can be calculated analytically for that particular representation. Heimann *et al.* used a gradient-descent method [78], and applied such models to 3-D segmentation of the liver [79] from clinical CT images, thereby achieving promising results for automatic segmentation.

Other work considers other choices for the representation of the reparameterization function; Horkaew and Yang [80] optimize the full MDL objective function with respect to a B spline-based representation of parameterization. They consider objects

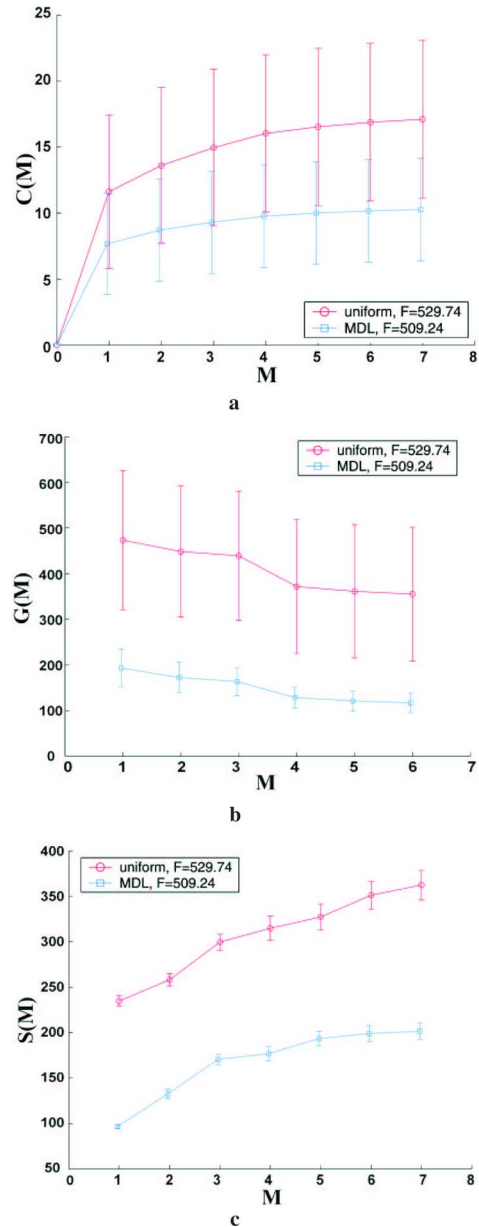


Fig. 17. Quantitative comparison of the brain ventricle models, with standard error bars. (a) Compactness. (b) Generalization ability. (c) Specificity.

that are topologically equivalent to a disk (the cardiac left ventricle). The model built from the optimized correspondence is used for 3-D dynamic segmentation of the left ventricle from 3-D MR time-series data. They show that the optimized model is more compact than the one that used uniform sampling, and that the dynamically segmented volumes agree well with manual annotation. They also note that the modes of the optimal model correspond with intrinsic physiological shape variation.

Styner *et al.* [68] have compared models built using various methods, including MDL and SPHARM. They used a number of different biomedical datasets, and concluded that for modeling purposes, MDL (or approximate MDL) outperformed other approaches. These combined results are in broad agreement with our own observation that such models are both practicable, superior in performance, and suitable for

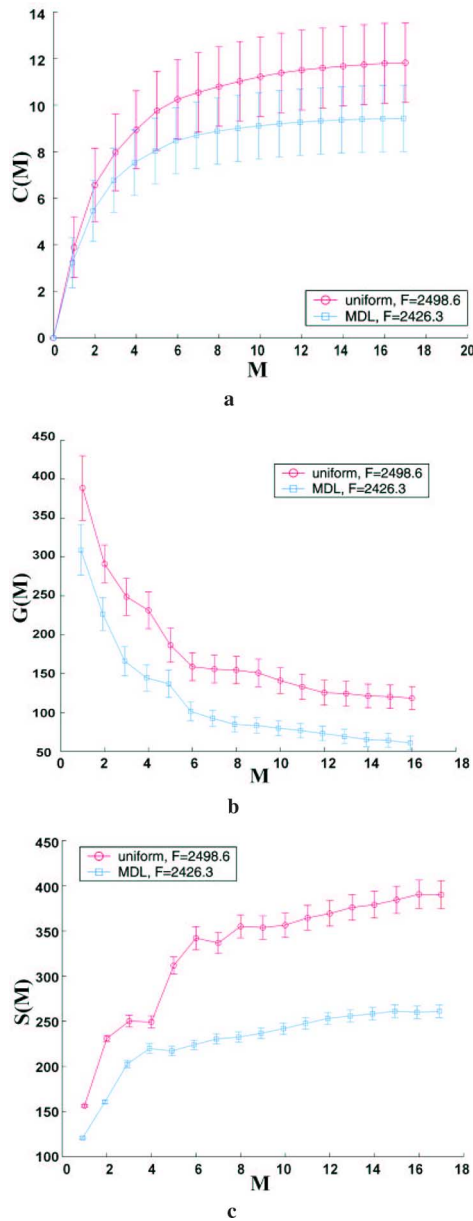


Fig. 18. Quantitative comparison of the rat kidney models. (a) Compactness. (b) Generalization ability. (c) Specificity.

real-world applications. Later, Styner *et al.* [81] looked at the performance of various model-building methods, including MDL, in morphometric analysis, but since there is no ground truth, and it is difficult to make meaningful comparisons between the methods.

A key assumption of the method presented here is that we can measure the “quality” of a model with a single suitable objective function. The approach that has been taken is based on the idea that a good model should allow a concise description of the members of the training set. As in any branch of science, this is the essential role of a model—to account for a possibly large number of observations as manifestations of some underlying pattern that is itself described as simply as possible. Other objective functions could also be incorporated into the framework described here. The determinant of the model covariance matrix [34], for example, produces plausible models.

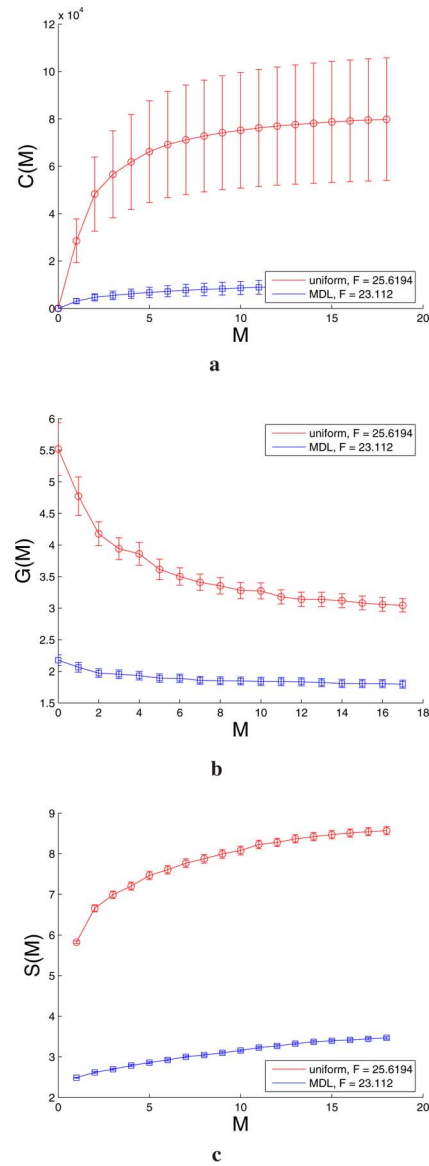


Fig. 19. Quantitative comparison of the human knee models. (a) Compactness. (b) Generalization ability. (c) Specificity.

We could also directly optimize an objective function based on the three objective measures described earlier (specificity, generalization ability, and compactness), but optimizing such an objective function will not necessarily lead to models with optimal values of all three properties.

We used the MDL principle to generate an objective function for model parameter selection, but, in principle, we could also use it to solve the model selection problem—that is, determine the optimal *class* of model. In this study, we restricted ourselves to the case of linear models where the pdf on shape space is represented as a single multivariate Gaussian. The MDL principle can also be used to generate analogous objective functions for any class of parametric statistical shape model (for example, mixture models [82], independent components analysis [83], or kernel principal components analysis [84], [85]). The restriction to linear models is due to computational, *not* theoretical reasons: in general, nonlinear models take longer to build than linear models, and since we have to rebuild the model each time

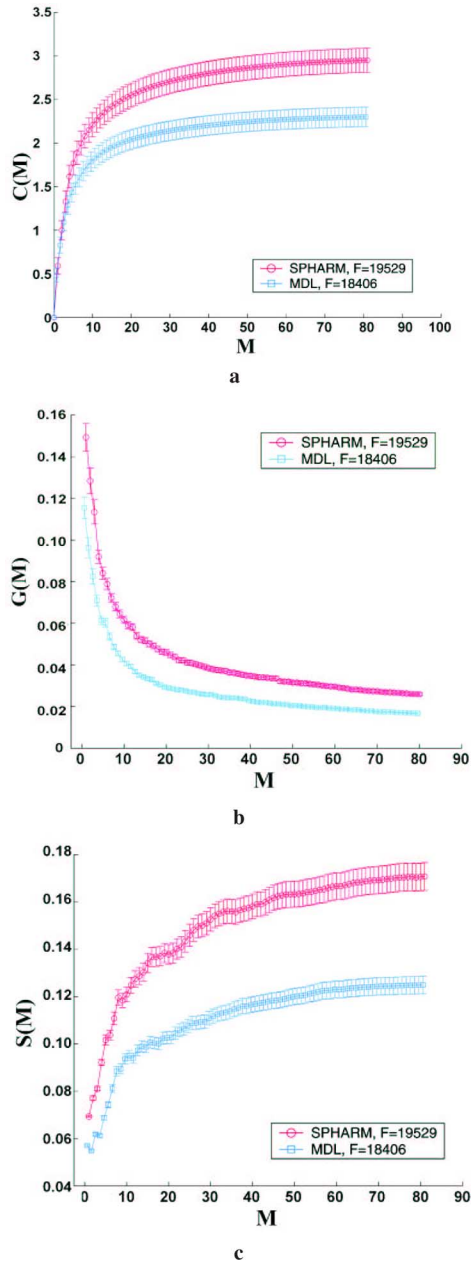


Fig. 20. Quantitative comparison of the hippocampus models. (a) Compactness. (b) Generalization ability. (c) Specificity.

we evaluate the objective function, the computational cost of the model-building algorithm would increase significantly, leading to longer convergence times.

Even the best optimization algorithm (Section III-D4) took a long time to converge; for example, using a MATLAB implementation on a low-specification PC (Intel Pentium IV—2.8 GHz, 512 Mb RAM), the ventricles (eight instances) converged after 25 min, but the hippocampi (82 examples) took over a day. Although model building is a one-off, offline process, the convergence times reported earlier are inconveniently long. If, however, the algorithm was implemented in a lower level language (e.g., C) on a modern PC, the convergence time would decrease by some orders of magnitude, allowing a large dataset (≈ 100 examples) to be optimized overnight. As the method scales approximately linearly with the number of

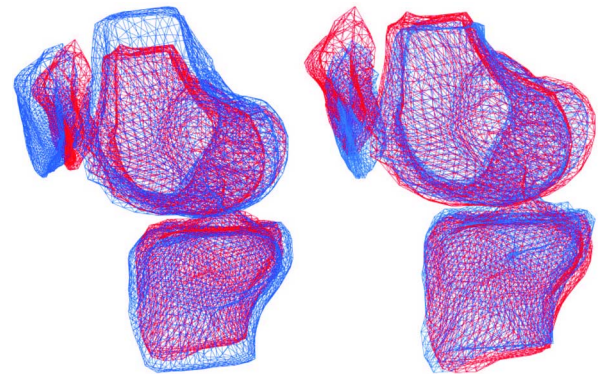


Fig. 21. First two modes of the human knee model in medio-lateral view. (Left) Two extremes of the first mode of variation ($\pm 3\sqrt{\lambda_1}$); the mode has captured relative positional variation between the three bones and a scaling of the femur and tibia. (Right) Extremes of the second mode of variation ($\pm 3\sqrt{\lambda_2}$); this mode has the effect of varying the position and shape of the patella with respect to the other two bones.

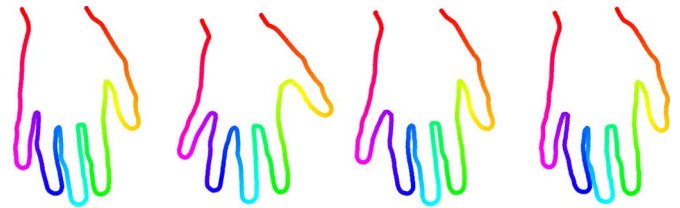


Fig. 22. Correspondence found by the MDL method for the Hand data (only some examples are shown). The color denotes the correspondence.

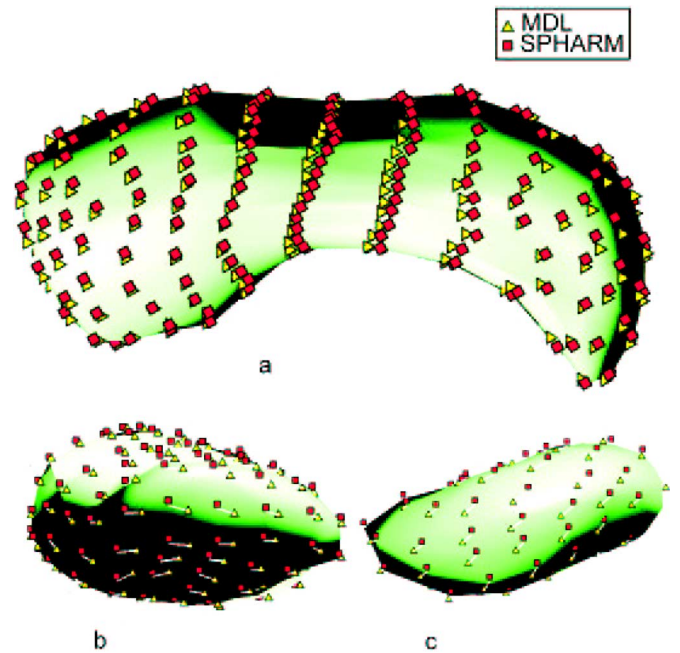


Fig. 23. Sparse subset of the SPHARM and MDL correspondences. (a) Side view. (b) Detail of head. (c) Detail of tail.

shapes, it can cope with large training sets, making it a practical method for model building.

Only the symmetric theta transformations were used in the experiments reported earlier. Although the asymmetric and shear/twist transformations (see the Appendix for details) can represent a more flexible set of re-parameterizations, they have

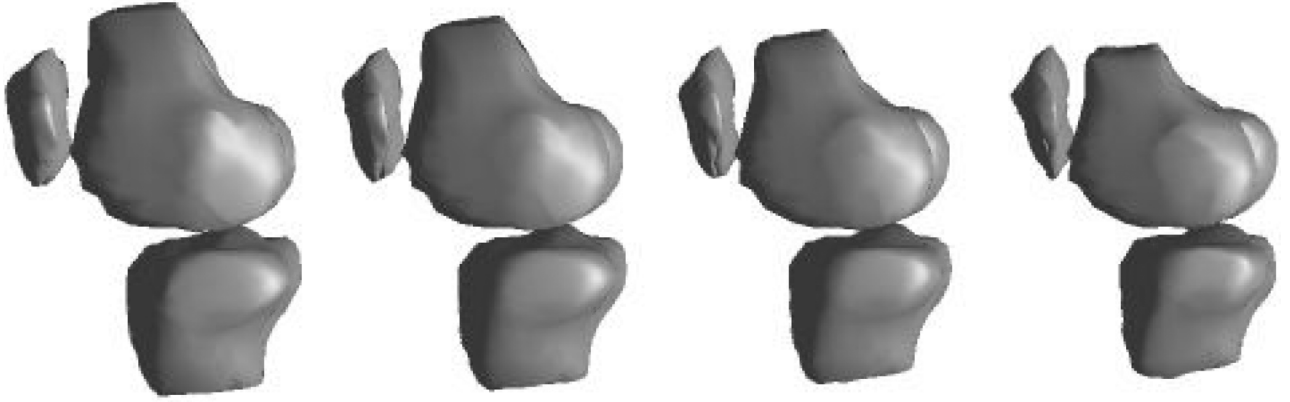


Fig. 24. First mode of the MDL human knee model. This is a medio-lateral view of the model, showing the first parameter being varied from $-3\sqrt{\lambda_1}$ (left) to $+3\sqrt{\lambda_1}$ (right). The structural variations are easier to appreciate in the wire frame view shown in Fig. 21.

more parameters to optimize, leading to longer convergence times. The asymmetric and shear/twist transformations will, however, be required for more complex shapes than those described here.

The model-building framework described here requires a parameterized transformation model for manipulating correspondence. This is straightforward for the representation used here, because the point-based representation of shape is local explicit, but finding a transformation model for implicit representations, such as M-REPS, is more difficult [32].

All objects considered in this paper had spherical topology. The extension of the method to “simple” topologies such as open surfaces is straightforward. More complex (but fixed) topologies could also be handled, provided that it is possible to find a suitable parameterization scheme (i.e., a mapping to the relevant topological primitive), as well as a set of parameterized homeomorphic mappings of that topological primitive. An example is presented in [86] where the ideas presented here are extended to produce an MDL shape model of the left side of the human heart, including the ventricle, atrium, aorta, and pulmonary veins. The case of objects with varying topology is more problematic, but mainly because the central concept of meaningful correspondence becomes difficult when considering the case of a single object that can split into two objects. But if required to do so, such a case could be handled within this general framework—for example, the correspondence between a single object with spherical topology and two such objects would mean defining a cut on the topological primitive, with each resulting hemisphere then being closed at the cut, and mapped onto a sphere.

As regards the process of reparameterization itself, we note that this has much in common with warp generation for nonrigid registration of images.⁵ B splines have been very successful in image registration [87], and as noted previously, have also been applied to reparameterizing shape [80]. This suggests that other nonrigid registration techniques, such as fluid regularization [88], might be usefully applied to the case of reparameterizing shape, and such investigations form the basis of our study.

⁵Note that we are not referring here to the approach of constructing shape models using *image* registration (e.g., [27] and [30]).

APPENDIX

FURTHER HOMEOMORPHISMS OF THE SPHERE

Although the symmetric theta transformation described in the main body of the text provides considerable representational power, here we consider a more general set of transformations.

Asymmetric Theta Transformations: In the text, we considered transformations of the form

$$\theta \mapsto f(\theta) \quad (26)$$

which mean that the transformation acts in a rotationally symmetric manner about the point $\theta = 0$ (that is, it is independent of the ψ coordinate). We can also consider more general asymmetric transformations of the θ coordinate. That is

$$\theta \mapsto f(\theta, \psi). \quad (27)$$

An asymmetric transformation around $\theta = 0$ can be achieved using the symmetric theta transformation (10) and making the amplitude A a smooth function of the ψ coordinate

$$A \rightarrow A(\psi'). \quad (28)$$

One such way to do this is to use the wrapped Cauchy distribution to obtain

$$A(\psi') = A_0 \left[\frac{1 - \beta^2}{1 + \beta^2 - 2\beta \cos \psi'} - \frac{1 - \beta^2}{(1 + \beta^2)^2} \right] \quad (29)$$

where $\beta(\beta \equiv e^{-b})$ is the width of the subsidiary Cauchy. We have chosen the formulation such that $A(\psi')$ has a minimum value of zero. An example of an asymmetric transformation is shown in Fig. 25.

Shear and Twist: We also consider transformations of the ψ coordinate. This is equivalent to shearing and twisting the sphere about the axis defined by a point P . So, for example, we could consider a reparameterization of the form

$$\psi \mapsto \psi + g(\theta) \quad (30)$$

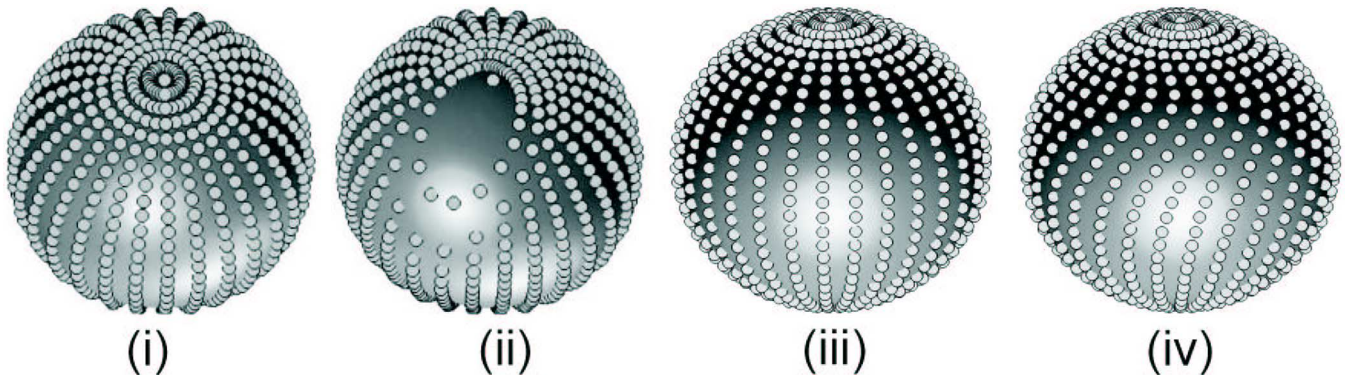


Fig. 25. (a) and (c) Original sphere from two different viewpoints. (b) Sphere after asymmetric theta transformation. (d) Sphere after shear transformation.

where

$$g(\theta) = \left(\frac{B}{2\pi} \right) \frac{1 - \gamma^2}{1 + \gamma^2 - 2\gamma \cos(\theta - \theta_0)} \quad (31)$$

with B being the amplitude, γ ($\gamma \equiv e^{-c}$, $c \in \mathbb{R}$) the width, and θ_0 the position of the center. This transformation is continuous with the identity at $B = 0$ (i.e., the transformation has no affect when $B = 0$). It can also be localized about $\theta = \theta_0$ in the limit of zero width. An example of such a transformation is shown in Fig. 25.

ACKNOWLEDGMENT

The authors would like to thanks AstraZeneca, who provided the original knee dataset, and T. Williams and C. Wolstenholme for performing the initial processing of the data. The hippocampus dataset and the SPHARM model were obtained in a study supported by the Stanley Foundation, and were kindly provided by G. Gerig, M. Styner, and co-workers from the University of North Carolina, Chapel Hill.

REFERENCES

- [1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.
- [2] A. D. Brett and C. J. Taylor, "Construction of 3D shape models of femoral articular cartilage using harmonic maps," in *Proc. Int. Conf. Med. Image Comput. Comput.-Aided Intervent. (MICCAI)*, 2000, pp. 1205–1214.
- [3] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," *Image Vis. Comput.*, vol. 12, no. 6, pp. 276–285, Jul. 1994.
- [4] S. Solloway, C. Hutchinson, J. Waterton, and C. J. Taylor, "The use of active shape models for making thickness measurements of articular cartilage from MR images," *Magn. Reson. Med.*, vol. 37, pp. 943–952, 1997.
- [5] A. L. Readhead, A. C. W. Kotcheff, C. J. Taylor, M. L. Porter, and D. W. L. Hukins, "An automated method for assessing routine radiographs of patients with total hip replacements," *Proc. Inst. Mech. Eng. H, J. Eng. Med.*, vol. 211, no. 2, pp. 145–154, 1997.
- [6] B. van Ginneken, A. F. Frangi, J. J. Stall, and B. ter Haar Romeny, "Active shape model segmentation with optimal features," *IEEE Trans. Med. Imag.*, vol. 21, no. 8, pp. 924–933, Aug. 2002.
- [7] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME—A flexible appearance modelling environment," *IEEE Trans. Med. Imag.*, vol. 22, no. 10, pp. 1319–1331, Oct. 2003.
- [8] P. Smyth, C. Taylor, and J. Adams, "Vertebral shape: Automatic measurement with active shape models," *Radiology*, vol. 211, no. 2, pp. 571–578, 1999.
- [9] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images: An improved active shape model," *IEEE Trans. Med. Imag.*, vol. 17, no. 6, pp. 1049–1067, Dec. 1998.
- [10] G. Gerig, M. Styner, M. Shenton, and J. Liberman, "Shape vs. size: Improved understanding of the morphology of brain structures," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2001, pp. 24–32.
- [11] R. H. Davies, C. J. Twining, P. D. Allen, T. F. Cootes, and C. J. Taylor, "Shape discrimination in the hippocampus using an MDL model," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 38–50.
- [12] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.
- [13] T. F. Cootes and C. J. Taylor, "Combining point distribution models with shape models based on finite-element analysis," *Image Vis. Comput.*, vol. 13, no. 5, pp. 403–409, 1995.
- [14] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor, "A minimum description length approach to statistical shape modelling," *IEEE Trans. Med. Imag.*, vol. 21, no. 5, pp. 525–537, May 2002.
- [15] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
- [16] A. Kelemen, G. Szekely, and G. Gerig, "Elastic model-based segmentation of 3D neurological data sets," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 828–839, Oct. 1999.
- [17] S. M. Pizer, D. Eberly, D. S. Fritsch, and B. S. Morse, "Zoom-invariant vision of figural shape: The mathematics of cores," *Comput. Vis. Image Understand.*, vol. 69, no. 1, pp. 055–071, 1998.
- [18] R. H. Davies, C. J. Twining, P. D. Allen, T. F. Cootes, and C. J. Taylor, "Building optimal 2D statistical shape models," *Image Vis. Comput.*, vol. 21, no. 13, pp. 1171–1182, 2003.
- [19] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor, "3D statistical shape models using direct optimisation of description length," in *Proc. 7th Eur. Conf. Comput. Vis. (ECCV)*, 2002, pp. 3–21.
- [20] C. Brechbühler, G. Gerig, and O. Kühler, "Parametrization of closed surfaces for 3D shape description," *Comput. Vis. Image Understand.*, vol. 61, no. 2, pp. 154–170, 1995.
- [21] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 11, pp. 1061–1075, Nov. 1992.
- [22] C. Brechbühler, G. Gerig, and O. Kübler, "Parameterisation of closed surfaces for 3D shape description," *Comput. Vis., Graph. Image Process.*, vol. 61, pp. 154–170, 1995.
- [23] C. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [24] G. E. Christensen, S. C. Joshi, and M. I. Miller, "Volumetric transformation of brain anatomy," *IEEE Trans. Med. Imag.*, vol. 16, no. 6, pp. 864–877, Dec. 1997.
- [25] G. Szekely and S. Lavalée, "Matching 3-D anatomical surface with non-rigid deformations using octree-splines," *Int. J. Comput. Vis.*, vol. 18, no. 2, pp. 171–186, 1996.
- [26] S. Duchesne, J. C. Pruessner, and D. L. Collins, "Appearance-based segmentation of medial temporal lobe structures," *Neuroimage*, vol. 17, pp. 515–531, 2001.
- [27] D. Rueckert, A. F. Frangi, and J. A. Schnabel, "Automatic construction of 3D statistical deformation models of the brain using non-rigid registration," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1014–1025, Aug. 2003.
- [28] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2000, pp. 316–323.

- [29] P. Golland, W. E. L. Grimson, M. E. Shenton, and R. Kikinis, "Small sample size learning for shape analysis of anatomical structures," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2000, pp. 72–82.
- [30] A. F. Frangi, D. Rueckert, J. Schnabel, and W. Niessen, "Automatic construction of multiple-object three dimensional statistical shape models: Application to cardiac modelling," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1151–1166, Sep. 2002.
- [31] N. P. M. Rousson and R. Deriche, "Implicit active shape models for 3D segmentation in MRI imaging," in *Proc. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, 1, 2004, pp. 209–216.
- [32] P. A. Yushkevich, H. Zhang, and J. C. Gee, "Parametric medial shape representation in 3-d via the poisson partial differential equation with non-linear boundary conditions," in *Proc. 20th Conf. Inf. Process. Med. Imag. (IPMI)*, 2005, pp. 162–173.
- [33] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995.
- [34] A. C. W. Kotcheff and C. J. Taylor, "Automatic construction of eigen-shape models by direct optimisation," *Med. Image Anal.*, vol. 2, no. 4, pp. 303–314, 1998.
- [35] A. Baumberg and D. Hogg, "Learning flexible models from image sequences," in *Proc. 3rd Eur. Conf. Comput. Vis. (ECCV)*, J.-O. Eklundh, Ed., Berlin, Germany, 1994, vol. 1, pp. 299–308, Springer-Verlag.
- [36] H. Lameckera, M. Seebab, H.-C. Hegea, and P. Deuhard, "A 3d statistical shape model of the pelvic bone for segmentation," in *Proc. SPIE*, 2004, pp. 1341–1351.
- [37] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig, "Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models," *Med. Image Anal.*, vol. 1, pp. 19–34, 1996.
- [38] D. Meier and E. Fisher, "Parameter space warping: Shape-based correspondence between morphologically different objects," *IEEE Trans. Med. Imag.*, vol. 21, no. 1, pp. 31–47, Jan. 2002.
- [39] M. Kaus, V. Pekar, C. Lorenz, R. Truyen, S. Lobregt, and J. Weese, "Automated 3-D PDM construction from segmented images using deformable models," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1005–1013, Aug. 2003.
- [40] C. Shelton, "Morphable surface models," *Int. J. Comput. Vis.*, vol. 38, pp. 75–91, 2000.
- [41] S. Sclaroff and A. P. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 6, pp. 545–561, Jun. 1995.
- [42] F. L. Bookstein, "Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape," *Med. Image Anal.*, vol. 1, no. 3, pp. 225–244, 1997.
- [43] A. D. Brett, A. Hill, and C. J. Taylor, "A method of automatic landmark generation for automated 3D PDM construction," *Image Vis. Comput.*, vol. 18, pp. 739–748, 2000.
- [44] Y. Wang, B. S. Peterson, and L. H. Staib, "Shape-based 3D surface correspondence using geodesics and local geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2000, pp. 644–651.
- [45] G. Subsol, J. Thirion, and N. Ayache, "A scheme for automatically building three-dimensional morphometric anatomical atlases: Application to a skull atlas," *Med. Image Anal.*, vol. 2, no. 1, pp. 37–60, 1998.
- [46] L. Garcin, A. Rangarajan, and L. Younes, "Non rigid registration of shapes via diffeomorphic point matching and clustering," in *Proc. Int. Conf. Image Process.*, 2004, pp. 3299–3302.
- [47] M. Vaillant and J. Glaunes, "Surface matching via currents," in *Inf. Process. Med. Imag. (IPMI)*, 2005, pp. 381–392.
- [48] R. Paulsen and K. Hilger, "Shape modelling using Markov random field restoration of point correspondences," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 1–12.
- [49] A. Pitot, H. Delingette, A. Toga, and P. Thompson, "Learning object correspondences with the observed transport shape measure," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 25–37.
- [50] A. Hill and C. J. Taylor, "Automatic landmark generation for point distribution models," in *Proc. 5th Br. Mach. Vis. Conf. (BMVC)*, E. Hancock, Ed., Sept. 1994, pp. 429–438, BMVA Press.
- [51] A. Baumberg and D. Hogg, "An adaptive eigenshape model," in *Proc. 6th Br. Mach. Vis. Conf. (BMVC)*, D. Pycok, Ed., Sept. 1995, pp. 87–96, BMVA Press.
- [52] A. Rangarajan, H. Chui, and F. L. Bookstein, "The softassign procrustes matching algorithm," in *Proc. 15th Conf. Inf. Process. Med. Imag. (IPMI)*, 1997, pp. 29–42.
- [53] R. H. Davies, T. F. Cootes, J. C. Waterton, and C. J. Taylor, "An efficient method for constructing optimal statistical shape models," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2001, pp. 57–65.
- [54] H. Thodberg, "MDL shape and appearance models," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 51–62.
- [55] H. Thodberg and H. Olafsdottir, "Adding curvature to MDL shape models," in *Proc. 14th Br. Mach. Vis. Conf. (BMVC)*, 2003, vol. 2, pp. 251–260.
- [56] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, "On the Laplace-Beltrami operator and brain surface flattening," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 700–711, Aug. 1999.
- [57] M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper, and D. A. Rottenberg, "Quasi-conformally flat mapping the human cerebellum," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 1999, pp. 279–286.
- [58] B. Fischla, M. I. Sereno, and A. M. Dale, "Surface-based analysis II: Inflation, flattening, and a surface-based coordinate system," *Neuroimage*, vol. 9, pp. 195–207, 1999.
- [59] K. V. Mardia, *Statistics of Directional Data*. New York: Academic, 1972.
- [60] J. R. Rissanen, "A universal prior for integers and estimation by minimum description length," *Ann. Statist.*, vol. 11, pp. 416–431, 1983.
- [61] J. R. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
- [62] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [63] T. Heimann, I. Wolf, and H. Meinzer, "Automatic generation of 3D statistical shape models with optimal landmark distributions," *Methods Inf. Med.*, vol. 46, no. 3, pp. 275–281, 2007.
- [64] I. Dryden and K. V. Mardia, *The Statistical Analysis of Shape*. London, U.K.: Wiley, 1998.
- [65] R. H. Davies, "Learning Shape: Optimal Models of Natural Variability," Ph.D. dissertation, Univ. Manchester, Manchester, U.K., 2002.
- [66] J. E. Dennis, "Non-linear least squares," in *Proc. State Art Numer. Anal.*, 1977, pp. 269–312.
- [67] B. Efron and B. Tibshirani, *Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.
- [68] M. Styner, K. Rajamani, L. Nolte, G. Zsemlye, G. Szekely, C. Taylor, and R. Davies, "Evaluation of 3D correspondence methods for model building," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 63–75.
- [69] C. J. Twining, T. F. Cootes, S. Marsland, V. Petrovic, R. Schestowitz, and C. J. Taylor, "A unified information-theoretic approach to group-wise non-rigid registration and model building," in *Proc. 20th Conf. Inf. Process. Med. Imag. (IPMI)*, 2005, pp. 1–14.
- [70] C. J. Twining and C. J. Taylor, "Specificity as a graph-based estimator of cross-entropy and KL divergence," in *Proc. 17th Br. Mach. Vis. Conf. (BMVC)*, 2006, vol. 2, pp. 459–468.
- [71] H. D. Tagare, D. O'Shea, and A. Rangarajan, "A geometric criterion for shape-based non-rigid correspondence," in *Proc. 5th Int. Conf. Comput. Vis. (ICCV)*, Jun. 1995, pp. 434–439.
- [72] J. C. Waterton, B. J. Middleton, R. Pickford, C. P. Allott, D. Checkley, and R. A. Keith, "Reduced animal use in efficacy testing in disease models with use of sequential experimental designs," *Dev. Animal Vet. Sci.*, vol. 31, pp. 737–745, 2000.
- [73] S. R. Morgan, A. F. P. Nash, J. C. Waterton, R. A. Maciewicz, J. E. Leadbetter, S. J. Gandy, R. J. Moots, P. Creamer, and A. J. Silman, "Magnetic resonance imaging measurement of knee cartilage volume in a multi-centre study," *Rheumatology*, vol. 42, pp. 1–4, 2003.
- [74] M. Styner, J. Liberman, and G. Gerig, "Boundary and medial shape analysis of the hippocampus in schizophrenia," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2003, pp. 464–471.
- [75] M. H. Chakos, S. A. Schobel, H. Gu, G. Gerig, D. Bradford, C. Charles, and J. Liberman, "Duration of illness and treatment effects on hippocampal volume in male patients with schizophrenia," *Br. J. Psychiatry*, vol. 186, pp. 26–31, 2005.
- [76] A. Ericsson and K. Åström, "Minimizing the description length using steepest descent," in *Proc. 14th Br. Mach. Vis. Conf. (BMVC)*, 2003, vol. 2, pp. 93–102.
- [77] J. Hladůvka and K. Bühler, "MDL spline models: Gradient and polynomial reparameterisations," in *Proc. 17th Br. Mach. Vis. Conf. (BMVC)*, 2006, pp. 869–878.

- [78] T. Heimann, I. Wolf, T. Williams, and H. Meinzer, "3D active shape models using gradient descent optimization of description length," in *Proc. 20th Conf. Inf. Process. Med. Imag. (IPMI)*, 2005, pp. 566–577.
- [79] T. Heimann, I. Wolf, and H.-P. Meinzer, "Active shape models for a fully automated 3D segmentation of the liver an evaluation on clinical data," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2006, vol. II, pp. 41–48.
- [80] P. Horkaew and G. Yang, "Optimal deformable surface models for 3D medical image analysis," in *Proc. 18th Conf. Inf. Process. Med. Imag. (IPMI)*, 2003, pp. 12–24.
- [81] M. Styner, S. Xu, M. El-Sayed, and G. Gerig, "Correspondence evaluation in local shape analysis and structural subdivision," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2007, pp. 1192–1195.
- [82] T. F. Cootes and C. J. Taylor, "A mixture model for representing shape variation," *Image Vis. Comput.*, vol. 17, no. 8, pp. 567–574, 1999.
- [83] A. Bell and T. Sejnowski, "An information maximisation approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.
- [84] S. Romdhani, S. Gong, and A. Psarrou, "A multi-view non-linear active shape model using kernel pca," in *Proc. 10th British Machine Vision Conference (BMVC)*, T. Pridmore and D. Elliman, Eds., Nottingham, U.K., Sep. 1999, vol. 2, pp. 483–492.
- [85] C. J. Twining and C. J. Taylor, "The use of kernel principal component analysis to model data distributions," *Pattern Recognit.*, vol. 36, pp. 217–227, 2003.
- [86] P. Horkaew and G.-Z. Yang, "Construction of 3D dynamic statistical deformable models for complex topological shapes," in *Proc. Int. Conf. Med. Image Comput. Comput. Aided Intervent. (MICCAI)*, 2004, pp. 217–224.
- [87] D. Rueckert, I. Somoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.
- [88] G. Christensen, R. Rabbitt, and M. Miller, "Deformable templates using large deformation kinematics," *IEEE Trans. Image Process.*, vol. 5, no. 10, pp. 1435–1447, Oct. 1996.