Automatic Construction of 2D Shape Models

Nicolae Duta, *Member*, *IEEE*, Anil K. Jain, *Fellow*, *IEEE*, and
Marie-Pierre Dubuisson-Jolly, *Member*, *IEEE*

Abstract—A procedure for automated $2D$ shape model design is presented. The modeling system is given a set of training example shapes defined by the coordinates of their contour points. The shapes are automatically aligned using Procrustes analysis and clustered to obtain cluster prototypes (typical objects) and statistical information about intracluster shape variation. One difference from previously reported methods is that the training set is first automatically clustered and those shapes considered to be outliers are discarded. In this way, the cluster prototypes are not distorted by outlier shapes. A second difference is in the manner in which registered sets of points are extracted from each shape contour. We propose a flexible point matching technique that takes into account both pose/scale differences as well as nonlinear shape differences between a pair of objects. The matching method is independent of the initial relative position/scale of the two objects and does not require any manually tuned parameters. Our shape model design method was used to learn 11 different shapes from contours that were manually traced in MR brain images. The resulting model was then employed to segment several MR brain images that were not included in the shape-training set. A quantitative analysis of our shape registration approach, within the main cluster of each structure, demonstrated results that compare very well to those achieved by manual registration; achieving an average registration error of about 1 pixel. Our approach can serve as a fully automated substitute to the tedious and time-consuming manual $2D$ shape registration and analysis.

Index Terms—Shape models, point correspondence, flexible registration, automatic landmarks, shape clustering.

1 Motivation

Object learning is an important problem in machine vision with direct implications on the ability of a computer to understand an image. Usually, an object is defined by its shape, color, and/or texture, and sometimes, by its relationships to other objects in the scene. A current trend in automatic image interpretation is to use model-based methods. Typically, the models are handcrafted based on the prior knowledge the user has about the object of interest. More recently, automatic model design has emerged as a powerful tool for learning object characteristics.

In this study, we will concentrate on creating $2D$ shape models: Given a number of 2D shapes, a model consists of a shape prototype along with statistical information about shape variation around the prototype (Fig. 1). Shape models are especially useful when the object of interest has a homogeneous appearance and can be distinguished from other objects mostly by its shape. One important application for shape-based object recognition is in medical image analysis. During the past decade, there has been a lot of work in shape-based approaches for the automatic segmentation of flexible structures [4], [5], [6], [7], statistical tests to differentiate between healthy and sick patients [1], and building anatomical atlases. Among the numerous shape models that have been used, the following approaches are well-known: Fourier [4], wavelet [6], and contour (eigen-shape) [5], [1]. However, regardless of the model used, the training data consists of a set of coordinates of some points along the contour of the object of interest from several images. It is usually desirable for a model to describe an *average/typical object* (prototype), to contain information about shape variation within the training set, and to be independent of the object pose. In a detailed comparison of Fourier, wavelet, and eigen-shape models, Neumann and Lorenz [6] demonstrate that if one does not separate shape information from pose or parametrization information, then the resulting model is unable to precisely describe the shape variation present in the training set. That is, the model parameters should be computed *after* the training shapes have been *aligned* in a common coordinate and parametrization frame.

Among the rich computer vision literature dealing with shapes, there are few studies that give a precise definition of the term *alignment*. However, most authors seem to implicitly agree that if $D(.,.)$ is a "distance" function between two sets of points, then a point set B is aligned to a point set A with respect to a transformation group G (e.g., rigid, similarity, linear, affine) if $D(A, B)$ cannot be further decreased by applying to B a transformation from G. The main difference between various alignment approaches is in the distance function used: Huttenlocher et al. [8] use the Hausdorff distance, Sclaroff and Pentland [9] use "strain energy," Ton and Jain [10] use "support functions," and Horn [11], Besl and McKay [12], Gold et al. [13], and the statistical shape community [2] use a least-squares type

- *N. Duta and A.K. Jain are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48823. E-mail: {dutanico, jain}@cse.msu.edu.*
- *M.-P. Dubuisson-Jolly is with the Imaging and Visualization Department, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540. E-mail: jolly@scr.siemens.com.*

Manuscript received 12 July 1999; revised 6 June 2000; accepted 28 June 2000. Recommended for acceptance by J.R. Beveridge.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110217.

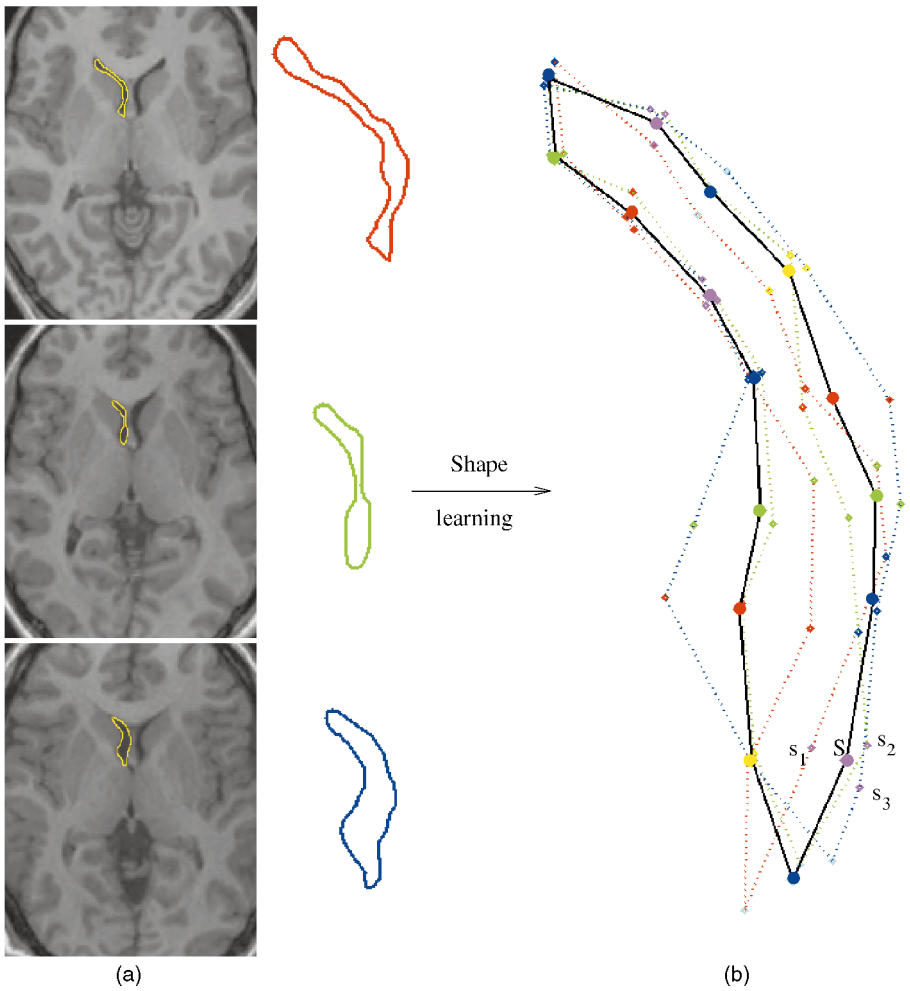


Fig. 1. Learning the shape of the right ventricle from MR brain images. (a) Manual tracing of the right ventricle performed by a neuroanatomist on three different patients. (b) A ventricle model consists of a shape prototype (drawn in black) along with statistical information about shape variation. The prototype vertices (drawn as colored circles) have been obtained by averaging the coordinates of the corresponding vertices on the three ventricles (drawn as colored diamonds) *after they have been aligned in a common coordinate frame* (e.g., vertex S is the average of s_1, s_2, and s_3). The three aligned ventricle shapes are shown in dotted red, green, and blue lines. This method for obtaining a shape prototype is called *Procrustes analysis* [1], [2], [3]. Note that, in order for this method to work, one needs to extract sets of corresponding points of equal cardinality (in this case, 16) from the three ventricle shapes. The shape variance is given by the 32×32 covariance matrix of the (x, y) coordinates of the vertices on the three ventricle shapes after alignment.

(Procrustes[1]) distance. Other differences are the types of transformations allowed and whether point correspondences are established during the alignment process. We are not aware of any comparative study that reports alignment results on a common data set for various distance functions and neither of a common quantitative evaluation criterion. Therefore, except for some theoretical assessments, it is difficult to claim that one alignment method dominates another for a given practical problem.

We use a least-squares type (Procrustes) distance whose choice was motivated by the following facts: 1) It provides a convenient way to compute a prototype (average shape) from a set of simultaneously aligned shapes (Procrustes analysis [3]), 2) once the point correspondences are found, there exists an analytical (exact) solution to the alignment problem [11], [15], and 3) it has been used quite frequently in medical image analysis [1], [5], [15], [13]. Unfortunately, least-squares alignment methods do not deal with parametrization and are usually applied to sets of *corresponding* points.[2] In practice, such sets of points have been obtained by a painstakingly manual inspection of the data of interest. This fact is illustrated in Fig. 2, where an expert defined the points (called pseudolandmarks) that characterize each of the three

1. Procrustes was a villainous son of Poseidon in Greek mythology who robbed travelers on the road from Eulesis to Athens. He offered travelers a room for the night and fit them into his bed by stretching them if they were too short or cutting off their legs if they were too tall (Webster's dictionary). Procrustes analysis compares the differences in shape between two point sets by transforming one point set in order to match the other. The transformations allowed in a standard analysis are the *similarity* (*shape-preserving*) ones: scale changes, rotations, and translations. One can regard by analogy, one point set as the bed and the other as the person being "translated," "rotated," and "rescaled" so as to fit as close as possible to the bed. After one of the point sets has been transformed to match the other, the sum of squared differences of the coordinates between them is called the *Procrustes distance*, while the shape instance defined by the average of their coordinates is called *Procrustes average shape* [1], [2], [3], [14].

2. An approach that unifies the correspondence estimation with the alignment problem for sets of points with unknown correspondences (called the *Softassign Procrustes Matching Algorithm*) was presented in [16].

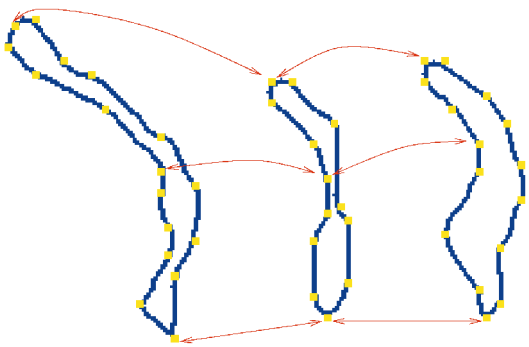


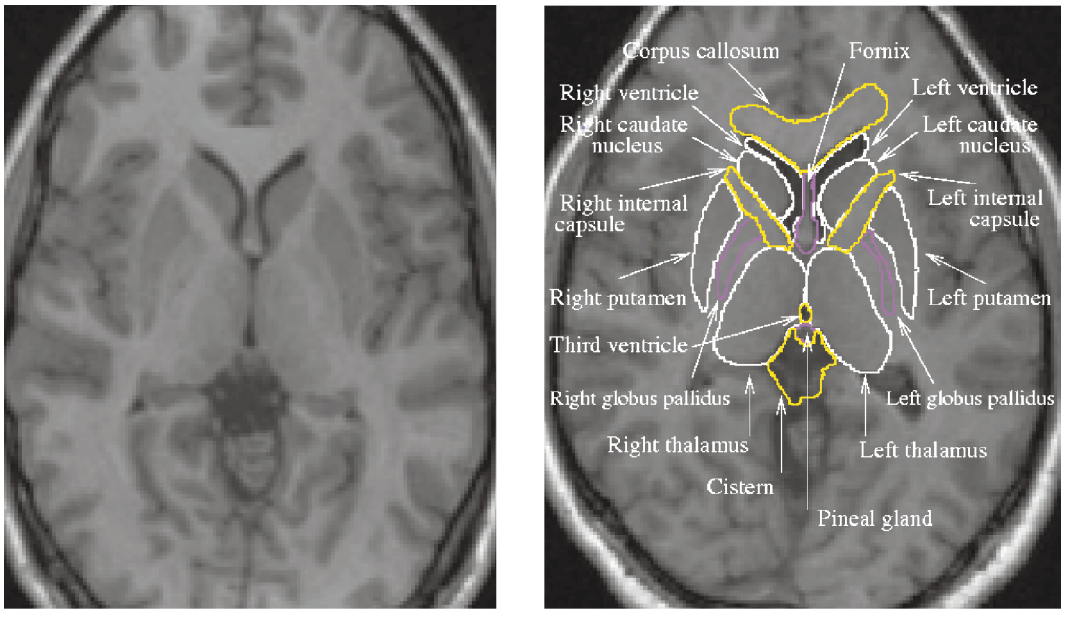
Fig. 2. Expert-defined pseudolandmarks (yellow squares) on the three shapes in Fig. 1 along with some obvious point correspondences (red arrows). Note that, if the pseudolandmarks are defined on each shape independently (as it was the case here) then, in most cases, it is very difficult to find corresponding points on other shapes.

ventricle shapes in Fig. 1. When these points are defined independently for each shape, it may be very difficult to exactly define point correspondences in the absence of anatomical landmarks (points for which correspondences can be defined based on prior knowledge). Other problems are human bias and lack of reproducibility, that is, different persons may extract different numbers of pseudolandmarks and even specify point correspondences in a different way.

There have been only a few attempts to automate the shape alignment/averaging process in the least-squares framework: Bookstein [1] used thin-plate splines, Hill et al. [17] used polygonal matching (subsequently extended to $3D$ by Brett and Taylor [18]), and Davatzikos et al. [19] used curvature registration on outlines produced by an active contour approach. In the thin-plate spline approach, the shape registration-reparametrization is only implicit and not completely automatic. Polygonal matching is based on the assumption that arc path-lengths between consecutive points are equal, which may be violated in case of severe shape differences. As pointed out by several studies, curvature is a rigid invariant of shape and its applicability is limited in case of nonlinear shape distortions. None of these methods attempt to reject a training shape if it is significantly different from the majority in the training set.

We present an alternate solution to the problem of shape reparametrization-alignment-averaging problem. One difference from most previous methods is that it does not necessarily compute a single average shape from the given training set, but rather it detects shape clusters in the data and provides a shape average and *variation* for each cluster. The idea of detecting shape clusters in the training set has been employed by Cootes and Taylor [15] and by Gold et al. [20]. However, in [15], the clusters were not detected in the initial shape space but in the parameter space obtained by Principal Component Analysis, while in [20], clustering is formulated as a complex optimization problem whose solution is computationally expensive.

Formally, we are attempting to solve the following problem: Given a set of m shape instances $S_k = \{(x_i^k, y_i^k)\}_{i=1..n_k}^{k=1..m}$ represented by a set of boundary points (shape S_k represented by n_k boundary points), partition it into a set of clusters and, for each shape cluster, compute a *prototype* (mean shape). The set of prototypes will be used as models for detection of object instances in new images by means of the deformable template segmentation. Working with average templates learned from examples results in a faster and more reliable segmentation. As a direct application of our method, we construct shape models for 11 different structures in *MR* brain images and demonstrate how the learned shapes facilitate subsequent image segmentation. Fig. 3 shows an example of an MR brain image taken in the coronal plane. Several neuroanatomic structures of interest whose contours were identified by a neuroanatomist are shown in Fig. 3b. Note that parts of the



(a) (b)

Fig. 3. Magnetic resonance image of the human brain, imaged in the coronal plane with in-slice resolution of 256×256 pixels. (a) Original image (cropped to show only the brain region). (b) Neuroanatomic structures of interest whose contours were identified by a neuroanatomist.

contours of some structures cannot be distinguished from the background based on gray level information alone. In such cases, only a model-based segmentation that uses prior knowledge about the shape of a structure of interest is able to identify the structures [7].

2 Shape Model Design

In this section, we first provide a short introduction to the shape related terminology that will be used throughout the paper. Then, we present an outline of the proposed shape model construction method and the ideas that stand behind it.

An *n-point* shape instance

$$B = \{s_i^B\}_{i=1..n} = \{(x_i^B, y_i^B)\}_{i=1..n}$$

is said to be *aligned* to $A = \{(x_i^A, y_i^A)\}_{i=1..n}$ if the *sum-of-squares*

$$SS(A,B) = \sum_{i=1}^{n}\left[\left(x_i^A - x_i^B\right)^2 + \left(y_i^A - y_i^B\right)^2\right]$$

cannot be decreased by scaling, rotating, or translating B. In this case, the quantity $SS(A,B)/n$ is called the *Mean Alignment Error* $(MAE(A,B))$. Practical algorithms for shape alignment can be found in [15], [3]. In general, the alignment procedure is not symmetric and, if $|A| = |B| \leq 2$, A and B can be aligned exactly (the alignment error is 0).

The *Procrustes average* of a set of shapes $\{A_k\}_{k=1..m}$ is a shape instance near the center of the empirical distribution of A_ks in the shape space defined by the coordinates of the contour points. The computation of Procrustes average is graphically illustrated in Fig. 1b. From the three ventricle shapes (dense sets of $2D$ points), sparser subsets of corresponding points (in this case, 16 points) were extracted. These subsets are aligned into a common coordinate frame and their coordinates are averaged in order to obtain the Procrustes average shape. For a detailed definition, properties, and methods of computing an average shape, see [1], [15], [3].

Let $A = \{(x_j^A, y_j^A)\}_{j=1..p}$ and $B = \{(x_k^B, y_k^B)\}_{k=1..r}$ be two shape instances defined by p and r contour points, respectively. A *match matrix* $M = \{M_{j,k}\}_{k=1..r}^{j=1..p}$ [13] is defined by:

$$M_{j,k} = \begin{cases} 1 & \text{if point } A_j \text{ corresponds to point } B_k \\ 0 & \text{otherwise.} \end{cases}$$

We consider 0-1 match matrices M corresponding to symmetric one-to-one links (point correspondences); that is, a point $A_j \in A$ can have at most one corresponding point $B_k \in B$, in which case, the correspondence is symmetric. The points from both sets that have no correspondence are called *outliers*. We denote by A_M and B_M the subsets of A and B matched by M and by $MAE(M) = MAE(A_M, B_M)$. Note that the definition of a match matrix M *does not assume the existence of a similarity transformation that would overlay the subsets* A_M *and* B_M *exactly* (this would be equivalent to $MAE(A_M, B_M) = 0$). However, in practice, it makes sense to use a match matrix M only when the distances between corresponding points matched by M are reasonably small. For a discussion of this issue, see Section 6.

The outline of our shape model construction method is as follows (see, also, Fig. 4):

Algorithm 1: Shape Model Design Outline

Input: a set of m shape instances $S_1, \ldots, S_m$, each represented by a sequence of boundary points.

1. *Polygonal approximation*: For each shape S_k in the training set, compute a polygonal approximation S'_k.
2. *Global and local similarity registration*: For each $j, k = 1..m$, perform a flexible one-to-one registration (mapping) of S'_k to S_j. If the registration succeeds, define $T_{j,k}$ as the subset of S_j that corresponds (was matched) to the points of S'_k, otherwise, set $T_{j,k} = \emptyset$.
3. *Intershape distance matrix computation*: Compute a *pairwise mean alignment error* matrix $\mathcal{D} = \{d_{j,k}\}_{j,k=1..m}$, where $d_{j,k} = MAE(T_{j,k}, S'_k)$ if $T_{j,k} \neq \emptyset$ or $d_{j,k} = \infty$, otherwise.
4. *Shape clustering and prototype computation*: Set the current training set equal to the original set of m shapes: $CTS = \{S_k\}_{k=1..m}$. While $CTS \neq \emptyset$ do
   a. Find the shape approximation S'_{i_0} that has the least average distance to the shapes $S_j \in CTS$ (the *best fit shape* to the current training set).
   b. Extract from CTS and put in a cluster all the shapes $S_{i_1}, .., S_{i_p}$ to which S'_{i_0} could be fit (see Sections 4 and 6.3).
   c. The cluster prototype is defined as the *Procrustes average* of $T_{i_1,i_0}, \ldots, T_{i_p,i_0}$. The shape variance inside the cluster is defined as the covariance matrix of the aligned sets $\{T_{i_k,i_0}\}_{k=1..p}$. The size of the covariance matrix is equal to twice the number of points of S'_{i_0}.

Step 1 of *Algorithm 1* finds a polygonal approximation of each shape using a method described in [21]. The distance between consecutive vertices of the polygonal approximations is about 2-3 pixels in order to smooth small shape artifacts, noise, and quantization effects. The polygonal approximation is only used to extract subsets of *corresponding* points from the *original* shapes. This makes the registration task *easier* and *implicitly* brings together the extracted subsets into a *common parametrization frame*. Indeed, if a point s_{i_0} on a polygonal approximation S' is registered to $s_{i_1} \in S_1$, $s_{i_2} \in S_2, \ldots, s_{i_m} \in S_m$ ($S_1, \ldots, S_m$ are original shapes that form a cluster) then, by transitivity, $s_{i_1}, s_{i_2}, .., s_{i_m}$ are correspondents on $S_1, .., S_m$ of *one vertex* of an average shape (the idea of registration by transitivity was previously used by Sclaroff and Pentland [9], though, in a different context). For example, in Fig. 1b, s_1, s_2, and s_3 are the corresponding vertices on the three example shapes of the prototype vertex S. It is not advisable to register two polygonal approximations because one can dilute information about the original point variation and encounter contention problems caused by the one-to-one mapping requirement. On the other hand, one should not attempt to directly register pairs of original shapes since it is more difficult to define and register local topological neighborhoods (see Section 3) because of the local noise (jagged contours) and point contention. By using a smoother and sparser approximation of a shape, we increase the likelihood that *every* point on it will eventually have a correspondent. If, after registration, there exists a point on S'_k that does not have a correspondent on S_j then we say that the registration between S'_k and S_j has failed and we set $d_{j,k} = \infty$.

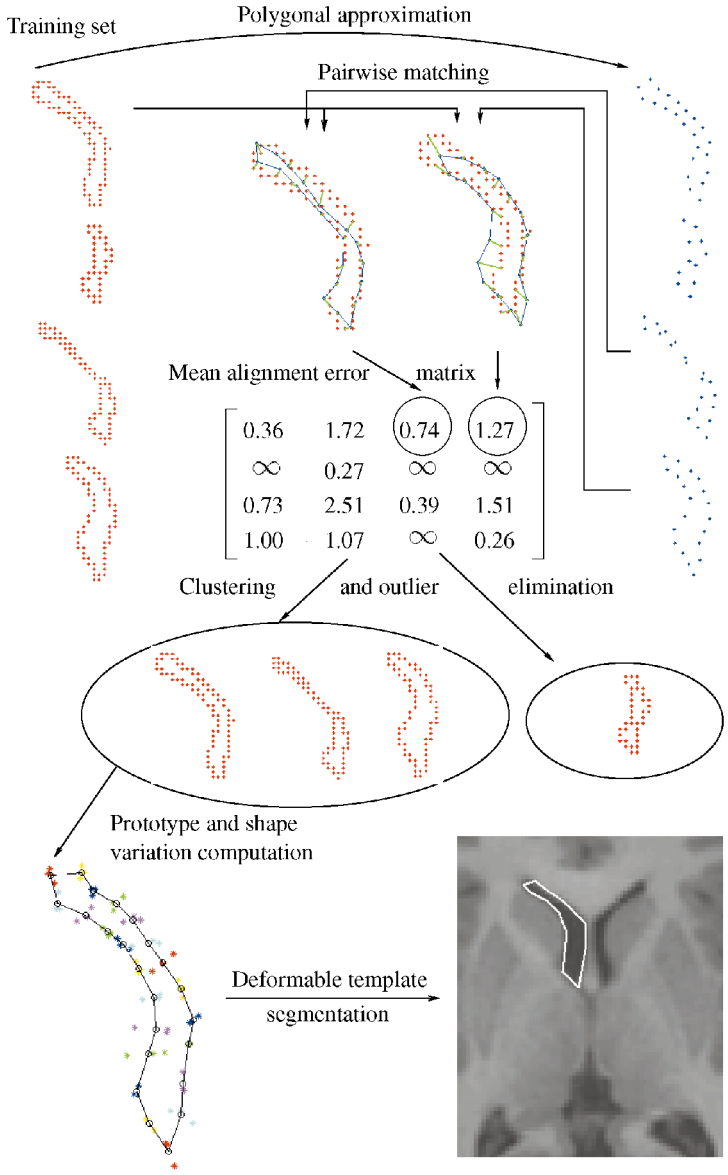


Fig. 4. The shape learning method (Algorithm 1).

In order to ensure that the shape variation present in the original data is *preserved*, one needs a precise automatic registration method. Hill et al. [17] reported that the mean-square-error (from a *ground truth*) of their registration method is about twice as large as the error in manual registration. They also reported that other methods were even less robust. Therefore, we have decided to combine several ideas from the literature [9], [13], [22], [23], [24] with new ideas in order to obtain a more precise registration method. Section 3 describes the shape registration process (Step 2), Section 4 describes the shape clustering and prototype computation mechanism (Step 4), while Section 5 shows an application of our model learning approach to automatic segmentation of MR images of the brain.

3 Shape Registration

Our shape registration method consists of two stages: 1) the global similarity registration of two arbitrary sets of points and 2) a nonlinear registration based on local similarity of two curves (ordered sets of points).

The global registration method attempts to find a similarity transformation corresponding to a one-to-one mapping of a subset A' of a shape instance A onto a subset B' of a shape instance B. This mapping is required to simultaneously fulfill two contradicting requirements: 1) the size n of the matched subsets is as large as possible and 2) the *mean-alignment-error* between A' and B' is as small as possible.

There is a plethora of point-based registration methods, see [25] for a recent survey. As noted in [13], when solving for best alignment transformation and its associated match matrix (set of point correspondences), one has to search one of the two complementary spaces: either the space of correspondence functions which is finite but exponential in the number of points from the two sets, e.g., as in Softassign [13] (and its recent $3D$ extension [26]), or the space of similarity transformations which is infinite, and can be only partially explored, e.g., as in ICP [12] or Chamfer Matching

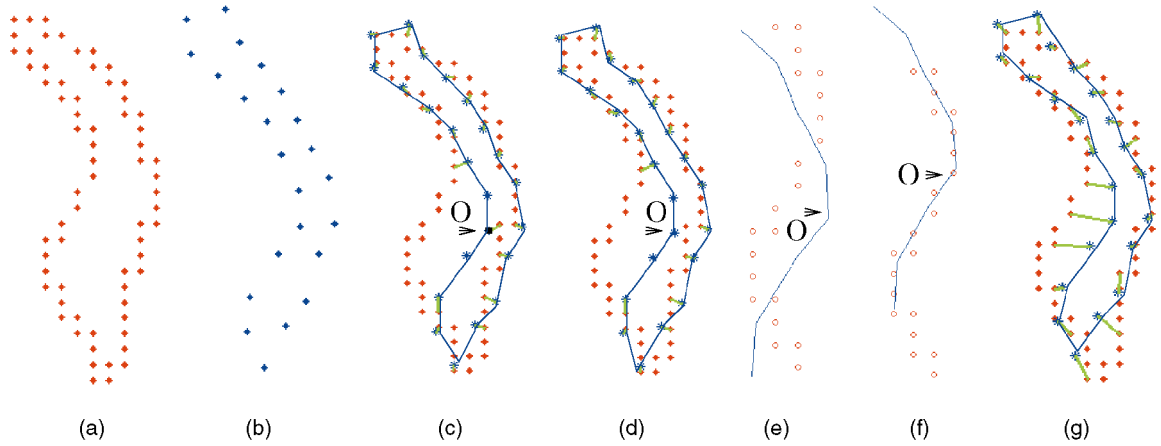


Fig. 5. The flexible registration of a shape approximation (b) to an original shape (a), with point correspondences drawn in green. (c) Global similarity registration—Algorithm 2. (d) Monotonic registration obtained from (c) by point reordering and inversion elimination (the point labeled O causes an inversion): Steps 1-3 of Algorithm 3. (e) Topological neighborhood corresponding to point O: Step 4a of Algorithm 3. (f) Similarity registration of the two topological neighborhoods in (e). (g) Final flexible registration.

[27]. We propose a registration procedure (Algorithm 2) based on a polynomial quasi-exhaustive exploration of the correspondence functions (match matrices) space. Its main novelty compared to techniques previously used in the literature [13], [22], [23], [24], [28] is the way it resolves the *shrinking* effect [24]: An unconstrained linear registration of two sets of points tends to "shrink" one set with respect to the other since, theoretically, the "best" alignment is obtained when one point set is rescaled to become a single point. Our problem formulation requires a small MAE between the two chosen subsets, using as many point correspondences as possible. Unfortunately, if we have less than three correspondences, the MAE is 0 and this should be compensated for. Therefore, we want to explicitly specify in the search criterion that a q percent increase in MAE with a p percent increase in the number of correspondences is accepted as long as no individual distance between a pair of corresponding points exceeds a given threshold. One of the simplest functionals that captures this trade-off is the ratio between a compensated MAE and the number of correspondences:

$$f(M) = [MAE(M) + K]/n, \tag{1}$$

where K is a constant depending on the percentages p, q, and the scale of the object (see Section 6 for the properties of this functional and how to choose K). If we also impose the constraint that the mapping is one-to-one, we implicitly solve the shrinking problem. With a large number of one-to-one correspondences (and the assumption that the two shapes are not sampled at very different rates), there can be no shrinking of one shape with respect to the other.

Algorithm 2: Global Similarity Registration

1. Set $V_{min} = \infty$.
2. For every pair of points $(a_{j1}, a_{j2}) \in A \times A$
   For every pair of points $(b_{k1}, b_{k2}) \in B \times B$, do Steps i through v
   i. Find the similarity transformation ψ that aligns the sets $\{a_{j1}, a_{j2}\}$ and $\{b_{k1}, b_{k2}\}$.
   ii. Apply ψ to all the points in B to obtain B'.
   iii. For every point b_k of B', find its nearest neighbor $NN(b_k)$ in A. If the distance between b_k and $NN(b_k)$ is smaller than a threshold T then set a correspondence between the two points. A match matrix M between A and B is constructed in this way. Since two points from B' can have the same nearest neighbor in A, we enforce the one-to-one correspondence requirement, that is, allow a point to be linked to its second to fifth nearest neighbor if the first one can be assigned to a closer point in B', and the length of the link does not exceed T. Recompute the transformation ψ that aligns the sets A and B according to the match matrix M.
   iv. Compute $f(M)$.
   v. If $f(M) < V_{min}$ then $V_{min} = f(M)$, $\psi_{min} = \psi$.
3. Apply ψ_{min} to all the points in B to obtain B'.
4. For every point b_k of B', find its nearest neighbor in A. If the distance between b_k and its nearest neighbor is smaller than T then set a correspondence between the two. A match matrix M' between A and B is constructed in this way and enforced to correspond to one-to-one links.
5. Find the linear transformation ψ_{final} that aligns the corresponding sets $A_{M'}$ and $B_{M'}$.

Fig. 5c shows an example of similarity registration of a ventricle shape approximation to a full ventricle shape. Although globally the registration is quite good, some point correspondences are wrong (e.g., the point labeled O) or have been missed (the two neighboring vertices of O).

We are interested not only in computing an average shape (which is robust to slight misregistrations) but also the shape variation present in the data set which is best described by the set of high curvature points. Since a *global* linear registration does not necessarily perform a good local registration (see [24] and Fig. 5c), we need to locally refine the results of the global registration such that the corresponding points of high curvature from the two data sets are matched together. However, some high-curvature points in A may not correspond to high curvature points in B, therefore, we do not enforce this requirement explicitly,

but rather through *local similarity registration* and *monotonicity*. We define the term "local" in a topological sense according to the natural point ordering along curves defined by A and B. A good registration should be *monotonic*, that is, preserve the topologies (point ordering) on the two shapes. A registration of two curves (sets of points along the contour) A and B (which can be regarded as a partial function f from A to B) is called *monotonic* if:

1. The points of A are cyclicly reordered such that point a_1 corresponds to point b_1.
2. There are no *inversions*, that is, if a_i and a_j correspond to b_k and b_l (in this order) and $i < j$ then $k < l$ (for example, an inversion is caused by point O in Fig. 5c).

Note that a monotonic registration of two sets of distinct points is one-to-one, therefore, we can perform a local registration and obtain one-to-one links by looking for a monotonic registration.

Algorithm 3: Monotonic, Local Similarity-Based Registration

Input: two sets of points A and B and a set $\mathcal{M}$ of one-to-one links between a subset A' of A and a subset B' of B obtained by global similarity registration.

1. Cyclicly reorder the points of A and B and the links in $\mathcal{M}$ such that a_1 corresponds to b_1.
2. If the number of inversions exceeds $|\mathcal{M}|/2$, reverse the ordering of the points in A.
3. Break the smallest number of links in $\mathcal{M}$ such that there are no more inversions (Fig. 5d). Note that we are left with a monotonic registration.
4. For $i = 1..|B|$ do
   a. Find a topological neighborhood of b_i, $[b_l, b_{l+1}, \ldots, b_i, \ldots b_{r-1}, b_r]$ (the actual size of the neighborhood depends on the curvature at b_i; the larger the curvature, the smaller the neighborhood) such that both b_l and b_r have correspondences in A, let them be $a_{l'}$ and $a_{r'}$ with $l' < r'$ (Fig. 5e).
   b. Perform a similarity registration between the sets $[a_{l'}, a_{l'+1}, .., a_{r'}]$ and $[b_l, b_{l+1}, .., b_r]$ (Fig. 5f).
   c. If B_i is linked to a different point in A than it was before, then record this change in $\mathcal{M}$.
5. Break the smallest number of links in $\mathcal{M}$ such that there are no more inversions.

4 Shape Clustering and Prototype Computation

Since the objects we deal with are complex manifolds of different dimensionality (depending on the number of points along the contour), shape clustering proved to be a difficult problem. We are aware of only two types of approaches to shape clustering: 1) projecting all shape instances to a common subspace (e.g., PCA, Fourier, modal space, etc.) and treating the projection coordinates as points in an Euclidean space [15] and 2) specifying a "distance" between two shapes and performing a distance-based clustering [20]. As such, shape clustering reduces to the general clustering problem for which numerous solutions have been proposed. One limitation of projection-based methods is that they exhibit an inherent loss of information due to the fact that the projection transformation is not one-to-one. That is, one point in the transformed space may correspond to several shapes whose visual appearance may be quite different (see the discussion in [7]).

In this study, we propose to solve the shape parameterization/correspondence/clustering problem in a unitary framework based on the MAE distance. The third step of Algorithm 1 defines a pseudodistance matrix $\mathcal{D}$ of *mean alignment errors* between a *polygonal approximation of a shape* and an *original shape* from the training set. A natural way for obtaining shape clusters based on $\mathcal{D}$, which is also helpful for cluster prototype (average) computation is a greedy, divide-and-conquer strategy related to *histogram mode seeking*, [29]:

1. Find a seed which is closest to the data (analogue of a mode of a histogram). This is done in Step 4a of Algorithm 1 by finding the *shape approximation* S'_{i_0} that *best fits* the current training set (based on the average distance to the rest of the shapes). S'_{i_0} will be used as a reference set for extracting corresponding sets of points of the same size from as many training shapes as possible, as discussed in Section 2.
2. Find all shapes S_j in the current training set that "fit" (are close enough to) S'_{i_0} (Step 4b of Algorithm 1). These shapes are removed from the training set and will form a new cluster (analogue to finding the two valleys adjacent to the histogram mode and grouping in one cluster all points between the valleys). For the mathematical details of what we mean by "*close enough*," see Section 6.3 We mention that it is very important that the cluster seed fits the shapes in the cluster as well as possible. Though there is no theoretical guarantee that all the points of the seed are perfectly registered to the shapes in the cluster, a poor fit can be due to one of the following two causes: either the training shapes are very different or the registration results are not accurate.

This cluster extraction procedure continues using a smaller current training set until all shapes have been assigned to a cluster. The clustering process makes only one pass through the data since each cluster starts from a shape (seed) which is close to the cluster center. We noticed that this seed is very similar to the cluster average, therefore, we do not believe that adding more passes (cluster reassignments) would modify the cluster membership (at least for well-separated clusters).

For each cluster, the cluster prototype is defined as the *Procrustes Average* of the subsets of registered points extracted from each shape in the cluster. The cluster variation is defined as the $2n \times 2n$ covariance matrix of the subsets of points used to compute the prototype (n is the number of points on the cluster prototype). This variation can be used by a segmentation method to reject shape deformations that have not been seen in the training set (see [5], [7]).

5 Experimental Results[3]

The method presented above was employed to design a shape model for 11 brain structures and its performance was assessed by a quantitative comparison to a "ground truth" model obtained manually. The training set consisted of observer-defined contours identified by a neuroanatomist in 28 individual T1-weighted contiguous MR images of the human brain, imaged in the coronal plane with in-slice resolution of 256×256 pixels. Fig. 5 shows the following five registration stages: global similarity registration—Algorithm 2 (Fig. 5c), point reordering and inversion elimination —Algorithm 3.1-3.3 (Fig. 5d), defining topological neighborhoods in Algorithm 3.4a (Fig. 5e) and their registration (Fig. 5f) and final registration (Fig. 5g). Fig. 6 shows the original manual tracings and clustering results for three structures, together with the *best fit shape* registration to the main cluster (the sets $T_{i_1,i_0}, \ldots, T_{i_p,i_0}$, as defined in Algorithm 1). The main cluster is drawn using multicolor dots, while the secondary clusters (which can be considered as outlier shapes) are drawn in red, green, and magenta. To emphasize the role of a good registration for extraction of $\{T_{i_k,i_0}\}_{k=1..p}$, we drew consecutive points on T_{i_k,i_0} using different colors, the same color for corresponding points on each shape. For example, corresponding bottom points on each right ventricle are drawn in red. Fig. 7 shows the main cluster prototypes for the 11 structures with the aligned shape examples overlaid. Consecutive point clouds are drawn in different colors to show that the clouds are nonoverlapping; the registration appears to be very precise.

In order to obtain a quantitative validation of our results, we used the method employed in [17]. From each structure prototype, we manually selected several points which we considered most important in defining its shape (points with the highest curvature) and we manually registered them to the training images. We defined the *ground truth* position of these points as the Procrustes average of the manually registered points (these point positions are shown for the right ventricle and globus pallidus in Fig. 8 as black circles). We computed and compared the *root-mean-square* (rms) distance of manually placed points from the ground truth and the rms distance of the automatically registered points from this ground truth, respectively. The rms distances for the right ventricle and globus pallidus are also shown in Fig. 8: for every point selected on each shape, each distance is displayed on the same y coordinate as the ground truth point it corresponds to. The average rms distances for the selected points are similar, though, on individual points they may be quite different. The very high curvature points (the extreme upper or lower points) are somewhat better registered manually while the intermediate points are better placed automatically. This was expected, since it is very difficult for a human to exactly place a point (identify landmarks) if there are no curvature or other anatomical cues. The average rms error for each of the 11 structures is between 0.7-1.2 pixels. For some structures, the automatic method produced a slightly smaller rms distance than the manual one, while for others the rms of the automatic method is slightly larger.

3. Due to space limitations, here, we present results from only one application, namely, model design for anatomic structures in coronal MR brain images. However, the method has successfully been employed in several application domains other than medical imaging. For a complete set of applications and results, see [30], [31], [32].

We employed the automatically extracted prototypes to train a knowledge-based Point Distribution Model (PDM) [5], [7] and segment the 11 neuroanatomical structures. Note that Algorithm 1 not only gives an average shape and variation for each cluster, but also provides the registration of the cluster prototype to all the shapes in the cluster. Therefore, PDM training becomes very fast and does not require any kind of human intervention once the shapes in the training set have been clustered.

The segmentation procedure is a top-down search based on the model fitting strategy. At each step of the fitting process, several model location hypotheses are considered and evaluated. Also, an outlier detection and replacement procedure has been developed to detect misplaced points and infer their new positions. The outlier detection improves the robustness and accuracy of the shape model fitting process. The searching procedure consists of the following steps [7]:

1. Model fitting using similarity transforms,
2. model fitting using piecewise similarity transforms,
3. outlier removal,
4. final point adjustment, and
5. final outlier removal.

We show the results of the automatic segmentation on four different MR images in Fig. 9. Since the automatically derived models are almost identical to the manually traced models, the segmentation results are as good as those reported in [7]: an average boundary positioning error of 0.8 ± 0.1 pixels with respect to the manual tracings, and maximum boundary positioning error of 4.3 ± 1.2 pixels.

6 Discussion

We will provide a brief discussion on the choice of parameters, criterion function, and the computational complexity of our algorithm:

6.1 Upper Bound on the Distance between Corresponding Points

Based on several thousand registration experiments using various real data sets for which there exists a ground truth match matrix, we found that *almost all* the distances between valid corresponding points are smaller than 10 percent of the *object scale*. The *scale* of a shape instance $A = \{(x_i, y_i)\}_{i=1..n}$ is defined as[4]

4. This definition of scale is different from that employed by most previous studies [1], [2], [3], [14], where $Scale(A) = \sqrt{\sum_{i=1}^{n}\left((x_i - \bar{x})^2 + (y_i - \bar{y})^2\right)}$ since we wanted it to be independent of the number of points and the sampling rate. Consider, for example, two circles centered at the origin of radius 1 and 0.001, respectively. Our definition of scale does not depend on whether the point set consists of exactly 10 points on the outer circle and 1,000 on the inner circle or vice versa. By contrast, the Euclidean norm-based definition of scale is highly dependent on the sampling rate which, for some applications, may be an undesirable property.

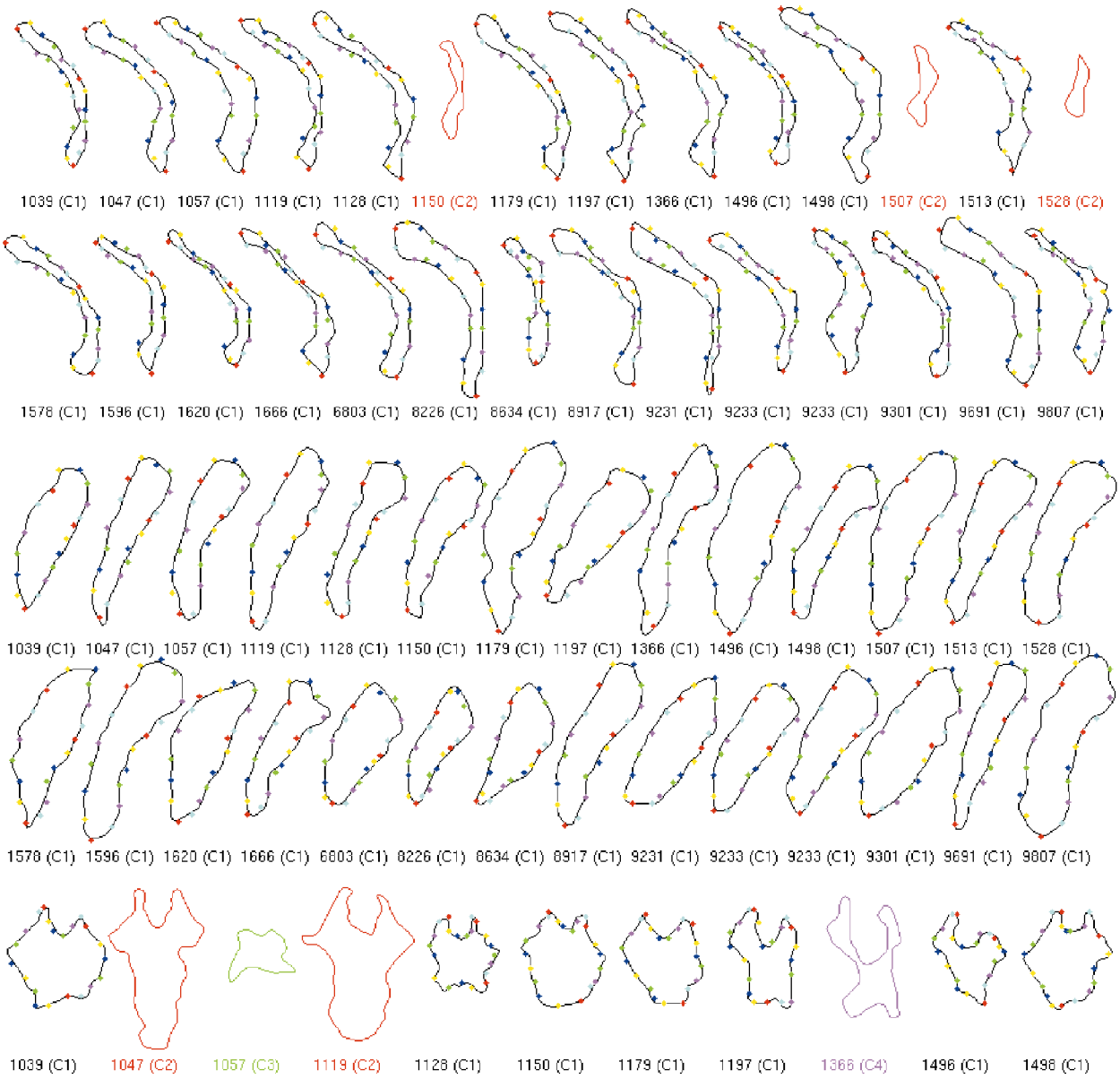


Fig. 6. Two training sets of 28 right ventricular (rows 1 and 2) and 28 globular shapes (rows 3 and 4) and a set of 11 cistern shapes (row 5) from different patients were automatically divided into clusters (main cluster (C1) drawn using multicolor dots and secondary clusters drawn in red, green, and magenta). The registration of the *best fit shape* ($\#1,047$ for ventricles, $\#8,917$ for globus pallidus, and $\#1,179$ for cisterns) to cluster C1 is overlaid as sets of colored points; corresponding points on different shapes are drawn using the same color. For example, corresponding bottom points on each right ventricle are drawn in red.

$$Scale(A) = \sqrt{\left(\max_{i=1..n}(x_i) - \min_{i=1..n}(x_i)+1\right)\cdot\left(\max_{i=1..n}(y_i) - \min_{i=1..n}(y_i)+1\right)}.$$

We enforce this *upper bound* when we set a point correspondence in Step 2.3 of Algorithm 2. In practice, if there exists a proper matching between two sets, most of the links are actually shorter than half of this upper bound. We also found it to be convenient to work in scale independence mode, therefore, before registering two point sets, we rescaled them such that the largest one has a scale equal to 10. In this case, the threshold T in Step 2.3 of Algorithm 2 is set to 1.

6.2 Properties of the Evaluation Function $f(M)$ (3)

Since the distance between two corresponding points is not larger than 1 (see above), we have that $K/n \leq f(M) \leq (K+1)/n$, $\forall$ match matrix M and $\forall$ n. If a match matrix M' has p percent more links than M, one has

$$K/[(1+p/100)n] \leq f(M') \leq (K+1)/[(1+p/100)n]$$

and M' is preferred to M if $(K+1)/[(1+p/100)n] < K/n \Leftrightarrow Kp > 100$. On the other hand, the choice between an n-link matrix and one with less than $(1+p/100)n$ links (if $p < 100/K$) is determined by the q percent increase in the MAE that we are willing to accept: The p percent increase in the number of links will be accepted iff

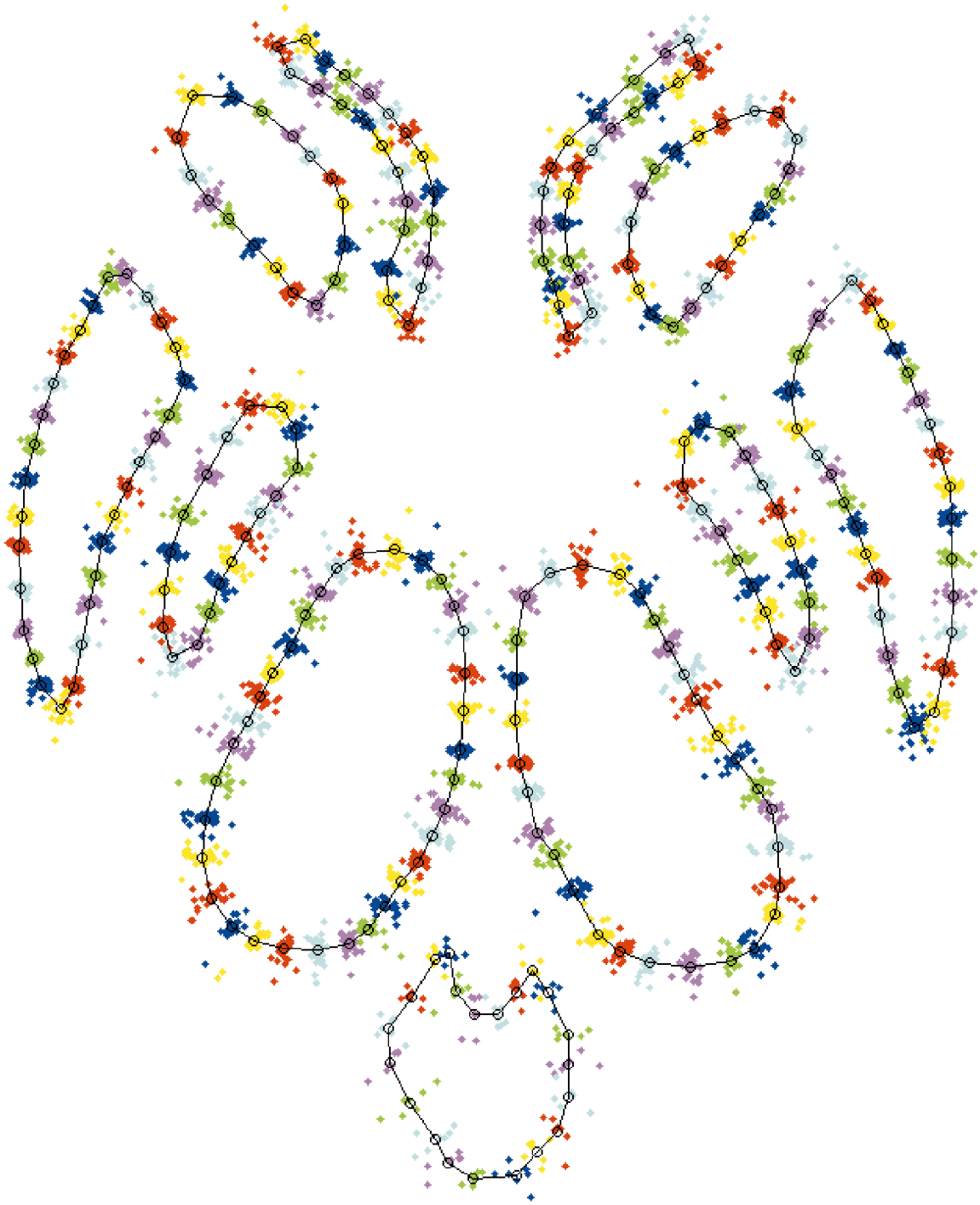


Fig. 7. Procrustes averages (prototypes) of the shapes in the main clusters corresponding to 11 brain structures with the aligned shape examples overlaid. The clouds of consecutive points are drawn in different colors to show the accuracy of the registration.

$$K/n + MAE(M)/n \geq K/[(1+p/100)n] + (1+q/100)MAE(M)/[(1+p/100)n] \Leftrightarrow q \leq p + Kp/MAE(M).$$

Since $MAE(M) \leq 1$, one has that, if $q \leq p(1+K)$ then $q \leq p + Kp/MAE(M)$. In particular, if $K = 2$, a 50 percent increase in the number of links will always be accepted no matter what the increase of the MAE is, while a 25 percent is accepted if the MAE increase is less than 75 percent (this does not mean that if $q > 75$ percent then, M' is always rejected; in practice, it might be accepted until q becomes about 100 percent).

6.3 Constructing a Shape Cluster (Step 4 of Algorithm 1)

One should note that a match matrix M obtained by global similarity registration satisfies $MAE(M) \leq 10\% Scale$ and this does not usually hold after the local registration. After

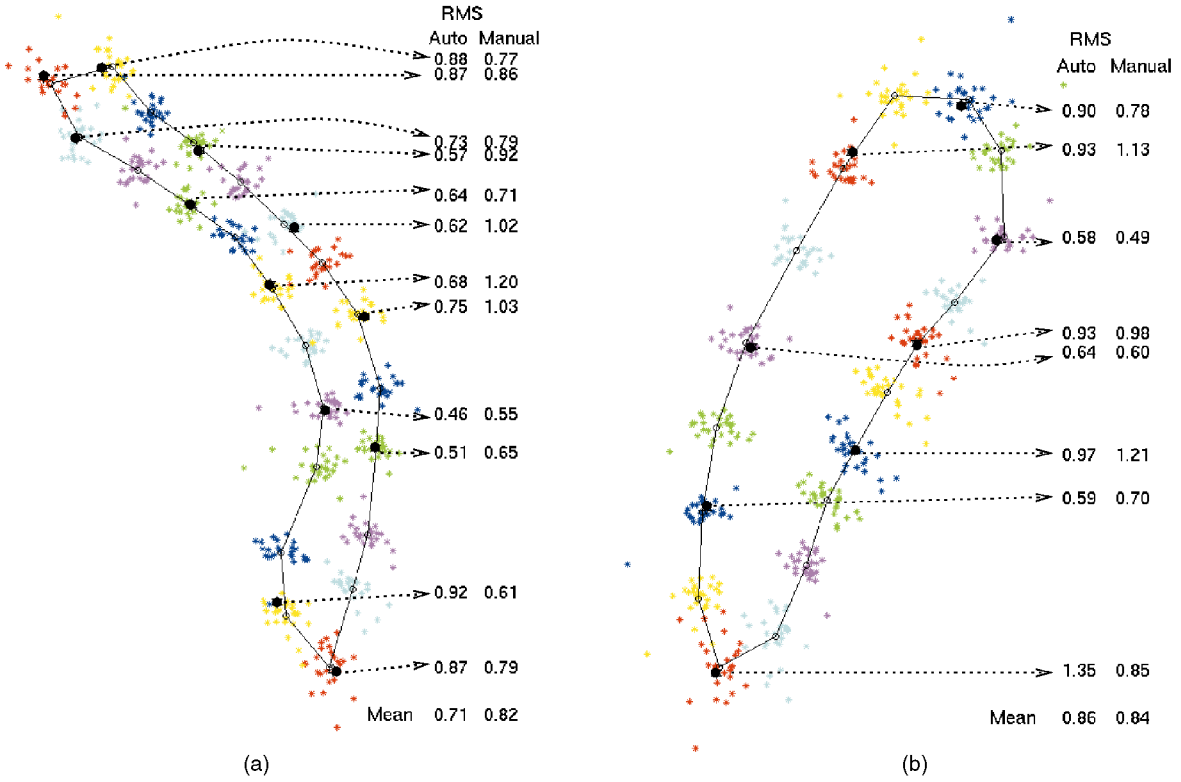


Fig. 8. Prototype of the 25 right-ventricle shapes in the main cluster (a) and prototype of the 28 right-globus pallidus shapes (b) with the aligned shape examples overlaid. The *ground truth* position for several points are shown using black circles. For each such point, we also show the *rms error* of the manual and automatic registrations.

performing all 8,687 possible pairwise registrations (between examples of the same structure), we found that in almost all the cases, a $MAE \geq 15\%Scale$ corresponds to a wrong matching. This is due to either large shape differences or cases when least-squares is not a good matching philosophy (for example, the hand shapes in [17], or handwritten characters). Our method always found a good least-squares matching if it was present. For the shape learning application reported here, we expected to find one main shape cluster and a few other small clusters with outlier shapes. Therefore, when computing the best fit to the training set in Step 4a, we did not compute the average *Mean Alignment Error* to all the shapes but only to the closest 70 percent (expecting that the furthest 30 percent might be outliers). On the other hand, two shapes were considered to be "*close enough*" in Step 4b if their $MAE \leq 15\%Scale$. We would also like to mention that the manner in which all the above parameters are chosen seems to be application independent. We obtained similar results for learning shapes coming from different applications like fingerprint matching or hand shape-based person verification, with no changes in the implementation.

6.4 Time Complexity

There are two aspects regarding the time complexity of our method. The first one concerns the shape registration complexity (Algorithm 2). It is easy to see that, if the two sets to be registered have n points each, then the complexity of Algorithm 2 is $O(n^5)$. This can be reduced to $O(n^4)$ if one data set is formed of ordered, relatively evenly sampled points along a continuous curve (and even further reduced to $O(n^3)$ if both data sets satisfy this property) and one does not expect large scale differences between the two objects or in the percentage of outliers found in the two data sets. The heuristic is based on the observation that an initial 2-point pairing hypothesis in which the two points from A (or those from B) are very close together produces a worse estimate of the transformation between the two sets than a hypothesis where the selected points are far away. In this case, we could form the initial hypothesis based only on the "diagonal" of the data set(s): $(s_i, s_{i+|A|/2})$. Though it might be argued that this complexity is still high, for applications using 2D contours extracted from images that usually have less than a few hundred points, it is sufficiently fast and gives the result in order of seconds (our $O(n^4)$ implementation needs about 10 seconds to register a 30-point approximation to a 100-point original contour on a Sun Ultrasparc (296 MHz processor)).

The second complexity aspect concerns the shape model construction itself. Algorithm 1 performs m^2 pairwise matchings, where m is the number of shapes in the training set. We mention that it is neither possible nor necessary to apply it *directly* to a large shape set. The idea behind trying all m^2 matchings is to find a shape in the training set that fits well with the remaining shape examples, and compute a mean shape based on it. The resulting mean shape is sufficiently smooth (see Fig. 7) and retains enough shape characteristics so that it can be used for direct matching to new shape examples. Therefore, we only need to apply the

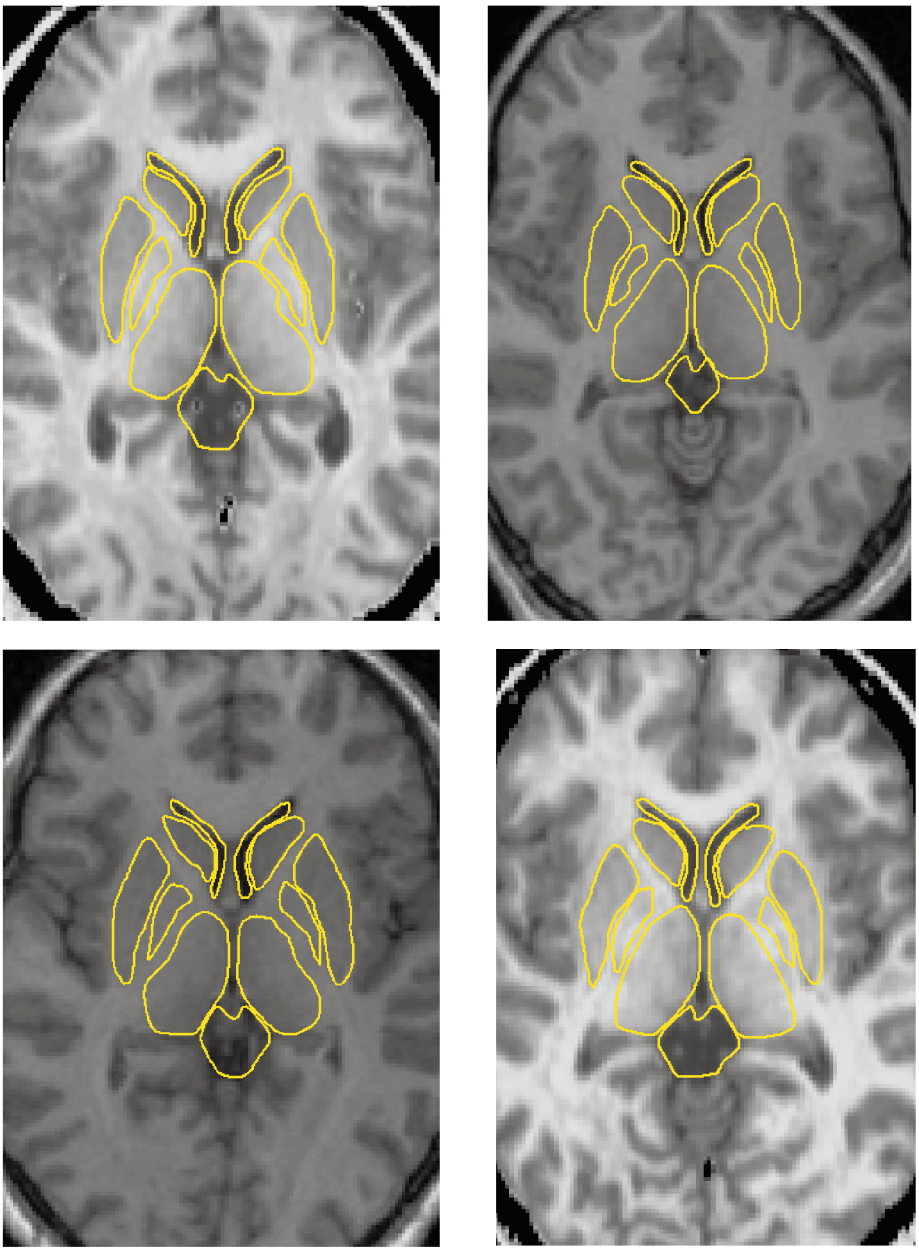


Fig. 9. Automatic segmentation of 11 brain structures in four different MR images using the constructed models.

quadratic pair-wise matching to a relatively small training set (a few tens of elements). After that, one can classify the remaining shapes only by registration to the estimates of the current clusters prototypes. Each time a shape does not fit any of the current prototypes, a new cluster is started. In this way, the learning process becomes incremental and linear in the size of the training set.

7 Conclusions

We have presented a fully automated method for constructing $2D$ shape models. Our approach is based on simultaneously solving the reparameterization/alignment/clustering problem for a set of training shapes and performing a Procrustes analysis on each cluster to obtain a cluster prototype and information about shape variation within each cluster. No prior information about the relative pose/parameterization of the shapes in the training set is necessary. As experimental results, we showed how to identify clusters and compute prototypes for 11 different neuroanatomic brain structures from training sets of 28 manual tracings per structure in MR images. However, our method has a larger range of applicability (see [30], [31], [32] and http://web.cse.msu.edu/~dutanico for more applications). It can also be employed for designing shape models for open contours [32] as well as for sets of points containing no connectivity information (e.g., fingerprint minutiae or palm feature points). In the latter case, only Algorithm 2 is applied for data registration, and the amount of nonlinear deformation that can be handled is smaller. In general, there is a trade-off (regulated by the threshold T described in Section 6.1) between the amount of nonlinear deformation accepted and the proportion of outlier points present in the data sets. If one does not expect to have many

outliers, then the threshold T can be increased and more nonlinear deformation (equivalent to a larger average distance between corresponding points) is accepted.

Algorithm 2 may fail to find the real point matchings between two point sets in cases where the proportion of outliers in at least one of the sets is very large (e.g., when matching one object outline to the edge points in a cluttered scene) and no additional information (constraints on the allowed similarity transformations or edge direction at each point) is used. However, this is not a failure of the search method in finding an "optimal" matching according to the optimality criterion in (1), but rather a failure of the data points in satisfying this optimality criterion. Since we used Algorithm 2 as a filter (declaring failure whenever the proportion of outliers resulted was high and stopping further processing), we have not noticed any failure of Algorithm 3 neither in the current application nor in [30], [31]. In the current application, Algorithm 4 used the assumption that we expect to have one main cluster along with some outlier shapes. This seems to be a reasonable assumption for shapes which describe natural variations across different subjects in medical or biometric images. However, we do not consider that this algorithm will necessarily fail on true multimodal distributions. If the 70 percent threshold defined in Section 6.3 is decreased, we believe that the method could accommodate multimodal distributions as well. We agree though, that this problem (along with that of whether a shape instance does indeed belong to a shape class) has to be further investigated.

Finally, the main use of the algorithms described in this paper was to train PDM models and perform an active shape type segmentation on images not included in the training set. The similarity registration described in Algorithm 2 could also be used alone for a *warping*-based type of segmentation [31] as well as for shape classification based on the MAE distance [32]. In conclusion, we believe that our approach can serve as a fully valid automated substitute to the tedious and time-consuming manual shape analysis.

ACKNOWLEDGMENTS

The authors would like express their thanks to Nancy Andreasen, MD, PhD, for providing both original and manually-traced MR brain data utilized in this study. We would also like to thank Professor Mario Figueiredo for providing us the spline approximation code and Professor Milan Sonka for helpful comments on the manuscript. This work was supported by a grant from Siemens Corporate Research, Princeton.

REFERENCES

[1] F.L. Bookstein "Landmark Methods for Forms without Landmarks: Morphometrics of Group Differences in Outline Shape," *Medical Image Analysis,* vol. 1, no. 3, pp. 225-244, 1997.

[2] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *J. Royal Statistical Soc. B,* vol. 53, no. 2, pp. 285-339, 1991.

[3] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis.* New York: Wiley, 1998.

[4] L.H. Staib and J.S. Duncan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 11, pp. 1061-1075, Nov. 1992.

[5] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam, "Use of Active Shape Models for Locating Structures in Medical Images," *Image & Vision Computing,* vol. 12, no. 6, pp. 355-366, 1994.

[6] A. Neumann and C. Lorenz, "Statistical Shape Model-Based Segmentation of Medical Images," *Computerized Medical Imaging and Graphics,* vol. 22, pp. 133-143, 1998.

[7] N. Duta and M. Sonka, "Segmentation and Interpretation of MR Brain Images: An Improved Active Shape Model," *IEEE Trans. Medical Imaging,* vol. 17, no. 6, pp. 1049-1062, 1998.

[8] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.

[9] S. Sclaroff and A. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 6, pp. 545-561, June 1995.

[10] J. Ton and A.K. Jain, "Registering Landsat Images by Point Matching," *IEEE Trans. Geoscience Remote Sensing,* vol. 27, no. 5, pp. 642-651, 1989.

[11] B.K.P. Horn, "Closed Form Solution of Absolute Orientation Using Unit Quaternions," *J. Optical Soc. Am. A,* vol. 4, pp. 629-642, 1987.

[12] P. Besl and N. McKay, "A Method for Registration of 3D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 239-256, Feb. 1992.

[13] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness, "New Algorithms for 2D and 3D Point Matching," *Pattern Recognition,* vol. 31, no. 8, pp. 1019-1031, 1998.

[14] C.G. Small, *The Statistical Theory of Shape.* Berlin: Springer-Verlag, 1996.

[15] T.F. Cootes and C.J. Taylor, "A Mixture Model for Representing Shape Variation," *Proc. British Machine Vision Conf.,* pp. 110-119, 1997.

[16] A. Rangarajan, H. Chui, and F. Bookstein, "The Softassign Procrustes Matching Algorithm," *Proc. Information Processing in Medical Imaging '97,* pp. 29-42, 1997.

[17] A. Hill, C.J. Taylor, and A.D. Brett, "A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 3, pp. 241-251, 2000.

[18] A.D. Brett and C.J. Taylor, "A Framework for Automated Landmark Generation for Automated 3D Statistical Model Construction," *Proc. Information Processing in Medical Imaging '99,* pp. 376-381, 1999.

[19] C. Davatzikos, M. Vaillant, S.M. Resnick, J.L. Prince, S. Letovsky, and R.N. Bryan, "A Computerized Approach for Morphological Analysis of the Corpus Callosum," *J. Computer Assisted Tomography,* vol. 20, pp. 88-97, 1996.

[20] S. Gold, A. Rangarajan, and E. Mjolsness, "Learning with Preknowledge: Clustering with Point and Graph Matching Distance Measures," *Neural Computation,* vol. 8, no. 4, pp. 787-804, 1996.

[21] M. Figueiredo, J. Leitao, and A.K. Jain, "Unsupervised Contour-Representation and Estimation Using B-Splines and a Minimum-Description Length Criterion," *IEEE Trans. Image Processing,* 2000.

[22] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.

[23] A. Gueziec and N. Ayache, "Smoothing and Matching of 3D Space Curves," *Int'l J. Computer Vision,* vol. 12, pp. 79-104, 1994.

[24] J. Feldmar and N. Ayache, "Rigid, Affine, and Locally Affine Registration of Free-Form Surfaces," *Int'l J. Computer Vision,* vol. 18, pp. 99-119, 1996.

[25] J.B. Maintz and M.A. Viergever, "A Survey of Medical Image Registration," *Medical Image Analysis,* vol. 2, no. 1, pp. 1-36, 1998.

[26] H. Chui, J. Rambo, R. Schultz, and A. Rangarajan, "Registration of Cortical Anatomical Structures with Robust 3D Point Matching," *Proc. Information Processing in Medical Imaging '99,* pp. 168-181, 1999.

[27] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, pp. 849-865, 1988.

[28] S. Ullman, "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition,* vol. 32, no. 3, pp. 193-254, 1989.

[29] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision, Volume I.* Reading, Mass.: Addison-Wesley, 1992.

[30] R. Fisker, N. Schultz, N. Duta, and J. Carstensen, "A General Scheme for Training and Optimization of the Grenander Deformable Template Model," *Proc. Computer Vision and Pattern Recognition*, 2000.
[31] A. Lundervold, N. Duta, T. Taxt, and A.K. Jain, "Model-Guided Segmentation of Corpus Callosum in MR Images," *Proc. Computer Vision and Pattern Recognition '99,* pp. 231-237, 1999.
[32] A.K. Jain and N. Duta, "Deformable Matching of Hand Shapes for User Verification," *Proc. Int'l Conf. Image Processing '99,* 1999.

Nicolae Duta received the BS degree in applied mathematics from the University of Bucharest (Romania) in 1991, the DEA degree in stochastic modeling from the University of Paris-Sud (France) in 1992, the MS degree in computer science from the University of Iowa in 1996, and the PhD degree in computer science and engineering with a concentration in computer vision and pattern recognition from Michigan State University in 2000. He is a staff scientist at BBN Technologies, GTE Internetworking, Cambridge, Massachusetts. He held temporary research positions at INRIA-Rocquecourt (France) in 1993 and Siemens Corporate Research (Princeton, New Jersey) from 1997 to 1999. He is a member of IEEE and ACM and his current research interests include computer vision, pattern recognition, machine, and biological learning.

Anil Jain is an university distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include statistical pattern recognition, deformable models, texture analysis, document image analysis, fingerprint matching, and 3D object recognition. He received the best paper awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997 from the Pattern Recognition Society. He also received the 1996 *IEEE Transactions on Neural Networks* Outstanding Paper Award. He was the Editor-in-Chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1990-94). He is the co-author of *Algorithms for Clustering Data*, Prentice-Hall, 1988, has edited the book *Real-Time Object Measurement and Classification*, Springer-Verlag, 1988, and co-edited the books, *Analysis and Interpretation of Range Images*, Springer-Verlag, 1989, *Markov Random Fields*, Academic Press, 1992, *Artificial Neural Networks and Pattern Recognition*, Elsevier, 1993, *3D Object Recognition*, Elsevier, 1993, and *BIOMETRICS: Personal Identification in Networked Society*, Kluwer in 1999. He is a fellow of the IEEE and IAPR. He received a Fulbright research award in 1998.

Marie-Pierre Dubuisson-Jolly received the BS degree in computer science from the Universite de Technologie de Compiegne, France in 1989. She received the MS and PhD degrees in computer science from Michigan State University in 1991 and 1995, respectively. Since 1995, Dr. Dubuisson-Jolly has been a researcher in the Imaging and Visualization Department at Siemens Corporate Research in Princeton, New Jersey. Her research interests include image segmentation, deformable templates and active contours, and medical image analysis. She is member of the IEEE.