# Learning 2D Shape Models

Nicolae Duta[1], Anil K. Jain[1], and Marie-Pierre Dubuisson-Jolly[2]
[1]Department of Computer Science and Engineering, Michigan State University
{dutanico, jain}@cse.msu.edu, http://web.cse.msu.edu/~dutanico
[2]Imaging and Visualization Department, Siemens Corporate Research, Princeton

## Abstract

*A new fully automated shape learning method is presented. It is based on clustering a set of training shapes in the original shape space (defined by the coordinates of the contour points) and performing a Procrustes analysis on each cluster to obtain cluster prototypes and information about shape variation. The main difference from previously reported methods is that the training set is first automatically clustered and those shapes considered to be outliers are discarded. The second difference is in the manner in which registered sets of points are extracted from each shape contour. As a direct application of our shape learning method, an 11-structure shape model of brain substructures was extracted from MR image data, an eigen-shape model was automatically trained, and employed to segment several MR brain images not present in the shape-training set. A quantitative analysis of our shape registration approach, within the main cluster of each structure, shows that our results compare very well to those achieved by manual registration; achieving an average rms error of about 1 pixel. Our approach can serve as a fully automated substitute to the tedious and time-consuming manual shape registration and analysis.*

## I. Motivation

Object learning is an important problem in machine vision with direct implications on the ability of a computer to understand an image. 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 learning $2D$ shape models. 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 automatic segmentation of flexible structures [3, 14, 16], statistical tests to differentiate between healthy and sick patients [2] and building anatomical atlases. Among the numerous shape models that have been used, the following are well known: Fourier [16], wavelet [14] and contour (eigen-shape) [2,3]. 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 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 [14] 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.

Most authors in the rich computer vision literature dealing with shapes 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 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. [12] use the Hausdorff distance, Sclaroff and Pentland [15] use "strain energy", and Horn [11], Besl and McKay [1], Gold et al. [8] and the statistical shape community [9] use a least-squares type (Procrustes) distance. Other differences are the types of transformations allowed and whether point correspondences are set during the alignment process.

We use a least-squares type (Procrustes) distance whose choice was motivated by the following facts: (i) it provides a convenient way to compute a prototype (average shape) from a set of simultaneously aligned shapes (Procrustes analysis [5]), (ii) once the point correspondences are found, there exists an analytical (exact) solution to the alignment problem [4, 11] and (iii) it has been quite frequently used in medical image analysis [2–4, 8]. Unfortunately, least-squares alignment methods do not deal with parametrization, and can be applied only to sets of *correspondent* points. In practice, such sets of points have been obtained by a painstakingly manual inspection of the data of interest. Even so, it is very difficult for a human to exactly define point correspondences in the absence of anatomical landmarks. The main problems are human bias and lack of reproducibility. According to our knowledge, there have been only two attempts to automate the shape alignment/averaging process in the least-squares framework: Bookstein [2] used thin-plate

0-7695-0149-4/99 $10.00 © 1999 IEEE

splines and Hill et al. [10] used polygon matching. In the thin-plate spline approach, the shape registration-reparametrization is only implicit and not completely automatic, while the second approach is based on the assumption of equal arc path-lengths between consecutive points which may be violated in case of severe shape differences. 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. Its main conceptual difference from previous methods is that it doesn't necessarily compute a single average from the given shape training set, but it rather detects shape clusters in the data and provides a shape average and *variation* for each cluster.

Mathematically speaking, 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, 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 deformable template segmentation. Working with average templates learned from examples results in a faster and more reliable segmentation.

## II. Shape learning method

A shape instance $B = \{s_i^B\}_{i=1..n} = \{(x_i^B, y_i^B)\}_{i=1..n}$ is called *aligned* to $A$ 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 *Mean Alignment Error* ($MAE(A, B)$). Practical algorithms for shape alignment can be found in [4, 5]. 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 $A_k$'s in the shape space defined by the coordinates of the contour points. For a detailed definition, properties and methods of computing an average shape, see [2, 4, 5].

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. A *match matrix* $M = \{M_{j,k}\}_{k=1..r}^{j=1..p}$ [8] 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 match matrix $M$ and by $MAE(M) = MAE(A_M, B_M)$.

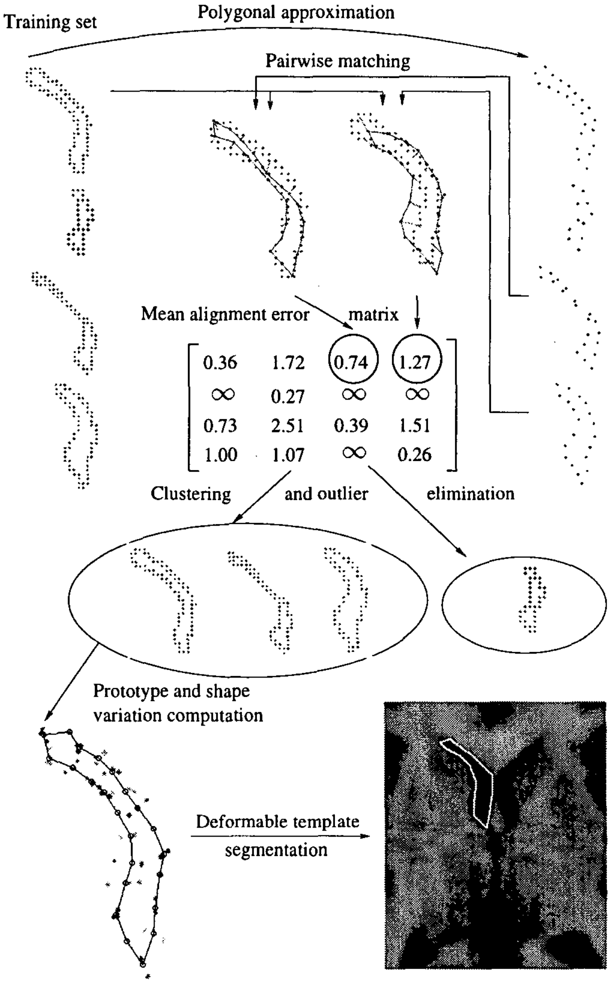
Training set
Polygonal approximation
Pairwise matching
Mean alignment error matrix
0.36 1.72 0.74 1.27
∞ 0.27 ∞ ∞
0.73 2.51 0.39 1.51
1.00 1.07 ∞ 0.26
Clustering and outlier elimination
Prototype and shape variation computation
Deformable template segmentation

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

The outline of our shape learning method is as follows (see also Fig. 1):

*Algorithm 1: Shape Learning Outline*

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. *Inter-shape 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 ($d_{i_k,i_0} < \infty$).

(c) The cluster prototype is defined as the *Procrustes average* of $T_{i_1,i_0},..., T_{i_p,i_0}$. The shape variance inside the cluster is defined as the covariance matrix of the aligned sets $\{T_{i_k,i_0}\}_{k=1..p}$.

Step 1 of the learning algorithm finds a polygonal approximation of each shape using a method described in [7]. These approximations are sampled at a rate of about 2-3 pixels in order to smooth small shape artifacts, noise and quantization effects and are only used to extract subsets of *corresponding* points from the *original* shapes, providing an *easier* task for the registration algorithm and *implicitly* bringing together the extracted subsets into a *common parametrization frame.* Indeed, if a point $s_{i_0}$ on a polygonal approximation $S'$ is registered to $s_{i_1} \in S_1$, $s_{i_2} \in S_2$, ..., $s_{i_m} \in S_m$ ($S_1,...,S_m$ are original shapes that form a cluster), then one can say that, 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 [15], though in a different context). It is not advisable to register two polygonal approximations because one can loose 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 considerably increase the chances 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 $S_j$ failed and set $d_{j,k} = \infty$.

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. [10] reported a mean-square-error from a *ground truth* about twice as large for their automatic method compared to manual registration. They also reported that other methods were even less robust. Therefore, we decided to combine several ideas from the existing literature [6,8] 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 mechanisms (Step 4), while section 5 shows an application of the model learned to automatic brain segmentation.

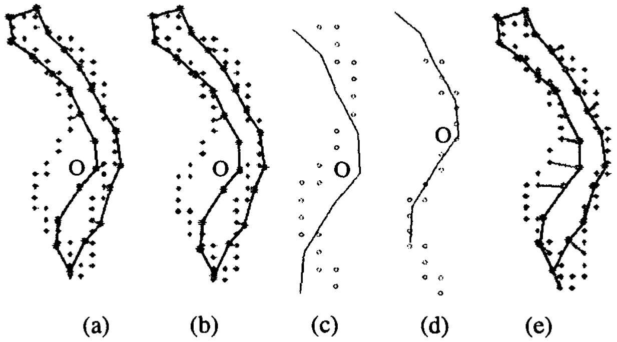

Fig. 2. Flexible registration of a shape approximation (solid outline) to an original shape (dotted outline), with point correspondences drawn as gray segments. a) Global similarity registration - Algorithm 2. b) Monotonic registration obtained from a) by point reordering and inversion elimination (the vertex labeled O causes an inversion): Steps 1-3 of Algorithm 3. c) Topological neighborhood corresponding to vertex O: Step 4a of Algorithm 3. d) Similarity registration of the two topological neighborhoods in c). e) Final flexible registration.

## III. Shape registration

Our shape registration method is based on two stages: (i) global similarity registration of two arbitrary sets of points and (ii) non-linear registration based on local similarity of two curves.

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: (i) the size $n$ of the matched subsets is large, and (ii) the *mean-alignment-error* between $A'$ and $B'$ is small. We propose a registration procedure (Algorithm 2) based on a polynomial quasi-exhaustive exploration of the correspondence functions (match matrices) space. Its main novelty from techniques previously used in the literature [6,8] is the way it resolves the *shrinking* effect [6]: 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 3 correspondences the MAE is 0 and this should be compensated for. Therefore, we want to explicitly encode in the search criterion that we are willing to trade a $q\%$ increase in MAE for a $p\%$ increase in the number of correspondences 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:

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

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 highly 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 $\psi$ that aligns the sets $\{a_{j1}, a_{j2}\}$ and $\{b_{k1}, b_{k2}\}$.
   ii. Apply $\psi$ 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. 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 $T$.
   iv. Compute $f(M)$.
   v. If $f(M) < V_{min}$ then $V_{min} = f(M)$, $\psi_{min} = \psi$.
3. Apply $\psi_{min}$ 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 $\psi_{final}$ that aligns the corresponding sets $A_{M'}$ and $B_{M'}$.
```

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 [6]), we need to locally refine the results of the global registration such that 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 $A$ 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 $k < l$.

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. (Note that we are left with a monotonic registration).
4. For $i = 1..|B|$ do
   (a) Find a topological neighborhood of $b_i$, $[b_l, b_{l+1}, ..., b_i, ...b_{r-1}, b_r]$ (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 with, say $a_{l'}$ and $a_{r'}$ in $A$ with $l' < r'$.
   (b) Perform a similarity registration between the sets $[a_{l'}, a_{l'+1}, .., a_{r'}]$ and $[b_l, b_{l+1}, .., b_r]$.
   (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.
```

## IV. Shape clustering and prototype computation

The third step of Algorithm 1 defines a pseudo-distance matrix $\mathcal{D}$ of *mean alignment errors* between an *approximation of a shape* and an *original shape* from the training set. A convenient way for obtaining shape clusters based on $\mathcal{D}$ which is also helpful for cluster prototype (average) computation is a *k-means* type clustering algorithm [13]:

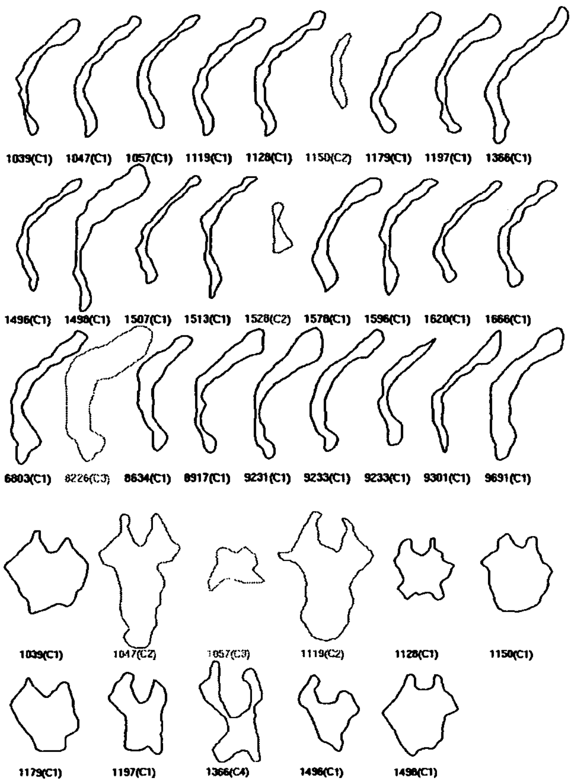

Fig. 3. Two training sets of left ventricle and cistern shapes from different patients were automatically divided into clusters (main cluster (C1) drawn in black and secondary clusters drawn in different shades of gray).

1. Find a seed which is closest to the data. 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 common ground for extracting corresponding sets of points of the same size from as many training shapes as possible, as discussed in Section 2.

2. Extract from the training set and put in a cluster all shapes $S_j$ that fit to $S'_{i_0}$ (Step 4b).

This cluster extraction procedure continues until all shapes from the training set have been assigned to a cluster. For each cluster, define the cluster prototype 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 is used by the segmentation method to reject shape deformations that have not been seen in the training set (see [3]).

## V. Experimental Results

The shape learning 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. Figure 2 shows the registration stages: global similarity registration - Algorithm 2 (Fig. 2a), point reordering and inversion elimination - steps 1 – 3 of Algorithm 3 (Fig. 2b), defining topological neighborhoods step 4.a of Algorithm 3 (Fig. 2c) and their registration (Fig. 2d) and final registration (Fig. 2e). Figure 3 shows the original manual tracings and clustering results for two structures. The main cluster is drawn in black, while the secondary clusters (which can be considered outlier shapes) are drawn in gray shades. Figure 4 shows the main cluster prototypes for 11 structures with the aligned shape examples overlaid. Consecutive point clouds are drawn in different shades of gray to show that the clouds are non-overlapping; the registration appears to be very precise.

In order to obtain a quantitative validation of our results, we used the method employed in [10]. From each structure prototype, we manually selected several points that were considered most important in defining its shape (the points with the highest curvature). These points were manually registered 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 thalamus in Fig. 4 as black squares). 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 thalamus are also shown in Fig. 4: for every point selected on each shape, each distance is displayed on the same $y$ coordinate as the ground truth point to which it corresponds. 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 if there are no curvature or other anatomical cues. The average *rms* error computed 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

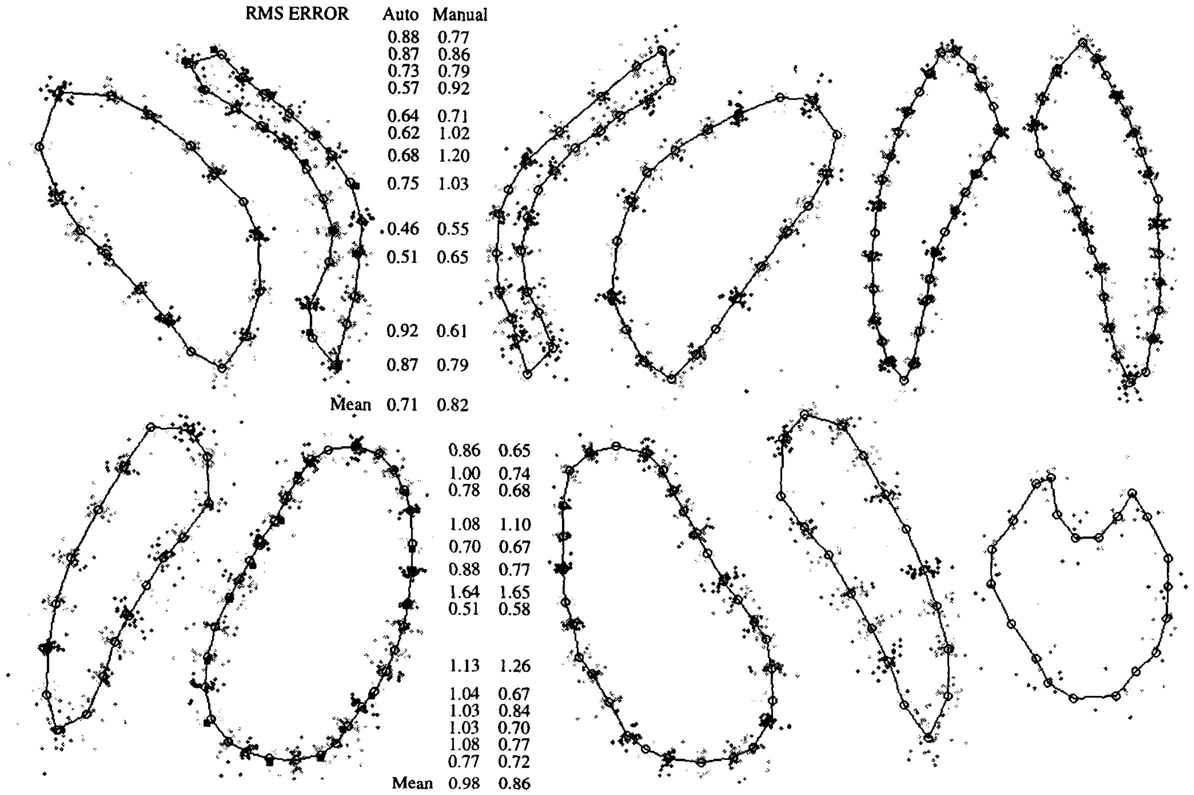

Fig. 4. Procrustes averages (prototypes) of the shapes in the main clusters for 11 brain structures with the aligned shape examples overlaid. The clouds of consecutive points are drawn in different shades of gray to show the accuracy of the registration. The *ground truth* position for several points on the right ventricle and thalamus are shown using black squares. For each such point we also show the *rms error* of the manual and automatic registrations.

*rms* is slightly larger.

We employed the automatically extracted prototypes to train a Point Distribution Model (PDM) [3] and segment 11 neuroanatomical structures. Note that Algorithm 1 not only gives an average shape and variation for each cluster, but also the registration of the cluster prototype to all shapes in the cluster. Therefore, PDM training becomes very fast and does not require any kind of human intervention once the shapes in training set have been clustered. We show the results of the automatic segmentation on two different MRI images in Fig. 5.

## VI. Discussion

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

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* distances between meaningful corresponding points are smaller than 10% of the object scale. Therefore, 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 this upper bound. We also found it to be convenient to work with scale independent objects, therefore, before registering two point sets, we rescale 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.

2. *Properties of the evaluation function $f(M)$ (Eq. 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 $p\%$ more links than $M$, one has $K/[(1+p)n] \leq f(M') \leq (K+1)/[(1+p)n]$

and $M'$ is prefered to $M$ if

$$(K+1)/[(1+p)n] < K/n \Leftrightarrow Kp > 1.$$

On the other hand, the choice between an n-link matrix and one with less than $(1+p)n$ links (if $p < 1/K$) is determined by the MAE: the $p\%$ increase in the number of links will be accepted iff

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

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% increase in the number of links will be always accepted no matter what the increase in MAE is, while a 25% is accepted if the MAE increase is less than 75% (this does not mean that if $q > 75\%$ then $M'$ is always rejected, in practice it might be accepted until $q$ becomes about 100%).

3. *Time complexity.* Algorithm 1 performs $m^2$ pairwise matchings, where $m$ is the number of shapes in the training set. Note that for a large training set, we only need to apply the quadratic pair-wise matching to a relatively small subset (a few tens of elements). After that, one can classify the remaining shapes only by registering them 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. Our implementation needs approximately 5 seconds of computing time on a Sun Ultrasparc 296 MHz to compute one pair-wise $2D$ shape registration.

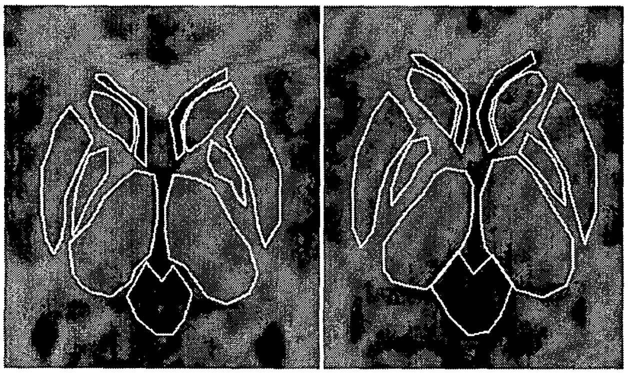

Fig. 5. Automatic segmentation of 11 brain structures in two different MRI images.

## VII. Conclusion

A new fully automated $2D$ shape learning method has been presented. It is based on clustering a shape training set in the original shape space and performing a Procrustes analysis on each cluster to obtain a cluster prototype and information about shape variation. The method was utilized to identify clusters and compute prototypes for 11 neuroanatomic brain structures from a training set of 28 manual tracings. A quantitative analysis of our shape registration approach shows that our results compare quite well to those achieved by manual registration; achieving an average *rms error* of about 1 pixel. Our approach can serve as a fully automated substitute to the tedious and time-consuming manual shape analysis.

## Acknowledgments

This work was supported by a grant from Siemens Corporate Research, Princeton. We would like to thank Mario Figueiredo for the spline approximation code.

## References

[1] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] F. L. Bookstein, "Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape," *Medical Image Analysis*, pp. 225–244, 1997.

[3] 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.

[4] T. F. Cootes and C. J. Taylor, "A mixture model for representing shape variation," in *Proceedings of British Machine Vision Conference*, pp. 110–119, BMVA Press, 1997.

[5] I. L. Dryden and K. V. Mardia, "Multivariate shape analysis," *Sankhya*, vol. 55, pp. 460–480, 1993.

[6] J. Feldmar and N. Ayache, "Rigid, affine and locally affine registration of free-form surfaces," *Int. J. of Comp. Vision*, vol. 18, pp. 99–119, 1996.

[7] M. Figueiredo, J. Leitao, and A. K. Jain, "Adaptive B-splines and boundary estimation," in *Proceedings CVPR-'97*, San Juan, PR, 1997.

[8] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching," *Pattern Recognition*, vol. 31(8), pp. 1019–1031, 1998.

[9] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Royal Stat. Soc. B*, vol. 53(2), pp. 285–339, 1991.

[10] A. Hill, A. D. Brett, and C. J. Taylor, "Automatic landmark identification using a new method of non-rigid correspondence," in *Proceedings of IPMI '97 Conference*, vol. 1230, pp. 483–488, Springer-Verlag, 1997.

[11] B. K. P. Horn, "Closed form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, 1987.

[12] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[13] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* New York: Prentice Hall, 1988.

[14] A. Neumann and C. Lorenz, "Statistical shape model based segmentation of medical images," *Computerized Medical Imaging and Graphics*, vol. 22, pp. 133–143, 1998.

[15] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 17, no. 6, pp. 545–561, 1995.

[16] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, 1992.