# Learning Shape Models from Examples using Automatic Shape Clustering and Procrustes Analysis

Nicolae Duta*, Milan Sonka** and Anil K. Jain*

*Department of Computer Science and Engineering, Michigan State University,
**Department of Electrical and Computer Engineering, The University of Iowa

**Abstract.** A new fully automated shape learning method is 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. As a direct application of our shape learning method, a 17-structure shape model of brain substructures was computed from MR image data, an eigen-shape model was automatically trained, and employed in our method for segmentation of those MR brain images not present in the shape-training set. Our approach can serve as a fully valid automated substitute to the tedious and time-consuming manual shape analysis.[1]

## 1 Motivation

Automated learning of shape models is an important problem in medical image analysis with direct implications in the area of medical image interpretation. We and others have previously demonstrated the utility of incorporating shape in medical image segmentation and interpretation [1]. However, training a shape-based segmentation system is mostly done manually following a tedious and therefore impractical process. In the work reported here, a novel approach to automated learning of shape models from examples is presented and its utility demonstrated in segmentation of MR brain images.

We have developed a novel solution to the problem of shape reparameterization–alignment–averaging problem. The main difference from previously reported methods [2, 3] 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.

## 2 Background and notation

A *shape instance* $A = \{s_i^A\}_{i=1..n} = \{(x_i^A, y_i^A)\}_{i=1..n}$ is a set of points in the 2-D Euclidean space. A shape instance $B$ is called *aligned* to a shape instance

[1] This work was supported by a grant from Siemens Corporate Research, Princeton. A complete paper and set of results can be found at http://web.cse.msu.edu/~dutanico.

$A$ 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 $SS(A,B)$ is called *Procrustes sum of squares* $PSS(A,B)$.

The *Procrustes average shape* 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. For a detailed definition, properties and ways of computing an average shape see [2].

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* $M = \{M_{j,k}\}_{k=1..r}^{j=1..p}$ 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*. Let $A_M$ and $B_M$ be the subsets of $A$ and $B$ matched by $M$ and $PSS(M) = PSS(A_M, B_M)$. We define a search criterion to be minimized over the match matrices space as: $f(M) = [PSS(M)/n + K]/n$, where $n$ is the number of links in $M$ and $K$ is a constant. This functional encodes the fact that we are willing to trade a $q\%$ increase in average PSS for a $p\%$ increase in the number of correspondences. It also helps avoid the *shrinking effect* described in [4].

## 3 Problem definition and solution outline

Mathematically speaking, we present a solution to the following problem: Given a set of $m$ shape instances $S_k = \{(x_i^k, y_i^k)\}_{i=1..n_k}^{k=1..m}$, partition it into a set of clusters and, for each shape cluster, compute a *prototype* (Procrustes mean shape). The set of shape prototypes will be used as models for detection of object instances in new images by means of deformable template segmentation. Our shape learning method consists of the following main steps:

*Algorithm 1: Shape Learning Outline*

1. For each (evenly sampled) shape $S_k$ in the training set compute a polygonal approximation $S'_k$.
2. 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 a set $T_{j,k}$ as the subset of $S_j$ that corresponds (was matched) to the points of $S'_k$, otherwise set $T_{j,k} = \emptyset$.
3. Compute a pseudo-distance matrix $\mathcal{D} = \{d_{j,k}\}_{j,k=1..m}$ where $d_{j,k} = PSS(T_{j,k}, S'_k)/|T_{j,k}|$ if $T_{j,k} \neq \emptyset$ or $d_{j,k} = \infty$ otherwise.
4. 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 $S'_{i_0}$ can 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 set $\{T_{i_k,i_0}\}_{k=1..p}$.

The shape approximations computed in Step 1 of the learning algorithm have about three times fewer points than the original shapes in order to smooth small shape artifacts, noise and quantitation 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 parameterization 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 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. This also ensures that the shape variation present in the original data is *completely* preserved if the registration process is precise.

The employed shape registration method consists of two stages: (i) Similarity registration of two arbitrary sets of points and (ii) Non-linear registration based on local similarity of two curves. Our similarity matching algorithm can be outlined as follows:

*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 (a) through (e)
   (a) Find the similarity transformation $\psi$ that aligns the sets $\{a_{j1}, a_{j2}\}$ and $\{b_{k1}, b_{k2}\}$.
   (b) Apply $\psi$ to all the points in $B$ to obtain $B'$.
   (c) 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 $T$ (automatically set equal to 10% of the scale of $B$) 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 on $M$ a 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 $T$.
   (d) Compute $f(M)$.
   (e) 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 $NN(b_k)$ in $A$. If the distance between $b_k$ and $NN(b_k)$ is smaller than $T$ then set the 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 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 [4]), 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.

*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 some subset $A'$ of $A$ and a subset $B'$ of $B$ obtained by global similarity registration.

1. Cyclically reorder the points of $A$, $B$ and the links in $\mathcal{M}$ such that point $a_1$ corresponds to point $b_1$.
2. If the number of inversions (pairs of points $a_i$ and $a_j$ corresponding to $b_k$ and $b_l$ -in this order- such that $i < j$ and $k \geq l$) 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 in $A$, let them be $a_{l'}$ and $a_{r'}$ 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.

The third step of Algorithm 1 defines a pseudo-distance matrix $\mathcal{D}$ of *normalized Procrustes sum of squares* between an *approximation of a shape* and an *original shape* from the training set. A convenient way for obtaining shape clusters based on $\mathcal{D}$ and at the same time helpful for cluster prototype computation is a *k-means* type clustering algorithm:

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). $S'_{i_0}$ is going to be used as a common ground for extracting corresponding sets of points of the same size from as many training shapes as possible.

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 we 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 [1]).

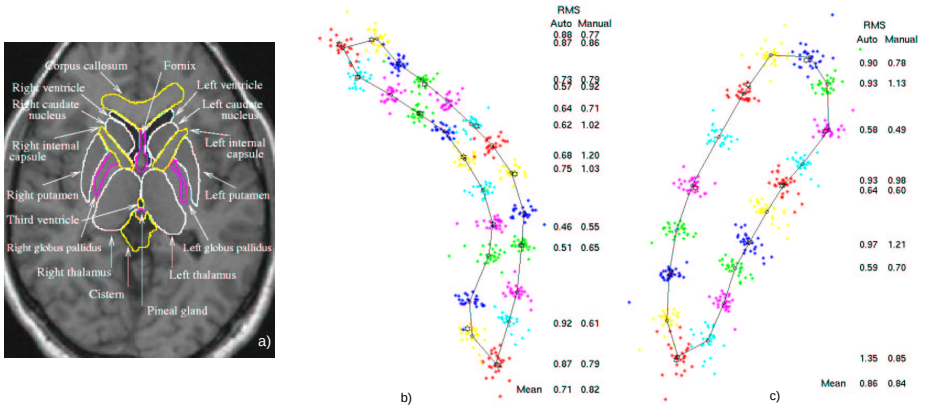


**Fig. 1.** The 17 neuroanatomical structures of interest (a). Procrustes average of 25 right-ventricle shapes (b) and 28 right-globus pallidus shapes (c) with the scatter of fits overlaid. The fits of consecutive points are drawn in different shades of gray to show the accuracy of the registration: consecutive clouds are non-overlapping.

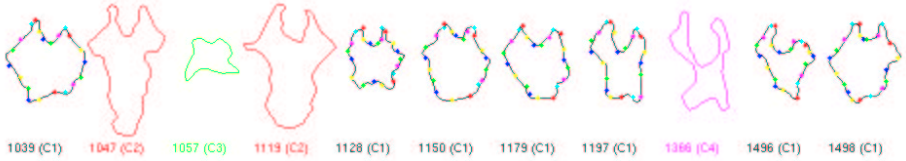


**Fig. 2.** A set of 11 cistern training shapes from different patients was automatically divided into clusters (main cluster (C1) and three secondary clusters). The registration of the *best fit shape* (1179) to cluster C1 is overlaid.

## 4 Experimental Results

The shape learning method presented above was employed to design a shape model for 17 brain structures (shown in Fig. 1a) and its performance was assessed

by a quantitative comparison to a manually-identified independent standard. The training set consisted of observer-defined contours identified by a neuroanatomist in 28 individual T1-weighted contiguous MR images of the human brain. Fig. 2 shows the original manual tracings and clustering results for cistern together with the *best fit shape* registration to the main cluster (the sets $T_{i_1,i_0}, ..., T_{i_p,i_0}$ as defined in Algorithm 1). Fig. 1b,c shows the Procrustes averages for the right ventricle and globus pallidus with the *scatter of fits* overlaid.

In order to obtain a quantitative validation of our results we used the method employed in [3]. From each shape model, we manually selected several points that were considered most important in defining its shape (the 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. We computed and compared the *root-mean-square* (rms) distance of manually placed points from the independent standard and the rms distance of the automatically registered points from the independent standard, respectively. The *rms* distances for the right ventricle and globus-pallidus are also shown in Fig. 1b,c: for every point selected on each shape, each distance is displayed on the same $y$ coordinate as the ground truth point it corresponds to. As a rule, 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. On the average, *all* rms errors are between $0.7 - 1.5$ pixels.

## 5 Conclusion

A new fully automated shape learning method was 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. A quantitative analysis of our shape registration approach demonstrated results well comparable to those obtained by manual registration; achieving an average *rms error* of about 1 pixel. Our approach can serve as a fully valid automated substitute to the tedious and time-consuming manual shape analysis.

## References


1. T. Cootes, A. Hill, C. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image & Vision Computing*, 12(6):355–366, 1994.
2. F. L. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–244, 1997.
3. A. Hill, A. Brett, and C. Taylor. Automatic landmark identification using a new method of non-rigid correspondence. In *Proceedings of IPMI '97*, pages 483–488.
4. J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. of Comp. Vision*, 18:99–119, 1996.