Optimal Embedding for Shape Indexing in Medical Image Databases

Xiaoning Qian and Hemant D. Tagare

Yale University, New Haven CT 06520, USA

Abstract. Fast retrieval using organ shapes is crucial in medical image databases since shape is a clinically prominent feature. In this paper, we propose that 2-D shapes in medical image databases can be indexed by embedding them into a vector space and using efficient vector space indexing. An optimal shape space embedding is proposed for this purpose. Experimental results of indexing vertebral shapes in the NHANES II database are presented. The results show that vector space indexing following embedding gives superior performance than metric indexing.

1 Introduction

Content-based retrieval in medical image databases is critically dependent on efficient indexing techniques. There are two common indexing techniques: *vector space indexing*, and *metric space indexing*. If a feature has well defined coordinates, then *vector space indexing* techniques are used. If the feature does not have coordinates, then *metric space indexing* techniques are applicable.

In this paper, we propose techniques for efficient indexing of shape for 2-D medical image databases. We show that the 2-D *shape space* can be embedded in a vector space in such a way that the vector space metric best approximates the partial Procrustes distance in the shape space. With the optimal embedding, shapes can be indexed by classical vector space indexing techniques. We provide experimental results that compare the performance of the embedding strategy versus metric trees and show that the embedding strategy gives superior results.

All experiments reported in this paper use images from the NHANES II database. NHANES II has about 17,000 spine x-ray images. Spine disease is often manifest as *osteophyte*, which is bony prominence along the vertebral boundary. Because osteophyte changes the shape of the vertebra, retrieval by vertebral shape is important to NHANES II. Indexing vertebral shape in NHANES II is our main application.

The vertebrae are segmented using a dynamic programming template matching algorithm [11]. The output of this algorithm is a fixed set of m points placed on the vertebral boundary in a homologous manner. These points may be taken as landmarks along the boundary. By "shapes of vertebrae" we mean the shapes of these "landmark" points.

This paper draws on indexing theory and shape space theory – two theories that are quite different. In the limited space of this paper, we have opted to treat indexing rather briefly and present the shape embedding in more detail.

J. Duncan and G. Gerig (Eds.): MICCAI 2005, LNCS 3750, pp. 377–384, 2005.

2 Literature Review

The literature on shape analysis is vast. We briefly mention some of the related work. Shape descriptors may be boundary based or region based. For boundary based descriptors, Fourier and wavelet descriptors [9], scale space techniques [6], and shape matching techniques [2,7] are used. For region based description, different moment invariants [9] are used. When landmarks are available, the shape of the landmarks are described as elements of an appropriate shape space. We refer the reader to [3,4] for a complete discussion.

For indexing, we note that classical indexing structures for vector spaces are in [10], while classical indexing structures for metric spaces are in [1].

3 Indexing for Content-Based Retrieval

Content-based retrieval uses *range queries* and *nearest neighbor queries.* In a range query, the user has an example image with a feature u and asks the database to retrieve all images with features v, such that $d(u, v) \leq T$ for some threshold T. Here $d()$ is a metric in the feature space. A nearest neighbor query asks for k nearest neighbors to the example u according to the metric d. We concentrate on the nearest neighbor queries in this paper.

Queries can be answered by a linear search through the database. *Indexing trees* refer to techniques that can speed up the search by organizing the database into hierarchical trees. A brief, relevant summary of indexing is as follows:

1. Indexing hierarchically partitions the feature space and creates a cover for each partition. The covers are arranged in a tree; each node of the tree representing a cover.

   For vector space features, the covers are cubes with sides perpendicular to the coordinate axis [10]. For metric spaces, the covers are metric spheres. All leaf nodes point to the data that are contained in its cover.
2. Retrieval starts from the root node and proceeds by testing whether the cover at a node intersects the metric sphere defined by the query. If a node passes this *node test*, then the procedure is applied to the children of the node. If the node fails the node test, then the entire subtree rooted at the node is rejected since its children cannot contain any data that intersect the query sphere.
3. One performance measure for indexing trees is the average number of node tests per query. This measures the computation cost during retrieval. A theoretical expression for the performance was derived in [12].
4. The performance of an indexing tree becomes poor if its nodes increasingly survive the node test. In [12], we proposed a *greedy* algorithm that traverses and eliminates inefficient nodes.

   In [8], we reported an algorithm that *optimally* eliminates nodes. The algorithm is a dynamic program over all possible node eliminations. We call these procedures *tree adaptation procedures* since they adapt the tree to the data distribution.

4 Shape Spaces and Shape Queries

4.1 Configuration, Preshape, and Shape Space

As mentioned in section 1, vertebrae in NHANES II are segmented by an algorithm that gives a set of m points along the boundary. Representing each point as a complex number, every boundary can be considered as an element of $\mathbb{C}^m$, the complex vector space of dimension m. $\mathbb{C}^m$ is the *configuration space*. Two boundaries $z_i, z_j \in \mathbb{C}^m$ have the same shape if there exists a translation, rotation, and (non-zero) scaling that aligns them, i.e. if there exist complex numbers t, μ, with $\mu \neq 0$, such that $z_i = \mu z_j + 1_m t$, where $1_m = (1 \cdots 1)^T$. Here t is the translation and $\mu = re^{i\theta}$ is scaling by r and rotation by θ.

Following Kendall [4], we first consider only the action of scale and translation. For every $z \in \mathbb{C}^m$, define $\tilde{z} = \frac{z - \frac{1}{m} 1_m^T z}{|z - \frac{1}{m} 1_m^T z|}$ to be the *preshape* of z. Then, $\tilde{z}$ is simply z translated so its center of mass is at the origin and scaled so that the resulting scale is unity. All shapes that differ only by translation and scaling are mapped to the same preshape. The *preshape space* is the set of all preshapes, and is easily seen to be the unit sphere in the configuration space $\mathbb{C}^m$.

The map $z \rightarrow \tilde{z}$ from configurations to preshapes factors out translation and scaling. To get shape, it remains to factor out rotation. Suppose that a configuration z has preshape $\tilde{z}$. Rotating z by θ gives the configuration $ze^{i\theta}$ which has the preshape $\widetilde{ze^{i\theta}}$. It is easy to show that $\widetilde{ze^{i\theta}} = e^{i\theta}\tilde{z}$. Thus, all configurations that have the same shape as z fall on the one dimensional *orbit of $\tilde{z}$ in the preshape space* defined as $\check{z} = \{e^{i\theta}\tilde{z}\}$. The orbit $\check{z}$ is the *shape of* z.

Since each orbit is a shape, the set of all orbits is the *shape space*. Kendall showed that the shape space of m landmarks is a complex projective space of complex dimension $m - 2$. This is a non-Euclidean manifold.

The different spaces and relations between them are illustrated in figure 1.

Shape space has many natural metrics. The specific one we use is the *partial Procrustes metric* $d_P(z_i, z_j)$. It is defined as the minimum Euclidean distance between the preshape of z_i and the orbit of the preshape of z_j:

$$d_P(z_i, z_j) = \inf_{\theta} \parallel \tilde{z}_i - e^{i\theta}\tilde{z}_j \parallel .$$

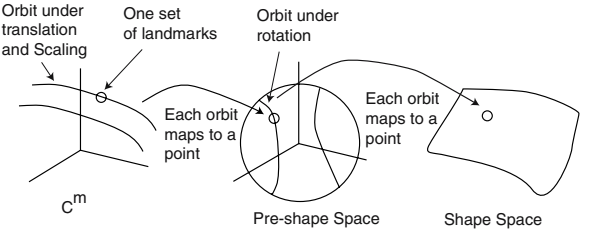

Fig. 1. Preshape and Shape Spaces

4.2 Metric Shape Indexing

Because shape spaces are curved manifolds and the partial Procrustes distance is non-Euclidean, one obvious choice for indexing shapes is to use metric indexing trees. Specifically, we use hierarchical clustering with the partial Procrustes metric to cluster shapes in a tree. Greedy node elimination and optimal tree adaptation are used to further increase the efficiency.

As mentioned in section 1, an alternative is to embed the shape space into a vector space and use vector space indexing. We discuss this next.

5 Shape Embedding

Let z_k be one of n configurations in the database, and let $\widetilde{z_k}$ and $\breve{z_k}$ be its preshape and shape. Recall that the preshape space is a unit sphere in $\mathbb{C}^m$ and that the only variation left in the preshape space is rotation. Hence, it is reasonable to consider choosing one point on the preshape orbit of $\breve{z_k}$ to represent $\breve{z_k}$. That is, we choose the preshape with a particular orientation (yet to be determined) as the shape embedding. This is our key idea and is illustrated in figure 2.

Suppose we embed $\breve{z_k}$ as the point on the preshape orbit that is given by $[z_k] = e^{i\theta_k}\widetilde{z_k}$ for some θ_k. This embedding gives a Euclidean shape distance:

$$d_s([z_i],[z_j]) = \|[z_i] - [z_j]\|, \tag{1}$$

where, $\|\ \|$ is the usual Euclidean norm in $\mathbb{C}^m$. In general, this shape distance will be different from the partial Procrustes distance, and we would like to choose an embedding such that the difference between them is as small as possible.

One measure of the difference between d_P and d_s is

$$J = \sum_i \sum_j \mid d_s^2([z_i],[z_j]) - d_P^2(z_i,z_j) \mid. \tag{2}$$

We would like to choose embeddings $[z_1] = e^{i\theta_1}\widetilde{z_1}, [z_2] = e^{i\theta_2}\widetilde{z_2}, \cdots$, or alternatively, choose the angles $\Theta = (\theta_1, \theta_2, \cdots, \theta_n)$ such that J is minimized as a function of Θ. From now on we will write J as $J(\Theta)$ explicitly showing dependence on Θ. We now derive an algorithm for minimizing $J(\Theta)$.

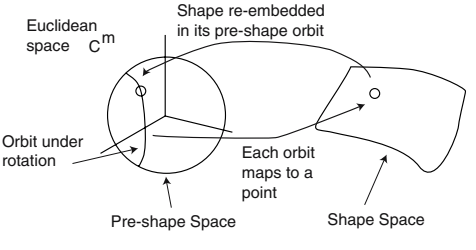

Fig. 2. Shape Embedding in Preshape Space

The first step is to show the following proposition:

Proposition 1: A Θ minimizes $J(\Theta)$ if and only if it minimizes

$$J_1(\Theta) = \sum_i \sum_j d_s^2([z_i],[z_j]). \tag{3}$$

Proof: First note that $d_s([z_i],[z_j])$ is the Euclidean distance between two fixed point $[z_i]$ and $[z_j]$ on the preshape orbits of $\widetilde{z_i}$ and $\widetilde{z_j}$. But $d_P(z_i,z_j)$ is the shortest distance between preshape orbits of $\widetilde{z_i}$ and $\widetilde{z_j}$. Thus, $d_s([z_i],[z_j]) \geq d_P(z_i,z_j)$, and therefore $\mid d_s^2([z_i],[z_j]) - d_P^2(z_i,z_j) \mid = d_s^2([z_i],[z_j]) - d_P^2(z_i,z_j)$. Note that the $d_P^2(z_i,z_j)$ term is independent of Θ and can be dropped from $J(\Theta)$, giving

$$J(\Theta) = \sum_i \sum_j d_s^2([z_i],[z_j]) = J_1(\Theta).$$

To proceed further, a simple algebraic manipulation of $J_1(\Theta)$ gives:

$$J_1(\Theta) = \sum_i \sum_j d_s([z_i],[z_j])^2 = 2n \sum_i \parallel [z_i] - \frac{1}{n} \sum_j [z_j] \parallel^2.$$

As a brief aside, consider a second objective function

$$H_1(\Theta,\mu) = 2n \sum_i \parallel [z_i] - \mu \parallel^2. \tag{4}$$

The minimizing μ of H_1 is known in the shape space literature as the *procrustean mean size-and-shape* of the preshapes $\widetilde{z_i}$. Conditions for a unique procrustean mean size-and-shape are given in [5]. Loosely speaking, a unique μ exists if the distribution of $\widetilde{z_i}$ is not too broad. In practice this condition almost always holds and a unique μ exists. We assume this to be the case and we have

Proposition 2: If $H_1(\Theta,\mu)$ has a minimizer (Θ^*,μ^*), Θ^* minimizes $J_1(\Theta)$.

Proof: For any fixed Θ, because $[z_i]$ are in the vector space $\mathbb{C}^m$, and $\parallel\ \parallel$ is the usual Euclidean norm, the function $H_1(\Theta,\mu)$ has a unique minimum with respect to μ, and the minimum is given by $\mu^* = \frac{1}{n}\sum_j [z_j]$. Thus,

$$\min_\mu H_1(\Theta,\mu) = 2n \sum_i \parallel [z_i] - \frac{1}{n} \sum_j [z_j] \parallel^2 = J_1(\Theta).$$

It follows that if $H_1(\Theta,\mu)$ has a minimizer (Θ^*,μ^*), Θ^* also minimizes $J_1(\Theta)$.

To obtain the optimal embedding, we minimize $H_1(\Theta,\mu)$ by alternately updating Θ and μ as follows:

1. Initialize $\Theta^{[0]} = (0,\cdots,0)$, and $\mu^{[0]} = \frac{1}{n}\sum_k [z_k]$.
2. Update $\Theta^{[l]} = \arg_\Theta \min H_1(\Theta,\mu^{[l-1]})$, where $\Theta^{[l]} = (\theta_1^{[l]},\cdots,\theta_n^{[l]})$ is given by

$$\theta_k^{[l]} = \arg \widetilde{z_k}^* \mu^{[l-1]}, \tag{5}$$

where, $\widetilde{z_k}^*$ is the complex conjugate transpose of $\widetilde{z_k}$.

Calculate $\mu^{[l]} = \arg_\mu \min H_1(\Theta^{[l]}, \mu)$. This is given by

$$\mu^{[l]} = \frac{1}{n}\sum_k [z_k] = \frac{1}{n}\sum_k e^{i\theta_k^{[l]}}\widetilde{z_k}. \tag{6}$$

3. Terminate if a fixed point is reached (i.e. if $(\Theta^{[l]}, \mu^{[l]}) = (\Theta^{[l-1]}, \mu^{[l-1]})$). Else, go to 2.

Let the terminating $(\Theta^{[l]}, \mu^{[l]})$ be denoted by $(\hat{\Theta}, \hat{\mu})$. Then, the optimal embedding is given by $[z_k] = e^{i\hat{\theta_k}}\widetilde{z_k} = \frac{\widetilde{z_k}^*\hat{\mu}}{\|\widetilde{z_k}^*\hat{\mu}\|}\widetilde{z_k}$ for all k.

5.1 Vector Space Indexing of Shape

Following optimal embedding, the embedded shapes can be indexed by any of the classical vector space indexing techniques. We choose to index by a kd-tree [10]. After embedding, shape similarity retrieval is carried out by using the d_s shape metric of equation (1).

6 Experiments

At the moment, a total of 2812 boundaries have been segmented. Each boundary is a consistent set of 34 landmarks.

We first evaluated the closeness of the embedding distance d_s to the partial Procrustes distance d_P. Recall that the optimal embedding was obtained by minimizing the absolute difference between d_s^2 and d_P^2 for all pairs of data in the database. To measure the similarity between the two we calculated fraction squared difference $FSD = |d_s^2([z_i],[z_j]) - d_P^2([z_i],[z_j])|/d_P^2([z_i],[z_j])$ as well as the fractional difference $FD = |d_s([z_i],[z_j]) - d_P([z_i],[z_j])|/d_P([z_i],[z_j])$. A set of 1000 vertebrae were randomly chosen from the database and the FSD and FD were calculated for all pairs of vertebrae from this set. The average and standard deviation of the FSD and FD are given in Table 1. From the table, it is clear that the Euclidean distance following embedding is very similar to the partial Procrustes distance.

We also compared the relative ranking of vertebrae according to the embedded Euclidean distance and the partial Procrustes distance. From the set of 1000 vertebrae used in the above experiment, 100 vertebrae were chosen as query vertebrae. For each query vertebra, the set of 20 nearest vertebrae was found according to the partial Procrustes distance d_P and the embedded Euclidean distance d_s. For 98 of 100 queries the sets of nearest neighbors were identical, and for 2 queries they differed by a single image.

Table 1. Mean and Variance of FSD and FD of pairs from a set of 1000 vertebrae

Quantity	Mean	Var.
FSD	9.19×10^{-4}	2.09×10^{-6}
FD	4.59×10^{-4}	5.21×10^{-7}

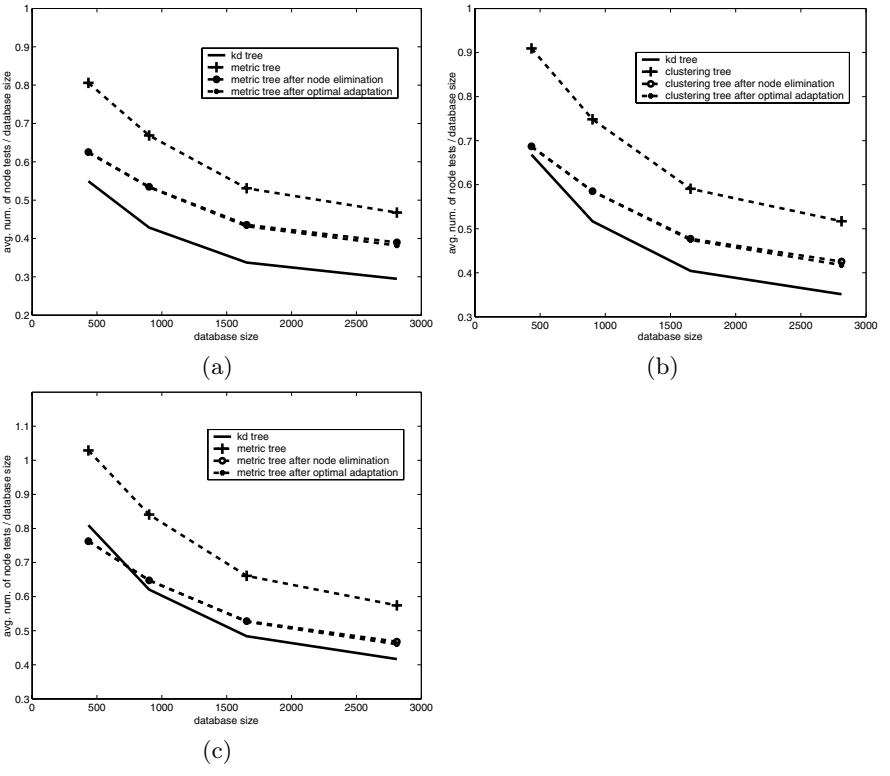

Fig. 3. Indexing performance comparison for (a) 5, (b) 10, and (c) 20-nearest neighbors

Finally, the 2812 shapes were randomly sampled into sets of size 434, 902, 1654 and 2812. Each set was indexed for shape with metric and vector indexing trees. The metric tree was used in its raw form, and to improve its performance also after greedy node elimination and optimal node elimination. Each vertebral shape in the database was used as query and 5, 10 and 20-nearest neighbor vertebral images were retrieved. The average number of node tests per query were recorded and expressed as a fraction of the total number of database points and plotted. Figure 3a-c show the results. From the figures, it is clear that

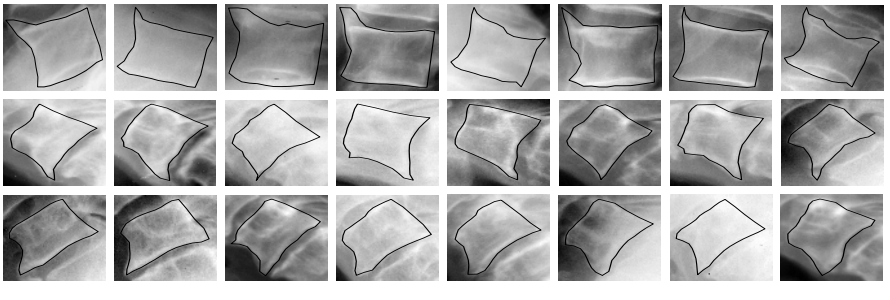

Fig. 4. Shape query samples

in all cases the kd-tree outperforms the raw metric tree. Further, except for the small database of 434 images and 20-nearest neighbors, the vector space indexing outperforms metric indexing with adaptation. Also, since the average number of node tests as a fraction of database size decreases with database size, kd-tree indexing has sub-linear complexity with respect to database size.

Three illustrative sample queries are given in figure 4. The left most image in each row is the query image and the successive images are the retrieved neighbors ranked in increasing shape distance from the query. The first query has an osteophyte near the top left corner. The other two have an osteophyte near the bottom left corner.

7 Conclusion

We proposed an embedding technique that optimally embeds shapes into a vector space. This allows the use of vector space indexing techniques for fast retrieval. Experiments show that the embedding does not significantly alter the metric or the nearest neighbor queries. Further, shape indexing efficiency using a kd-tree is significantly higher compared to the raw metric tree. It remains higher even when the metric tree is adapted.

Acknowledgment

This research was support by the grant R01-LM06911-05 from the National Library of Medicine.

References

1. E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquin: Searching in Metric Spaces. *ACM Computing Surveys*, Vol. 33, No. 3, Sept., 2001, pp. 273–321.
2. T.F. Cootes and C.J. Taylor: Statistical models of appearance for medical image analysis and computer vision. *Proc. SPIE Medical Imaging*, 2001
3. I. L. Dryden and K. Mardia: Statistical Shape Analysis. *J. Wiley*, 1998.
4. D. G. Kendall, D. Barden, H. Le: Shape and Shape Theory. *Wiley Series*, 1999
5. H.-L. Le: Mean Size-and-Shapes and Mean Shapes: a Geometric Point of View. *Advances in Applied Probability*, 27, 1995, pp. 44–55
6. F. Mokhtarian, S. Abbasi, J. Kittler: Robust and Efficient Shape Indexing Through Curvature Scale Space. *Proceedings of BMVC*, 1996, pp. 53–62
7. S. M. Pizer, T. Fletcher, A. Thall, M. Styner, G. Gerig, S. Joshi: Object Models in Multiscale Intrinsic Coordinates via M-reps. *IVC*, 2000
8. X. Qian, H. D. Tagare: Optimal Indexing Trees for Medical Image Databases. *ISBI* 2002
9. Y. Rui, T. S. Huang, and S.-F. Chang: Image retrieval: current techniques, promising directions and open issues. *JVCIR*, Vol. 10, 1999, pp. 1–23
10. H. Samet: The Design and Analysis of Spatial Data Structures. *Addison-Wesley*
11. H. D. Tagare: Deformable 2-D Template Matching Using Orthogonal Curves. *IEEE Trans. on Med. Imaging*, Vol. 16(1), 1997, pp. 108–117
12. H. D. Tagare: Increasing Retrieval Efficiency by Index Tree Adaptation. *IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997