

Indexing of Complete and Partial 2-D Shapes for NHANES II

Xiaoning Qian¹, Hemant D. Tagare¹, Robert K. Fulbright¹, Rodney Long² and Sameer Antani²

¹ Yale University, New Haven, CT 06510, USA

² National Library of Medicine, Bethesda, MD 20894, USA

Abstract. NHANES II is a database of cervical and lumbar x-ray spinal images and is a nationally important database maintained at the National Library of Medicine. This paper gives an overview of our efforts to index NHANES II images for osteophyte severity which is one of the leading causes of back pain. We first use a template deformation algorithm to extract vertebra boundaries. Shapes of vertebrae are then conveniently described in shape spaces. Our major contribution here is a fast retrieval scheme for both complete and partial shapes in shape space with indexing after optimal embedding. Our experiments first evaluate the performance of different proposed shape distances and demonstrate the optimality of the embedding algorithm. The proposed indexing scheme with the NHANES II database achieves the sub-linear retrieval complexity.

1 Introduction

The ability to retrieve images using shapes of organs is important for medical image databases. In medical practice, it is common to characterize shapes of organs by the locations of a few key landmarks and the main aim of this paper is to use such landmarks to index *complete* and *partial shapes* of organs. By indexing “partial shape” we mean indexing the shape of only a part of the contour. Indexing partial shapes is important because many diseases affect organ shape locally, changing only a part of the boundary.

The NHANES II is a national health survey of the U.S. population. Because of the prevalence of neck and back pain, the survey collected approximately 17,000 spine x-ray images: 10,000 of which are cervical spine x-rays and 7,000 are lumbar spine x-rays. Fig. 1(a) and (b) show two example images from NHANES II.

Osteophytes are important markers of spine disease. They form in response to repetitive strain and manifest as bony prominences along the anterior, lateral and posterior aspects of the vertebral body [1]. Workshops held by the National Institutes of Health (NIH) and the National Institute of Arthritis and Musculoskeletal and Skin Diseases (NIAMS) identified anterior osteophytes as the one of most important features of NHANES II images. Osteophytes are visible as the sharp prominences at the bottom left of the outlines in Fig. 1 (c) to (f). It is quite clear from these figures that the presence and severity of osteophytes affects

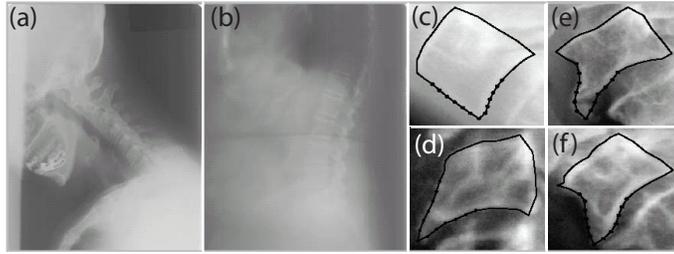


Fig. 1: Example images from NHANES II and several vertebrae with osteophyte severity increasing from (c) to (f)

the shape of the vertebral boundary. Thus retrieval by shape of the vertebral boundary is one means for retrieval by osteophyte severity.

Our aim is to construct a retrieval-by-similarity capacity for NHANES II where a user can select or provide a query shape q and the system retrieves images with k most similar shapes u according to a shape distance $D(q, u)$. This is the *k-nearest neighbor* query. We expect the retrieved vertebrae to have osteophyte severity that is similar to the osteophyte severity in the query vertebra.

Boundaries and Shapes In NHANES II: Boundaries of vertebrae in NHANES II images are available as a result of previous interactive segmentation using a dynamic programming template deformation algorithm [20]. The segmentation yields a set of 34 ordered points around the boundary. By the *shape of a vertebra* we mean the shape of the 34 ordered points in the plane obtained by the segmentation. The precise definition of shape and shape distance is given in section 2.

Similarity Retrieval and Indexing: The shape space of a finite set of landmarks in the plane is known to be a Riemannian manifold – it is in fact a complex projective space with a natural metric [7]. Because this shape space is non-Euclidean, standard indexing algorithms, which are designed for similarity retrieval in Euclidean spaces, cannot be used for similarity retrieval in it. On the other hand, all the metric access methods proposed recently are designed for the similarity retrieval using a fixed distance metric. It is not clear how those metric access methods can be implemented for similarity retrievals using different distance metric functions based on users’ requirements. To emphasize the pathological difference with respect to the osteophyte severity, we need an indexing structure to support both complete and partial shape retrievals.

Our key contribution is to solve the problem of creating indexing structures, which allow adaptable similarity queries (complete or partial), for non-Euclidean shape spaces. We first optimally embed shape space in a vector space and index the embedding space with standard Euclidean indexing techniques. Our indexing structures in the embedding shape space support both complete and partial shape retrieval efficiently.

Related Work: This paper focuses on the research of shape-based image retrieval. We briefly mention some of the related work. Content-based Image

Retrieval (CBIR) provides a flexible way to browse or retrieve images using visual information and the literature on this topic is vast. Lack of space prevents us from reviewing it exhaustively. We simply refer readers to the review papers in medical applications [19, 13]. For the shape extraction methods, we refer readers to the surveys of powerful active contour and level set formulations [10, 11].

Shape descriptors of extracted curves may be boundary based [12, 17] or region based [8, 17]. When landmarks are available, the shape of the landmarks are described as elements of an appropriate shape space [9, 7]. Many researchers have also proposed schemes to compare partial shapes e.g. [14]. They are based on a dynamic programming alignment of curve fragments. The corresponding indexing methods [16] are computationally expensive.

Fast nearest-neighbor searches in Euclidean spaces have a rich history (e.g. [4, 5]). Metric access methods ([2, 6]), indexing schemes proposed for generic metric spaces, have also attracted attention from the researchers. A user-adaptable similarity retrieval has been proposed to support different distances in Euclidean spaces [18]. However, there is not yet an indexing algorithm that works for user-adaptable similarity retrieval in non-Euclidean spaces. In particular, we are not aware of any algorithm that indexes in shape spaces.

The rest of the paper reads as follows: A brief introduction on shape, shape and pre-shape spaces is presented in Section 2. Section 3 discusses the idea of embedding the shape space in the pre-shape space and proposes the weighted partial shape distances. The resulting indexing structure is discussed in Section 4. Experimental results are given in Section 5. Section 6 concludes the paper.

2 Shape, Shape Space, and Shape Metrics

We begin our technical description by briefly reviewing shape and shape spaces. Suppose we are given m point landmarks in a plane. The *shape* of these points is taken to be the property of the landmarks that is independent of translation, rotation, and scaling. To be more precise, we consider the action of the similarity group on the landmarks. Two sets of landmarks are considered equivalent if they can be mapped onto each other exactly by some element of the similarity group. This partitions the set of all m landmarks into equivalence classes. Each equivalence class represents a shape. The *shape space* is the quotient space of the set of all m landmarks under the above equivalence relation. Kendall showed that for m landmarks in the plane the shape space is a familiar manifold – it is the complex projective space of complex dimensions $m - 2$ [9, 7].

The above is a direct route to the definition of shape spaces. However a slightly different description is more illuminating and suggestive. The quotient space under the similarity group can be constructed by first finding the quotient space under translation and scaling, followed by the quotienting of this space under rotations (Fig. 2). Let the m 2-D landmarks be represented as a complex position vector $z = [x_1 + jy_1, x_1 + jy_1, \dots, x_m + jy_m]^T \in C^m$, where $j^2 = -1$; (x_i, y_i) is the coordinate for the i^{th} landmark, and C^m is the vector space of m complex variables. The quotient space of C^m with respect to translation

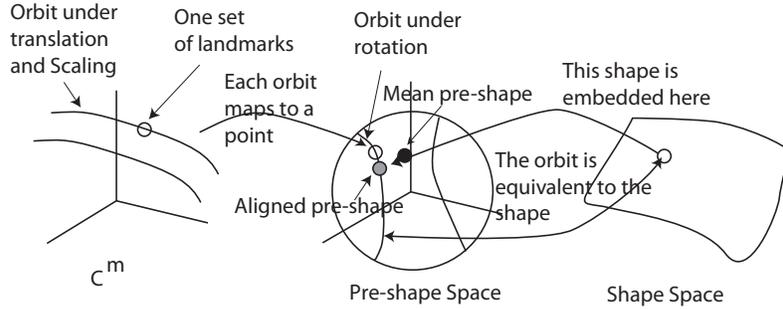


Fig. 2: Embed shape space back to pre-shape space

and scaling can be easily shown to be a unit complex sphere in C^m (Fig. 2). This space is called the *pre-shape space* PS^m . The *pre-shape* of z is $Z = (z - z^T \mathbf{1}_m) / \|z - z^T \mathbf{1}_m\|$, where $\mathbf{1}_m = [1/m \cdots 1/m]^T$. Every point in the pre-shape space generates an orbit under rotation of the original plane. Each orbit is a shape $[Z]_s$ and the set of all orbits forms the shape space SS^m as shown in Fig. 2.

Each vertebral boundary in the database maps onto its shape in the shape space and similarity retrieval corresponds to retrieving the k closest shapes to the query shape using a distance in the shape space. One of the natural Riemannian metrics is known as *partial Procrustes distance* $d_P^2([Z_k]_s, [Z_l]_s) = \min_{\theta} \|Z_k - e^{j\theta} Z_l\|^2$, where θ is the rotation factor.

3 Embedding Shape into a Vector Space

Since shape space is a non-Euclidean manifold, classical indexing techniques cannot be used with it. We index the shape space by first embedding it in a vector space as illustrated in Fig. 2. Recall that the pre-shape space PS^m is the unit sphere in C^m and assume that the map $\rho : PS^m \rightarrow SS^m$ takes the entire equivalence class of pre-shapes to the shape $[Z] = \rho(Z)$. One natural way to embed the shape space SS^m into the vector space C^m is to “invert” the map ρ . That is, we re-embed the shape $[Z]_s$ in the orbit of the pre-shape Z at some point $e^{j\theta} Z$ (Fig. 2).

Let z_k , $k = 1, \dots, n$ be the configurations in the database, and let Z_k and $[Z_k]_s$ be their pre-shapes and shapes respectively. We embed $[Z_k]_s$ at the point in the pre-shape orbit given by $e^{j\theta_k} Z_k$ for some θ_k . After embedding, the Euclidean distance between shapes $[Z_k]_s$ and $[Z_l]_s$ is $d_E([Z_k]_s, [Z_l]_s) = \|e^{j\theta_k} Z_k - e^{j\theta_l} Z_l\|_{C^m}$, where $\|\cdot\|_{C^m}$ is the usual Euclidean norm in C^m . In general, this shape distance is 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 the partial Procrustes distance d_P and the Euclidean distance defined above is

$$J = \sum_k \sum_l |d_E^2([Z_k]_s, [Z_l]_s) - d_P^2([Z_k]_s, [Z_l]_s)|. \quad (1)$$

We would like to choose the embeddings $e^{j\theta_1}Z_1, e^{j\theta_2}Z_2, \dots, e^{j\theta_n}Z_n$, or alternatively, choose the angles $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$ such that J is minimized as a function of Θ .

In [15], we proved that this minimization problem is actually equivalent to minimize another function

$$H_1(\Theta, \mu) = 2n \sum_k \| e^{j\theta_k} Z_k - \mu \|_{C^m}^2. \quad (2)$$

where $\mu^* = \arg H_1(\Theta, \mu)$ is a Fréchet mean of the shapes $\{[Z_1]_s, \dots, [Z_n]_s\}$, which minimizes $E[d_P([Z_k]_s, [Z_l]_s)^2]$ in statistical sense. To get the optimizing Θ^* , we only need to align all pre-shapes with μ^* in optimal positions:

$$\theta_k^* = \arg \mu^* Z'_k, \quad (3)$$

where Z'_k is the complex conjugate transpose of Z_k . This problem has a unique solution. [9]

Using the embedding map, all database shapes can be embedded into a vector space, which is decided by μ^* . A summary of the procedure is as follows: i. Convert each database point z_k to its pre-shape $Z_k = (z_k - z_k^T \mathbf{1}_m) / \|z_k - z_k^T \mathbf{1}_m\|$; ii. Calculate the mean pre-shape μ^* to minimize the objective function (2) by an iterative algorithm [15]; iii. Align each pre-shape Z_k along its orbit to the mean pre-shape according to (3). We note in passing that any shape, including those out of the database, can be embedded as $\gamma([Z]_s) = e^{j\theta} [u]_p$, where $\theta = \arg \mu^* Z'$ in C^m as long as u^* exists.

We now introduce distances to compare partial shapes. By comparing partial shapes, we mean that we have in possession the full set of landmarks, but are interested in only comparing the relative shape of a subset of the landmarks.

Weighted Partial Procrustes Distance: Let W be a $m \times m$ positive square full rank matrix. Then a weighted Procrustes distance between $[Z_k]_s$ and $[Z_l]_s$ can be defined according to $d_{WP}([Z_k]_s, [Z_l]_s) = \min_{\theta} |W Z_k - W Z_l e^{j\theta}| = \sqrt{Z'_k W' W Z_k + Z'_l W' W Z_l - 2|Z'_k W' W Z_l|}$.

Weighted Euclidean Distance in the Embedding Space: After the embedding procedure, all shapes in the database are embedded in C^m . We can represent each embedded shape as a vector $\gamma([Z]_s) = e^{j\theta^*} Z$ for any configuration z , and the Euclidean distance $d_E([Z_k]_s, [Z_l]_s)$ can be taken as the shape distance after embedding. The *weighted shape distance* is defined as

$$d_{WE}([Z_k]_s, [Z_l]_s) = \sqrt{(\gamma([Z_k]_s) - \gamma([Z_l]_s))' W' W (\gamma([Z_k]_s) - \gamma([Z_l]_s))},$$

where W is again a weighting matrix with positive entries. In particular, if we set W equal to the diagonal matrix with some entries equal to 1 and the rest equal to a small positive number (say 0.1), then the weighted shape distance will mostly compare the shapes of those landmarks that correspond to the weight of 1. This a partial shape comparison emphasizing the portion of shapes that is more pathologically important.

4 Indexing Shapes

Because shapes are embedded in a Euclidean space and the relevant distance between them is a (possibly weighted) Euclidean distance, standard indexing techniques with adaptable similarity retrieval can be used to index shapes now. Specifically we use a kd-tree in C^m for indexing [4]. The kd-tree recursively partitions C^m and arranges the partitions in a tree. Each node of the tree represents a cube in C^m and the subtree represents the repartitioning of the cube. The leaves of the kd-tree contain data (the vectors $\gamma([Z_k]_s)$ in our case).

In this paper, users can choose different W weighting matrices to define various distances for shape retrieval using both complete shapes and different portions of shapes. Considering the retrieval efficiency, it is necessary to find a uniformly efficient indexing methods for different distance metrics that users choose. To support the user adaptability, we use the kd-tree, a space partitioning tree instead of metric access methods where they are only efficient for a fixed distance which is used for tree construction. Similarity retrieval in our kd-tree, supporting all the different distance metrics, is efficient since kd-tree is organized by non-overlap tight covers. For both complete and partial shape retrieval, the cubes that bound the embedded shapes in the kd-tree node have to be tested for intersection with the query ellipsoid centered at the query object, which differs in the weighting matrix W . We implement the user-adaptable similarity retrieval [18] in our kd-tree.

For performance comparison, we use two standard performance measures. The first is the average number of node tests per query and the second is the average number of surviving leaf nodes. The former represents the amount of calculation in the tree while the latter represents the amount of disk accesses.

5 Experimental Results

The key point of this paper is to support complete and partial shape retrievals efficiently with a kd-tree after embedding. Before we evaluate the efficiency of the indexing, we first demonstrate the validity of the new proposed partial distance metrics and the optimality of our embedding algorithm.

At the moment the vertebral boundaries in 546 images in NHANES II have been segmented. A total of 2812 boundaries are available. A subset of 94 images have been graded by an expert with respect to osteophyte severity. The grading is from 0 to 5, where “0” represents normal vertebrae without osteophyte; “1” indicates sharp protuberance that is barely visible; “2” means a short osteophyte with length less than 1/2 disk spacing; “3” implies longer and thicker osteophyte with length greater than 1/2 disk spacing; “4” and “5” are rare cases of large osteophytes that can bridge or extend to the next vertebra but have osteophyte that are straight or bent respectively.

Performance of Partial Shape Distances: As we mentioned before, osteophytes appear as localized bony outgrowths along the boundary of a vertebra. The severity of an osteophyte depends only on the shape of this outgrowth and

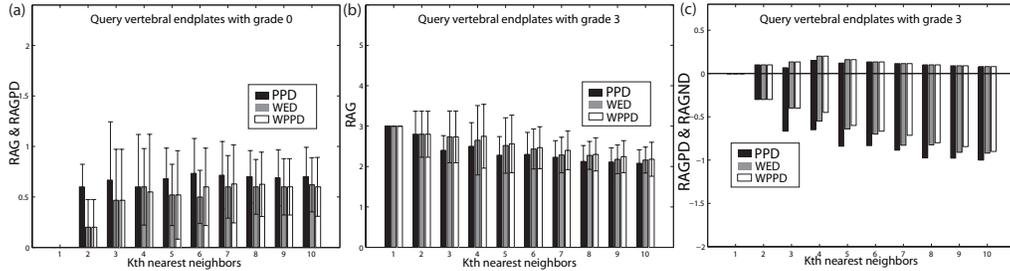


Fig. 3: Comparison of the efficacy of different distance metrics: PPD stands for partial Procrustes distance; WED stands for weighted Euclidean distance in the embedding space; and WPPD stands for weighted partial Procrustes distance.

not on the overall shape of the vertebra. Thus a retrieval system that is capable of using partial shape information (only from the osteophyte) is useful for NHANES II. In order to demonstrate the advantage of our partial shape distance metrics, we compare the retrieval results using partial Procrustes distance, weighted partial Procrustes distance and weighted Euclidean distance in the embedding space. Two groups of vertebrae, each containing 5 vertebrae with osteophyte grade 0 and 3 respectively, are chosen as query examples. We choose grade 0 and 3 vertebral shapes as queries because they are representative and we have an adequate number of them within the ranked image set.

In each group the vertebrae are ranked using different distances from the query and we study the expert grades of the retrieved vertebrae. For each $k \geq 1$, the average expert grade of all retrieved vertebrae with rank less than or equal to k is calculated. When compared to the query grade, this grade indicates for each k whether or not the retrieved vertebrae had a grade similar to the query. We call this average grade the running average grade (RAG). For each k , we also calculate the average positive difference between the grades of the retrieved vertebrae and the query grade (RAGPD) and the average negative difference between the grades of the retrieved vertebrae and the query grade (RAGND). These indicate, for every k , the bias in the retrieval, i.e. whether or not the shape similarity distance tends to favor more severe grades or less severe grades.

Fig. 3 shows the comparison for Grade 0 and 3 groups respectively. Since the first nearest neighbor is always the query example itself, we only analyze the trend of the rest of retrieved results. Since Grade 0 is the lowest grade, there is no negative difference for it and the average positive difference is the same as running average grade. We plot the mean RAGs and their standard deviations in Fig. 3(a) for grade 0. Comparing the results from complete and partial shape retrievals, we find that partial shape retrievals give better results. RAGs from both partial shape retrievals are lower than RAGs from complete shape retrieval for all k th nearest neighbors. Fig. 3(b) and (c) show the RAG and the RAGPD and RAGND for the grade 3 retrievals. In Fig. 3(c), we see that complete shape distance tends to retrieve more lower grade vertebrae than partial shape distance. This is because the complete shape distance uses the entire set of landmarks and considers any shape variation from the other parts

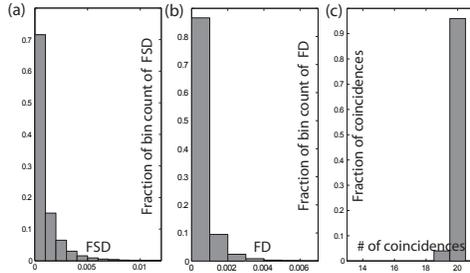


Table 1: Mean and Variance of FSD and FD of pairs from a randomly sampled set of 1000 vertebrae

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

Fig. 4: (a) Histogram of FSD (b) Histogram of FD (c) Histogram of $S_P \cap S_E$

Fig. 4: Metric distortion after optimal shape embedding

of the vertebra just as relevant. This experiment quickly reveals that the weighted partial shape distances outperformed the complete shape distance: The weighted shape distance groups together more vertebrae with similar expert grades. This is reasonable since the expert grade depends only on the local osteophyte shape rather than the overall shape of the vertebra.

Comparing Embedding Distance with Partial Procrustes Distance:

Next, we experimentally evaluate the closeness of the Euclidean distance after embedding to the partial Procrustes distance.

To measure the similarity between the two distances we calculate fraction squared difference $FSD = (d_E^2([Z_k]_s, [Z_l]_s) - d_P^2([Z_k]_s, [Z_l]_s)) / d_P^2([Z_k]_s, [Z_l]_s)$ and fractional difference $FD = |d_E([Z_k]_s, [Z_l]_s) - d_P([Z_k]_s, [Z_l]_s)| / d_P([Z_k]_s, [Z_l]_s)$ for a randomly sampled set of 1000 vertebrae from the database. The average and standard deviation of the FSD and FD are given in Table 1 and histograms of FSD and FD are shown in figure 4(a), (b). From the table and the figures, it is clear that the Euclidean distance following optimal embedding is very similar to the partial Procrustes distance.

Next, we compare the nearest neighbor structure of the data set before and after embedding as follows: From the set of 1000 vertebrae used in the above experiment, 100 vertebrae are chosen as query vertebrae. For each query vertebra, the set of 20 nearest vertebrae is found according to the partial Procrustes distance d_P and the embedded Euclidean distance d_E . Let these sets be S_P and S_E respectively. Then $S_P \cap S_E$ is the set of vertebrae that are common to both retrievals in the query. The distribution of the number of elements in $S_P \cap S_E$ over the 100 queries is shown in figure 4(c). For 97 of 100 queries the sets S_P and S_E were identical, and for 3 queries they differed by a single image.

These experiments show that the embedded Euclidean distance is a good approximation to the partial Procrustes distance and can be used in shape retrieval queries.

Retrieval Performance of KD-Tree: As mentioned before, the performance of the indexing kd-tree is measured using the average number of node tests and surviving leaf nodes. We randomly sample the 2812 shapes into sets of size 434, 1089, 1654 and 2812. Each set is indexed for shape with the algorithms described above. Every point in the database is used as a query image and 10-nearest neighbor vertebral images are retrieved. Fig. 5 plot the perfor-

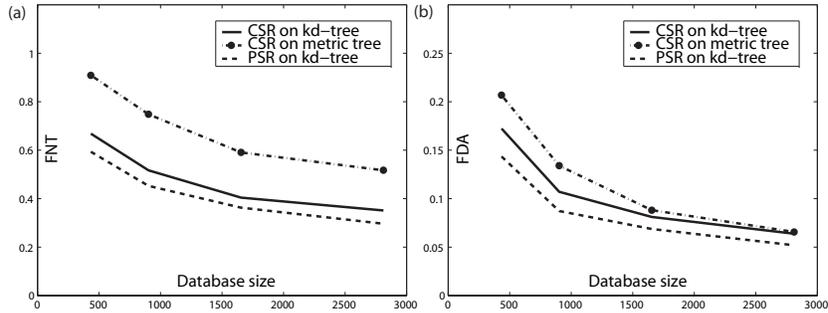


Fig. 5: Comparison of indexing performance:(a) Ratio of average number of node tests to the database size (FNT); (b) Ratio of average number of surviving leaf nodes to the database size (FDA). CSR stands for complete shape retrieval; PSR stands for partial shape retrieval.

mance measures as a function of the database size. The performance measures are expressed as percentages. The percentages should remain constant for an indexing scheme with linear complexity and should decrease with the size of the database for sub-linear complexity. Both figures suggest that the indexing procedure has sub-linear complexity. Thus, the procedure is effective in indexing shapes in the embedding space and scales up well with the size of database. For complete shape retrieval, we also compare the performance from our kd-tree and a hierarchical clustering tree. We find that our kd-tree is more efficient. For the performance of user-adaptable similarity retrieval on the kd-tree, we compare the plots for complete and partial shape retrieval. The plots demonstrate that our kd-tree support both complete and partial shape retrieval and shape indexing in the embedding space is efficient.

Figure 6 shows a query example using our algorithm. The leftmost image in each row is the query image and the successive images are the neighbors retrieved by the indexing algorithm ranked in increasing shape distance from the query. Here too it is clear that partial shape retrieval gives better results.

6 Conclusions

The space of all shapes of landmarks is strongly non-Euclidean. Nevertheless, the shapes can be indexed for complete and partial shape similarity by embedding them in a vector space. In this paper, we proposed such an embedding and demonstrated experimentally that the proposed indexing has sub-linear complexity for both complete and partial shape similarity retrieval.

Several extensions of this idea are possible. One alternative is to use density-based clustering methods [3] to derive a piecewise embedding for data which are not tightly clustered. We hope to investigate these and other possibilities in the future.

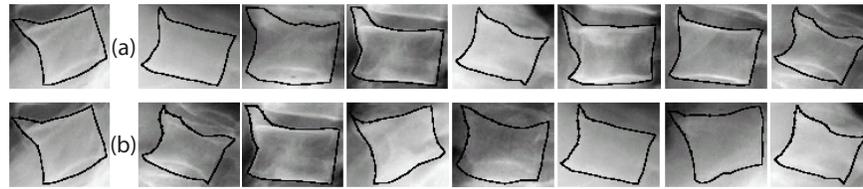


Fig. 6: Shape query example: (a) Complete shape retrieval; (b) Partial shape retrieval

References

1. Bick EM. Vertebral osteophytosis: pathologic basis of its roentgenology. *AJR*73:979-993,1955.
2. E. Chávez, et. al., *Searching in Metric Spaces*, *ACM Comp. Surveys*, 2001.
3. M. Ester, et. al., *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, *KDD 1996*.
4. J. Friedman, J. Bentley, and R. Finkel, *An algorithm for finding best matches in logarithmic expected time*, *ACM Trans. Math. Software*, pp. 209-226, 1977.
5. V. Gaede, O. Gunther, *Multidimensional Access Methods*, *ACM Comp. Sur.*,1998.
6. G. R. Hjaltason and H. Samet, *Properties of Embedding Methods for Similarity Searching in Metric Spaces*, *IEEE Trans. on PAMI*, pp.530-548, 2003.
7. D. G. Kendall, D. Barden, L. He, *Shape and Shape Theory*, *Wiley Series*, 1999.
8. Z. Lei, et. al., *Computationally Fast Bayesian Recognition of Complex Objects based on Mutual Algebraic Invariants*, *Proc. IEEE Int. Conf. on Image Proc.*, 1995.
9. I. L. Dryden and K. Mardia, *Statistical Shape Analysis*, *J. Wiley*, 1998.
10. R. Malladi, J. A. Sethian, B. C. Vemuri, *Shape Modeling with Front Propagation: A Level Set Approach*, *IEEE Trans. on PAMI*, 17(2), pp. 158-175, 1995.
11. T. McInerney and D. Terzopoulos, *Deformable Models in Medical Image Analysis: A Survey*, *Medical Image Analysis*, 1(2), pp. 91-108, 1996.
12. F. Mokhtarian, et. al.. *Robust and Efficient Shape Indexing Through Curvature Scale Space*, *Proceedings of BMVC*, pp. 53-62, 1996.
13. H. Muller, et. al., *A review of content-based image retrieval systems in medical applications - clinical benefits and future directions*, *Int. J. Med. Inform.*, 1-23, 2004.
14. E. Petrakis, et. al., *Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming*, *IEEE Trans. on PAMI*, Vol. 24, No. 11, 2002, pp. 1501-1516.
15. X. Qian, H. D. Tagare, *Optimal Embedding for Shape Indexing in Medical Image Databases*, *MICCAI 2005*.
16. G. Robinson, et. al., *Medical Image Collection Indexing: Shape-Based Retrieval Using KD-Trees*, *Comp. in Med. Img. Graph.*, 20(4), pp. 209-217, 1996.
17. Y. Rui, et. al., *Image retrieval: current techniques, promising directions and open issues*, *Journal of Vis. Comm. Img. Rep.*, pp. 1-23, 1999.
18. T. Seidl, H-P Kriegel, *Efficient User-Adaptable Similarity Search in Large Multimedia Databases*, *VLDB 1997*.
19. A. W. M. Smeulders, et. al., *Content-based image retrieval at the end of the early years*, *IEEE Trans. on PAMI*, pp. 1349-1380, 2000.
20. H. D. Tagare, *Deformable 2-D Template Matching Using Orthogonal Curves*, *IEEE Trans. on Med. Imaging*, Vol. 16(1), pp. 108-117, 1997.