Indexing Vertebrae in Shape Space

Medical Informatics Training Program
Final Report

Xiaoning Qian
Mentor: Rodney Long

I. Introduction

It has now become common practice to convert text, sound, images, and even videos into digital libraries. As these digital libraries from different resources are communicated and shared across computer networks, an important problem arises: how to efficiently interpret and extract useful information from the massive data for different users. Efficient content-based similarity retrieval in large image databases gains considerable research attention as an important functionality for many applications.

There are at least the following three problems to solve for developing a complete content-based image retrieval system: 1) A robust feature space should be selected to represent the visual appearance of images; 2) Some appropriate similarity/distance measure should be defined between images in this selected feature space; 3) An indexing and retrieval scheme is necessary to speed up similarity retrieval on large size image databases.

Images always contain much more information than text. But how to represent visual information presented in images in digital format is a problem. If the visual content of images can be fully captured by computers, we will have a flexible and intelligent image retrieval system. But the representations of images are complex and the procedure of extracting the representations from images are time consuming. There will be always a trade-off between the information we want to extract, which determines image retrieval capabilities, and the model complexity of feature representation, which affects the time required for the extraction procedure and later the similarity search. The computer vision community is still focusing on this image understanding problem. But still, we are not at the stage to provide a genuine solution. Therefore, most CBIR systems still rely on lower level, computationally attainable image features (colors, textures, shapes of objects in images, etc.) to measure the similarities between images.

Most current CBIR systems still use exhaustive, sequential search for similarity search. This is definitely not efficient as the size of image database increases. An efficient indexing structure will support the development of an effective image database. But most feature spaces we select are high dimensional spaces, where the retrieval performance of indexing structures could even be worse than the exhaustive search if the data is uniformly distributed. This phenomenon is known as the " curse of dimensionality".

In this paper, we concentrate on indexing spine X-ray images in the NHANES II data set. In these images, the most dominant visual information is shape. It is known that some shape features are crucial for clinical diagnosis for related diseases. For example, in Figure 1, we showed a cervical vertebra with a lower anterior osteophyte, which is significant to osteoarthritis researchers. Before we find the shape representations, we first need to extract contours of organs or abnormal regions presented in medical images. Thus, we obtain a collection of boundary points recorded as a sequence of x and y coordinates. Applying shape space theory, we can measure the rotation, translation, scaling, and starting point shift invariant distance between shapes and develop an efficient indexing tree to support content-based similarity retrieval. Eventually, such systems may become significant tools in the clinical diagnosis of human back pain. Research shows that the shape distribution in shape space is the main factor affecting the retrieval performance of our indexing tree. If our shapes are uniformly distributed in shape space, our retrieval performance would be severely constrained by the curse of dimensionality. Fortunately, similarity retrieval on our indexing tree is always more efficient than doing a sequential search since the vertebral shapes in our database is non-uniformly distributed and have low intrinsic dimension.

In the following, each step of our shape indexing technique is described in detail.

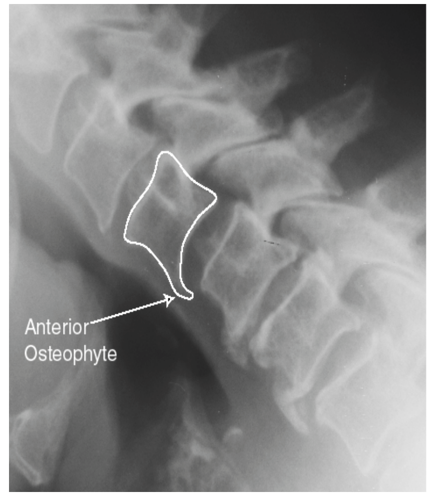


Figure 1. A cervical vertebrate with anterior osteophyte

II. Modified Active Contour Segmentation of spine X-ray images

We first need to extract shape information from spine X-ray images, which generally have low contrast and low image quality. To achieve this, we developed a dynamic programming algorithm for the extraction of vertebral contours. The algorithm integrates prior shape information, given by a small number of expert landmarks (Figure 2), into the active contour segmentation (ACS) framework. The procedure of extracting vertebral contours in X-ray images, consisting of feature detection, dynamic programming searching and later user validation and visualization, has been streamlined and automated.

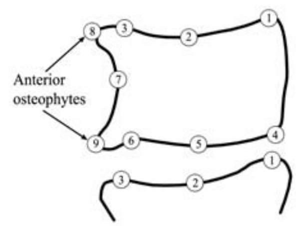


Figure 2. Expert landmarks

Active Contour Segmentation deforms an a priori curve to find the optimal contour which minimizes an objective function, consisting of external energy and internal energy terms:

$$E(c) = E_{ext}(c) + E_{\text{int}}(c) = w_{ext}\int_0^1 -\nabla I(c)d\alpha + w_{\text{int}}\int_0^1 \|c\| d\alpha + w_{prior}\int_0^1 \|c - c_{prior}\| d\alpha$$

where c is the deformed contour, c_{prior} is the prior shape information and the w's are appropriate weights for the respective energy terms. It is time consuming to find the optimal contour if the size of image is large. Therefore, we constrain our contour search grid to orthogonal curves as suggested in [1]. By writing the above energy function in a discrete form, we find that the net energy on one sample point along the deformed curve is computable by using only its neighboring sample points. Therefore, we can speed the minimizing procedure by dynamic programming. We will give some segmentation results in section V.

III. Shape Space

Shape in this paper is defined as the geometric information after removing translation, rotation, scaling effects. Given a set of m landmarks extracted from a contour, with the contour being in p dimensional space, we treat the set as a matrix X_{pm}. The "shape set", a set of sets of landmarks with the same shape, is generated by translation, rotation and scaling. With this abstraction, we define a shape as the shape set: $\{S == \{\mathrm{s}R\mathrm{X} + \mathrm{t}, \mathrm{s} \in \mathrm{R}, \mathrm{t} \in \mathrm{R}^{\mathrm{p}}, R \in SO(p)\}\}$.

Since any polygon (such as a set of landmarks) is represented by a matrix, we can remove the translation factor by moving the origin to the mass center of the polygon. This can be realized by pre-multiplying by a Helmert matrix [4]. We can then remove the scaling factor by normalization. After these two steps, we obtain the so-called *pre-shape space*. This pre-shape space is actually the orbit space of the configurations of m landmarks in R^{p} under the action of translation and scaling. Its dimension is $(p-1)m-1$ and it can be considered as a sphere. Finally, we need to remove the rotation information to get the *shape*. In order to do that, we identify all rotated versions of the pre-shape with each other. The *shape space* now is the space of orbits on the pre-shape space and its dimension is $(p-1)m-1-\frac{m(m-1)}{2}$. For planar shapes, shape space in fact is the well defined complex projective space CP^{2m-2}, which is not an ordinary Euclidean space.

After having a rough idea of what the shape space is, we need to further explore how to define the distance between two shapes. For our vertebra, we consider $p=2$. The problem to calculate the distance between two 2-D shapes is formulated as a complex linear regression problem. If m landmarks are used to represent the object contour, the contour becomes the complex vector $C=[x_1+jy_1, x_2+jy_2, \ldots, x_m+jy_m]^T$, where (x_i, y_i) is the ith coordinate for one landmark on the contour. Thus the problem can be written as following:

$$d_F(C_1, C_2) = \inf_{\Gamma} \left| C_2 - \Gamma(C_1) \right|,$$

where C_1 and C_2 are the $m-D$ complex vectors representing contours of m points; $\Gamma(C_1) = \lambda C_1 + t, \lambda, t \in \mathrm{C}$. $\Gamma(C_1)$ represents translating, rotating and scaling the contour C_1. We can therefore define the *shape* as a set of all the possible contours represented by $\Gamma(C_1)$. The distance we obtained is called *full Procrustes distance* and has closed form [4]: $d_F = \sqrt{1-|\sum_{j=1}^{m} z_1^j \bar{z}_2^j|^2}$. z_1, z_2 are known as pre-shapes derived from C_1 and C_2; $\bar{z}_2^j$ is the conjugate of z_2^j. The Procrustes distance $\rho = a\sin(d_F)$ is used in our spine image database application described later and is easy to compute. The Procrustes distance is actually a Riemannian metric in shape space, which in fact is the well-known *Fubini-Study* metric in classic geometry. This distance can also be considered as the geodesic distance between shapes in shape space. It is also provable that we can find a starting point invariant distance metric based on this Procrustes distance.

IV. High Dimensional Indexing

Our vertebral shape database supports both range query and k nearest neighbor (k-NN) query. Given a query shape q, we want to retrieve all the "similar" vertebral shapes from a data set embedded in shape space. Before defining the query problem, we first need to have a mathematical metric $d(,)$ to measure the distance or similarity between two shapes. We use the Procrustes distance.

(Radius-)range query retrieves data points within distance r_T to the query example q:

$$\{u \mid d(u,q) < r_T, u \in DB\}.$$

The k-NN query finds k closest data points to q in the database, that is,

$$\{u_i \in A \mid d(u_i,q) < d(x,q), \forall x \in DB \setminus A, A \subset DB, |A| = k\}.$$

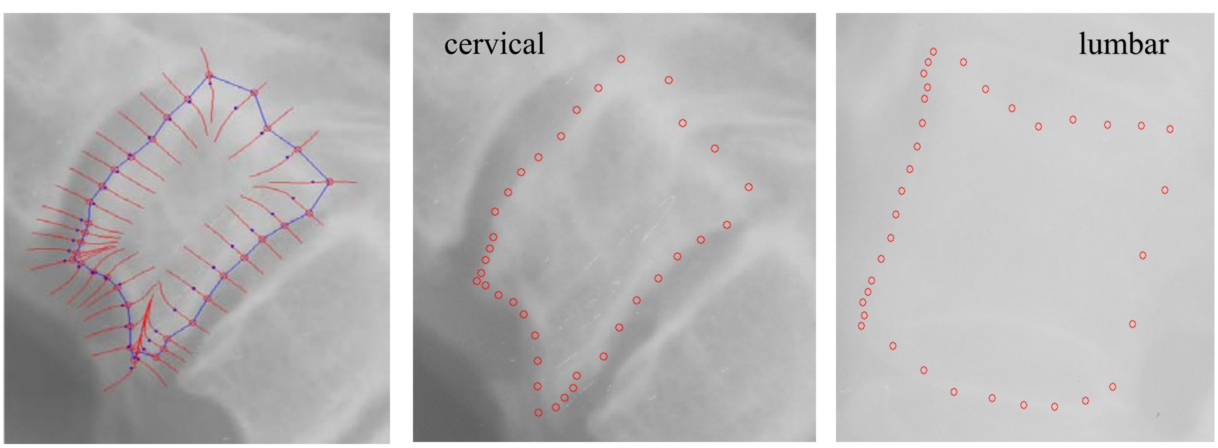
Since shape space is not an ordinary Euclidean space, only indexing trees in metric spaces can be applied to index shapes in shape space. We use the agglomerative cluster tree in our experiments. The indexing problem can be considered as the problem of organizing a data set DB with size n in shape space. The data set will be organized hierarchically for storage; later desired data may be efficiently retrieved. The objective of using indexing trees is to retrieve desired data with retrieval time sub-linear to n. The procedure of the range query on indexing trees traverses the tree from the root to the leaf nodes by *node test*, which evaluates the lower bound of the distance from the data points contained in the intermediate nodes to the query example q and compares it to r_T. By the triangle inequality of our shape distance metric, we are able to find all the points satisfying the query. If the lower bound is greater than the threshold r_T, this node cannot contain any points satisfying the query, and consequently the sub-tree rooted at this node need not be searched further. If the lower bound calculated by the node test is less than or equal to r_T, the subset represented by the node may contain points that satisfy the query and must be explored further. The number of node tests required for the retrieval can be used to evaluate the retrieval efficiency. In high dimensional spaces, the expected nearest neighbor distance increases and the variance of distances is very small. Hence, those intermediate nodes of indexing trees will have high probability of passing the node tests for given queries. However, we have experimentally determined that our vertebral shapes are not uniformly distributed. The intrinsic dimension of vertebral shapes is much lower than the external dimension of shape space. Hence, using our shape indexing tree, the similarity retrieval is still more efficient than doing sequential search.

Since we needed a C program for shape indexing to efficiently handle indexing data structures, we needed to find a way of interfacing it to our MATLAB CBIR package. To accomplish this, we wrote two MEX wrappers for constructing the indexing tree and for similarity retrieval, respectively. We also needed one "gateway function" for maintaining the C data structure in memory during program execution. The related technical report can be requested from CEB.

V. Results

A. Shape Extraction

We improved the previously-implemented ACS algorithm by generating a new search grid, with better geometry to avoid segmentation problems at corners of vertebrae. We generated the contour search grid by solving 2 partial differential equations using the method described in [1]. After a conformal map, we can get an "orthogonal search grid" (orthogonal curves) for the



(a) (b) (c)

Figure 3. (a) A generated orthogonal search grid; (b) A segmentation of cervical vertebrate; (c) A segmentation of lumbar vertebrate

contour search space in the ACS algorithm. We currently use 9 expert landmarks to construct the initial templates and search grids for all images. Figure 3 shows a generated orthogonal search grid and two segmentation examples for both cervical and lumbar images. We also implemented a GUI to use several representative templates and the corresponding search grids to speed the segmentation procedure. (Also, we haven't pre-processed the X-ray images before the segmentation. Integrating image enhancement is expected to improve segmentation results.)

B. Modified CBIR Package

The CBIR software has a MySQL database to store the available NHANES II data set. We integrated two C applications for shape indexing and used a gateway function to maintain the indexing tree in memory for the similarity retrieval. Current CBIR software now supports both text queries and shape queries. These are applied separately since they are in different logical spaces. The system will be more efficient if we can find a seamless way to integrate them.

We also developed Graphic User Interfaces in MATLAB to support shape queries. Using this GUI, we output similar images (vertebral shapes) together with the user-required text information retrieved from the MySQL database.

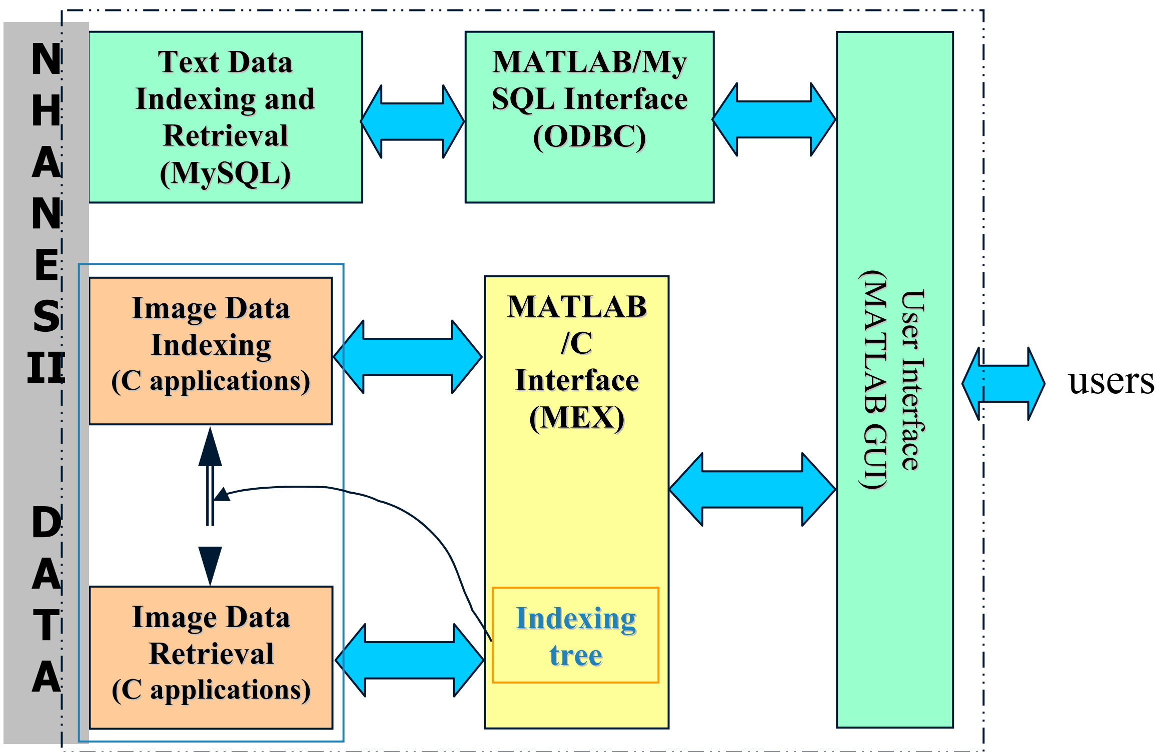


Figure 4. Sketch of modified CBIR package

C. Accuracy of Shape Indexing

We did several experiments for testing the accuracy of our shape indexing technique. 34 sample points were used to represent vertebral shapes in the following experiments. We provide one set of 10-NN query results in Figure 5. Most of the 10 images retrieved are similar to the query example. With one exception all of the retrieved shapes are lumbar vertebrae, like the query example. This indicates that our shape distance is a good discriminator for the global difference between cervical and lumbar vertebrae.

By sequentially aligning and comparing every shape in the database with the query example after alignment, we found that ranking of results obtained with our indexing tree is geometrically correct (Figure 6). This was predicated, since our distance measure is a metric. However, perceptually, the user might not be satisfied with these results. For example, the user might be interested in finding osteophytes in the images in Figure 5 and may not be happy with the last two images. This is because our shape distance still only measures the difference of *global* shape

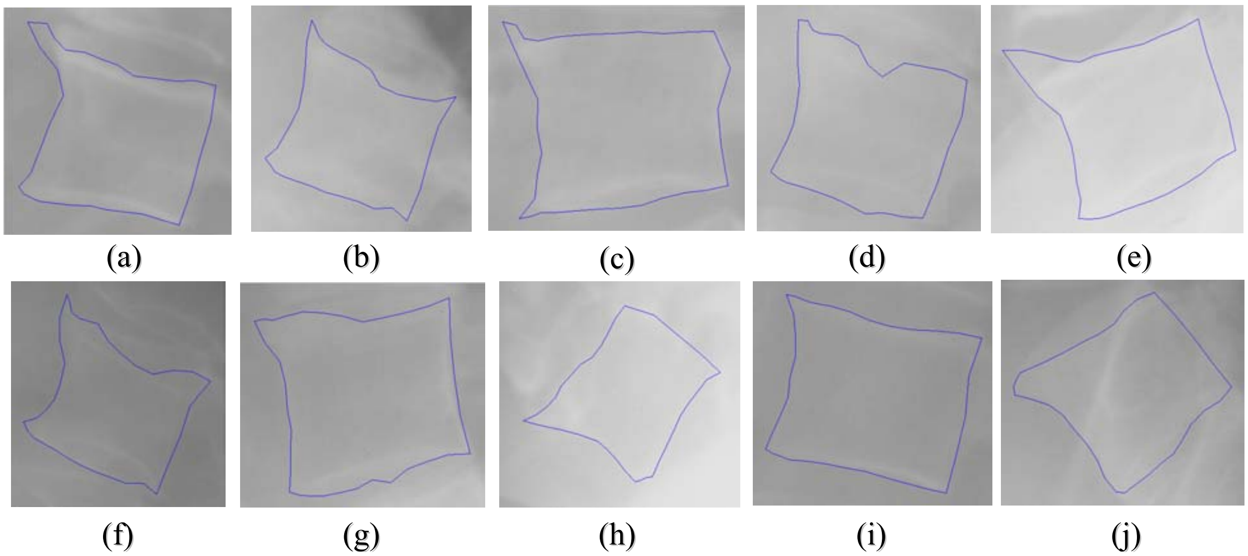


Figure 5. 10-NN results: (a) Query shape and 1-NN (d = 0); (b) 2-NN (d = 0.073); (c) 3-NN (d = 0.08); (d) 4-NN (d = 0.082); (e) 5-NN (d = 0.083); (f) 6-NN (d = 0.087); (g) 7-NN (d = 0.088); (h) 8-NN (d = 0.089); (i) 9-NN (d = 0.0898); (j) 10-NN (d = 0.0901)

properties. We suggest those improvements to the system: First, improve the correspondence between segments of different contours after segmentation. Second, let user be able to adapt the distance measure according to his specific criteria for the similarity retrieval. For instance, using partial shape matching, which only considers matching the (user-defined) interesting parts of different shapes, will surely produce retrievals more satisfying to the user.

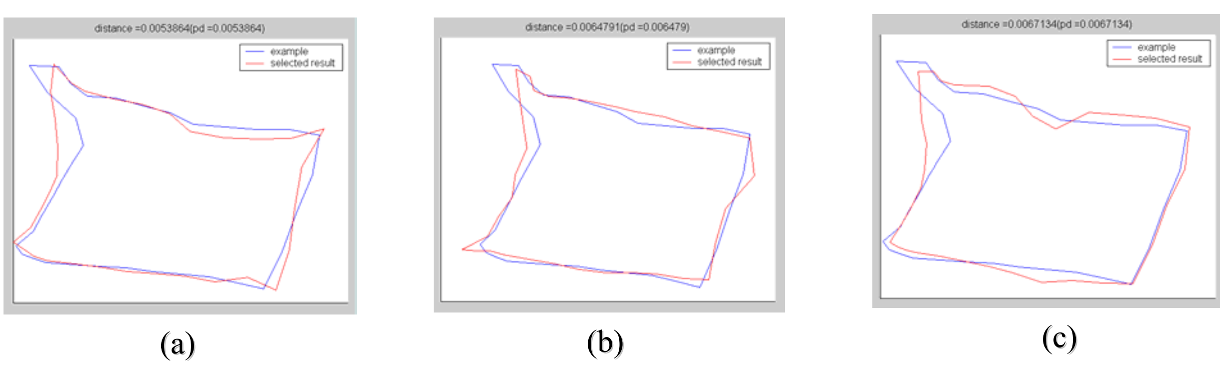


Figure 6. Sequential comparison between database shapes and query shape: (a) 2-NN in figure 5; (b) 3-NN in figure 5; (c) 4-NN in figure 5

D. Efficiency of Shape Indexing

For testing efficiency of our shape indexing tree, we measured the average number of node tests and average number of candidate leaf nodes. These are related, respectively, to the CPU distance calculation time and the number of disk accesses. In this example, we have 1298 shapes in our database. We tested several 13-NN and 40-NN queries to get the average numbers. We found that the numbers obtained are lower than if we did sequential search. We also tested two databases with different sizes and found that the indexing tree performance appears to scale up well with the size of database.

1298 shapes	Efficiency	
k-NN (k)	Average number of node tests	Average number of candidate leaf nodes (disk accesses)
13	996.5 (linear search: 1298)	176 (linear search: 1298)
40	1250.5 (linear search: 1298)	315.25 (linear search: 1298)

Table 1. Efficiency of shape indexing tree (CPU cost and number of disk accesses)

From the results, we believe that the intrinsic dimension of shape distribution is not very high. Recall that every vertebral contour is represented by 34 sample points; therefore the dimension of our shape space is 64. Considering "the curse", search with our indexing tree will definitely be worse than sequential search if the shapes are uniformly distributed. However, our results show that the search using our tree is better than linear, i.e. better than sequential search. Based on this observation, we can further improve our indexing technique by considering the clustering property of shapes. Also, if we apply the tree adaptation procedure given in [5], the number in the first column can be further reduced. In that case, our shape indexing will be always better than linear search.

VI. Conclusion and Future Work

We applied shape space theory to analyze different vertebral shapes, and we applied one distance metric defined in shape space for constructing shape indexing tree and developing the corresponding similarity retrieval algorithm. Using our shape indexing tree, we showed that the similarity retrieval performance is always better than linear search.

There are still many issues we could pursue. First, an implicit assumption for using shape space theory is that we have a one-to-one correspondence between sample points along different vertebral contours. But in practice, it is not easy to find the correct correspondence between those sample points, and solving the correspondence problem is time consuming. Integrating a correspondence invariant distance measure in shape indexing is a potential new research project. Second, partial shape matching could be useful for user-adaptive similarity retrieval. But for retrieval efficiency, we need to find the efficient indexing method for partial shapes. We have recently derived a weighted Procrustes distance metric, and we are still researching an appropriate indexing technique.

For spine X-ray images, other features like disk space narrowing and subluxation are also important for clinical use. Those features are more closely related to spatial relationship between multiple vertebrae in images. To CBIR to these features, it is crucial to extract and represent the corresponding shape information and develop an effective retrieval algorithm using an adaptive similarity measure.

Acknowledgements

I would like to express my gratitude to Dr. Alexa McCray and Dr. George Thoma for allowing me to work on this interesting project and in this excellent working environment. I would like also thank Mr. Rodney Long and Dr. Sameer Antani for their help and guidance. The well-written original CBIR software makes my work much easier. I would also like to thank Ms. May Cheh and other staff for making my visit at LHNCBC an enjoyable experience.

References

1. Hemant D. Tagare, "Deformable 2-D Template Matching Using Orthogonal Curves", In IEEE Trans. On Medical Imaging, Vol. 16, No. 1, pages 108-117, 1997.
2. A.K. Jain and A. Vailaya. "Shape-Based Retrieval: A Case Study with Trademark Image Databases." Pattern Recognition, 31(9):1369-1390, 1998.
3. A. Del Bimbo and P. Pala. "Shape Indexing by Multi-Scale Representation." Image and Vision Computing, 17(3-4):245-261, 1999.
4. D. G. Kendall, D. Barden, Carne, L. He, " Shape and Shape Theory, " Wiley Series, Reading, 1999.
5. Xiaoning Qian, Hemant D. Tagare, "Optimally Adapted Indexing Trees for Medical Image Databases," ISBI, Washington DC, 2002