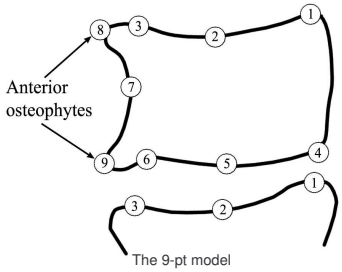
Pre-Indexing for Fast Partial Shape Matching of Vertebrae Images

Xiaoqian Xu, D.J. Lee
Dept. of Electrical & Computer Engineering
Brigham Young University
Provo, UT, USA

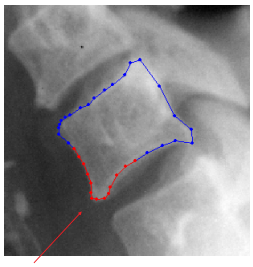
S. Antani, L.R. Long
Communication Engineering Branch
National Library of Medicine, NIH
Bethesda, MD, USA

1



The 9-pt model

2



A partial query

3

Partial Shape Matching

1. **It allows the user to pick a partial query, which could be a pathological part; and it retrieves similar partial shapes from database.**
2. **Multiple Open Triangles is employed as partial shape representation.**

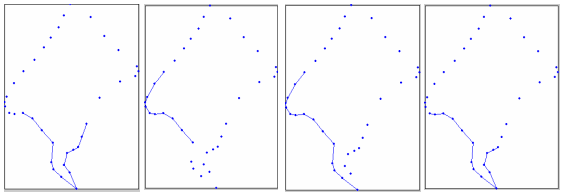
   Xiaoqian Xu, D.J. Lee, S. Antani, and L.R. Long, "A Spine X-ray Image Retrieval System Using Partial Shape Matching", *IEEE Transactions on Information Technology in Biomedicine*, in revision.
3. **Dynamic Programming is implemented as the searching strategy for retrieval.**

Challenges:

4

Problems:
1. **As the database grows, sequential search becomes time-consuming.**
2. **The flexibility in selecting a partial query makes it hard for indexing.**

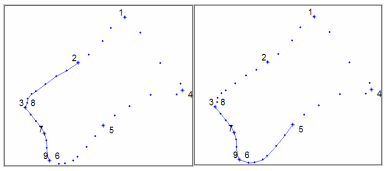
5



There is an unmanageably large number of partial queries, which makes the pre-calculation infeasible.

6

Solution:



Xiaoqian Xu, D. J. Lee, S. Antani and L. R. Long, "Localizing Contour Points for Indexing an X-ray Image Retrieval System", in Proc. 16th IEEE Symposium on Computer-Based Medical Systems, New York, pp. 169 – 174

Based on 9-pt model, Two partial queries (superior and inferior corners) are selected from each shape for pre-indexing

$$P = \sum_{i=1}^{n} \left\| \begin{bmatrix} S\cdot\cos\alpha & -\sin\alpha & T_x \\ \sin\alpha & S\cdot\cos\alpha & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_A - \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}_B \right\|^2$$

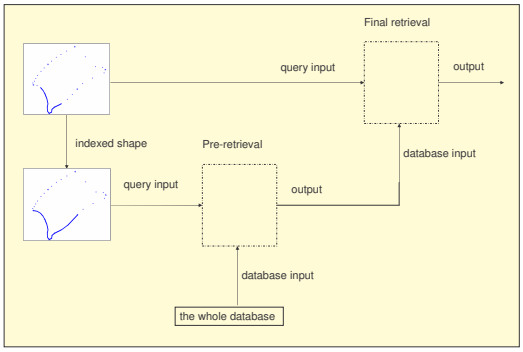
Procrustes distance expressed above is calculated between each pair of the selected partial queries.

7

Agglomerative clustering algorithm

1. **Initially, put each indexed shape in its own cluster.**
2. **Among all current clusters, pick the two clusters with the smallest distance.**
3. **Replace these two clusters with a new cluster.**
4. **Repeat Step 2 and 3 until there is only one remaining cluster.**

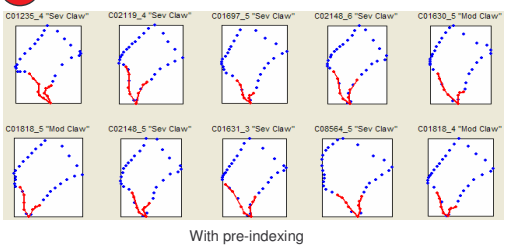
8



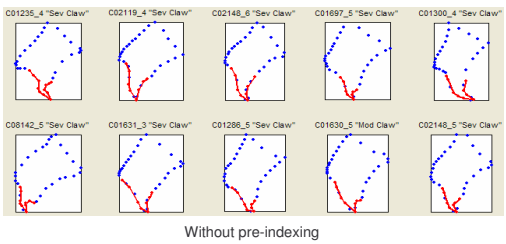
Retrieval process with pre-indexing

Results:

9



With pre-indexing



Without pre-indexing

10

	Severity				*Type*		
	Severe	Moderate	Slight	All queries	Claw	Traction	All queries
With Pre-indexing	55%	61%	81%	69%	80%	86%	84%
Without Pre-indexing	60%	65%	85%	73%	94%	79%	84%

Comparison of Recall Results of Both Approaches

	With Pre-Indexing	Without Pre-Indexing
Sec/query	20	58

Time Efficiency