Enlarging Nodes to Improve Dynamic Spatial Approximation Trees

Marcelo Barroso - Universidad Nacional de San Luis

Nora Reyes – Universidad Nacional de San Luis

Rodrigo Paredes – Universidad de Talca

Outline

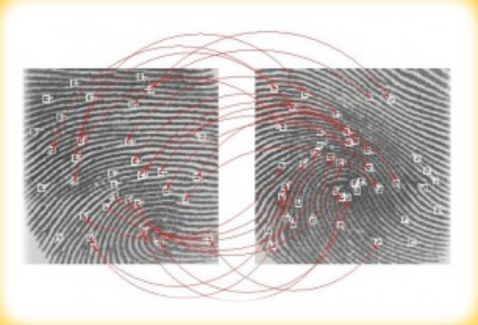
- Introduction
- Dynamic Spatial Approximation Trees: *DSA-tree*
- Our Proposal: *DSACL-tree*
- Experimental Results
- Conclusions

Introduction

- Similarity searching has applications in many fields, such as multimedia databases, text retrieval, etc.

Introduction

- To answer similarity queries the dataset is preprocessed so as to build an index that reduces query time.
- Most of the existing indexes are static.
- Similarity computation can be expensive.

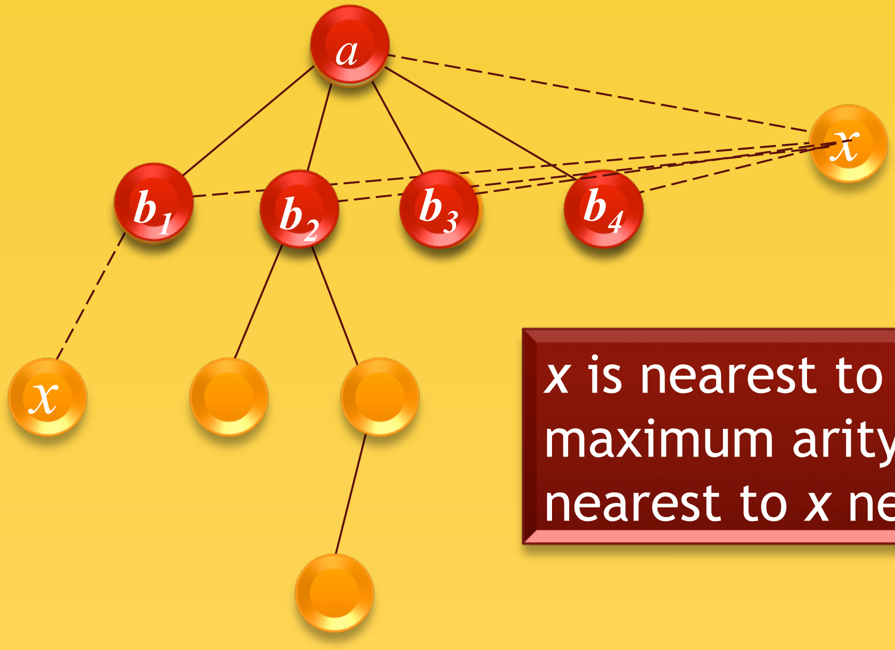


Dynamic Spatial Approximation Trees (*DSA-tree*)

- We consider the version called *timestamp with bounded arity*.
- The DSA-tree is built incrementally via insertions.
- Range searching replicates the insertion process of relevant elements.

DSA-tree

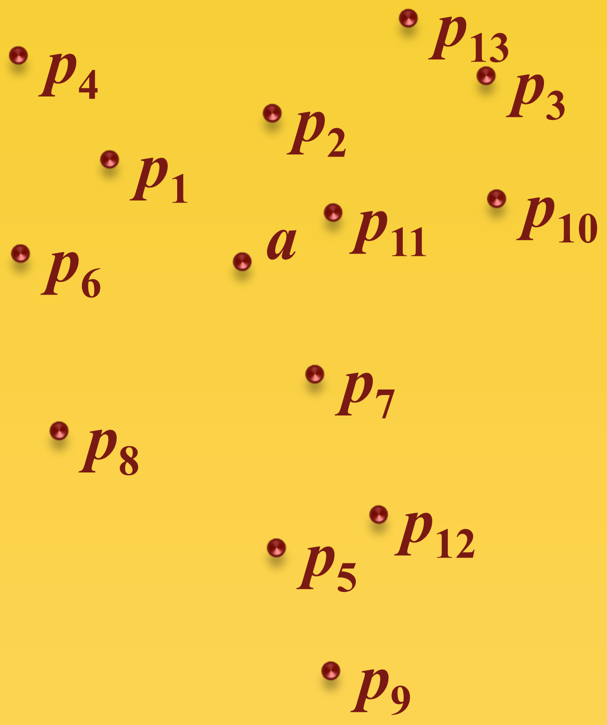
- Insertions:

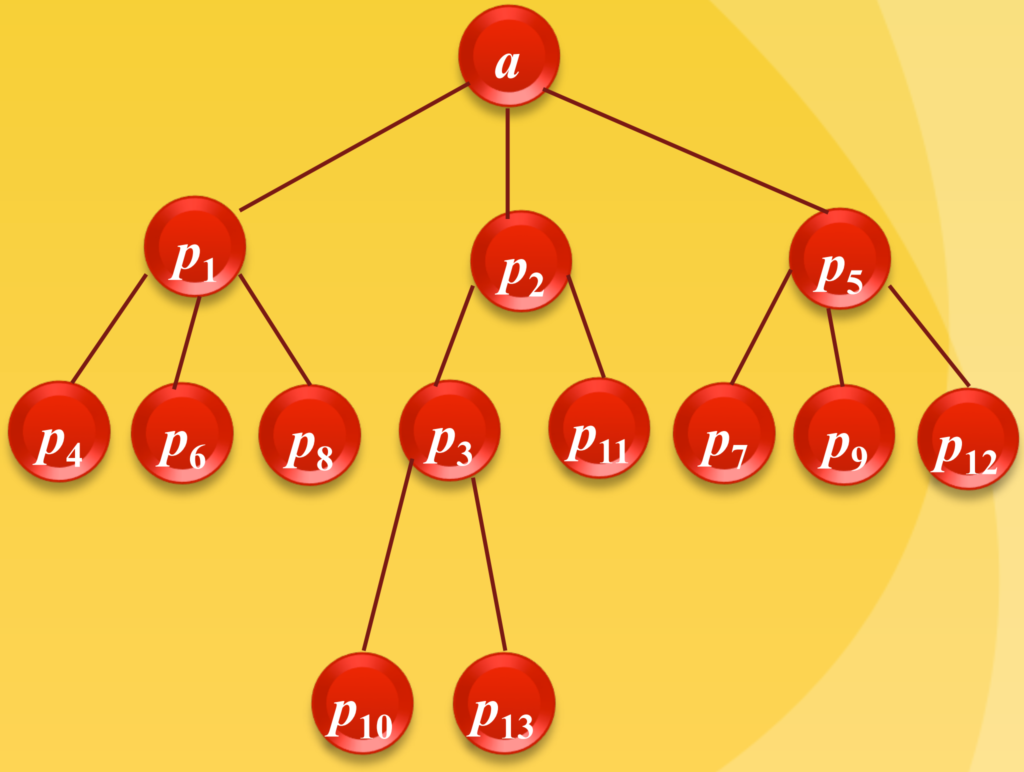


$A = 4$

x is nearest to node *a*, but *a* has maximum arity. So, it chooses the nearest to *x* neighbor of *a*.

DSA-tree: Example





$A = 3$

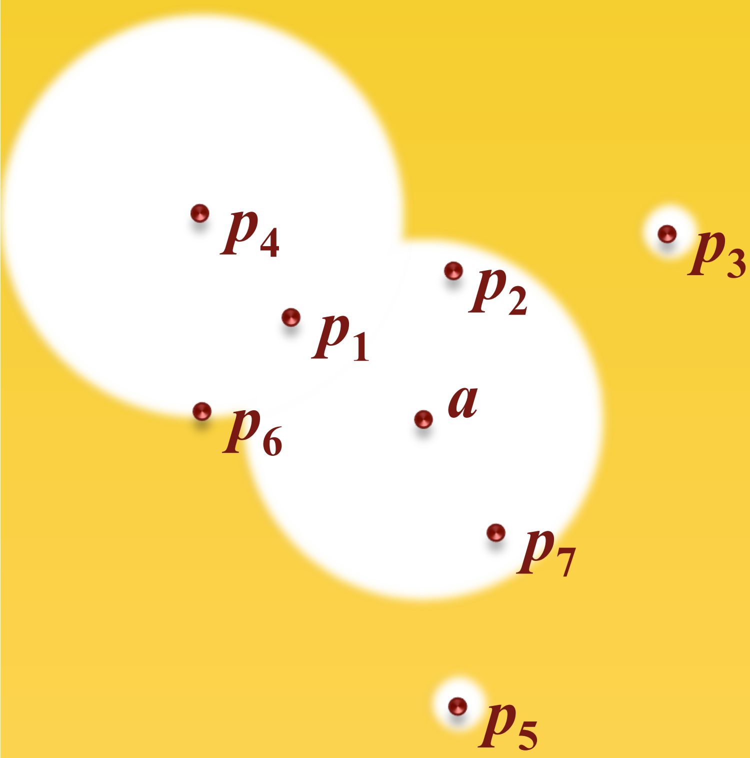
Our Proposal: *DSACL-tree*

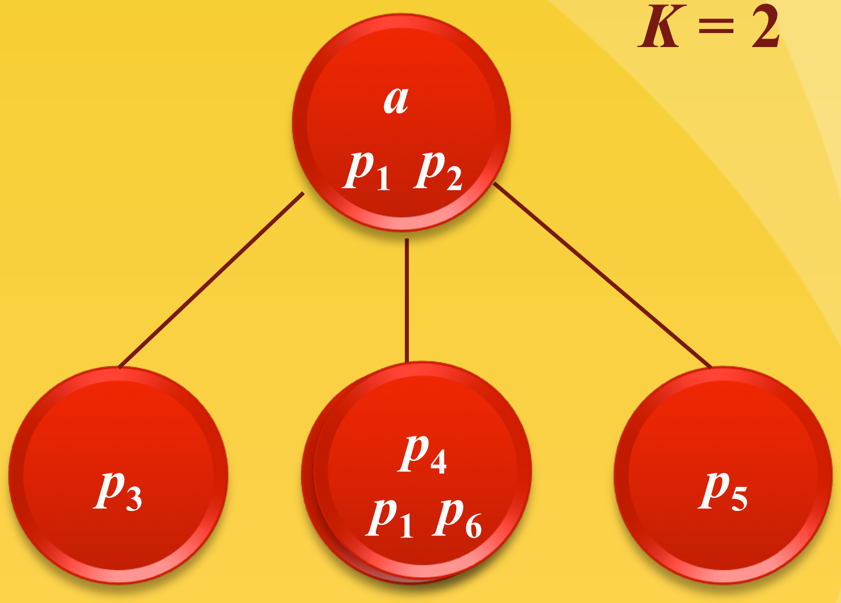
- It performs the spatial approximation on clusters of objects.
- Each node represents a cluster of very similar objects.
- We need to set the maximum arity and the maximum number of elements in a cluster.

DSACL-tree: Example

$A = 3$

$K = 2$

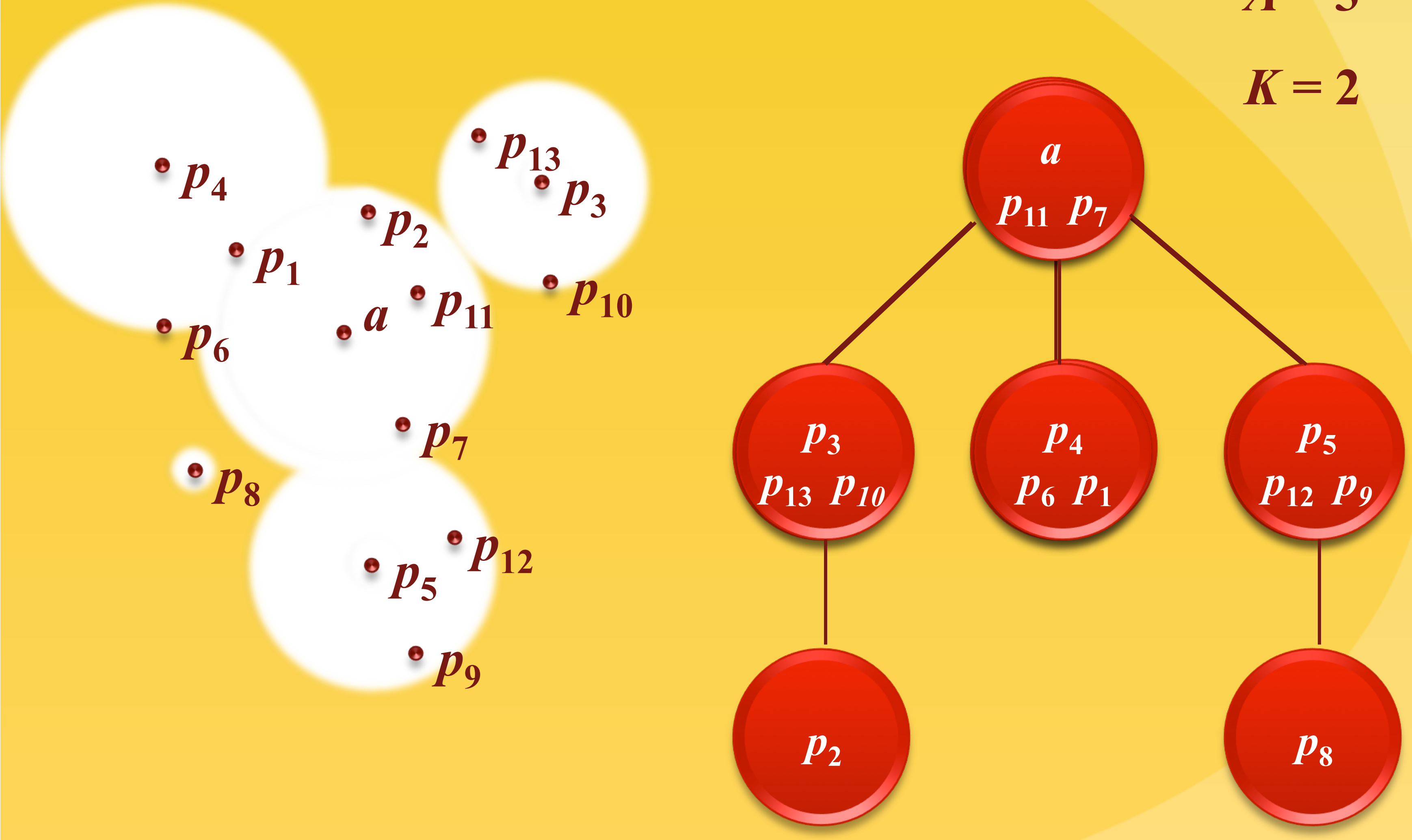




DSACL-tree: Example

$A = 3$

$K = 2$

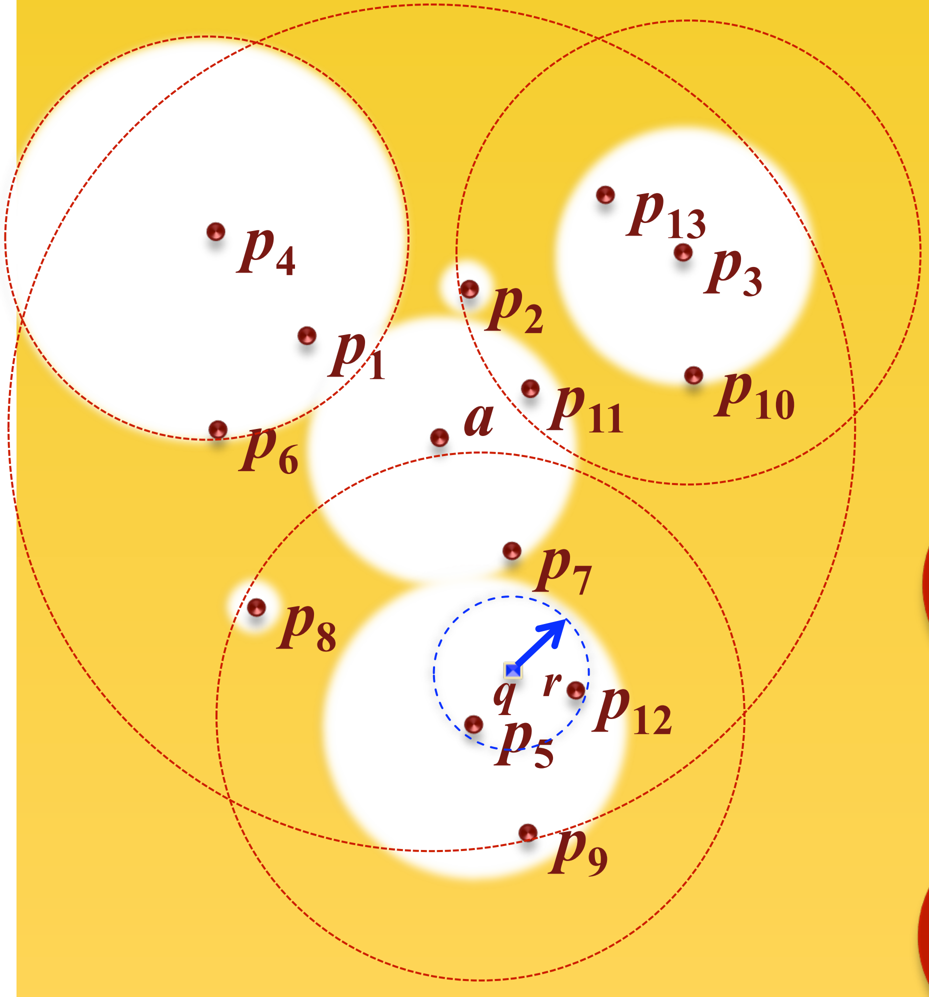


DSACL-tree: Searches

- During searches we perform the spatial approximation to the query via the centers of nodes.

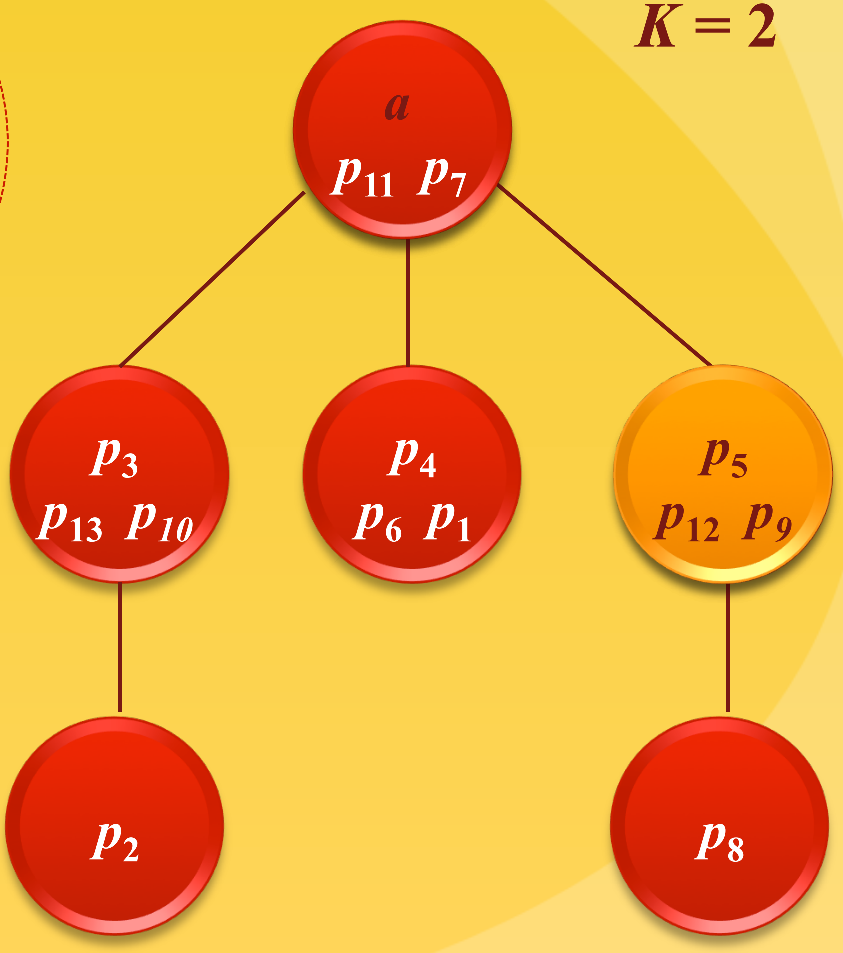
- We can prune searches by using timestamps, covering radii, cluster radii and the stored distances between the center and the elements of the clusters.

DSACL-tree: Example



$A = 3$

$K = 2$



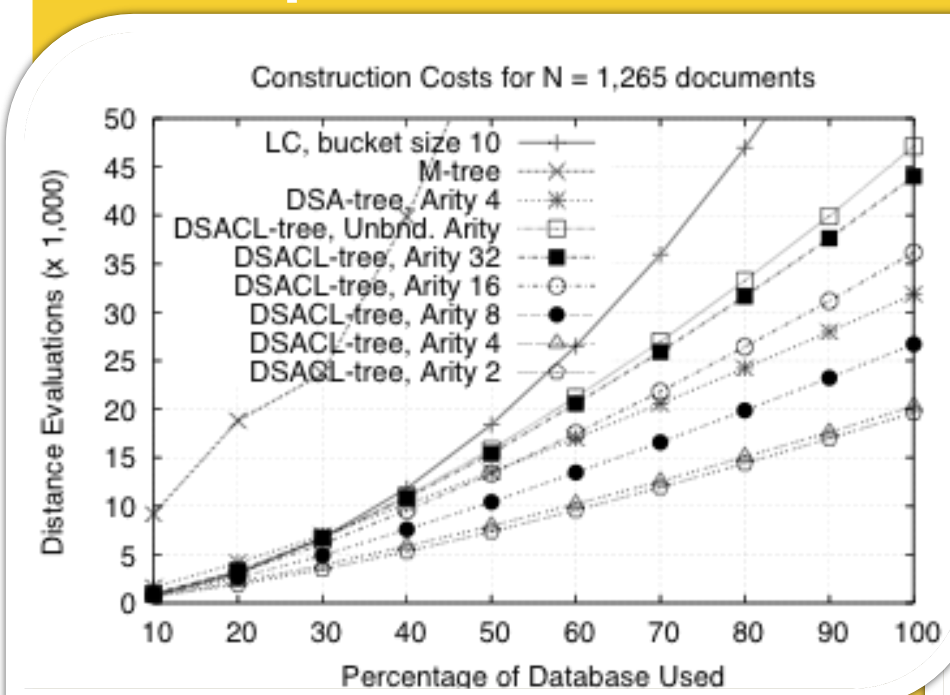
Experimental Results

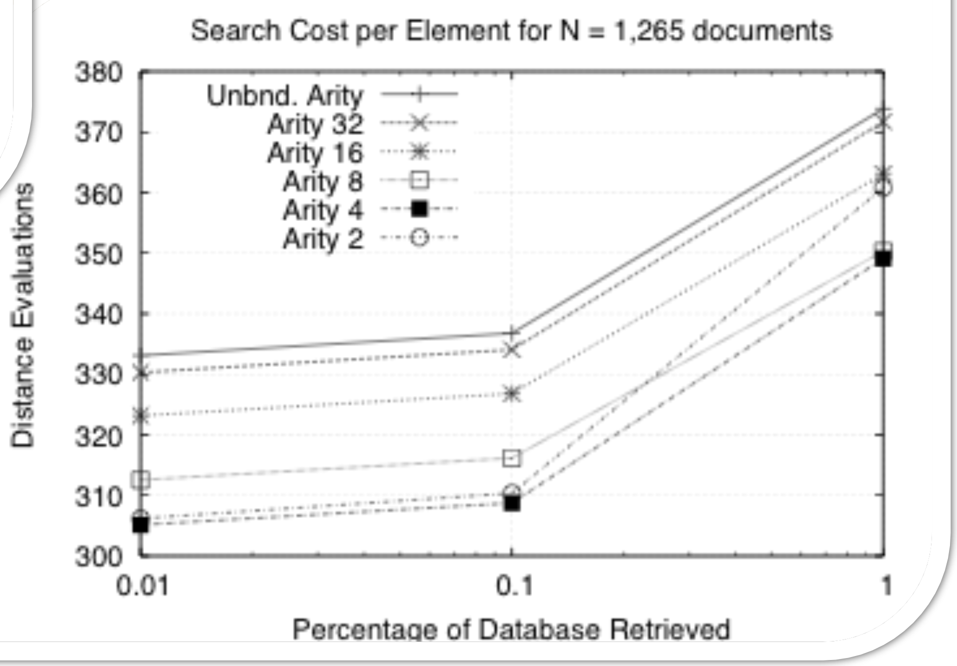
- We have selected four widely different metric spaces, all from the SISAP Metric Library.

- All our results are averaged over 10 index constructions using different permutations of the datasets.

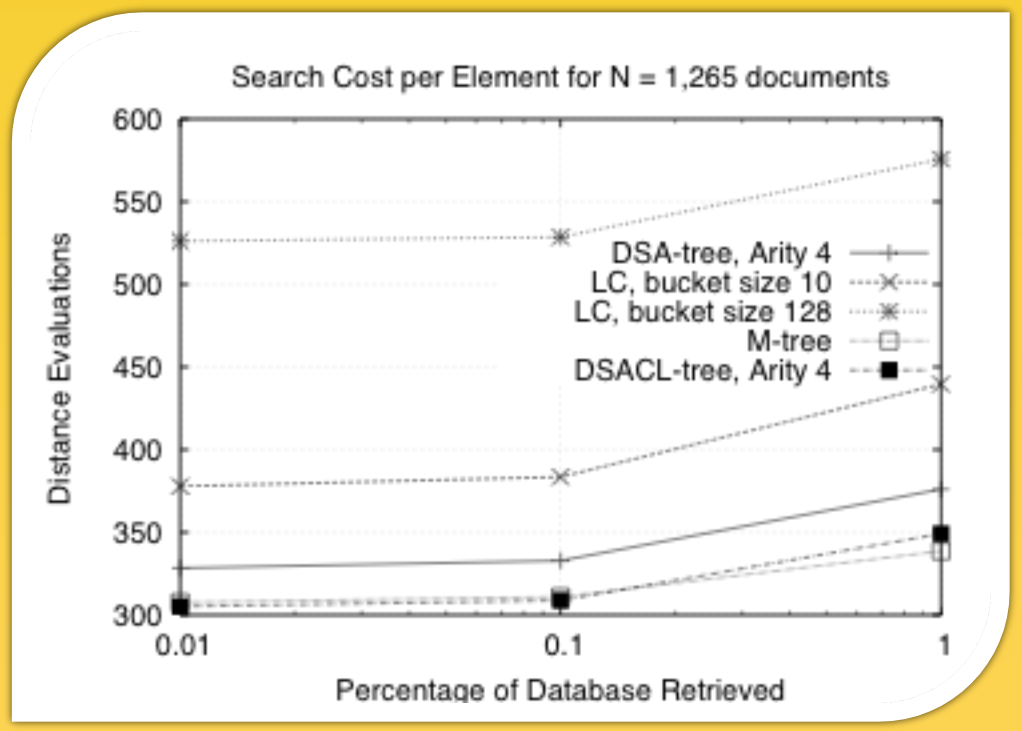
- We compare *DSACL-tree* with *M-tree*, *DSA-tree*, and *List of Clusters*.

Experimental Results: Documents

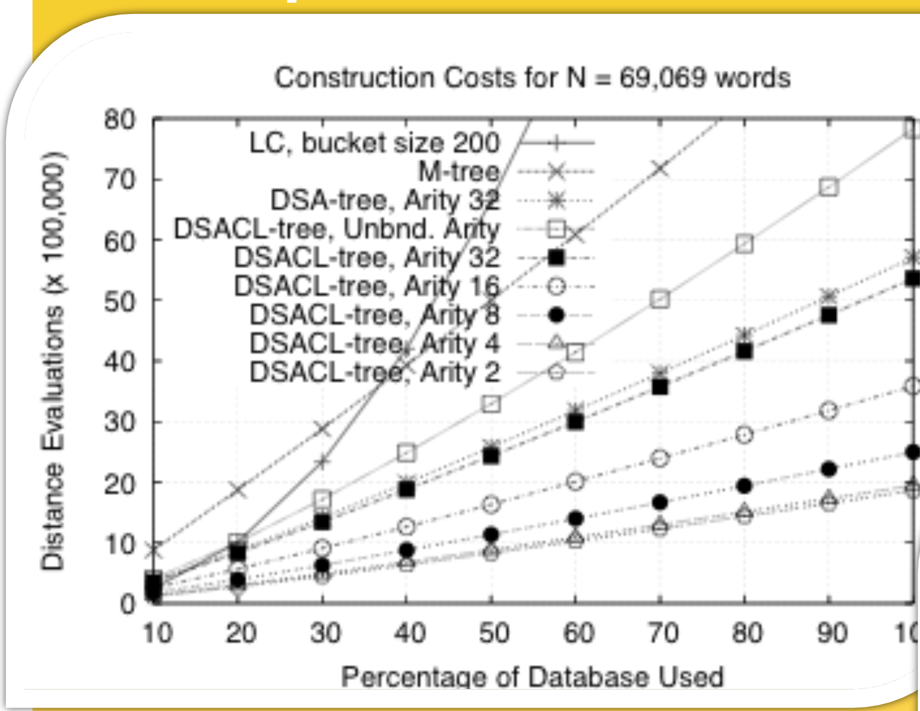


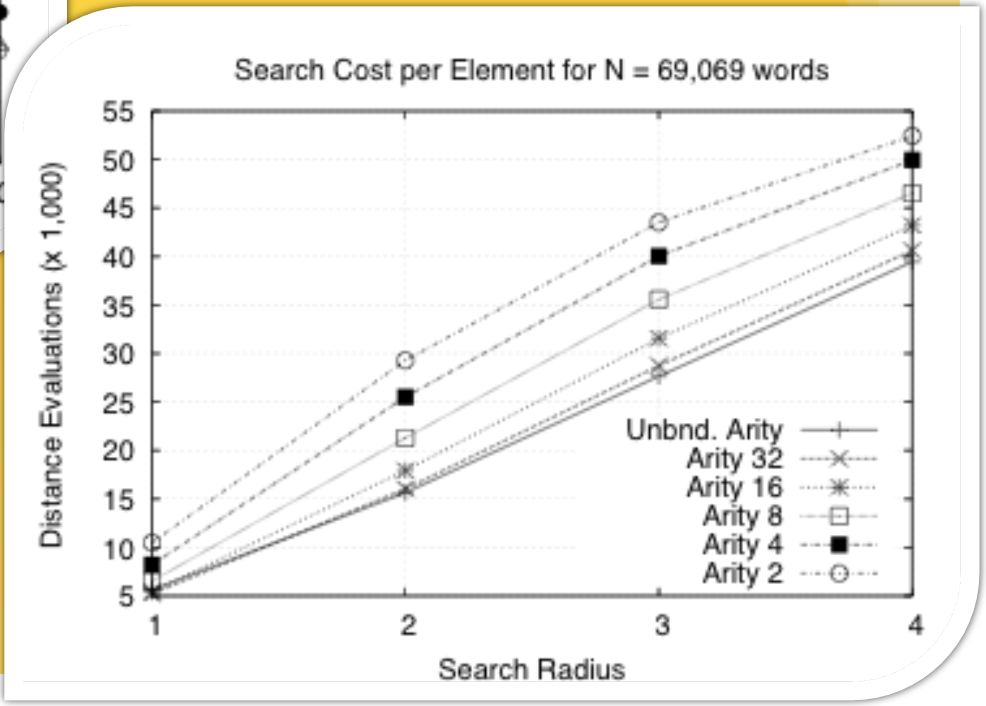


Experimental Results: Documents

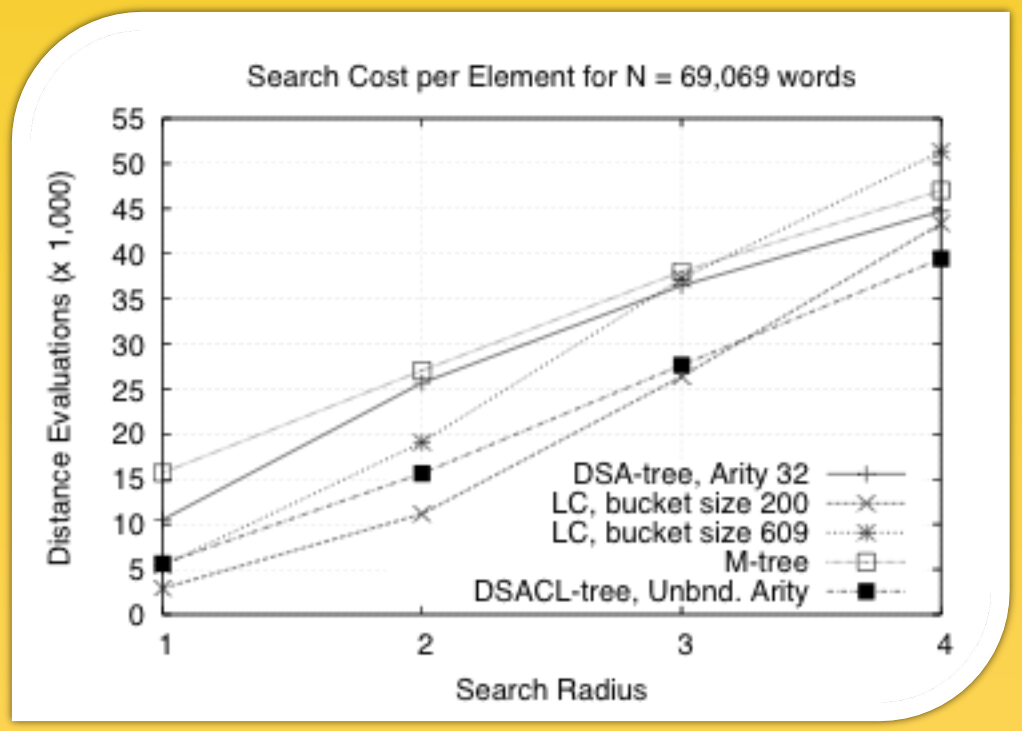


Experimental Results: Dictionary

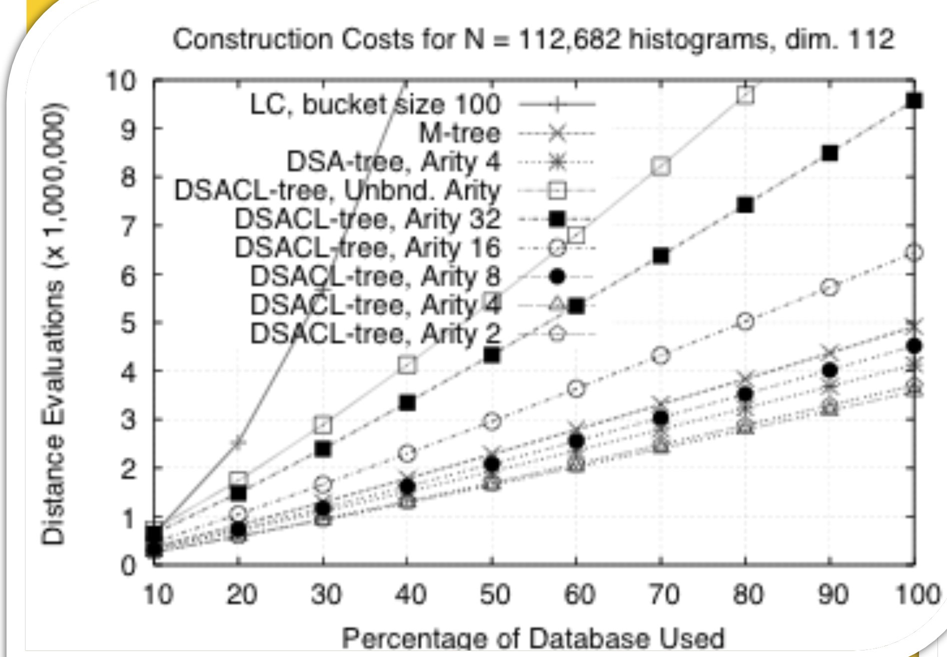


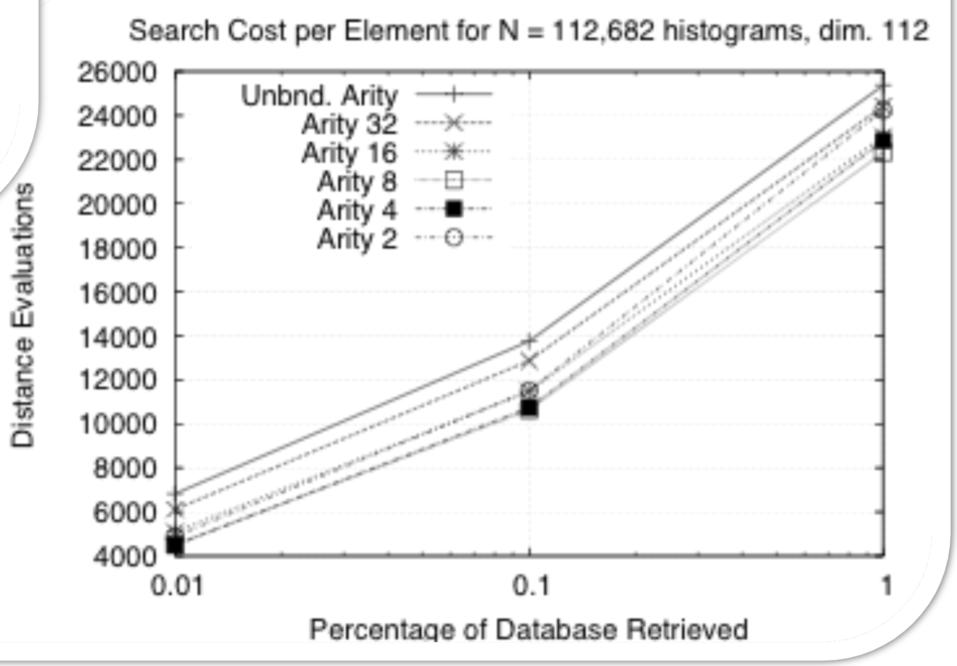


Experimental Results: Dictionary

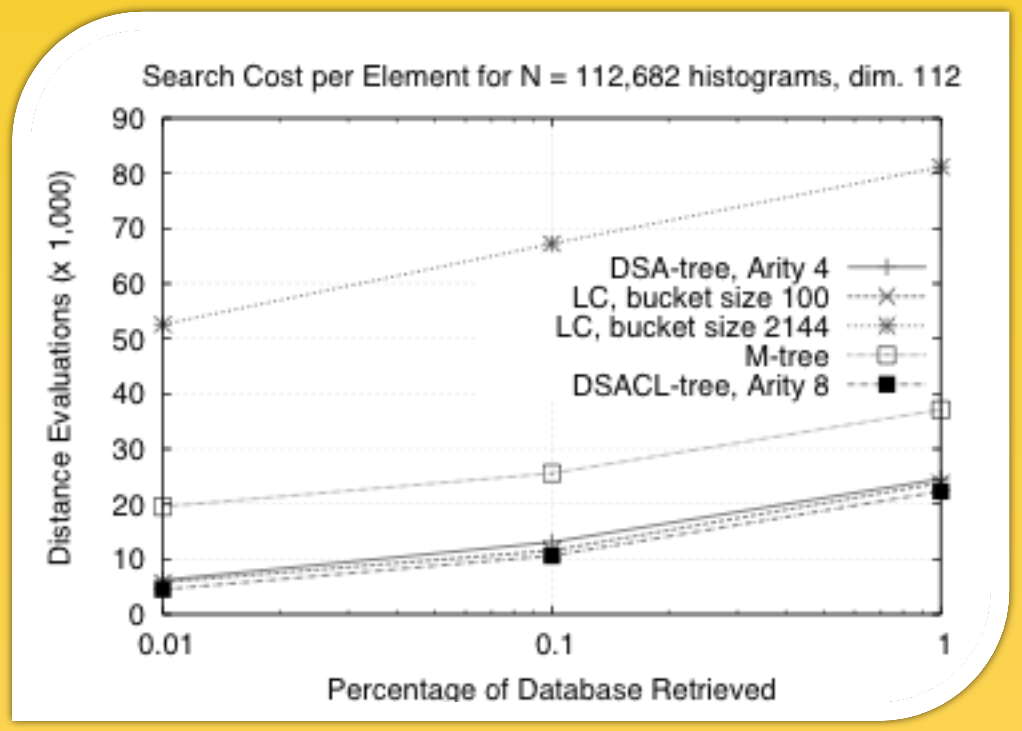


Experimental Results: Histograms

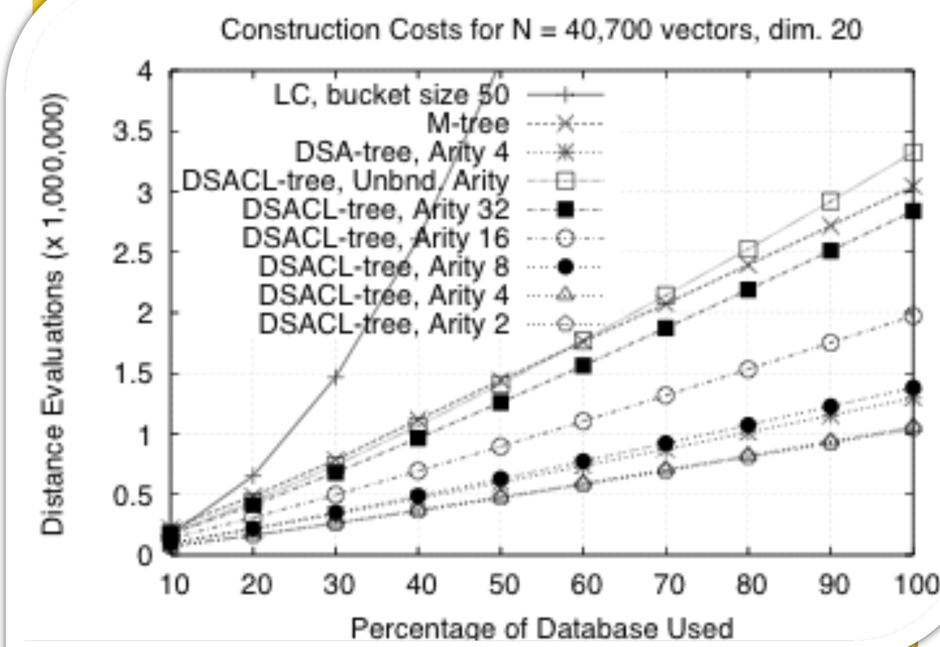


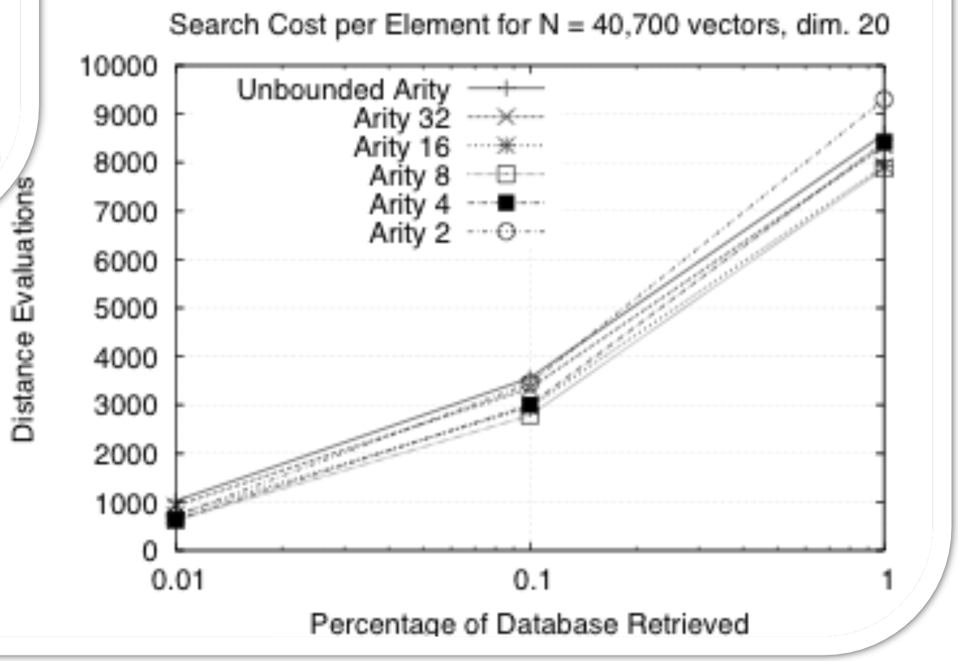


Experimental Results: Histograms

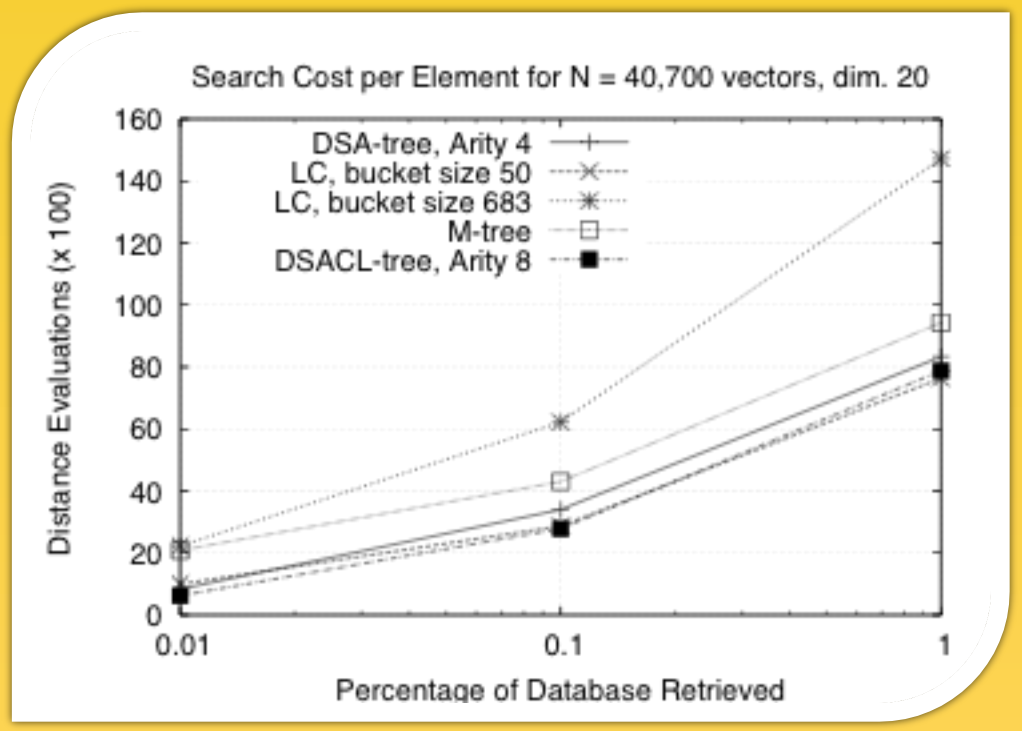


Experimental Results: NASA images





Experimental Results: NASA images



Conclusions

- *DSACL-tree* enhances the good features of the DSA-tree by taking into account the element clusters present in the metric space.

- We may reduce the backtracking in the tree improving the cost of retrieval relevant elements when performing a proximity query.

Future Works

- We are considering a secondary memory version of the *DSACL-tree*.
- We plan to evaluate the quality of the clusters produced in the *DSACL-tree*.
- Deletions have to be implemented in order to achieve total dynamism.

Thanks for your attention!
Ilginiz için teşekkürler!

Istambul, September 2010

18 de Septiembre