Ahmet Sayar, Süleyman Eken, Okan Öztürk, 2015. Kd-tree and quad-tree decompositions for declustering of 2D range queries over uncertain space. *Frontiers of Information Technology & Electronic Engineering*, **16**(2):98-108. [doi:10.1631/FITEE.1400165]

Front Inform Technol & Electron Eng

Kd-tree and quad-tree decompositions for declustering of 2D range queries over uncertain space

Key words: Kd-tree, Quad-tree, Space partitioning, Spatial indexing, Range queries, Query optimization

Contact: Ahmet Sayar

E-mail: ahmet.sayar@kocaeli.edu.tr

ORCID: http://orcid.org/0000-0002-6335-459X

Introduction

- The aim of our study is to evaluate the feasibility and efficiency of well-known space partitioning approaches (kd trees and quad trees) through performance tests on 2D range queries applied to point data in declustering applications.
- Declustering is a process used in distributed computing or clustered computing environments to reduce processing and query execution times by applying load balancing and workload sharing approaches.
- The aims are to create efficient indexing over uncertain space, and divide the workload for a range query into sub-ranges carrying equal sizes of query payload.

Front Inform Technol & Electron Eng

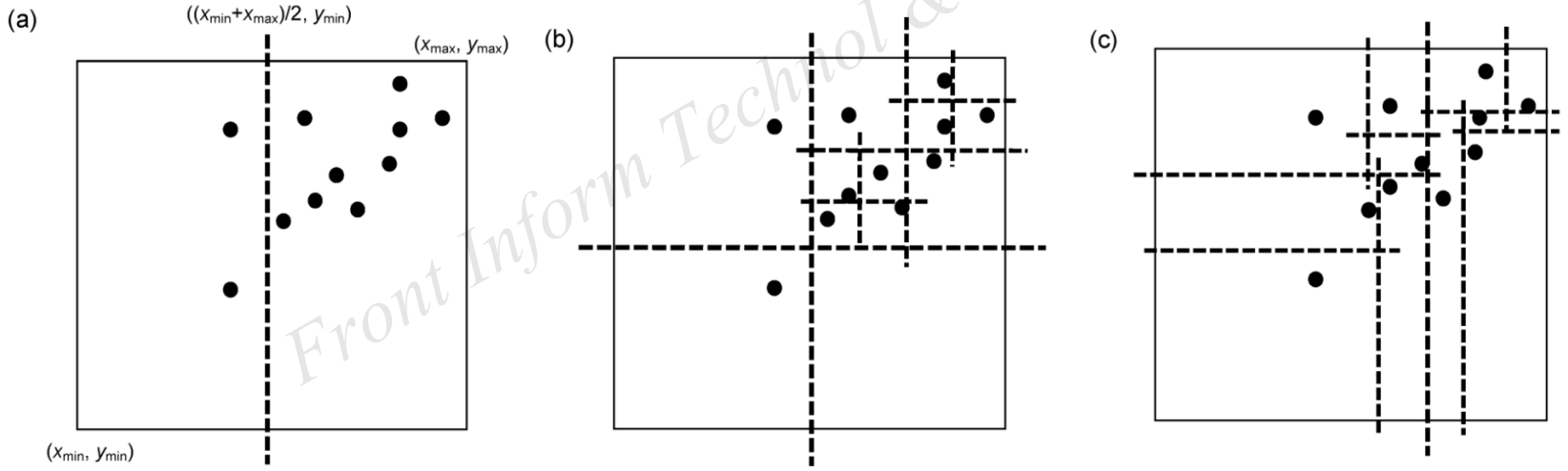
Problem statement

- The classical approach for dividing the workload into smaller pieces and assigning them to worker processors does not always help to improve performance in uncertain spaces.
- The aim is to eliminate data and execution skews for efficient I/O parallelization in uncertain spaces. Data sets are defined in a 2D plane as points, and are queried by 2D range queries (window queries).
- A range query is a general process to analyze and display spatial data defined by their coordinates in space, and is used in many science and application domains including GIS, astronomy, CAD/CAM, and computer graphics.

Front Info Technol & Electron Eng

Problem statement (Con'd)

- An illustration of the problem with (a) naive partitioning, (b) quad-tree partitioning, (c) kd-tree partitioning



Front Inform Technol & Electron Eng

Analysis of kd-tree and quad-tree decompositions in declustering of uncertain space

- *R* is a set of all rectangles created by one of the two indexing techniques. *S* consists of all the rectangles meeting the requirements. When we position query *Q*

  on the index table (set *R*) we find all the intersection

  rectangles. $S=Q\cap R=\{r_i: (r_i\in Q)\wedge(r_i\in R)\}$
- The query is given in a rectangular format. So, it is a trivial calculation to compute the area of the query, Area(*Q*).

$$\text{Area}(r_i)=(x_i^2-x_i^1)(y_i^2-y_i^1)$$

- The motivation behind selecting the cost metric is given as

$$\text{cost}=\text{Area}(S)-\text{Area}(Q)$$

Front Inform Technol & Electron Eng

Evaluation and test scenarios

- A GUI enables the creation of users' custom data by mouse clicks. The GUI was divided into two sections, one (left-hand side) related to kd trees and the other (right-hand side) related to quad trees.
- Test scenarios were determined based on the data size (relatively large or small) and distribution (random or skewed) characteristics. Randomization could be applied both to the location and to the size of the queries. Skewed data were created according to the user's preferences by using interactive GUI tools. The test scenarios are listed below:

  Case 1: Large range queries over random data space

  Case 2: Large range queries over skewed data space

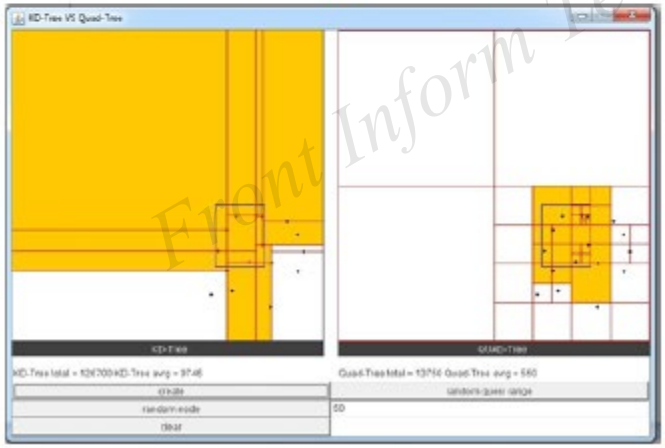
  Case 3: Small range queries over random data space

  Case 4: Small range queries over skewed data space

Front Inform Technol & Electron Eng

Evaluation and test scenarios (Con'd)

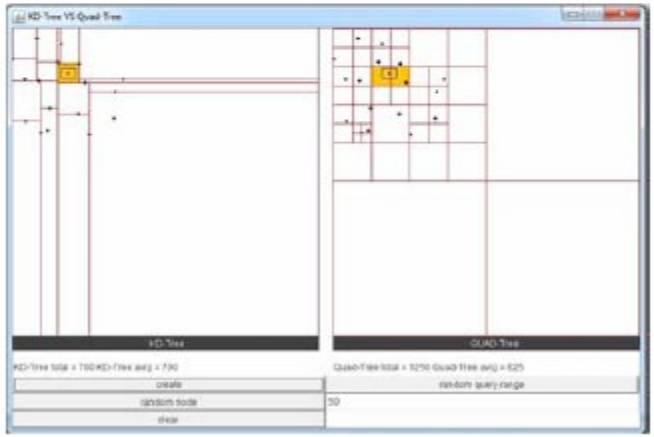
(a)



(b)

Fig. 3 Large range queries over (a) random data space, (b) skewed data space

(a)



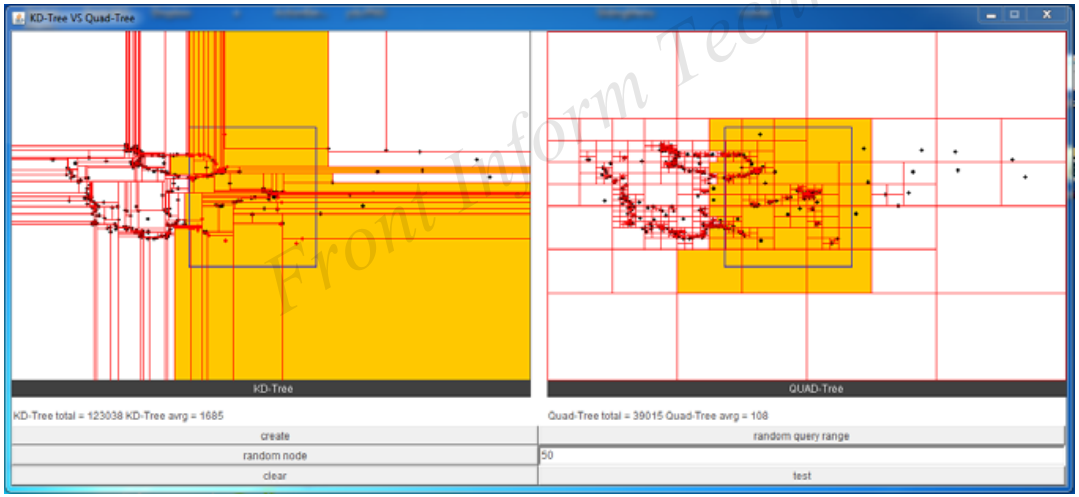
(b)

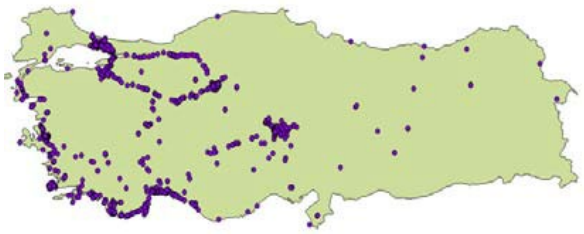
Fig. 4 Small range queries over (a) random data space, (b) skewed data space

Front Inform Technol & Electron Eng

Evaluation and test scenarios (Con'd)

- The same tests were performed on real data. We used Turkey's points of interest data in a shapefile format. The shapefile used in this test shows points of interest such as monuments, attractions, ruins, and archaeological buildings. It has about 3272 data points.





Large range queries over Turkey's points of interest

From Inform Technol & Electron Eng

Performance comparison

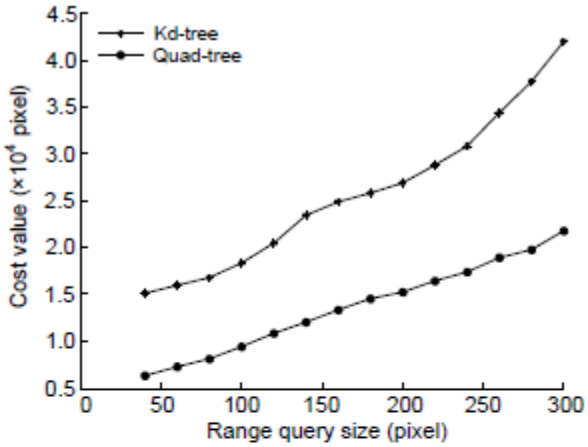


Fig. 5 Comparison of cost values for case 1

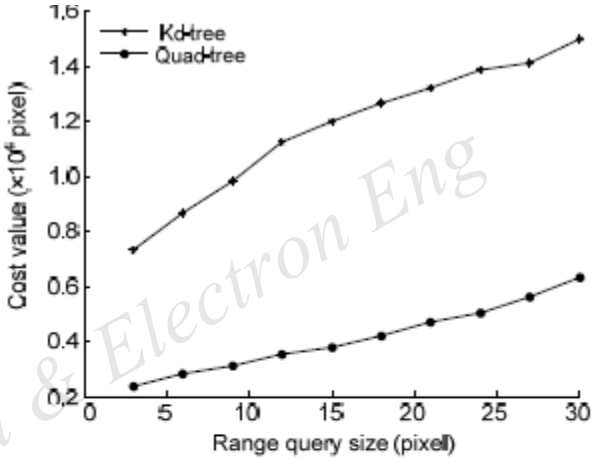


Fig. 7 Comparison of cost values for case 3

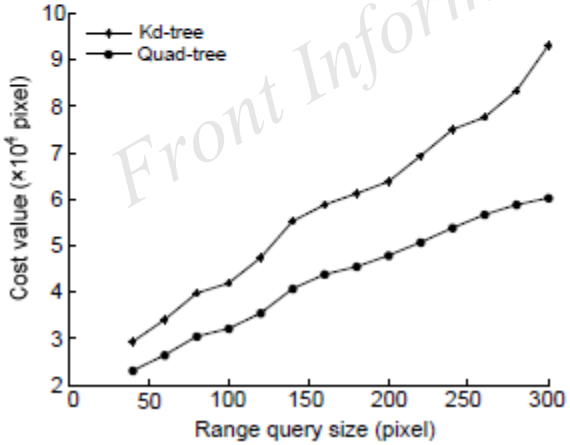


Fig. 6 Comparison of cost values for case 2

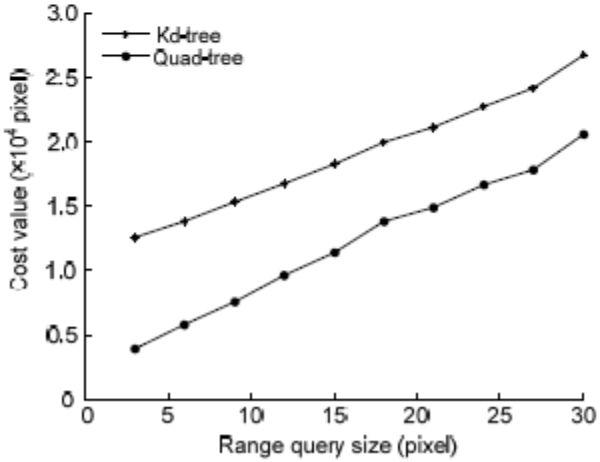


Fig. 8 Comparison of cost values for case 4

Front Inform Technol & Electron Eng

Conclusions

- Spatial data are mostly distributed, and their distribution is uncertain and unexpected based on their locations in a given space. This causes data and execution skews and is not a trivial problem in terms of the declustering and parallelization of this type of data.
- The proposed analysis helps with the selection of the best combination of partitions created by index tables that will minimize the response time of a given range query.
- It appears that quad-tree indexing gives better parallelization. This is due mostly to the fact that quad trees reveal the spatial locations of data more clearly.
- In the future, we will study the situation in which the I/O parallelization of query/rendering is increased on other more complex spatial data types such as line-strings and polygons.

Journal of Information Technology & Electronic Engineering