

# Indexing and Ranking in Spatial Database

**Krishnaveni S**

Dept. of Computer Science & Engineering  
College of Engineering College  
Chennai-600 025

**TamizhSelvi S P**

Dept. of Computer Science & Engineering  
College of Engineering College  
Chennai-600 025

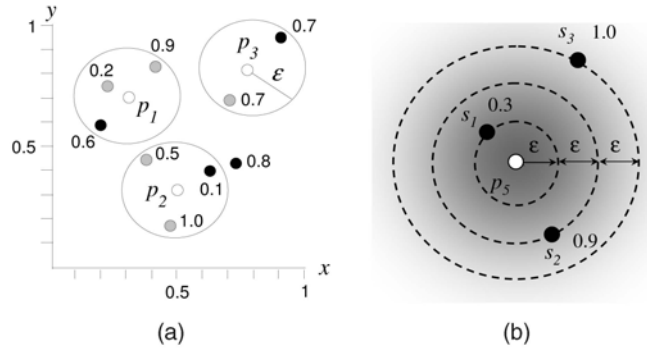
**Abstract :** In reality spatial objects (e.g., Dams) not only have spatial locations but also have quality attributes (e.g., height, reservoir capacity). Given a spatial location  $S$ , Quality vector  $\psi$  and a set of spatial objects  $D$ , a spatial query which retrieves and ranks the objects that intersect the region  $S$  and satisfies the quality vector. We proposed an efficient index called SKR tree and algorithms which performs 1) spatial filtering ,2) textual filtering and 3) object ranking in a fully integrated manner. The SKR tree is compared with the R tree.

**Keywords:** *R-tree, SKR-tree ,Query Processing, Spatial database.*

## I. INTRODUCTION

Spatial database systems manage large collections of geographic entities, which apart from spatial attributes contain non spatial information. Spatial objects in reality are associated with multiple quality attributes in addition to their spatial locations. Traditional spatial queries and joins focus on manipulating only spatial locations and distances, but they ignore the importance of quality attributes. The dominance comparison is suitable for comparing two objects with respect to multiple quality attributes. For the sake of simplicity, we assume that the domain of each quality attribute is fully ordered (e.g., integer domain). An object  $A$  is said to dominate another object  $B$ , if  $A$  is no worse than  $B$  for all quality attributes and  $A$  is better than  $B$  for at least one quality attribute. In this system, we study an interesting type of spatial queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood. Given a set  $D$  of interesting objects (e.g., candidate locations) and quality vector, a top –  $k$  spatial preference queries retrieves the  $k$  objects in  $D$  with the highest scores. The score of an object is defined by the quality of features (e.g., facilities or services) in its spatial neighborhood. As a motivating example, consider a database containing all information of dams. Here “feature” refers to specific facilities or services. A customer may want to rank the contents of this database with respect to the quality of their locations, quantified by aggregating non spatial characteristics of other features (e.g., height of dam, reservoir capacity etc.,).

As another example, the user (e.g., a tourist) wishes to find a hotel that is close to a high quality restaurant and a high quality cafe. For the ease of discussion, the qualities are normalized to values in  $[0, 1]$ . The score  $T(p)$  of a hotel is defined in terms of: 1) the maximum quality for each feature in the neighborhood region of  $p$ , and 2) the aggregation of those qualities.



In Figure 1a: a simple score instance, called the range score, binds the neighborhood region to a circular region at  $p$  with radius (shown as a circle), and the aggregate function to SUM. For example, the maximum quality of  $p_1$  (dam1) are 0.9 and 0.6. Hence the  $\tau(p_1)$  is  $0.9+0.6=1.5$ . Similarly for the dam  $p_2$ ,  $\tau(p_2)$  is  $1.0+0.1=1.1$ . Hence the dam  $p_1$  is returned as top result.

Figure 1b: shows a dam  $p_5$  and three features  $s_1, s_2, s_3$  (with their quality values). The circles have their radii as multiples of  $\tau$ Now, the score of  $s_i$  is computed by multiplying its quality with the weight  $2^{-j}$ , where  $j$  is the order of the smallest circle containing  $s_i$ . [1]

Traditionally, there are two basic ways for ranking objects: 1) spatial ranking, which orders the objects according to their distance from a reference point, and 2) non spatial ranking, which orders the objects by an aggregate function on their non spatial values. The top-k spatial preference query integrates these two types of ranking in an intuitive way.

## II. PROBLEM STATEMENT

Let  $c$  be the number of (numeric) quality attributes. Each object  $p$  in the object dataset  $D$  is composed of a set of quality attributes  $c$ , and is associated with a location  $L_p$ . A quality vector is a point  $\psi$  in the  $c$ -dimensional space where each dimension refers to a quality attribute. As a shorthand notation, we use  $\psi[i]$  to represent the  $i$ th (quality) attribute value of  $\psi$ . The notion of dominance is used to compare quality vectors. A quality vector  $\psi$  is said to dominate another one  $\psi'$  (denoted as  $\psi < \psi'$ ), if  $\exists 1 \leq i \leq c, \psi[i]$  is better than  $\psi'[i]$  and  $\forall 1 \leq i \leq c, \psi[i]$  is not worse than  $\psi'[i]$ .

A location is a pair  $(x, y)$  in the Euclidean space, where  $x$  and  $y$  are the coordinates values. A spatial object  $p = \langle loc, \psi \rangle$  consists of both a location  $o.loc$  and a quality vector  $o.\psi$ . The notation  $dist(p, p')$  denotes the Euclidean distance between the locations of the spatial objects  $p$  and  $p'$ . Given two spatial objects  $p$  and  $p'$  is said to be a dominator of  $p'$  when  $p.\psi < p'.\psi$ . Given a query  $q$  that specifies a set of query keywords  $W_q$  a query spatial scope  $S_q$ , textual relevance and spatial relevance of object  $p$  to  $q$  are formalized in definitions 1 and 2, respectively.

**Definition 1 (Textual relevance)** An object  $p$  is said to be textually relevant to a query  $q$  if  $o$  contains queried keywords i.e.,  $W_p \cap W_q \neq \emptyset$ . To quantify the relevance of  $p$  to  $q$ , a weighting function denoted by  $\tau_q(o)$  is adopted. Thus for a given  $q$ ,  $\tau_q(p1) > \tau_q(p2)$  means object  $p1$  is more textually relevant to  $q$  than object  $p2$ .

**Definition 2 (Spatial relevance)** An object  $o$  is said to be spatially relevant to a query  $q$  if the location of object  $o$  overlays with the query spatial scope of  $q$ , i.e.,  $L_p \cap S_q \neq \emptyset$ . Let  $\tau_q(o)$  be a scoring function to quantify the spatial relevance of  $o$  to  $q$ . Thus for a given  $q$ ,  $\tau_q(p1) > \tau_q(p2)$  means object  $p1$  is more spatially relevant to  $q$  than object  $p2$ .

Accordingly,  $k$  objects are retrieved from  $D$  objects that are both textually and spatially relevant to a given query and ranks objects based on the quality features of their objects.

The spatial relevance of an object  $p$ , denoted as  $\tau(p)$  depends on the types of the spatial relationships defined between an object location  $L_p$  and a spatial scope  $S$ . Commonly adopted relationships include

1. **Enclosed.**  $\tau(p)$  is set to 1 if the corresponding location is fully enclosed by the query scope, i.e.,

$$\tau(p) = \begin{cases} 1, & \text{if } L_p \subseteq S \\ 0, & \text{otherwise} \end{cases}$$

2. **Overlapping.**  $\tau(p)$  is set to the fraction of the object location that is covered by the spatial scope,.

$$\tau(p) = \frac{\text{Area}(L_p \cap S)}{\text{Area}(L_p)}$$

3. **Proximity.**  $\tau(p)$  is represented by the inverse of the distance between the center of  $L_p$  and that of  $S$ , i.e.,

$$\tau(p) = \begin{cases} \frac{1}{\text{dist}(L_p, S)}, & \text{if } L_p \subseteq S \\ 0, & \text{otherwise} \end{cases}$$

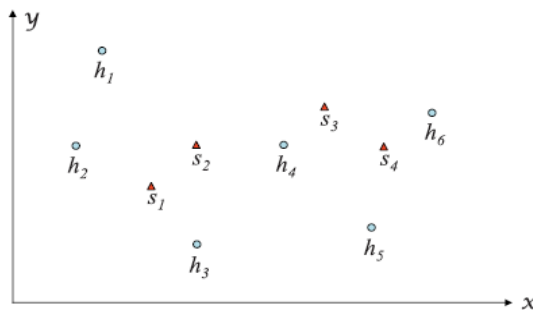


Figure2: Example of Farthest Dominated Location

Dams	Height m	Reservoir capacity
h 1	180	4
h 2	150	3
h 3	190	4
h 4	250	3
h 5	190	4
h 6	220	5

(a)

Loc	NN	ND
S1	h3	h3
S2	h4	h3
S3	h4	h5
S4	h6	h5

(b)

Figure3:Lists of dams and candidate locations, (a) Qualities of dams, (b) Candidate location at  $\psi(200,4)$

**Definition 3 (Nearest dominator, Nearest Dominator Distance).** Given a location  $S$  .its quality vector  $\psi$ , and a set of spatial objects  $D$ , the nearest dominator of  $S$  in  $D$  is defined as

$$ND(S,\psi,D)= \underset{p \in D, p.\tau < \psi}{\operatorname{argmindist}}(S,p)$$

i.e., the nearest neighbor of  $S$  in  $D$  among those that dominate  $\psi$ . The nearest dominator distance  $ndd(S,\psi,D)=\operatorname{dist}(S,ND(S,\psi,D))$ . Refer to the example in Fig 2 and 3, the ND of  $S$  is the dam  $h_j$  that minimizes the  $\operatorname{dist}(S_j,h_j)$  value, among those dams dominating the design competence  $\psi$ . In Fig 2.b. lists the NN and ND of each location  $S_j$ . It is important to note that NN is not necessarily the same as ND. For example, the NN of  $S_2$  is  $h_4$  which, however, does not dominate  $S_2$  with respect to its design competence. Whereas its next nearest neighbor  $h_3$  does, which exactly is  $S_2$ 's ND. By considering the distance of each location  $S_j$  from its ND, we pick the largest one(i.e.,  $\operatorname{dist}(S_3,h_5)$ ), and take its location(i.e.,  $S_3$ ) as the result location for building the new dam.

**Definition 4 (Farthest Dominated Location Query)**

Given a set of (competitors’) spatial objects  $D$ , a set of (candidate)locations  $L$ , and a quality vector  $\psi$  as the design competence, the farthest dominated location query returns from  $L$  a location  $S$  such that the distance  $ndd(S,\psi,D)$  is maximized, i.e.,

$$\forall S' \in L, ndd(S, \psi, D) \geq ndd(S', \psi, D)$$

Refer to the dam example in Figs. 2 and 3. There are  $c = 2$  Quality attributes (i.e., height, reservoir capacity). The set of objects is  $D=\{h_1,h_2..h_6\}$  and the set of locations is  $L=\{S_1,S_2..S_4\}$  dam  $h_1$  is a spatial object, with a fixed location in the Euclidean space and the quality vector  $h_1.\psi=(180, 4)$ . Let the design competence be  $\psi(200, 4)$ . Location  $s_3$  is the farthest dominated location and its nearest dominator is  $h_5$ .

**III. SPATIAL QUERY EVALUATION ON R TREE**

The most popular spatial access method is the R-tree[3], which indexes minimum bounding rectangles (MBRs) of objects. Figure 4 shows a set  $D\{p_1,p_2..p_8\}$  of spatial objects (e.g., points) and an R-tree that indexes them.

R-trees can efficiently process main spatial query types, including spatial range queries, nearest neighbor queries, and spatial joins. Given a spatial region S, a spatial range query retrieves from D the objects that intersect.

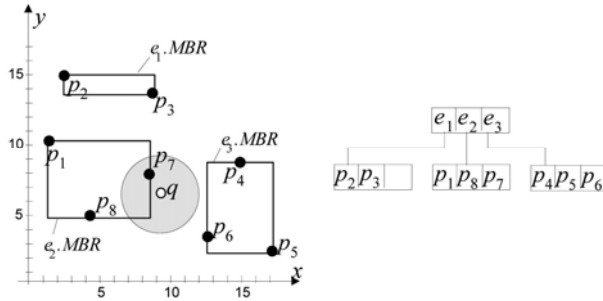


Figure4: Spatial Query Evaluation on R trees

Table I List of Notations

Notations	Meaning
$e$	An entry in the R-tree
D	The object dataset
M	The number of features
$p$	An object point of D
$\tau_c(p)$	The c-the score of p
$mindist(p, e)$	Minimum distance between p and e
$T(e)$	Upper bound score an R-tree entry
$\epsilon$	Radius
Wk	Min Heap and Max Heap
$L_p$	Location of object p
S	Spatial Scope

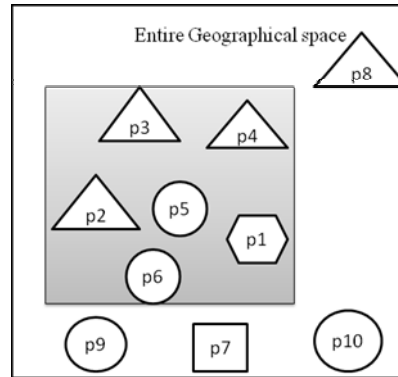
For instance, consider a range query that asks for all objects within the shaded area in Fig.4 Starting from the root of the tree, the query is processed by recursively following entries, having MBRs that intersect the query region. For instance, e1 does not intersect the query region, thus the sub tree pointed by e1 cannot contain any query result. In contrast, e2 is followed by the algorithm and the points in the corresponding node are examined recursively to find the query result p7.

The upper bound score of an object is calculated using the formula

$$\tau(p) = \sum_{c=1}^M \begin{cases} \tau_c(p) & , \text{if } \tau_c(p) \text{ is known.} \\ 1 & \text{otherwise.} \end{cases}$$

**IV. SPATIAL QUERY EVALUATION ON SKR TREE**

In Figure 5 assume if a user Alice in Boston issues a geographic query “Boston’s pollution control irrigation” and the top-3 dams are to be returned. In this query, “pollution control” and “irrigation” are query keywords, and “Boston” represents a location/area of her interest. Regardless of the order of their relevance, this example shows a set of candidate objects {p2, p3, p4, p5, p6}, with respect to both textual relevance and spatial relevance. Here, p1, although being within “Boston,” is not relevant because it contains neither “pollution control” nor “irrigation.” On the other hand, {p7, p8, p9, and p10} although being textually relevant to the query, are not within “Boston.” The top-3 dams in the candidate set that are most relevant are returned.



- 
- 

Figure5: Spatial Data Distribution

For this we proposed an efficient indexing scheme called SKR tree which indexes both the textual and spatial contents of objects to support data retrievals based on their combined textual and spatial relevance, which, in turn, can be adjusted with different relative weights. Fig.6. Shows the structure of SKR tree where each node has both spatial and non spatial information of the data object.  $N_0$  is the Root node which has two child nodes  $N_1$  and  $N_2$ . The child node has the keyword  $A_{N_1}$ , and spatial data  $D_{N_1}$ . This indexing scheme performs textual filtering and spatial filtering and hence it is efficient when compared with the R tree indexing scheme.

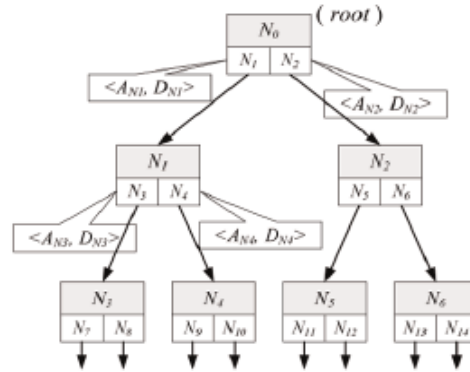


Figure6: Structure of SKR Tree

## V. ALGORITHMS FOR SPATIAL PREFERENCE QUERY

### A. Range Score Algorithm

Given a set of objects, the range score algorithm calculates the Nearest neighbour distance and ranks the objects.[4].

**Input:** A set of data D, query location, Root of R tree and the radius  $\epsilon$ .

**Output:** Top k data with shortest distance

**Procedure:**

- 1: Group\_Range(NodeN)
- 2: foreachentry  $e \in N$  do
- 3: If N is nonleaf then
- 4: readthechildnode  $N^1$  pointed by e;
- 5: Group\_Range( $N^1$ );
- 6: else
- 7: foreach  $p \in V$  such that  $\text{dist}(p, q) \leq \epsilon$  then
- 8: update  $W_k$  by e;

### B. Branch and Bound Algorithm

Branch and bound algorithm which calculates the score and ranks the objects based on the non spatial quality features of the objects.

**Input:** A set of data D and Query location and root of R tree

**Output:** Top k data with the highest score.

**Procedure:**

- 1: Cal\_Score(NodeN)
- 2: foreachentry  $e \in N$  do
- 3: If N is nonleaf then
- 4: readthechildnode  $N^1$  pointed by e;
- 5: Cal\_Score( $N^1$ );

```

6: else
7:  foreach  $p \in V$  such that
8:  for  $c := 1$  to  $m$  do
9:  compute  $T_c(p)$  for all  $e \in V$  concurrently;
10: update  $W_k$  by  $e$ 

```

### C. SKR Tree Construction Algorithm

**Input:** Set of Objects  $D$

**Output:** Root of SKR tree

**Procedure:**

```

1:  $N_e \leftarrow 0$ 
2: For each  $p \in D$  do
3:  geocode  $p$  and represent  $L_p$  with MBB  $m_p$ 
4:  if for some  $e \in N_e$ ,  $m_e = m_p$  then
5:  add  $p$  to  $e$ 's dataset  $D_e$ ;
6:  else
7:  create a new entry  $e$ ;
8:  set  $m_e \leftarrow m_p$  and  $D_e \leftarrow \{p\}$ ;
9:   $N_e \leftarrow N_e \cup \{e\}$ ;
10: End if
11: End for
12: For each  $e \in N_e$  do
13:  While  $|N_e| > n_{\max}$  do
14:  Cluster the data according to min/max into nodes
15:   $N_e \leftarrow N_e'$ 
16:  End while
17: Create the root node to cover  $N_e$  and their data summary

```

### D. NDD algorithm

This algorithm which finds the nearest dominated locations that satisfies the quality vector  $\psi$ .

**Input:** Root of SKR tree, set of locations  $L$  and quality vector  $\psi$

**Output:** A set nearest dominators

**Procedure:**

```

1:  $fdl := \text{null}$ ;  $ndd := 0$ 
2: for each location  $s \in L$  do
3:   $s.ND := \text{null}$ ; initialize a min-heap  $H$ 
4:  for each entry  $e'$  in  $R_p$ 's root do
5:   if  $e', \psi < \Psi$  then
6:    enheap( $H, \langle e', \text{mindist}(s, e') \rangle$ )
7:  while  $H$  is not empty and  $s.ND$  is null do
8:    $e := \text{deheap}(H)$ 
9:   if  $e$  is a non-leaf entry then
10:    for each child  $e'$  of  $e$  do
11:     if  $e', \psi < \Psi$  then
12:      enheap( $H, \langle e', \text{mindist}(s, e') \rangle$ )
13:   else
14:     $s.ND := e$ 
15:    if  $\text{dist}(s, s.ND) > ndd$  then
16:      $ndd := \text{dist}(s, s.ND)$ ;  $fdl := s$ 
17: return  $fdl$ 

```



### E. NDD Score algorithm

This algorithm which finds the score of all nearest dominated locations and ranks the objects.

**Input:** Root of SKR tree , set of locations  $L$  and quality vector  $\psi$

**Output:** Top  $k$  objects with highest score

**Procedure:**

```
1: initialize a min-heap  $H$ 
2: for each entry  $e'$  in  $R_P$ 's root do
3:   if  $e'.\psi \prec \Psi$  then
4:     enheap( $H, \langle e', mindist(e_L, e') \rangle$ )
5: while  $H$  is not empty do
6:    $e :=$  deheap( $H$ )
7:   if  $e$  is a non-leaf entry then
8:     for each child  $e'$  of  $e$  do
9:       if  $e'.\psi \prec \Psi$  then
10:        enheap( $H, \langle e', mindist(e_L, e') \rangle$ )
11:   else
12:     return  $maxdist(e_L, e)$ 
```

## VI. EXPERIMENTAL EVALUATION

In this section, we conduct experiments on real object and feature data sets in order to demonstrate the application of top- $k$  spatial preference queries. We obtained real spatial data from <http://www.fao.org/nr/water/aquastat/gis/index2.stm> location in these data sets correspond to (longitude and latitude) coordinates in Africa and Middle East countries.

This Spatial dataset containing information about dams in South Africa and middle east countries which has several features like irrigation, water supply, flood control, Hydro electricity, navigation , recreation, pollution control, livestock rearing and others. We used SQL server 2008 for spatial database design. Spatial query visualizer to visualize the spatial query and spatial uploader to upload the shape files of Africa and middle east countries. The proposed indexing scheme is evaluated with the R tree.

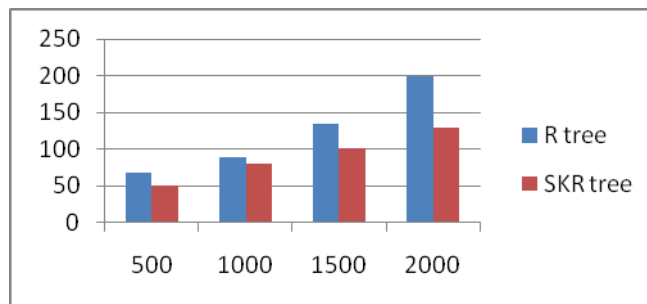


Figure7:Effect of IO Access Time for Different Trees in ms



## VII.CONCLUSION

In this paper, we studied top-k spatial preference queries, which provide a novel type of ranking for spatial objects based on qualities of features in their neighborhood. Given a spatial location  $S$ , Quality vector  $\psi$  and a set of spatial objects  $D$ , a spatial query which retrieves and ranks the objects that intersect the region  $S$  and satisfies the quality vector. We proposed an efficient index called STR tree and algorithms which performs 1) spatial filtering ,2) textual filtering and 3) object ranking in a fully integrated manner.

In the future, we will study the top-k spatial preference query on a road network, in which the distance between two points is defined by their shortest path distance rather than their euclidean distance. The challenge is to develop alternative methods for computing the upper bound scores for a group of points on a road network.

## REFERENCES

1. M.L.Yiu, X.Dai,N. Mamoulis and M. Vaitis, “Top k Spatial Preference Queries,” Proc. IEEE Intl Conf. Data Eng.(ICDE),2007.
2. Man Lung Yiu, Hua Lu, Nikos Mamoulis, and MichailVaitis “Ranking spatial data by quality preferences” IEEE Trans.on Data Eng. March 2011.
3. A.Guttman, “R-Trees:A Dynamic Index Structure for Spatial Searching, ” Proc. ACM SIGMOD, 1984.
4. Y. Chen and J.M. Patel, “Efficient Evaluation of All-Nearest-Neighbor Queries,” Proc. IEEE Int’l Conf. Data Eng. (ICDE), 2007.
5. N. Mamoulis, M.L. Yiu, K.H. Cheng, and D.W. Cheung, “Efficient Top-k Aggregation of Ranked Inputs,” ACM Trans. Database Systems, vol. 32, no. 3, p. 19, 2007.
6. David Noack, “Spatial variation in search engine Results”,Int’lconf on system sciences-2010.
7. Zhicheng Li, Hu Xu, Yansheng Lu, Ailing Qian“Aggregate Nearest Keyword Search in Spatial Databases “Int’l Asia specific web conf. 2010.
8. RamondT.Ng, Jiawei Han, ”Effective and efficient clustering in spatial datamining”, proceedings of 20<sup>th</sup> VLDB conf, santiago, chile.
9. Muhammad Aamir Cheema , Xuemin Lin , Haixun Wang , Jianmin Wang , Wenjie Zhang “A unified Approach for Computing Top-k Pairs in Multidimensional Space.” IEEE Int’l Conf. Data Eng. (ICDE),2011.
10. Xiang Lian and Lei Chen “Ranked Query Processing in Uncertain Databases”, IEEE Trans Data Eng.