Incremental Quality Based Reverse Ranking for Spatial Data

A.Sasi Kumar* and G. Suseendran
School of Computing Sciences, Vels University, Chennai – 600 117, Tamil Nadu, India;
askmca@yahoo.com, suseendar_1234@yahoo.co.in

Abstract

Background/Objectives: The scope of this proposed work is to minimize the search time and complexity in spatial database. Methods: Improves the query processing in spatial database by using the existing R Tree, IR Tree and Reverse Ranking. A comparative analysis is made between the existing methods and the proposed method Incremental Quality Reverse Ranking (IQRR). The proposed method effectively evaluates to find the top-k spatial objects in multiple query processing. Findings: To evaluate the performance of the proposed approach, a comparative study has been performed in this work. The R tree and IR tree are compared with the proposed work namely Incremental Quality Reverse Ranking (IQRR). The evaluation parameters are radius, time, location, directions and number of dams. Applications: A spatial preference query ranks objects (e.g. Dams) based on the qualities of features (irrigation, water supply, flood control, hydroelectricity, navigation, recreation and pollution control) in their spatial neighborhood. In future, according to the user specification, it may be developed for any spatial network application. This application can be deployed in the cloud server and cloud will provide a service to the user.

Keywords: Indexing Structures, IR-tree, Query Processing, R-tree, Spatial Databases

1. Introduction

Spatial objects in reality are associated with multiple quality attributes such as irrigation, water supply, flood control, hydroelectricity, navigation, recreation, pollution control and livestock rearing in addition to their spatial locations. Traditional spatial queries and join queries 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. An object A is compared with another object B, based on the quality attributes such as irrigation, water supply, flood control, hydroelectricity, and pollution control. In this system, an interesting type of spatial queries has been studied, which selects the best spatial location with respect to the qualities in its spatial neighborhood. Given a data set D of interesting objects (candidate locations) and a quality vector, the 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 in its spatial neighbourhood. For instance, consider a database containing all information on 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 its locations, quantified by aggregating the non-spatial characteristics of other features (e.g., height of dam, reservoir capacity etc.).

In this case, the data pertaining to the constructed dam would be updated and added as a field to the spatial index table, thus facilitating query retrieval by reducing the search time.

*Author for correspondence
2. Materials and Methods

2.1 Spatial Query Evaluation on IR Tree

Spatial keyword queries take a location and a set of qualities as arguments and return the content that best matches these arguments. For instance, dam p1 is described by the quality features: ‘Irrigation’ and ‘Water Supply’. In the event of the user Q, issuing a query (q1, q2 ...qn) to retrieve the nearest dam that serves the purposes of irrigation and flood control, the spatial query processor returns the information of the dam p2, which is the closest one that contains all the keywords given by the user query q1. Based on the multiple quality attributes of the query, the data is retrieved accordingly.

Each non-leaf node R in the IR tree contains a number of entries of the form (cp, rect, cp.di) where cp is the address of a child node of R, rect is the Minimum Bounding Rectangle (MBR) of all rectangles in the entries of the child node and cp.di is the identifier of a pseudo text description, that is the union of all text descriptions in the entries of the child node. Figure 1 contains 9 spatial objects p1, p2, p3...p9 and Table1 shows the words appearing in the description of each object. Figure 2 illustrates the corresponding IR tree and Table 2 shows the contents of the inverted files associated with the nodes. Thus the objects are retrieved with the requested quality attributes using the IR tree.

Spatial data deals with geographic entities. To consider quality along with distance, the IR tree had been proposed. The IR tree is a combination of spatial and textual description given as a result in the form of an inverted file.

![Figure 1. Tree based object locations.](image1)

![Figure 2. Example of IR tree.](image2)

<table>
<thead>
<tr>
<th>Table 1. Object description with qualities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objects</strong></td>
</tr>
<tr>
<td>P1</td>
</tr>
<tr>
<td>P2</td>
</tr>
<tr>
<td>P3</td>
</tr>
<tr>
<td>P4</td>
</tr>
<tr>
<td>P5</td>
</tr>
<tr>
<td>P6</td>
</tr>
<tr>
<td>P7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Content of the inverted files in IR tree node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invfile-root</td>
</tr>
</tbody>
</table>

2.2 Nearest Dominator Locations

Man Lung Yiu formalized the top-k spatial preference query, which returns the k spatial objects with the highest ranking scores. Objects are ranked, based on an aggregate score function that is defined for the feature qualities in their spatial proximity. Such score functions, however, do not support multidimensional dominance.
relationship. Therefore, the top-k spatial preference query is essentially different from the Farthest Dominated Location (FDL) query. Li⁴ defined the concepts of Nearest Dominators (ND) and combined dominance relationship with spatial distance which provides the solution for complex location selection problems. Li⁴ further describes the problem in different semantics. First, the desirable objects are selected. Second, the quality attributes are applied. Design competence vector and Euclidean distance were used to reduce the search space. But this work cannot be directly applied to solve the generic distance. In contrast, the technical solutions have been provided for such generic cases.

2.3 Nearest Dominators

The dominance comparison is suitable for comparing two objects with respect to multiple quality attributes. Dominance comparison combined with distance always produces satisfying results. The Figure 3 shows the locations of a set of competitors’ dams (dots) and a set of candidate locations (triangles) for building the dam.

![Figure 3. Example of dominated locations.](image)

The quality attributes of existing dams have been listed in Table 3, where irrigation and water supply values are preferable. Therefore, in case of a new dam being constructed within a specific area, comprising of other existing dams, the quality attributes must satisfy with the design competence.

Unfortunately, both the skyline query and traditional spatial queries fail to find a desirable location for building the new dam. By issuing a skyline query on the quality attributes of hotels shown in Figure 3.4, the obtained result set is {p1, p2, ..., p6}, which, however, offers no recommendation at all regarding the candidate locations {s1, s2, s3, s4}. Therefore, the skyline query is unable to select an appropriate location for building the new dam. Existing spatial queries are also not helpful here.

Let dist(s_i, p_j) denote the Euclidean distance between a location s_i and a dam p_j. The nearest neighbour of s_i is the dam p_j having the minimum value of dist(s_i, p_j). Table 4 shows the NN of each s_i. One may suggest finding the location s_i (for the new dam) such that its distance to its NN is maximized; however, this totally ignores the quality attributes of the dams.

![Table 3. List of dam locations with qualities](image)

<table>
<thead>
<tr>
<th>Dam</th>
<th>Irrigation</th>
<th>Water Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>P2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>P3</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>P4</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>P5</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>P6</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

![Table 4. List of locations with nearest neighbour and dominator](image)

<table>
<thead>
<tr>
<th>Location</th>
<th>NN</th>
<th>ND</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>P3</td>
<td>P3</td>
</tr>
<tr>
<td>S2</td>
<td>P4</td>
<td>P3</td>
</tr>
<tr>
<td>S3</td>
<td>P4</td>
<td>P5</td>
</tr>
<tr>
<td>S4</td>
<td>P6</td>
<td>P5</td>
</tr>
</tbody>
</table>

2.4 Nearest Dominator Score Calculation

Ranking with score is based on the quality attributes (e.g., facilities or services) in its spatial neighbourhood. Assigning higher weights to the features is based on their proximity to the object. The maximum and aggregation score for each feature in the neighbourhood region of the object is calculated. Generalization is the new technique k-NN query, which returns the ‘k’ closest objects to ‘q’ given a positive integer k. Nearest Neighbour queries can be efficiently processed using the best first algorithm, provided D is indexed by an R tree. A simple score binds the neighbourhood 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 p1(dam1) are 0.9 and 0.6. Hence the τ(p_1) is 0.9+0.6=1.5. Similarly
for the dam p_2, \(\tau(p_2) = 1.0 + 0.1 = 1.1\). Hence the dam p_1 is returned as the top result. The minimum and maximum latitudes are calculated using the following formula (2.1) and (2.2). Similarly, the longitude is calculated using (2.3), (2.4) (2.5) and (2.6).

\[
\text{latitude}_{\text{min}} = \text{latitude} - \text{radius of earth (r)} \\
\text{latitude}_{\text{max}} = \text{latitude} + \text{radius of earth (r)} \\
\text{latitude}_T = \arcsin(\sin(\text{latitude})/\cos(r)) \\
\text{longitude}_{\text{min}} = \text{longitude} - \Delta \text{longitude} \\
\text{longitude}_{\text{max}} = \text{longitude} + \Delta \text{longitude} \\
\Delta \text{longitude} = \arccos(\frac{\text{cos}(r) - \sin(\text{latitude}_T) \cdot \sin(\text{latitude})}{\cos(\text{latitude}_T) \cdot \cos(\text{latitude})}) \\
\]

### 2.4.1 NDL Score Algorithm

Given an object dataset O and a set of c feature datasets \(\{F_i | i \in [1, c]\}\), the top-k spatial preference query returns the k data objects \(\{p_1, \ldots, p_k\}\) from O with the highest score. The score of a data object \(p \in O\) is defined by the scores of feature objects \(t \in F_i\) in its spatial neighborhood. Each feature object \(t\) is associated with a non-spatial score \(w(t)\) that indicates the goodness (quality) of \(t\) and its domain of values is the range \([0, 1]\).

The score \(\tau(p)\) of a data object \(p\) is determined by aggregating the partial scores \(\tau_i^{\theta}(p)\) with respect to neighborhood condition \(\theta\) and the i-th feature dataset \(F_i\): \(\tau(p) = \text{agg}\{\tau_i^{\theta}(p) | i \in [1, c]\}\). The aggregate function \(\text{agg}\) can be any monotone function (such as sum, max, min), but sum is used in the equation internally. The partial score \(\tau_i^{\theta}(p)\) is determined by feature objects that belong to the i-th feature dataset \(F_i\) only, and in addition satisfy the user-specified spatial constraint \(\theta\). More specifically, the partial score \(\tau_i^{\theta}(p)\) is defined by the non-spatial score \(w(t)\) of a single feature object \(t \in F_i\). This feature object \(t\) is determined by feature objects that belong to the i-th feature dataset \(F_i\) only, and in addition satisfy the user-specified spatial constraint \(\theta\). More specifically, the partial score \(\tau_i^{\theta}(p)\) is defined by the non-spatial score \(w(t)\) of a single feature object \(t \in F_i\). The following list provides intuitive definitions of partial score for different neighborhood conditions \(\theta\) (where d() denotes the distance function):

- The range (rng) score of \(p\), given a radius \(r\):
  \[
  \tau_i^{\text{rng}}(p) = \max\{w(t) | t \in F_i : d(p, t) \leq r\} 
  \]
- The nearest neighbor (nn) score of \(p\):
  \[
  \tau_i^{\text{nn}}(p) = \max\{w(t) | t \in F_i, \forall v \in F_i : d(p, v) \leq d(p, t)\} 
  \]
- The influence (inf) score of \(p\), given a radius \(r\):
  \[
  \tau_i^{\text{inf}}(p) = \max\{w(t) \cdot 2^{-d(p, t)/r} | t \in F_i\} 
  \]

The NDL score algorithm calculates score for each feature and computes the upper bound score of an object and finally ranks the objects.

**Algorithm: NDL Score**

**Input**: Root of node N, Set of locations S and quality vector.

**Output**: Top-k objects with highest score.

- Step 1: Assume the initial root node as the node N.
- Step 2: Every edge e depends on its parent node N.
- Step 3: If node N is a non-leaf then
- Step 4: Read the child node N pointed by e;
- Step 5: Calculate the score value for the node N from the equation (2.7).

- Step 6: Else
- Step 7: P is considered as the object to be searched, V is the vertices. The objects are compared based on the query to find the best dam name.
- Step 8: For each feature data set from 1 to m do.
- Step 9: Compute the score of an object for all vertices V and edges e.
- Step 10: Update all the entry e in the minheap and maxheap. Assume W_k is used to represent the minheap and maxheap value.
- Step 11: End.

In this NDL score procedure; e represents an entry in the IR tree. Node is given as N. If N is a non-leaf, then insert a child node which is pointed by the edge and calculate the score for all nodes. Heap sort is used to sort the top rank. Two types of heap sort is implemented. To get the minimum value as a top k result min heap is used and to get the maximum value, max heap is used.

### 2.4.2 Direction Search Algorithm

This algorithm provides solution to the queries on direction.

**Algorithm: Direction Search**

**Input**: Root of node N, Set of locations S, Objects P and directions

**Output**: Object selected based on directions

- Step 1: Assume the initial root node as the node N
- Step 2: Every edge e depends on its parent node N
- Step 3: If node N is a non-leaf then
- Step 4: Read the child node N pointed by e;
- Step 5: Assume x and y as latitude and longitude values in spatial region
- Step 6: If the object P with latitude (p.x) is greater than the location object S with latitude (s.x) then
Step 7: Search the object P (dam) in east direction
Step 8: else
Step 9: Search the object P (dam) in west direction
Step 10: If the object P with longitude (p.y) is greater than the location object S with longitude (s.y) then
Step 11: Search the object P (dam) in north direction
Step 12: Else
Step 13: Search the object P (dam) in south direction
Step 14: Retrieve the object selection based on direction.
Step 15: End.

2.5 Ranking Spatial Objects

The spatial objects are ranked based on the qualities of their features. Traditionally, there are two basic ways of 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. The score of an object is set as the input. The branch and bound algorithm is used for ranking the objects. It reduces the number of objects to be examined and top-k objects are retrieved. A min-heap H which organizes R tree entries based on the (minimum) distance of their MBRs to q is initialized with the root entries.

2.5.1 Reverse Ranking

The reverse rank of an (dam) object o with respect to a given query objects q (that measures the relative nearness of q to o) is said to be k when q is the k-th nearest neighbour of o in a geographical space. For a given location, the dams are retrieved using IR tree. The distances between the retrieved objects to all other query locations are calculated. From this, find the kth nearness of the retrieved dams to the given location.

2.5.2 Reverse Ranking Query

Given a query object q and an object set O, an RR query RRQ(q,O,t) retrieve t objects from O with the smallest expected reverse ranks; formally,

\[ RRQ(q,O,t) = \{ r | r \in R \land \forall o \in (O-R)R_{\tau} < R_{-o} \} \quad (2.10) \]

Where \( R \subseteq O \land |R| = t \). The reverse rank algorithm retrieves the objects based on the reverse rank.

Algorithm: Reverse Rank

Input: Root of node N, Set of locations S, Result of objects P and directions, Quality attributes Q.

Output: Object selected based on reverse rank.

Step 1: Each object p belongs to the set of objects P (i.e., p \in P)
Step 2: Calculate minimum distance between p and s (i.e., min_dist(p,s)).
Step 3: Set the reverse rank for each object p by using equation 2.10.
Step 4: Retrieve the reverse rank for the dam objects.
Step 5: End.

Nearest Neighbour (NN) queries retrieve (dams) objects and sort them according to their distances from a given query object in a spatial database. Alternatively, it can be an important and interesting perspective to search objects that have a given query object as their k-NNs and to rank them. Owing to asymmetric nearest neighbour relationships, it is not straightforward to identify those objects based purely on their distances from a query object. Therefore, the notion of reverse ranks has been recently defined. When a query object q is the k-th nearest neighbour of an object o, the reverse rank for ‘o’ with respect to q is defined as k. Based on k’s, reverse ranking queries (or RR queries) t objects are retrieved, ordered by their k’s with respect to given query point, where t is specified at the query time to limit the result set size.

For a given location, the dams are retrieved using R tree. The distances between the retrieved objects to all other query locations are calculated. From this it can find the kth nearness of the retrieved dams to the given location.

3. Incremental Quality Based Reverse Ranking for Spatial Data

3.1 System Overview

The architecture for the proposed efficient evaluation of spatial information is shown in Figure 4.
Proposed Work:

Incremental Quality Reverse Ranking (IQRR) Technique

Construct IR Tree
Generate map
NDL score calculation
Find Nearest Dominator

Performance Evaluation

Figure 4. System architecture of the proposed evaluation of spatial information using superior preferences.

In the proposed work, the R tree which is for spatial query and the IR tree which is for spatial as well as non-spatial query have been combined to increase the performance and robustness. The main objective of this work is to reduce the search time in finding the top-k spatial objects based on the proposed work called Incremental Quality Reverse Ranking (IQRR) and to improve the effectiveness in multiple query processing.

3.2 Comparative Analysis of R tree and IR Tree

Basically, the R tree uses a parameter such as distance (based on radius), whereas the IR tree uses the parameters, distance and keywords. But the proposed work focuses mainly on the additional parameters, such as quality attribute for incremental quality based on reverse ranking. The double refined process works in an efficient way and retrieves information, which is needed by the user. The performance of the proposed work has been improved to 72.5% better retrieval accuracy than that of the R tree and IR tree on processing query retrieves effectively, and produces more efficient and accurate results thereby reducing and the response time when compared with the existing trees.

3.3 Proposed Incremental Quality Reverse Ranking (IQRR) Algorithm

This algorithm retrieves an object along with incremented quality and reverse ranking.

Algorithm: Proposed Incremental Quality Reverse Ranking

Input: Root of the node, Set of locations S, Result of objects P and directions

Output: Object selected based on reverse rank with incremental quality attributes

Step 1: Each object p belongs to the set of objects P (i.e., p \( \in \) P).

Step 2: Calculate minimum distance between p and s using quality attributes (i.e., \( \text{min}_\text{dist}(p, s, Q) \)).

Step 3: With the given node N, Location S, Object P, the direction is found (Refer Direction Search Algorithm in 2.5.2).

Step 4: Retrieve the dams based on directions.

Step 5: Calculate the removal of imprecise data based on benchmark value according to the formula

\[ B_1 = B_0 - V \]

Where B1 is the number of dams retrieved after benchmark value, B0 is the number of actual dams retrieved and V is the benchmark value.

To find the error rate E:

\[ E = \frac{B_1}{B_0} * 100 \]

Step 6: Apply the reverse ranking algorithm (Refer Reverse Rank Algorithm in 2.6.2) to retrieve the object based on the rank by using quality attributes.

Step 7: Set the reverse rank for each object based on incremental qualities.

Step 8: Retrieve the reverse rank with incremental qualities.

Step 9: Retrieval of dams with qualities.

Step 10: End.

For a given location, the dams are retrieved using R tree. The distances between the retrieved objects to all other query locations are calculated. From this it can find the kth nearness of the retrieved dams to the given location.

3.3.1 Sample Spatial Queries

// To create Spatial Index create index tbl_dams_africa_idx ontbl_dams_africa(system.get_long_lat_pt(ddlong,ddlat)) indextype ismdsys.spatial_index;
create index tbl_dams_middleeast_idx ontbl_dams_middleeast(system.get_long_lat_pt(dd_long,dd_lat)) indextype ismdsys.spatial_index;

// To calculate distance selectnamesdo_geom.sdo_distance(system.get_long_lat_pt(c.ddlong,c.ddlat), system.get_long_lat_pt(15.150755125,-12.5719879375), 0.005)*111 distance from tbl_dams_africa c where name = ‘biopio’; select c.name, c.ddlong, c.ddlat from
tbl_dams_africa c where sdo_within_distance(system.get_long_lat_pt(c.ddlong,c.ddlat), sdo_geometry(2003, null, null, sdo_elem_info_array(1, 1003, 1), sdo_ordinate_array(30.3, 29.1)), ‘distance=1000’) = ‘true’;

selectsdo_geom.sdo_distance(system.get_long_lat_pt(“+ lon + “, “+ lat + “), “+ system.get_long_lat_pt(“+ lon-search + “, “+ latsearch + “), 0.005)*111 distance from dual;

4. Experiments and Results

To evaluate the performance of the proposed approach, a comparative study has been performed in this work. The R tree and IR tree are compared with the proposed work namely Incremental Quality Reverse Ranking (IQRR). The evaluation parameters are radius, time, location, directions and number of dams.

4.1 Search Results for the given Location and Quality

Based on the experimental results, Figure 5 shows that the performance of IQRR when it is compared with the existing R tree and the IR tree with respect to radius and number of dams.

In Figure 5, with 1000 kms radius, the R tree retrieves 210 dams (distance), IR tree retrieves 150 dams (distance, keyword) and the proposed method IQRR retrieves 30 dams (distance, keyword and qualities). So, it clearly shows that the retrieval of dams is based on radius and keyword along with quality attributes, results in efficient work by the proposed method.

4.2 Retrieval Response Time Comparison

According to the Figure 6 has been constructed which shows the comparison of existing and the proposed methods based on response time along with distance for searching the data. In terms of distance i.e., with 1000 kms, R tree requires 138 ms to retrieve the query. IR tree requires 72 ms and the proposed method requires 68 ms to retrieve the query. Hence, the proposed method retrieves the query more efficiently than the other two existing methods.

4.3 Retrieval Response Time Comparison using Score Value

By using incremental quality attributes based on score value, IR tree takes 65 ms for 2000 kms. But for R tree it takes 155 ms and IQRR takes only 43 ms. The comparison performance evaluation is shown in Figure 7.

Based on the distance and the incremental quality attributes comparative study is done with R tree, IR tree and proposed method IQRR. The basic factor to be considered, Figure 8 is the response time without score value.
4.4 Performance Measures using Removal of Imprecise Data

The Comparative search results between proposed and existing tree structures for removing imprecise data based on distance for dam retrieval is shown in Figure 9 based on the result values the Figure 9 predicts that R tree does not remove any imprecise data if it is available in the database whereas IR tree removes some imprecise data at a reduced rate only. Whereas the proposed method Incremental Quality Reverse Ranking (IQRR) removes the imprecise data efficiently and only exact data retrieves for a given query.

In Figure 9, for 1000 kms, R tree does not remove any imprecise data and it retrieves around 200 dams. But IR tree removes some imprecise data and it retrieves 106 dams. In the proposed method IQRR removes more imprecise data and it retrieves 80 dams i.e., number of dams is accurate in IQRR tree with reverse ranking when compared with the other two existing methods.

4.5 Measurement of Accuracy

The proposed work, IQRR accuracy is quantized based on the performance rate which is 72.5% high for the proposed work when compared with other existing works and is shown in Figure 10.

For finding the accuracy, the following formula is evaluated.

\[
\text{Accuracy} = 100 - \frac{\text{Error rate}}{\text{Number of actual dams retrieved}}
\]

For example,

Number of actual dams retrieved by R tree in 1000 kms = 200.

Let assume benchmark value as 50.

Number of dams retrieved after benchmark = 200 – 50 = 150

Error rate = \frac{150}{200} \times 100 = 75

Accuracy = 100 - 75 = 25

Similarly, follow the above steps to find out the accuracy for R tree, IR tree and proposed IQRR tree with different benchmark value.

4.6 Search Result for Different Direction Search

Figure 11 shows the number of dams displayed for different directions for a location 'Algeria' with qualities. It is evaluated using R tree, IR tree and the proposed method.
The number of data displayed using IQRR is less than R tree and IR tree.

Basically, IR tree retrieves the value along with keyword and distance. Proposed method IQRR retrieves the value along with keyword, distance and quality attributes. Hence the input is given as directions and the two quantities such as irrigation and water supply is specified. Based on the directions and quality, the IR tree retrieves 74 dams for north direction. But for the same north direction R tree retrieves 170 dams and the proposed method IQRR retrieves only 23 dams. The proposed method retrieves less number of dams as the retrieval is based on incremental quality attributes thus leading to more specific retrievals and reducing the search time. Similarly for south, east and west directions, the number of dams retrieved for R tree and IR tree comparison with IQRR is shown in Figure 11 which illustrates that retrieval of number of dams using in comparison with R tree and IR tree.

### 4.7 Search Result with Different Qualities based on Keywords

Figure 12 shows the IQRR method to retrieve the number of dams based on two countries Algeria and Benin. It searches based on the quality, i.e. keyword. With single keyword, Algeria displays 70 dams and Benin displays 320 dams. So whenever the quality, i.e. keyword increases, the number of dams displayed is less.

Based on the values are plotted in the Figure 13. This shows the time taken to search the number of dams which satisfies the quality. With single keyword retrieval required 20 and 36 ms and with two keywords 26 and 47 ms are taken for Algeria and Benin. So when the quality (keywords) increases the response time also increases.

### 5. Conclusion

In this paper, a novel approach called Incremental Quality Reverse Ranking (IQRR) for efficient evaluation of spatial information and performance of top-k spatial preference query processing has been proposed. The proposed approach focuses on the quality features with
reverse ranking and compares it with the existing R tree and the IR tree and provides the best results and reduces the search time and complexity. The proposed algorithm, Incremental Quality Reverse Ranking (IQRR) has been efficiently evaluated thus leading to 72.5% accuracy in retrieval of dams and reduction in search time.

6. References