International Journal of Computer Applications Technology and Research
Volume 2– Issue 2, 176 - 179, 2013, ISSN: 2319–8656

Spatial Keyword Ranking (SKR) based Dominated Location with Safe Zone

Krishna Balan
Pondicherry Engineering College
Pondicherry, India

Karthiga
Pondicherry Engineering College
Pondicherry, India

Sakthi Priya
Pondicherry Engineering College
Pondicherry, India

Abstract: Generally, spatial objects (e.g., Hotels) not only have spatial locations but also have quality attributes (e.g., star, price etc). In wild animal rehabilitation, an appropriate location is selected from a set of options for returning an animal to the nature, after its treatment in a rehabilitation center. A location is preferred if it is far away from (competing) animals with multiple better abilities (e.g., speed, weight, age). Since those animals are much stronger in fighting for essential resources, e.g., water and food, they will endanger the rehabilitated animal that is still unaccustomed to wildlife. By maximizing the distances to potential nearest dominators (NDs), the preferred location improves the survival chance of the rehabilitated animal in the nature. Hence the safe zone is created to retrieve result within the region. If the query moves from the safe zone another safe zone is created fot that query, the intersected part are purned. These reduce the database storage.

Keywords: Anonymizer; Database Management; Safe zone; Spatial Database; Spatial Objects;

1. INTRODUCTION

In reality, spatial objects (e.g., hotels) not only have spatial locations but also have quality attributes (e.g., star, price). An object p is said to dominate another one p_0, if p is no worse than p_0 with respect to every quality attribute and p is better on at least one quality attribute. Traditional spatial queries (e.g., nearest neighbor, closest pair) ignore quality attributes, whereas conventional dominance-based queries (e.g., skyline) neglect spatial locations. From this Farthest Dominated Location (FDL) [1] retrieves the results, includes both quality attributes, and spatial objects with sufficient R-Tree algorithm to retrieve the data. For each query, Location based Server (LBS) need to analyze the query objects and the query location. Then LBS search its entire database to give respective queries result. This process is effective for static object. If it is for dynamic object means the server sends the result for one location point that is not sufficient if the object moves to another location. For this, the proposed system include safe zone. This zone creates a circular zone with range for the query, and location will be analyzed. We propose an efficient index called Spatial Keyword Ranking (SKR) Tree data structure which performs 1) Spatial filtering, 2) Textual filtering and 3) Object ranking in a fully integrated manner.

2. RELATED WORKS

Terminal devices, which are the monitored moving objects, obtain their own location information from the GPS system and transmit it to the server via a wireless communication network. The whole system's timeliness and efficiency is affected by the wireless communication bandwidth. The location information updates are often the bottleneck because of the limited wireless bandwidth and the high sampling rate in the traditional uniform time/distance interval strategies.The idea behind the rectangular safe region (RSR) algorithm is to define a rectangular safe region for every object according to the registered query and the latest location obtained. As long as the object's motions do not exceed its safe region, all the query result sets of the object remain unchanged.

The terminal device is informed of the safe region assignments dynamically. Hence when a terminal device finds that it has exceeded the safe region, it will report its new location information.

2.1 Naive Iterative Incremental (NII) NN Search

Naive Iterative Incremental NN (nearest neighbour) search is for processing the FDL query. NII takes as input 1) a two dimensional R-tree Rp on the spatial object set P, 2) a set of locations L, and 3) a c-dimensional design competence [4]. It examines each location s of set L and computes the nearest dominator of s, by performing the incremental nearest neighbour search [3] of s on the tree Rp.

During the iterative search, the largest ndd (nearest dominated distance) is maintained together with the corresponding location. After all locations are examined, the maintained location is returned as the query result.

2.2 Best-First Search (BFS) Algorithm

The Best-First Search Algorithm is a basic graph-searching algorithm. Best-First makes use of a heuristic (or quickly computed estimate of the cost to reach the goal from each node), called $\hat{h}$, to guide its search. The idea behind using a heuristic to guide the search is that the algorithm will not waste time probing paths that do not seem likely to lead to the goal state (node). However, this means that an inaccurate $\hat{h}$

function can misguide the search, which can result in the search finding a path to the goal that is not optimal. For this reason, one of the greatest challenges in graph searching is to develop a heuristic function that can be quickly computed and that will be a very close estimate of the actual cost to the goal.

2.3Spatial Join Based (SJB) Algorithm

Spatial join algorithm can be classified into three categories. Assume that it have spatial join relation on R1 and R2

2.3.3 Nested Loop

In this algorithm, for each tuple of R1, entire R2 is scanned; any pair of tuples of R1 and R2 which satisfies the spatial join predicate is added to the result.

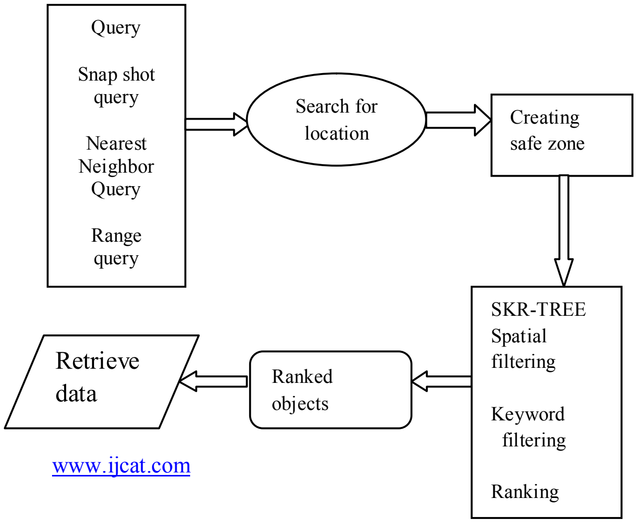
2.3.4 Tree Matching

Tree matching algorithm can be applied when indices are available on both the relations. For this discussion, it will assume that R-tree index is available. When index exists for only one relation, the index on the other relation is built on the fly and tree-matching technique is applied.

2.3.5 Partition-Based Spatial Merge Join

In this algorithm, first both of the relations are divided into *p* partitions if both of them do not fit in main memory. After that partition *i* of R1, where $1 \le i \le p$, is compared with corresponding partition *i* of R2. This strategy is very good when no indices are present on both the relation.

3. DESIGN FOR PROPOSED WORK



4. SKR TREE

We propose an efficient indexing scheme called SKR tree (Spatial Keyword Range tree),which indexes both the textual and spatial contents of objects to support data retrievals based on their combine textual and spatial relevance, which, in turn, can be adjusted with different relative weights. In fig.1 structure of SKR tree has nodes which have both spatial and non spatial information of the data object.

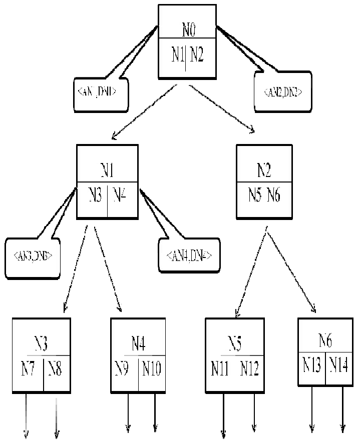


Figure 1. Structure of SKR tree

N_1 and **N_2** are the child nodes,
AN_1 - keyword, and **DN_1** - spatial data.

SKR Tree Construction Algorithm

Input: Set of Objects D
Output: Root of SKR tree

Procedure:

```
Ne ← 0
    For each p €D do
        geo code p and represent Lp with MBB mp
        if for some e €Ne, me = mp then
            add p to e's dataset De;
        else
            create a new entry e;
            set me ← mp and De ← {p};
            Ne ←Ne U {e};
        End if
    End for
    For each e € Ne do
        While Ne> n max do
            Cluster the data according to min/max into nodes
            Ne ← Ne′
        End while
    End for
```

5. SAFE REGION BASED LOCATION UPDATES

Figure 2 demonstrates the infrastructure of a moving object query system. The kernel of the system is the control center (the main server of the system) in the center of the figure which runs the Moving Object Device (MOD) engine, collects location information, handles continues queries and provides query results to the application servers to the right of the figure. Therefore, the major computation workload is applied to the main server/control center of a MOD system. For simplicity, we refer to the main server/control center as server. Terminal devices, which are the monitored moving objects, obtain their own location information from the GPS system and transmit it to the server via a wireless communication network. The whole system's timeliness and efficiency is affected by the wireless communication bandwidth. The location information updates are often the bottleneck because of the limited wireless bandwidth and the high sampling rate in the traditional uniform time/distance interval strategies. As long as the object's motions do not exceed its safe region, all the query result sets of the object remain unchanged shown in **Figure 3**.

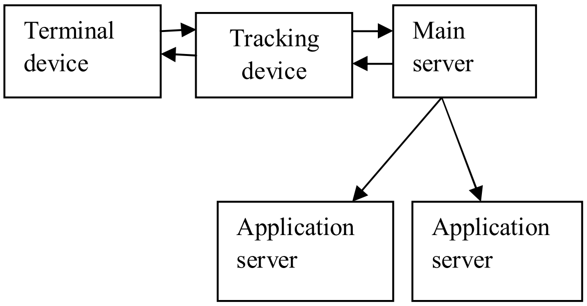


Figure 2. Infra structure of object system

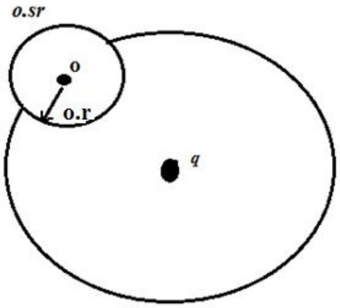


Figure 3. Location update of a moving object with circular safe region under continuous query.

The safe region of object *o* (referred to as *o.sr*) as a circle with the center at the location of the object and the radius of *o.r* (**Figure 3**). By assuming the maximum speed of the object is *o.maxspd*, then the continuous query result of query *q* will not be affected within the time interval of *o.r*/*o. maxspd*. Hence the server can issue a location report query to the terminal device at the time of (*o.r*/*o.maxspd* -δ)
where

δ is the sum of communication and computation delays.

o.r is Radius of the object safe region.

o.sr is Object safe region.

o.maxspd is Maximum speed of the moving object.

5.1 Circular Safe Region Calculation and Updates for Continuous kNN Queries

Following is the formula to update the safe region radius for the i^{th} object in the object set ascending sorted by distance to ordered kNN query q. The first object in the result set is q and the extra object o_{k+1} is kept for calculation of safe region radius of o_k.

$$o\mathrm{i.r} = \begin{cases} min\left\{oi.r, \dfrac{dist(oi,oi-1)}{2}, \dfrac{dist(oi,oi+1)}{2}\right\}, if\ o < i \leq k \\ min\{oi,r, dist(oi,q) - quar(q), if\ i > k \end{cases}$$

where *quar*(*q*) is the radius of the quarantine region for query *q* which surround and only surround the safe regions of all objects in the result set. Therefore, *quar*(*q*) = *dist*(o_k, *q*) + o_k.

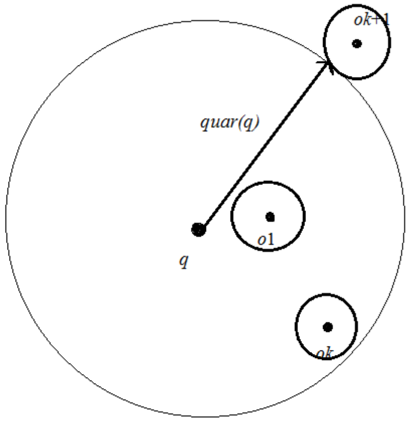


Figure 4. Quarantine region assignment for an ordered kNN query.

Figure 4 shows how such a quarantine region is assigned for such an ordered kNN query *q*. Hence the following property: either in or out of the kNN query result set (inside or outside the quarantine region), none of the safe regions of objects overlaps with each other. Therefore, when all the objects are moving inside their own safe regions, the result set and its order are not affected.

6. CONCLUSION

Safe zone is created to get the optimal location while comparing with farthest dominated location. Safe zone is created so as to reduce the search space and time for server because it retrieves the data within that region specified by the user. A quarantine region is also assigned for one or more objects present in the same zone. This project proposes an efficient index SKR-tree and algorithms which perform spatial filtering, textual filtering and object ranking in a fully integrated manner is expected to have reduce the database storage and reduction of CPU time.

7. REFERENCES

[1] Hua Lu and Man Lung Yiu, “On Computing Farthest Dominated Locations”, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 6, pp. 928-941, 2011.

[2] M.A. Cheema, L. Brankovic, X. Lin, W. Zhang, and W. Wang, “Multi-Guarded Safe Zone: An Effective Technique to Monitor Moving Circular Range Queries”, Proceedings of IEEE 26th International Conference on Data Engineering (ICDE), pp. 189-200, 2010.

[3] G. Hjaltason and H. Samet, “Distance Browsing in Spatial Database”, ACM Transaction on Database Systems, Vol. 24, No. 2, pp. 265-318, 1999.

[4] N. Roussopoulos, S. Kelley, and F. Vincent, “Nearest Neighbour Queries”, Proceedings of ACM International Conference on Management of Data, pp. 71-79, 1995.

[5] Y. Du, D. Zhang, and T. Xia, “The Optimal-Location Query”, Proceedings of International Symposium on Spatial and Temporal Databases, pp. 163-180, 2005.

[6] X. Huang and C.S. Jensen, “In-Route Skyline Querying for Location-Based Services”, Proceedings international Workshop on Web and Wireless Geographical Information Systems, pp. 120-135, 2004.

[7] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An Optimal and Progressive Algorithm for Skyline Queries”, Proceedings of ACM International Conference Management of Data, pp. 467-478, 2003.