

Spatial Keyword Search by using Point Regional QuadTree Algorithm

Ms Harshitha A.R¹, Smt. Christy Persya A²

PG Scholar, Department of ISE (SSE), BNMIT, Bangalore – 560085, Karnataka, India¹
Associate Professor, Department of ISE, BNMIT, Bangalore – 560085, Karnataka, India²

Abstract: Due to the current advances in geo-positioning technologies and geo-location services, the amount of spatio-textual objects collected in numerous applications are quickly expanding. These spatio-textual items are collected from area based administrations and informal organizations, in which an item is described by its spatial location that is “latitude” and “longitude” and a set of keywords describing the items. Consequently, the study of spatial keyword search which explores both location and textual description of the objects has attracted great attention from the commercial organizations and research communities. In this paper, we mainly concentrate on fundamental problem in the spatial keyword search queries of the top k spatial keyword search method. In previous approaches, for extracting Top k result out of data in online business directory is not having efficiently relevance and faster result. In this paper, we are going to propose the method that will give the unique index structure, called point regional quadtree, which is carefully designed to exploit both spatial and keyword based pruning techniques to effectively reduce the search space. Given a set of spatio-textual objects, a query location and a set of query keywords, the generic model of the point regional quadtree retrieves the closest k objects each of which contains all keywords in the query. In this paper, how the proposed technique is used to calculate distance and direction for the nearest spatial objects is also discussed.

Keywords — Spatial, Keyword, Point Regional Quadtree, Spatial database, Indexing .

I. INTRODUCTION

In the recently increasing pervasiveness of the geo-positioning technologies and geo-location services, there is a lot of spatio-textual objects accessible in numerous applications. For instance, in the local search service, online business directory (e.g., yellow pages, Mojo Pages) provide the location information and also short descriptions of the organizations (e.g., hotels, eateries). In addition, several social network services (e.g., Facebook, Flickr), a large number of geo-tagged photos are accumulated regularly, which is be geo-tagged by users [1]. These transferred photographs are typically connected with various content names. As a result, in recent years various spatial keyword

query models and techniques have emerged such that users can effectively exploit both spatial and textual information of these spatio-textual objects.

In the paper, we explore the issues of leading TOP-K spatial keyword search [2]; that is, given a group of spatio-textual objects, a query location q and a group of keywords, mean to recover the k nearest objects each of which contains all keywords in the query. The top k spatial keyword search is major in spatial queries and has a wide range of utilizations. In numerous real applications, the query workload may fluctuate every now and then, and the framework may experience a burst of queries summoned by a specific event. In this circumstance, the framework is poor if countless are handled one by one. Motivated by this, a large body of existing work has been done to examine how to enhance the framework all through with the point regional quadtree handling methods to such that a large number of queries in the queue can be processed with a reasonable delay[2] [3].

The roadmap, The rest of the paper, is organized as follows.

Section 2 literature survey. Section 3 presents a proposed system of general model point region quadtree. Section 4 implementation of proposed system. Section 5 concludes the paper.

II. RELATED WORKS

A. Information retrieval R-tree

Information retrieval R-tree is tree data structures utilized for getting to spatial information. R-Trees can arrange any-dimensional information by representing minimum bounding nodes. Each node bounds its children. Every node of an R-tree has a variable number of sections. Each entry within non-leaf node stores two bits of information a method for distinguishing a child node, and the bounding box of all entries within this child node. R-tree possibility of overlapping and minimum bounding box [4]. Information R-tree was sufficient method when there is only one query keyword .It keeps away from numerous path during searching. The performance of the R-tree is little when the quantities of keywords are increasing in the query and furthermore, there is no different ordering for the spatial and textual object.

B. Information retrieval R+ tree

R+ -tree is information structure. It trades off impediments of R-trees. R+ tree avoids overlapping of internal nodes by inserting an object into many leaves. R+ tree keeps up standard ordering for both spatial and textual data. Here nodes are not overlapped with each other. The r+ tree has a few downsides such as development and upkeep of R+ trees is more complicated than the design and maintenance of R trees, if any insertion and deletion operation is performed means restructuring full index and R+ is not balanced [4][5].

C. Keyword Search on spatial Databases

Numerous applications require discovering objects closest to a particular area that contains a collection of keywords. For instance, on-line phone directory empowers clients to determine an address and a group of keywords. Consequently, the client acquires a stock of organizations whose depiction contains this keyword, ordered by their distance from the required address. The issues of closest neighbor search on spatial data and keyword search on content data are extensively studied individually [6]. However, to the simplest of our information there's no practical philosophy to answer spatial keyword queries, that is, queries that determine each an area and a collection of the keyword.

D. Distance Browsing in spatial Databases

R-tree spatial data structure stores a collection of spatial objects. There are two distinct procedures of perusing through a collection of spatial objects put away in an R-tree spatial information structure on the premise of their distance from a discretionary spatial query object are compared. The ordinary approach is one that makes utilization of a k-closest neighbour algorithm where k is known before the invocation of the calculation. In this way if $m > k$ neighbors are required, the k-closest neighbor calculation should be re invoked for m neighbors, accordingly conceivably playing out some repetitive calculations[7]. The second approach is incremental as in having gotten the k closest neighbors; the k + first neighbor can be acquired without calculating the k+1 closest neighbor from scratch. The incremental approach discovers utilize when preparing complex inquiries where one of the conditions includes spatial proximity (e.g., the closest city to san francisco with a population greater than a million), in which case a query search engine can make utilization of a pipelined methodology[8].

E. Efficient processing of top-k spatial keyword queries

In a spatial location and a set of keywords, a top-k spatial keyword query gives back the k best spatio-

textual objects positioned by their query location and the query keywords. There are numerous applications dealing with tremendous measures of geo labeled information, for example, Twitter and Flickr that can profit by this query. Unfortunately, the state-of-the-art approaches require non-insignificant preparing cost that brings about in long reaction time. They propose a novel file to enhance the execution of top-k spatial keyword queries named Spatial Index (S2I) [9][10]. Their list maps each distinct term to a set of objects containing the term. The items are put away diversely according to the document frequency of the term and can be retrieved efficiently in decreasing order of keyword relevance and spatial proximity. Moreover, S2I algorithms exploit top-k spatial keyword queries efficiently. Finally, S2I Efficiently retrieved in decreasing order, but it is highly expensive [10].

F. Quadtree

A Quadtree is a tree information structure in which each interior hub has precisely four kids. Quadtrees are the two-dimensional analog of octrees and are regularly used to partition a two-dimensional space by recursively subdividing it into four quadrants or area. Quadtree hubs are based on the Morton code (Z-arrange)[11]. In Morton code, the hubs are encoded in light of its split succession, i.e., the way of the hub in the Quadtree, and the system of a particular hub (district) in the space are one of a kind. This is basic on the grounds that many Quadtrees with various shapes are utilized. Quadtree structure is kept as the space segment based mark of the objects, and consequently, the level of a hub in the Quadtree is accessible in a period of query processing [11][12].

III. PROPOSED METHODOLOGY

To overcome downsides of other technique we propose an entirely different index structure, specifically generic model of point region quadtree, to prepare the spatio-textual objects. By taking advantage of the generic model of PR Quadtree new algorithmic program is developed to find the nearest TOP-K spatio-textual objects. Point regional quadtree aims to retrieve the K-closest object which contains all keywords in the query. Section 3.1 describes the inspiring cases of the keyword constraints. Section 3.2 presents the system overview of proposed system. Section 3.3 presents an efficient algorithm for the implementation of a generic model of the quadtree.

A. Inspiring of existing system

The Fig. 1 shows the inspiring cases of the keyword constraints [13]. Assume there are set of organizations whose areas (represented by squares) and administration records (a set of the keyword) are enrolled in the online business catalog of a local search service provider. At the point client needs to locate a close-by eatery to have a south Indian food and Chinese food, the client may send the local search server two keywords, south Indian food, Chinese food. In light of the customer's current area (e.g., the point q in Figure. 1) and the two query keyword, business p1 is returned by the server. Take note of that in spite of the fact that organizations p3 and p2 are nearer to q than p1; they don't fulfill the keyword constraints.

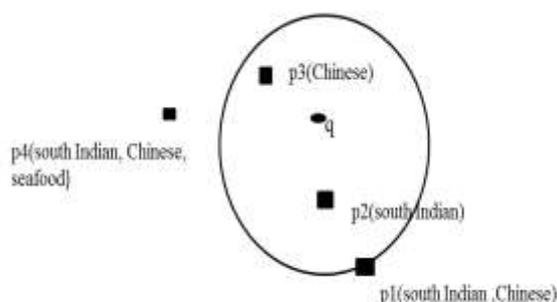


Figure. 1: Online Yellow Pages Example

B. System Overview

In this section, high-level summary of the system is discussed. Figure 2 shows the system architecture of the generic model of the proposed system. On the client side, the user can find the TOP-K nearest places with the required keywords. Here user supposed to set the current location and enter a query. Spatial keyword search which investigates both area and a textual description of the item to be searched. Based on the composite index structure of the point regional quadtree, both spatial and textual keyword is searched and it also actually reduces the search space. Point regional quadtree maintains separate indexing for the spatial and textual object.

C. Point Regional Quadtree Based Algorithm

In this section, algorithms of point regional quadtree are discussed. Implementation of the generic model of point regional quadtree as two algorithms.

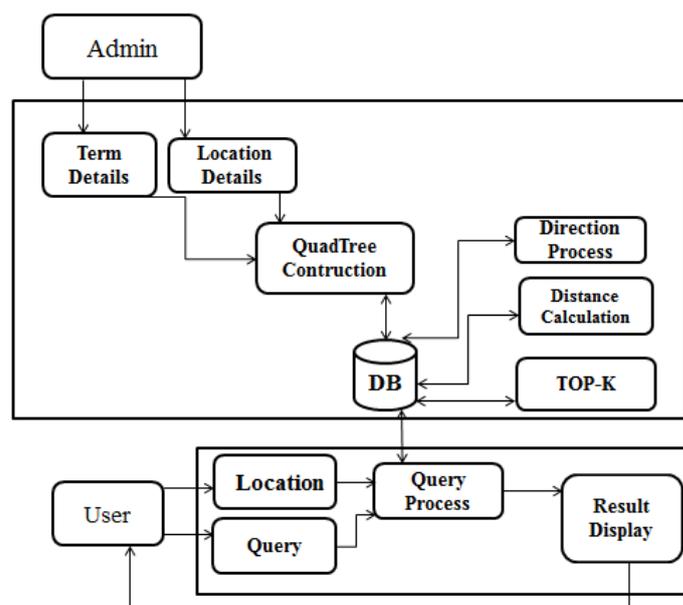


Figure.2: System Architecture

Algorithm1. PR Quadtree Construction

Input:

N: number of places

M: number of centers

T: terms details

R: root node

Children's(C): NE, NW, SE, SW

P = Process

Output: PR: point regional quadtree

1. $R \leftarrow$ Set the root node in the tree;
2. Partition root node into 4 regions or children's \rightarrow NE, NW, SE, SW;
3. for each m values inserted into tree for i 1 to m;
4. if (boundary. Centre (range));
//Check object appear within the range
5. Subdivide (m);
Centroids node again recursively divided into four regions
// partition the tuples and grows subtrees for each partition
6. else return (empty list);
7. for each n values ,add points whichever the region will accept it ;
8. Update (n) values;
9. Return PR;

Algorithm 1 illustrates the details of the construction of Point regional quadtree .Where R is the root node in the tree. In Line 1, set the root node in the tree .Partition of root node into four region or children's NE, NW, SE, SW in Line 2. For each value of m (centroid) inserted into quadtree. Check if the value of m appears within the boundary range (Line 4) of the root node. Centroids(m) is recursively subdivided into four regions. Following the split, sequence helps to grow subtrees for each partition

(Line 5). Line 6 return empty list if centroid value not within boundary range. N be a number of places, add each n value whichever the region will accept it (Line 7-8). The algorithm terminate final result is a generic model of point regional quadtree.

Algorithm 2. PR quadtree search Algorithm

Let S is the Whole System Consist of
 $S = \{I, P, O\}$
 I: Input.
 $I: \{U, Q, D\}$, U: User, $U: \{u_1, u_2, \dots, u_n\}$, Q: Query Entered by user, $Q: \{q_1, q_2, q_3, \dots, q_n\}$, keyword: $\{KW\}$
 D: Dataset
 P: Process
 O: R:TOP-k query result w

1. Input GPS(U) & keyword (KW);
2. M be number of centres;
3. Initialize array D(M);
4. for $i=1$ to M;
5. $GPS(U) == GPS(C)$;
6. Check current entry GPS(U) is the leaf node of current PR quadtree;
7. Input keywords(KW) == place keywords(KW');
8. $D(j) < \text{dist}(GPS(U), GPS(P))$;
9. $\min D(J)$;
10. Filter the place with KW;
11. sort the value in array D;
12. return R;

Algorithm 2 illustrates the details of the PR quadtree search algorithm. It aims to retrieve the K-closest object which contains all keywords in the query. Initially, the user will enter the query keyword (KW) and also set the current location GPS (U) in Line 1. After entering a query, the following operations will be performed in Line 2. Initialize array D(M), for centroids node. While searching compare GPS (U) with the GPS(C). Check whether current entry GPS(U) is the leaf node of current PR quadtree in Line (5 6). Compare query keyword with bussiness places keyword in the database, after filtering out the keyword then find the distance between user current location to the business location which satisfies the keyword constraints. Finally, calculate the distance between user current location to organization location. The algorithm terminates, and results are kept in R.

IV. IMPLEMENTATION

In this section, implementation of the generic model of the point regional quadtree is discussed. To find a nearest spatio-textual object in the online business directory is efficiently achieved through implementing a generic model of the PR quadtree .This implementation consisting of two modules: Admin and User module.

A. Admin Module

In this module, admin is supposed to maintain the keywords and location details. Based on these features point regional quadtree is constructed. A quadtree is implemented by using quadtree construction algorithm proposed in section 3.3.

| | |
|---------|--------|
| NW (10) | NE(11) |
| SW(00) | SE(01) |

Figure 3a Morton code

In this section how the generic model of the quadtree is implemented as shown in figure 3. Now we describe how to drive the Morton code of a node based on the split sequence in two dimension space. As shown in Fig. 3(a), here quadtree resulting from a division are numbered in the order SW, SE, NW and NE, which are represented by 00, 01, 10 and 11 respectively. Fig. 3 shows the space partition of a simple point regional quadtree for a given set of points $\{p_1, \dots, p_n\}$.

Set the root node in the tree that is point $p_1(3,7)$. Based on value ,quadtree resulting from splits into four regions are NE,NW,SE,SW as shown in figure 3(b). One's the root node is set ,then insert the centroid nodes(center places of the city) into the tree. For example figure 3(c) show the point p_2 inserted into point regional quadtree. By considering p_1 as parent node, point $p_2(6, 6)$ is coming under the south east region of the tree. Again node $p_2(6,6)$ is recursively divided into four area or children. Before inserting a node into the tree check whether a point is within a boundary or not . Then add next node into tree point $p_3(7, 4)$ starting from root node p_3 comes under NE region, already node p_2 is there in NE region now p_3 considered the p_2 as parent node and inserted into the tree Same procedure will follow for adding n number of centers points into the tree. By these, all the centroid nodes are recursively subdivided into four regions and grow subtrees for each partition as shown in figure 3c, 3d, 3e. After adding centroid node of cities then add places (business organizations) of the city. Insert the n locations into tree whichever the region will accept it. Here admin maintains details of spatial and textual data of a business organization.

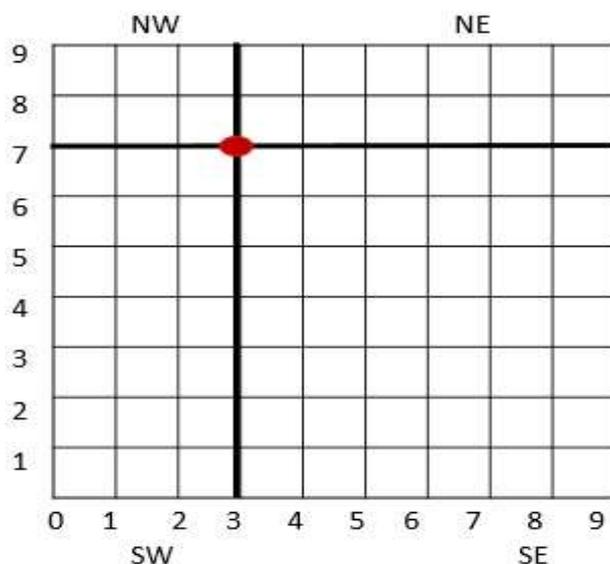


Figure 3b

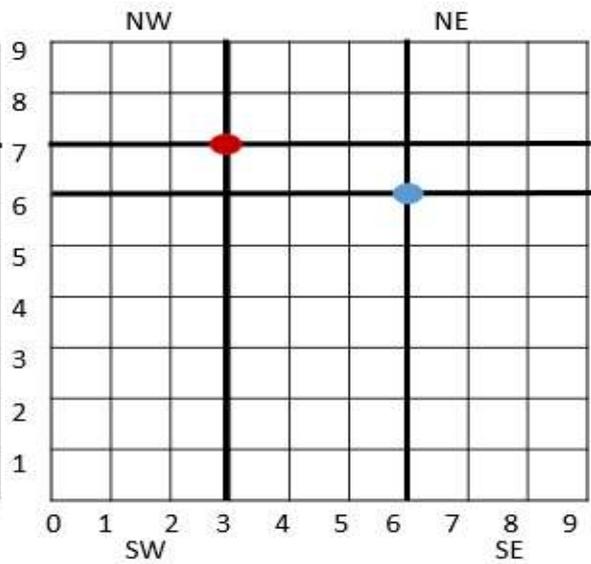


Figure 3c

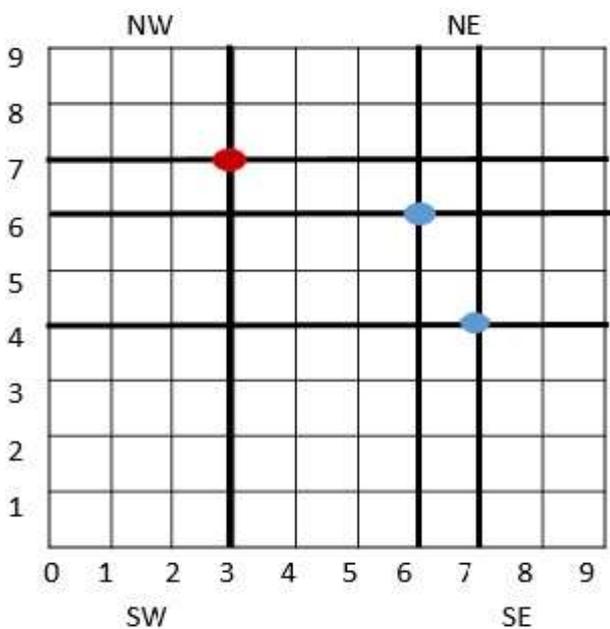


Figure 3d

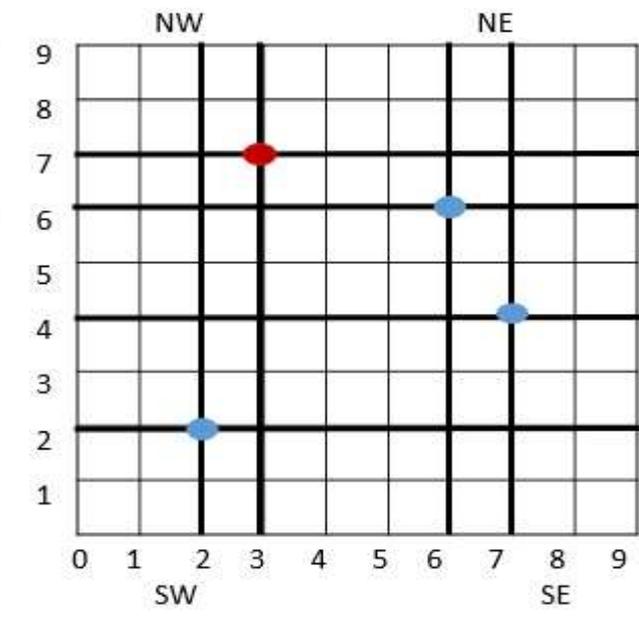


Figure 3e

For example list of business like a restaurant with its location (latitude and longitude) and set of keywords (south Indian meals, north Indian meals). On the other side, admin can upload these n number of details maintained in the Excel data sheet as shown in Figure 4.

B. User Module

In this module, here user will enter the query keyword and also set the current location as illustrated in figure 5a, 5b. By using quadtree search algorithm proposed in section 3.2.1 user finds the

results of TOP-K spatial keywords of the online business directory.

C. Experimental settings

The overall project was implemented in Java. Tom cat 1.6 were used as web server, and requirements are Windows 7/UBUNTU operating system, IDE Eclipse tool, MYSQL database, 40GB Hard Disk, 1GB Ram.

| ID | Places | Terms | Latitude | Longitude |
|----|---------------------------------|--------------|----------|------------|
| 1 | platter | 13-5-10-7- | | 12.927104 |
| 2 | jcubez | 10-11- | | 77.5627765 |
| 3 | 1947 | 5-6-12-13- | | 12.9176923 |
| 4 | big bazar | 14 | | 77.5606471 |
| 5 | Unilet Store | 9 | | 12.9264532 |
| 6 | Mega mart | 15 | | 77.5488348 |
| 7 | ezroma | 9 | | 12.9251568 |
| 8 | Hana Fine Dine restaurant | 12-6- | | 12.9229041 |
| 9 | Tandoori Knight | 12-13-7- | | 77.557939 |
| 10 | San Churro Café | 1-2- | | 12.9242273 |
| 11 | Roing Feast | 12-13-5-6-7- | | 77.5407721 |
| 12 | Ibaco | 17 | | 12.9220687 |
| 13 | PolarBear | 17 | | 77.5620607 |
| 14 | Sheesha Shack | 2 | | 12.9258004 |
| 15 | New Imperial Restauran resturan | 12-5- | | 12.9267938 |
| 16 | The Krishna Grand | 12-5-6- | | 12.9267938 |
| 17 | Sea Spice by 7 Star | 18- | | 77.5612472 |
| 18 | Mantex creations | 3 | | 12.9289694 |
| | | | | 77.5718488 |
| | | | | 12.9214204 |
| | | | | 77.5688222 |
| | | | | 12.91901 |
| | | | | 77.5666303 |
| | | | | 12.9294708 |
| | | | | 77.5654922 |

Figure 4 ExcelData Sheet



Figure 5a Query search



Figure 5b Location map

V. CONCLUSION

In the overall concept of getting a TOP-K result, which is having efficient and accurate Spatial and textual data information. Our proposed framework is having a proficient outcome at whatever point the query is being dynamically asked. Point regional quad tree works for region based spatial keyword search and have a good indexing structure, it increases the performance and also reduces the query processing time.

REFERENCES

[1] I.D. Felipe, V. Hristidis, and N. Rishe, “Keyword search on spatial databases,” in *ICDE*, 2009.
 [2] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen, “Joint top k spatial keyword query processing,” *TKDE*, 2011.

[3] S. Ding, J. Attenberg, R. A. Baeza-Yates, and T. Suel, “Query processing for web search engines,” in *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, 2011, pp. 137–146. [Online]. Available: <http://doi.acm.org/10.1145/1935826.1935858>.
 [4] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen, “Joint top k spatial keyword query processing,” *TKDE*, 2011.
 [5] G. R. Hjaltason and H. Samet, “Distance browsing in spatial databases.” *TODS*, vol. 24, no. 2, pp. 265–318, 1999.
 [6] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel, “Text vs. space: efficient geo-search query processing,” in *CIKM*, 2011.
 [7] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvoag, “Efficient processing of top-k spatial keyword queries,” in *SSTD*, 2011.

- [8] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatial-keyword (sk) queries in geographic information retrieval (gir) systems," in SSDBM, 2007.
- [9] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," PVLDB, vol. 2, no. 1, 2009.
- [10] D. Wu, G. Cong, and C. S. Jensen, "A framework for efficient spatial web object retrieval," VLDB J., 2012.
- [11] A. Cary, O. Wolfson, and N. Rishé, "Efficient and scalable method for processing top-k spatial boolean queries," in SSDBM, 2010, pp. 87–95.
- [12] Z. Li, K. C. K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang, "Ir-tree: An efficient index for geographic document search," IEEE Trans. Knowl. Data Eng., vol. 23, no. 4, pp. 585–599, 2011.
- [13] D. Zhang, K.-L. Tan, and A. K. H. Tung, "Scalable top-k spatial keyword search," in EDBT, 2013, pp. 359–370.
- [14] G. Li, J. Feng, and J. Xu, "Desks: Direction-aware spatial keyword search," in ICDE, 2012.
- [15] S. B. Roy and K. Chakrabarti, "Location-aware type ahead search on spatial databases: semantics and efficiency," in SIGMOD Conference, 2011.
- [16] J. B. Rocha-Junior and K. Nørsvåg, "Top-k spatial keyword queries on road networks," in EDBT, 2012.