IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 12, December 2020

DOI 10.17148/IJARCCE.2020.91205

A Survey on Location-Based Spatial Proximity Query Processing on Real Road Networks

Debajyoti Ghosh

Assistant Professor, Computer Science Department, BML Munjal University, Gurgaon, India

Abstract: Smartphones are revolutionary mobile devices, can be found with almost every person of all age groups, become an integral part of everyday life. Use of the smart mobile devices increases by leaps and bound with the growth of location-based services (LBS). With the ease of availability of internet and GPS enabled mobile devices with high accuracy and precision, location-based queries become very popular among mobile users on the road network. In this paper we have intensively surveyed cutting-edge solutions approaches for types of spatial query, mainly, location-based proximity queries (nearest neighbour queries, k-nearest neighbour queries, and range queries) on the real road network.

Keywords: Proximity query, Spatial road network, Location-based services, Range query, Nearest Neighbour query, k Nearest Neighbour query.

I. INTRODUCTION

We are living in a new era of mobile information. In our everyday life, extensive use of smartphone-based mobile applications to acquire information about the desired object of interest or points of interest (POI) is indispensable. Depending on the location and context of the mobile device and mobile user, LBS provides requested information of the users. Rapid popularity and growth in LBS become mainly due to the advancements in the field of mobile information systems, mobile communication, portable mobile devices (smartphones, laptops, tablets etc.), wireless internet connection and geo-positioning. LBS provide value-added information and entertainment services using real-time geo-data from a mobile device or smartphone through the mobile network. Location means spatial coordinates <latitude, longitude>, i.e. geographical position on the earth. Mobile users often seeking information on shortest travel routes, points-of-interest, turn-by-turn-navigation, traffic conditions, weather information, etc. using google maps. Google Maps is arguably the most popular and widely used location-based app used for a route planner. Searching is one of the fundamental problems of computer science. The conventional searching problem can be extended to search for desired POIs in the road network. The majority of smartphone users use the Google Maps app to search for intended POI around their current location and get turn by turn directions of destination places quickly. Social networking sites like Facebook recently released their location-based services, geolocation and mobile marketing feature, known as Facebook Places, to check into certain nearby locations via their mobile phones. Location-based feature enables Facebook users to find friends who are in the proximity range of their current location. WhatsApp's location-sharing feature allows to locate/track a friend's location. Ola, uber online cab booking apps offer to book cabs nearby your location at the lowest fares.

LBS [22, 23] provide spatial information (value-added information and entertainment services) anywhere, anytime depending on the current geo location of mobile users having GPS supported handy devices via the mobile network. A set of spatial queries that retrieve location-specific information depending on the mobile user's current location, i.e. spatial coordinate (latitude, longitude) of the inquirer/requester is referred to as Location Based (or dependent) Spatial Queries (LBSQ) [25]. Perhaps the most challenging task is to implement LBSQ on portable devices having limited memory capacity. In LBS, searching for spatial objects of interest or points of interest (POIs) in respect of the user's current query location q is a very common task. A comprehensive pointer of research direction in LBS, a wonderful survey paper has written by Ilarri et al. [24]. Unlike traditional queries (nonspatial/non location-related) like shortest path queries, SQL queries on Database, etc. where query answers neither determined by location nor depend on extraction of location information, a spatial query returns query result with respect to current spatial/geographic location of the query issuer and the locations of spatial POI.

II. PROXIMITY QUERIES

Proximity mean how close one object is to another object. A proximity-based spatial query computes the results based on the proximity (closeness) between the objects. One can say that Gurugram is in proximity to Delhi. Among the location-based spatial proximity queries, the nearest neighbour queries, k-nearest neighbour queries

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 12, December 2020

DOI 10.17148/IJARCCE.2020.91205

and range queries are very common and well-studied computational problem, have been actively re-searched in various LBS applications. Depending on the location of the two objects, i.e. location of the user requests the query and the location of the POI, responses of location-based spatial proximity queries vary. Focus on research in spatial proximity queries is to efficient pruning the search space to prevent the exhaustive search on road networks. Examples of spatial proximity queries include retrieve closest medicine store near to my current location, retrieve three closest medicine stores near to my current location, retrieve all medicine store within 2km to my current location, book the taxicab closest to a passenger's current query location, etc.

RANGE QUERIES

Given a query location q on the road network and a pre-specified search radius distance range r > 0 as input, range search or range query problem [26-29] is to pre-process a set P of objects of interest (POI) into a data structure so that quickly find all POI's (if any) on road networks within a maximum distance of r from the query location q, i.e. we have to find all p's for which d(q, p) <= r. For example, find hospitals within 2 km from my current location. In this example, location information is implicitly specified.

PAST WORK

Theodoridis and Papadias [1] provided the topological relations meet and inside, direction relations east and northeast and some distance relations, their retrieval schemes and implementations using spatial data structures like R-tree and B-tree indexing schemes to support range queries. Analytical models are proposed to predict the expected cost of retrieval of spatial relations and the analytical formulas provided for the performance estimation of R-trees and B-trees indexing mechanisms to retrieve spatial relations. The spatial relations supported by the spatial query optimizers of database systems can use proposed formulas as guidelines according to estimate the cost of spatial queries. An et al. [2] introduced a novel set of five analysis techniques on spatial data to estimate the performance of range query. In query optimization analysis is crucial. Two of these schemes can be used for both rectangular and point data sets and the rest three can analyze point data sets. A density file data structure (histograms) is used as an supporting structure. The density file contains adequate information. During the creation of index structure for a given query, a density file can be constructed quickly. During a given query, the density file is looked up quickly, and acquire the required information. They have shown that Cumulative Density (CD) method gives very precise results compared to previously proposed analysis techniques. They have conducted extensive experiments with several rectangles and point data sets to evaluate the performance of different schemes. Bae et al. [3] introduced the technique to process range queries on the spatial web data using k-NN searches. Their aim was to find all POIs within a given rectangular region. Query region is divided into subregions to minimize the number of k-NN searches. They have proposed the Quad Drill Down (QDD) algorithm, a recursive approach, which uses the characteristics of the quad-tree and a greedy and incremental approach called Dynamic Constrained Delaunay Triangulation (DCDT) algorithm to solve the range query problem using the Constrained Delaunay Triangulation (CDT). They have conducted empirical experiments and compared the efficiency of QDD and DCDT algorithms using both synthetic data sets and real GIS data sets. Stougiannis et al. [4] have developed a new tool named ZOOM to visualize neuroscience data and to execute spatial range queries efficiently. ZOOM is an interactive graphical user interface based on the execution strategy of (FLAT, a novel spatial index based on B+ Tree, use to process spatial range queries) that ensure efficient query execution. In ZOOM tool they have executed range queries for a small set of real neuroscience data, the query result is visualized. Sun et al. [5] introduced a new spatial index for air, called Network Partition Index (NPI), efficiently process spatial kNN queries and spatial range queries in road networks via wireless broadcast environment. 'Air' here mainly refers to the fact that indexes broadcast via a wireless channel(s) are not physically stored/available within any storage. They are available only when they are broadcast. Their idea was to partition the road network into smaller regions and construct the NPI index to carry certain pre-computation data. To demonstrate the efficiency of NPI scheme, extensive experiments are conducted and performance is evaluated to support kNN queries and range queries in real road network data sets.

NEAREST NEIGHBOUR QUERIES

Given a set P of spatial POI on the road network, in Nearest Neighbour (NN) queries (a. k. a. closest point problem) [30-37], the task is to organize the set P into a data structure such that consequent queries asking for the closest spatial object (POI) to a given query location q (on same road network) can be found quickly. For example, Amazon or Flipkart often must find the closest warehouse locations to a customer location/query point q quickly. Finding nearby POIs (facilities, e.g., filling stations, restaurants, ATMs, supermarkets/malls, etc) are among the most popular queries issued on google maps in a road network. Consider the query "Retrieve closest filling station closest to my current location", this query retrieves the closest filling station based on the location of the query issuer. There is a similar

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 12, December 2020

DOI 10.17148/IJARCCE.2020.91205

application of locating the closest post office to a query user, Donald Knuth called it the post office search problem [26].

PAST WORK

There are two very common existing approaches of finding nearest neighbour is: earlier one is an incremental approach and modern one is an index-based pre-computing approach. In the case where the query POI is far from query location, the incremental approach takes a long time to find closest POI, is impractical for large road network as starting from the query location, it expands the search area slowly but is fast in the average case. The pre-computation based approach uses spatial indices to prune large POIs. Yianilos [6] presented techniques for finding nearest neighbour from databases under application-specific metrics. The author has introduced vp-tree (vantage point tree, in Euclidian space, independent of the coordinate system) together with associated algorithms for construction and search processes. Presented a pruning based recursive branch-and-bound tree search algorithm which describes search in a vp-tree. Experiments on nearest neighbour search queries are performed on six different database images and a pair of database images. Roussopoulos and Vincent [7] presented an efficient pruning/branch-and-bound based search algorithm to locate the closest POI to a given query location for the R-trees, and then generalize it to finding the k closest POIs. They have implemented their k-NN algorithm, presented the results of several experiments obtained on both real-world and synthetic data sets to explain the performance and scalability of their method. Papadias et al. [8] presented the comprehensive method in spatial network databases (SNDB) to process query faster. Proposed an adaptable architecture for SNDB that integrates road network and Euclidean distance information. They have developed a network expansion and a Euclidean restriction framework based on this architecture that efficiently prunes the search space by taking benefit of connectivity and location. To improve the performance of spatial query execution we need appropriate indexing scheme. They have indexed POI dataset using an R-tree (Multidimensional extension of B-tree for indexing, combined most of the nice features of both B-trees and quadtrees). Proposed some new algorithms for the basic spatial queries (nearest neighbours search, finding closest pairs, and range search). Introduced Incremental Euclidean Restriction (IER) and Incremental Network Expansion (INE) methods, which also support kNN queries in SNDB. Starting from a query location q, IER computes NN using Euclidean distance, afterward compute similar network distances between nearest neighbour and query location. To find kNN, then incrementally finds the next closest neighbours according to non-decreasing order of Euclidean distance and continue the process of locating one neighbour whose Euclidean distance is greater than the recent k closest neighbour in terms of network distance. IER utilize spatial pruning techniques, it prunes unpromising expansions of nodes by taking the Euclidean distance as a bound. INE algorithm simply extends the Dijkstra's algorithm by expanding search area by including adjacent nodes incrementally from the query location until NN is found. INE overcome limitation of IER. In terms of average performance INE has been found better than IER. IER and INE both are referred to 'blind' algorithms as they do not prune unnecessary POIs efficiently. Similarly, proposed a Range Network Expansion (RNE) and a Range Euclidean Restriction (RER) algorithm to process range queries in SNDB. Authors implemented their algorithms, experimentally compared all algorithms with regard to I/O cost and CPU-time on both synthetic data sets and publicly available real-world spatial data sets. Sankaranarayanan et al. [9] introduced a new framework (SILC) for a spatial network to represent the path and distance information between every pair of vertices compactly. SILC pre-calculates the shortest distance and paths between all vertex pairs on a spatial road network using quadtree. SILC is appropriate for both secondary and primary memory datasets. A variation of the Best First Search (BFS) technique is also proposed for a spatial network to locate the closest neighbours to a query location. However, in a small region if there are large numbers of POIs clustered, SILC is inefficient. Moreover, a large road network consumes more storage space and pre-processing overhead remains high, so, SILC is impractical. They also presented an experimental evaluation of their technique by publicly available spatial road network datasets of sufficiently large size. Huang et al. [10] proposed an interesting idea for computing k closest neighbour efficiently on spatial road networks, named the Islands method. An island is a sub-network with edges and nodes situated within a certain radius distance away from a POI. The Islands approach consists of precomputation and re-computation: precomputation is carried out inside islands, and re-computation is carried out in between islands. Proposed Islands method does not precompute the actual shortest paths but finds the kNN along with their distances. The Islands approach uses a simple priority queue data structure and a natural search algorithm. Experimental result of the comparison of query performance of Islands approaches to existing techniques, mainly Network Voronoi Diagram based algorithm and Incremental Network Expansion algorithm shows that the Islands approach perform better than the existing techniques. Hu et al. [11] introduced a new index-based approach for network reduction on road networks for kNN search. Reduce the network into a set of tree-based interconnected structures, termed Shortest Path with Inequality Edges (SPIE). For each SPIE, a faster nd (nearest descendants) index is devised. kNN search algorithms on an SPIE avoid most of the costly network expansions by following a predetermined tree path. Dickerson et al. [12] proved several new properties of Voronoi diagram on multi-type nearest neighbour round-trip distance function defined in terms of triangle inequality for an undirected geographical network. Used those lemmas to develop asymptotically better algorithms than previously known algorithms (graph-theoretic Voronoi diagrams and geometric Voronoi diagrams). The round-trip distance function is

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 12, December 2020

DOI 10.17148/IJARCCE.2020.91205

defined as the minimum length tour distance visiting all three vertices from a single vertex to a pair of vertices and ending at the starting vertex. Provided algorithms (a brute force and an improved brute force) to find the round-trip distance function Voronoi diagram using the properties proved in their paper to prune the search space of the brute force algorithm. Author further revised their prune and search-based algorithm and gave a Dynamic Variation of their algorithm to minimize the shortest round-trip distance by further reduce the search space. Provided empirical analysis for Multi-Type Round Trip Nearest Neighbour (MTRTNN) search and by doubling densities of various types of POIs on road networks (Hospitals, ATMs, Fuel stations, Fire stations, Coffee houses, Restaurants or Grocery stores). They have implemented their algorithms and ran experiments on real road networks. Lee et al. [13] presented a framework, christened ROAD – an acronym of two modules, namely, **R**oute **O**verlay and **A**ssociation **D**irectory, is very effective pruning technique, and flexible to evaluate various kinds of LDSQs (such as nearest neighbour query and range query) on road networks. As an extension of the idea of Dijkstra's network expansion-based algorithm, a huge road network is arranged as a hierarchy of locally connected sub-networks (named Rnets). A road network recursively partitions into sub-networks and organizes them in a hierarchical manner, then pre-calculates the shortest distance paths within a sub-network (termed shortcuts). A sub-network which do not contain a POI can be skipped. However, on large networks, the performance of ROAD is poor. As ROAD partially captures global distance information, is called a 'half-blind' algorithm. Presented several useful properties of the Rnet hierarchy and construction of Rnet hierarchy. They have conducted comprehensive empirical experiments to evaluate kNN search and range search on real world road network to evaluate their proposed ROAD framework.

k NEAREST NEIGHBOUR QUERY

Given a set of POI on the road network, in k Nearest Neighbour (kNN) query [31, 38-40] the job is to find the specified k closest POI by network distance from a given query location on the road network. Consider the query "Retrieve three closest fuel stations near to my current location", according to the current location of query issuer, this query retrieves three nearest fuel stations.

PAST WORK

Kolahdouzan and Shahabi [14] first introduced the Voronoi-based Network Nearest Neighbour (VN^3) method to efficiently compute kNN queries in spatial network databases over the POIs. The large network is partitioned into smaller and manageable regions, called Voronoi polygons, and then both within and across the regions pre-compute some network distances. VN^3 is a simple approach and easy to implement using simple data structures like lookup tables and R-tree. Conducted extensive experimental evaluation with some real-world network data sets and comparison result shown that VN^3 perform better than INE. Safar [15] introduced a pre-processing based approach, named Progressive Incremental Network Expansion (PINE) which pre-compute some network distances and the network Voronoi polygons (NVP) to efficiently process kNN queries in spatial network databases. For large networks with a small number of POIs PINE approach suites is more. Empirical experiments have conducted with real-world network data sets, comparison shown that the PINE outperforms INE and VN^3 approaches, has better query response time. Almeida and Guting [16] presented index structures-based storage schema to support efficient processing of kNN queries in road network by slightly modifying Dijkstra's algorithm. By experimental evaluation over the real road network data set, they have shown that their proposed method is better than the INE and VN^3 approach. They have compared their algorithm in respect of the number of disk accesses and I/O cost for each query. Sharifzadeh and Shahabi [17] introduced an interesting index structure, termed VoR-tree (an R-tree is used to index the Voronoi diagram, i.e. an R-tree of point data augmented with the point's Voronoi diagram). VoR-tree concept has an advantage from the best of both worlds (the hierarchical structure of R-trees and the neighbourhood exploration capability of Voronoi diagrams). They proposed VoR tree-based efficient algorithm for kNN queries. They have evaluated the performance of their algorithm through extensive experiments using three real-world datasets, the result shows that VoR trees enable I/O-efficient processing of kNN. Bao et al. [18] proposed an efficient algorithm to process k-Range Nearest Neighbour (kRNN) query in road networks. The redundant searching overhead in the existing solution has been eliminated. They have used real road network data sets in their simulation-based experiment. According to experiment, their approach always outperforms the existing solution in respect of query response time. Nutanong and Samet [19] proposed a novel incremental algorithm to process kNN query, termed the Single Wavefront Heuristic search (SWH). The focus of their algorithm design is to process spatial queries efficiently with regard to the in-memory processing costs as well as access cost. The solution incrementally retrieves Euclidean NNs in reference to the query location q by utilizing the best-first NN algorithm. Their algorithm is optimal in terms of the graph traversal cost (i.e. the number of visited nodes). SWH search has compared with the LBC (Lower Bound Constraint) algorithm and INE. They have evaluated performance using a real road network data sets and as per the experimental results, their proposed algorithm outperforms existing LBC and INE. Abeywickrama et al. [20] investigated five state-of-the-art methods (INE, IER, Distance Browsing, ROAD and G-tree) of the kNN query on road networks. INE and ROAD are expansion-based methods, IER, Distance Browsing and G-tree are based on best-first search heuristic. They have tried for improvements if applicable to these methods. The widespread use of map-based services in smartphones, in-memory query processing

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 12, December 2020

DOI 10.17148/IJARCCE.2020.91205

becomes challenging. They have suggested to pay careful attention while designing and implementing algorithms for main memory. Presented efficient implementations in main memory and evaluated these five of the most notable methods on real-world POIs. Abeywickrama and Cheema [21] introduced Landmarks-based approach to compute tighter lower bounds for kNN search. Landmarks is a lower bounding technique. To improve kNN query performance, Landmark Lower Bounds (LLBs) can be used. They have presented two Pre-Processing based techniques to efficiently retrieve the kNN query on road networks using landmarks. One approach is to use a combination of Network Voronoi Diagrams (NVDs) and landmarks and another is to use ordered object lists for every landmark. Proposed the comparable kNN procedure to IER using LLBs and Incremental Lower Bound Restriction (ILBR).

III. CONCLUSION

In this survey, we have intensively investigated state-of-the-art approaches to solve most prominent spatial proximity queries on real road networks. Earlier studies of such queries were impractical as they considered Euclidean distance between the user's query location and location of the query POIs on the road network. Actual road network distance is the realistic measure of the distance between any two locations on the road network. Road network distance is the shortest path distance between two locations on the road network. So, past algorithm's output was inaccurate. We have also studied empirical experiments conducted on the studied methods.

IV. FUTURE WORK

Theodoridis and Papadias [1] identified that their analytical models can be applied on specific spatial relations such as nearest-neighbour, can be applied for other spatial indexing methods like K-D-B trees (k-dimensional B-tree), Grid files and their efficiency can be evaluated with respect to the specific spatial relations. An et al. [2] suggested that their proposed schemes can be extended for more complicated spatial join queries, also can be merged with a spatial database query optimizer. Bae et al. [3] suggested that their work can be extended the work to design algorithms when the value of k is not fixed. Roussopoulos and Vincent [7] suggested further work on Dynamic NN and kNN queries. Papadias et al. [8] suggested that can further work on continuous NN query, NN query for moving user, or Dynamic NN queries. Sankaranarayanan et al. [9] suggested that one can reduce the storage requirements of their proposed framework for efficient path retrievals. SILC may apply to moving objects. Huang et al. [10] suggested that the Island approach still may take into account live traffic maps and road alerts, such as roadblocks, traffic conditions, bad weather, fog, or flood and can be extended for dynamic road networks. Hu et al. [11] suggested that one can further improve the performance of their proposed approach by redesigning the structures of SPIE. Dickerson et al. [12] suggested that the time complexity can be further reduced by efficient pruning. Lee et al. [13] suggested that one can devise algorithms to support location-based spatial queries other than range search and nearest neighbour search. Kolahdouzan and Shahabi [14] suggested that their approach can be extended for continuous and group kNN. Safar [15] suggested that one can address constraint kNN and group kNN by extending PINE. The categorized kNN query can be defined as the POI data set can be partitioned into groups and the queries can be executed over any of these groups or even over a set of these group. Almeida and Guting [16] had planned to extend their storage schema to implement categorized k-nearest neighbour query. Nutanong and Samet [19] suggested using landmark method to improve the efficiency of their search process. Abeywickrama et al. [20] suggested improving kNN search heuristics by more efficient G-tree. Though their study is limited to in-memory query processing, can be extended on streaming and external memory index structure. Apart from the above mentioned proximity queries, one can consider other proximity queries like range NN query, range kNN query, incremental NN query, incremental kNN query, all NN query, all kNN query, range-aggregate proximity queries, in-route NN query, in-route kNN query, group NN query, etc.

REFERENCES

[1]. Yannis Theodoridis and Dimitris Papadias, "Range Queries Involving Spatial Relations: A Performance Analysis", Proceedings of the 2nd International Conference on Spatial Information Theory, part of LNCS book series, vol. 988, pp. 537-551, 1995.

[2]. Ning An, Ji Jin, and Anand Sivasubramaniam, "Toward an Accurate Analysis of Range Queries on Spatial Data", IEEE Transactions on Knowledge and Data Engineering, vol. 15(2), pp. 305-323, 2003.

[3]. Wan D. Bae, Shayma Alkobaisi, Seon Ho Kim, Sada Narayanappa, and Cyrus Shahabi, "Supporting Range Queries on Web Data Using k-Nearest Neighbour Search", GeoInformatica, vol. 13(4), pp.483-514, 2009.

[4]. Alexandros Stougiannis, Farhan Tauheed, Thomas Heinis, and Anastasia Ailamaki, "Accelerating Spatial Range Queries", Proceedings of the 16th International Conference on Extending Database Technology, pp. 713–716, 2013.

[5]. Weiwei SUN, Chunan CHEN, Baihua ZHENG, Chong CHEN, and Peng LIU, "An Air Index for Proximity Query Processing in Road Networks", IEEE Transactions on Knowledge and Data Engineering, vol. 27(2), pp. 382-395, 2015.

[6]. Peter N. Yianilos, "Data Structures and Algorithms for Nearest Neighbour Search in general metric spaces", Proceedings of the 4th annual ACM-SIAM symposium on Discrete algorithms, pp. 311–321, 1993.

[7]. Nick Roussopoulos and Kelley Vincent, "Nearest Neighbour Queries", Proceedings of the ACM SIGMOD international conference on Management of data, pp. 71–79, 1995.

[8]. Dimitris Papadias, Jun Zhang, Nikos Mamoulis and Yufei Tao, "Query Processing in Spatial Network Databases", Proceedings of VLDB Conference, pp. 802-813, 2003.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 12, December 2020
DOI 10.17148/IJARCCE.2020.91205

[9]. Jagan Sankaranarayanan, Houman Alborzi and Hanan Samet, “Efficient Query Processing on Spatial Networks”, Proceedings of the 13th annual ACM international workshop on Geographic information systems, pp. 200–209, 2005.
[10]. Xuegang Huang, Christian Jensen, and Simonas Saltenis, “The islands approach to Nearest Neighbour Queries in Spatial networks”, Proceeding of 9th International Symposium on Spatial and Temporal Databases, pp. 73–90, 2006.
[11]. Haibo Hu, Dik Lun Lee, and Jianliang Xu, “Fast Nearest Neighbour Search on Road Networks”, International Conference on Extending Database Technology, LNCS, vol. 3896, pp. 186-203, 2006.
[12]. Matthew Dickerson, Michael Goodrich, and Thomas Dickerson, “Round-Trip Voronoi Diagrams and Doubling Density in Geographic Networks”, Transactions on Computational Science, LNCS, vol. 6970, pp. 211-238, 2011.
[13]. Ken C. K. Lee, Wang-chien Lee, Baihua ZHENG, and Yuan Tian, “ROAD: A New Spatial Object Search Framework for Road Networks”, IEEE Transactions on Knowledge and Data Engineering, vol. 24(99), pp. 547-560, 2012.
[14]. Mohammad Kolahdouzan and Cyrus Shahabi, “Voronoi-Based K Nearest Neighbour Search for Spatial Network Databases”, Proceedings of the Thirtieth international conference on Very large data bases, vol. 30, pp. 840–851, 2004.
[15]. Maytham Safar, “K nearest neighbour search in navigation systems”, Mobile Information Systems, vol. 1(3), pp. 207-224, 2005.
[16]. Victor Teixeira de Almeida and Ralf Hartmut G¨uting, “Using Dijkstra’s Algorithm to Incrementally Find the K Nearest Neighbours in Spatial Network Databases”, Proceedings of the ACM symposium on Applied computing, pp. 58–62, 2006.
[17]. Mehdi Sharifzadeh, Cyrus Shahabi, “VoR-tree: R-trees with Voronoi diagrams for efficient processing of spatial nearest neighbour queries”, Proceedings of the VLDB Endowment, vol. 3(1), pp. 1231–1242, 2010.
[18]. Jie Bao, Chi-Yin Chow, Mohamed F. Mokbel, and Wei-Shinn Ku, “Efficient Evaluation of k-Range Nearest Neighbour Queries in Road Networks”, 11th International Conference on Mobile Data Management, pp. 115-124, 2010.
[19]. Sarana Nutanong and Hanan Samet, “Memory-Efficient Algorithms for Spatial Network Queries”, In Proceedings of the 29th IEEE International Conference on Data Engineering, pp. 649-660, 2013.
[20]. Tenindra Abeywickrama, Muhammad Aamir Cheema, and David Taniar, “k Nearest Neighbours on Road Networks: A Journey in Experimentation and In Memory Implementation”, Proceedings of the VLDB Endowment, vol. 9(6), pp. 492-503, 2016.
[21]. Tenindra Abeywickrama and Muhammad Aamir Cheema, “Efficient Landmark-Based Candidate Generation for kNN Queries on Road Networks”, International Conference on Database Systems for Advanced Applications, LNCS, vol. 10178, pp. 425-440, 2017.
[22]. Jochen Schiller and Agnès Voisard, Location Based Services, 1st ed., Morgan Kaufmann, Massachusetts, United States, 2004.
[23]. Richard Ferraro and Murat Aktihanoglu, Location Aware Applications, Manning Publications, New York, United States, 2011.
[24]. Sergio Ilarri, Eduardo Mena, and Arantza Illarramendi, “Location-dependent query processing : where we are and where we are heading”, ACM Computing Surveys, vol. 42(3), pp. 1–67, 2010.
[25]. Debajyoti Ghosh, Soharab Hossain Shaikh, and Prosenjit Gupta, “A survey on Location Based Spatial Queries on Road Networks”, Proceeding on 5th International Conference on Intelligent Computing and Applications, Advances in Intelligent Systems and Computing, Springer, 2019.
[26]. Jon Louis Bentley and Hermann Maurer, "Efficient Worst-Case Data Structures for Range Searching", Acta Informatica, vol. 13(2), pp. 155–168, 1980.
[27]. Jiří Matoušek, "Geometric range searching", ACM Computing Surveys, vol. 26(4), pp. 421–461, 1994.
[28]. Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, Computational Geometry : Algorithms and Applications, 1st ed., Springer, Berlin, Germany, 1997.
[29]. Pankaj Kumar Agarwal, Jeff Erickson, "Geometric Range Searching and Its Relatives", Advances in Discrete and Computational Geometry, pp. 1–56, 1999.
[30]. Donald Knuth, The Art of Computer Programming, Volume 3, 1st ed., Addison Wesley, Boston, United States, 1969.
[31]. Micheal Ian Shamos and Dan Hoey, “Closest-point problems”, 16th Annual Symposium on the Foundations of Computer Science, pp. 151-162, 1975.
[32]. G Yuval, “Finding nearest neighbours”, Information Processing Letters, vol. 5, pp. 63-65, 1976.
[33]. Jon Louis Bentley, Bruce Weide, and Andrew Yuo, “Optimal expected time algorithms for closest point problems”, ACM Transaction of Mathematics and software, vol. 6, pp. 563-580, 1979.
[34]. Iraj Kalantari and Gerard McDonald, “A Data Structure and an Algorithm for the Nearest Point”, IEEE Transactions on Software Engineering, vol. 9(5), pp. 631 - 634, 1983.
[35]. Sunil Arya, Nearest Neighbor Searching and Applications, Ph.D. thesis, University of Maryland at College Park, 1995.
[36]. Alexandr Andoni, Nearest Neighbor Search - the Old, the New, and the Impossible, Ph.D. dissertation, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2009.
[37]. Debajyoti Ghosh and Prosenjit Gupta, "Round-Trip Nearest Neighbors on Road Networks for Location Based Services", Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, pp. 310-313, 2016.
[38]. Keinosuke Fukunage and Patrenahalli Narendra, "A Branch and Bound Algorithm for Computing k-Nearest Neighbors", IEEE Transactions on Computers, vol. 24(7), pp. 750-753, 1975.
[39]. Der-Tsai Lee, On finding k nearest neighbors in the plane, M.S. Thesis, University of Illinois, Urbana, 1976.
[40]. Der-Tsai Lee, "On k-Nearest Neighbor Voronoi Diagrams in the Plane", IEEE Transactions on Computers, vol. 31(6), pp. 478 - 487, 1982.