Spat. Inf. Res.
https://doi.org/10.1007/s41324-019-00256-2

Improving the performance of location based spatial textual query processing using distributed strip index

M. Priya[1] · R. Kalpana[2]

Received: 14 October 2018 / Revised: 22 February 2019 / Accepted: 26 February 2019

Abstract Location Based Services are information retrieval services that offer accurate information required by the end user. These services are the query based services accessed mainly through mobile devices and have number of uses in social networking for providing entertainment, business and healthcare information. In health care system, if a person wants to get immediate medical help at any place, he needs to access a medical database with the help of location-based query. Sometimes, location-based query can associate with the text information, such as the user wants to find the nearest hospital with the facility of pharmacy or ambulance. This type of query has to resolve both location and textual information. This paper proposes a new distributed index structure to resolve location-based query, and introduces a new probabilistic mechanism to correct the typographical errors when retrieving the documents. The experimental results show that the distributed strip index structure produces better performance than the existing distributed R tree structure.

Keywords Data space · Location-based query · Medical databases · Box space · Divide and conquer · Maximum-a-likelihood

✉ M. Priya
shyamnithy@gmail.com

R. Kalpana
rkalpana@pec.edu

[1] Bharathiyar College of Engineering and Technology, Karaikal, India

[2] Pondicherry Engineering College, Puducherry, India

1 Introduction

Nowadays, the mobile devices have become the most popular way to get access to the internet. According to the survey of internet statistics, nearly 60% of the internet access made through the mobile devices such as smart phones and tablets. Among them, 90% of the search volume is about finding the location-based information. These devices are using location-based services to retrieve the information. Location-based services are the services which use smart phone's GPS technology to retrieve the location information. In the geographical system, each element called spatial object represented by a pair of attributes called latitude and longitude. The mobile users can issue various location-dependent queries to access the spatial object.

A range query is the basic type of query used to resolve the location-based access. Sometimes the spatial query associated with the text information, to retrieve the spatial objects that exactly satisfy the needs of users, called spatial textual query. Device information can be used for the different purpose such as finding the nearest restaurants; performing the proximity based marketing and retrieving the travel information and getting medical help. Mobile devices play an important role in getting the medical help timely due to its fastest communication. For example, a user can issue the query such as "List all the hospitals within 2 km with the facility of the ambulance" to get immediate medical help. This type of query is called as the spatial textual query.

Index structures are constructed to speed up the searching process of data retrieval operations. The spatial textual query requires index structures that support for searching both the spatial information and the text information. The factors such as an increase in the size of the

Published online: 09 March 2019

datasets and volume of the record get accessed can affect the performance of the query processed. The distributed based solution is introduced to face the challenges in scaling out the volume of data.

Sometimes the data to be analyzed for query processing are not in the appropriate format or they may contain errors that reduce the performance of query processing. Moreover, there is a possibility of classifying the misspellings and analyzing the error types automatically. The expected incidence of misspellings in different databases is about 0.2%. Nearly 90% of the errors have only a single mistake such as substitution or transposition. The spatial textual query has to correct these typographical errors.

This paper proposes a new distributed index structure to resolve Location Based Spatial Textual Query (LBSTQ). LBSTQ can be resolved by resolving the spatial and the textual predicates. The distributed based R tree index with vantage point transformation technique is used to resolve the spatial predicates. An Aho_Corasick tree based index structure is used to resolve the textual predicates. It also discusses a new probabilistic learning mechanism to correct the typographical errors when retrieving the documents. It also discusses the following two improvements in processing the spatial textual query.

- A probabilistic learning approach proposed to handle the context sensitive error.
- A new distributed index structure introduced to resolve the spatial textual query.

The Sect. 2 of the paper discusses the literature survey, Sect. 3 proposes the model for distributed index structure, Sect. 4 conducts an experimental evaluation and Sect. 5 concludes.

2 Related work

The query processing for moving objects is sectioned into two types, called snapshot queries and continuous queries. The snapshot queries are [1] the one in which the query return results based on the current location and the time. The continuous query [2] is the one that uses the incremental algorithm in which the query results are based on location of the moving object. These queries perform a range search or nearest neighbor search to retrieve the object.

The different index structure supports for resolving the spatial query. R tree is considered to be one of the best indexing structures for deliberating the spatial query, but the performance of the R tree index structure decreases with the increasing dimensionality. An efficient index structure such as an X-tree(eXtended node tree), VA-tree(Virginia Attribute tree) provides better performance than R tree to derive high dimensional data. The data objects of the multidimensional index structure cannot be processed efficiently by R index, as they have less correlation among them. Spatial index structures are not suitable for hierarchical data partitioning due to the inequality in few partitions.

Naresh et al. [3] discussed an air indexing method for the spatial query which uses the grid to store and process the data object and devise a new algorithm to handle the snapshot query in a wireless environment. The main objective of this index is to reduce the IO cost and communication cost, but unfortunately it has left certain factors regarding result updating. Snapshot spatial-temporal incremental indexing algorithm [4] provides an improvement in both spatial and temporal query processing.

The recent improvement in databases in terms of database size insists a method of handling the query through the distributed and parallel processing. VegaGiStore [5] indexing which is processed on the MapReduce [6] programming framework is used to handle the concurrent processing of spatial query on Big data. Two tier based distributed spatial index [7, 8] is applied to handle the parallelism with Quad tree index for global index and space filling curve for the local index. SpatialHadoop [9] is an extended version of Hadoop, which enables the adequate processing of spatial query over the large scale data. It adds a spatial language called Pigeon. Spatial indexes and spatial operations like range query, spatial join, and a suite of computational geometry operations.

A distributed Voronoi index [10] is used to handle Geo spatial query in a parallel manner. It uses a MapReduce model to designate the range query, the nearest neighbor query, and the reverse nearest neighbor query. Voronoi based index produces an appropriate query response time in 2D space when compared to MapReduce based R tree index. A prototype EGIS (Engineering Geographic Information Systems) [11, 12] uses the GPS technology for real time monitoring of spatial information technology of 3D space.

A dynamic indexing scheme [13, 14] initiates a graph based approach to answer the query on latest snapshot. It reconciles the queries based on exact method and approximation method. The exact method induction scheme is used to resolve the queries that contain no errors. An approximation method is used to resolve the queries that contain errors. A geo-hash based index [15] is proposed to access the spatial data object. Geo hash means obtaining the interleaved bits from the latitude and longitude pair and uses it as an index for identifying the spatial object in GIS. It improves the performance of query processing of spatial data.

3 Distributed model

Let MD be a Medical Database which consists of 'n' objects. Each object O € MD has feature values $\{f_1, f_2,\ldots f_n\}$ along n-dimensional space. It can be represented as a point in n-dimensional space R^n. For retrieving the hospitals within the particular range from the current location L, the distance of each object O from the current location L has to be computed. If the distance falls within the range specified, then the corresponding objects are retrieved. After retrieving an object that satisfies the spatial predicates, then check for string predicates. The distributed index structure with an aho_corasick tree is constructed to resolve the spatial textual query. The overall architecture Spatial Textual Query Processing is shown in Fig. 1. The proposed system architecture consists of four important phases, namely, Probabilistic Learning Technique, Data transformation, Construction of Distributed index and String Matching algorithm.

3.1 Probabilistic learning technique

The basic requirement of the search engines is error detection and correction mechanisms. A probabilistic-based approach proposed here to support this activity shown in Fig. 2. The objective of this module is to create a rule dictionary. This module consists of two phases namely rule generation and learning phase.

In the generation phase, divide and conquer algorithm are applied as shown in Fig. 3. The algorithm applied over the string pair(s,t). For example, a string pair("beacon", "baacon") is considered. Apply divide and conquer, get the substring (("bea", "con"), ("baa", "con")). The bigram of the string pair such as (("be, ea", "ba, aa"),("co, on", "co, on") is found out. The dice coefficient for the above pair is (0,1). The rule from the pair is $be \rightarrow ba, ea \rightarrow aa$. Now remove the common prefix and suffix the rule will be $e \rightarrow a$. The Fig. 3 describes the flow of rule generation algorithm.

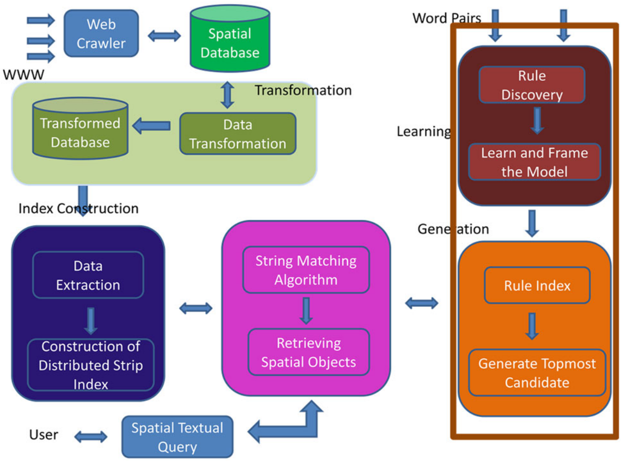


Fig. 1 Architecture of location based spatial textual query processing

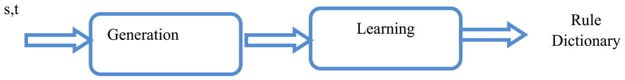
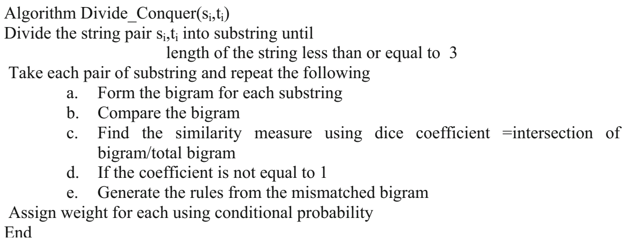


Fig. 2 Construction rule dictionary



```
Algorithm Divide_Conquer($s_i$,$t_i$)
Divide the string pair $s_i$,$t_i$ into substring until
        length of the string less than or equal to 3
 Take each pair of substring and repeat the following
     a.  Form the bigram for each substring
     b.  Compare the bigram
     c.  Find the similarity measure using dice coefficient =intersection of
         bigram/total bigram
     d.  If the coefficient is not equal to 1
     e.  Generate the rules from the mismatched bigram
 Assign weight for each using conditional probability
End
```

Fig. 3 Flow of rule generation algorithm

Let $T_r(s_i,t_i)$ be the set of possible rules created for the string pair (s_i,t_i). The mapping between the input string, output string and the rules are expressed with conditional probability. The linear model defines the probability for the $T_r(s_i,t_i)$ and t_i for the given s_i as follows

$$P(t_i, T_r(s_i, t_i)|s_i) = \frac{Exp\left(\sum_{i=1}^{k} w_i\right)}{z(s_i)}$$

k is the number of elements in weight vector. The value of this probability always remains positive. The normalizing constant is

$$z(s_i) = \sum_{t_i'} Exp\left(\sum_{i=1}^{k} w_i\right)$$

In the learning phase, the maximum-a-likelihood function is applied to select the best rule from the set of rules generated. The log likelihood function can be defined as,

$$P(t_i|s_i) = \sum_{T_r(s_i,t_i)} P(t_i, T_r(s_i, t_i)|s_i)$$

$$L(\omega) = \sum_{k} \log P(t_i|s_i)$$

3.2 Distributed index structure

The main issue in distributed index structure is how to partition the data space equally. The existing range and hash partition have suffered from the problem of skew distribution and selecting an appropriate hash function respectively. To overcome these issues, data transformation technique is applied.

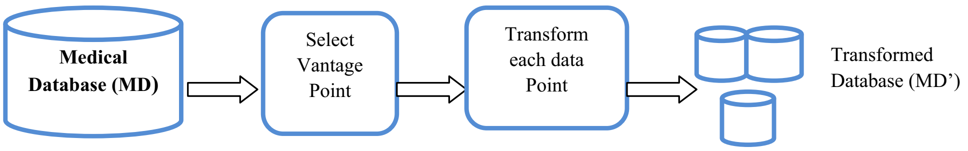


Fig. 4 Vanatge point transformation

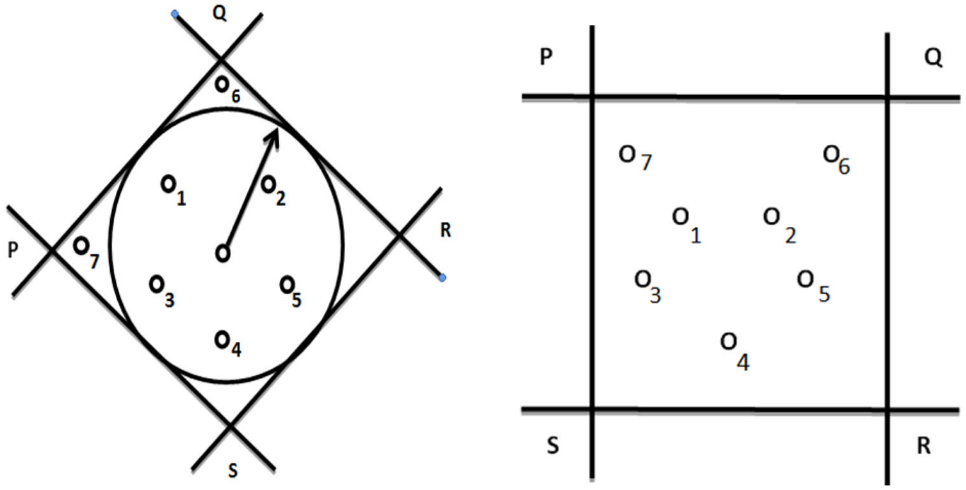


Fig. 5 Transformation of query from range to box

3.2.1 Vantage point data transformation

Data transformation technique is the process of converting the data from one form to another that is convenient to the destination. Transformation is technique of replacing a variable x by a function of that variable f(x) that changes the shape of a distribution. L^P space or Minkowski distance is a functional space used to measure the distance between two real points.

$$L^p = \left(\left(\sum_{i=1}^{n} |x_i - y_i| \right)^p \right)^{1/p}$$

When p = 2 the distance is called Euclidean distance. The Euclidean distance based transformation technique is used to transform each data point into simple data values. Each data point in the original database is transformed into a point based on a pre computed global point called vantage point (V) as shown in Fig. 4.

Let d be the distance of the object O from the current location L. Each object point O in MD with distance d in Euclidean space transforms with distance d' from the vantage point V. The database gets transformed from $MD \rightarrow MD'$. This type of transformation not only changes the data space, it also change the query shape. This method of transformation produces better results than existing method of 2DTransformation in terms of false positives. Vantage point transformation has advantages over the transformation based on the current location.

The Fig. 5 shows that when vantage point transformation applied, the query shape is changed into box shape.

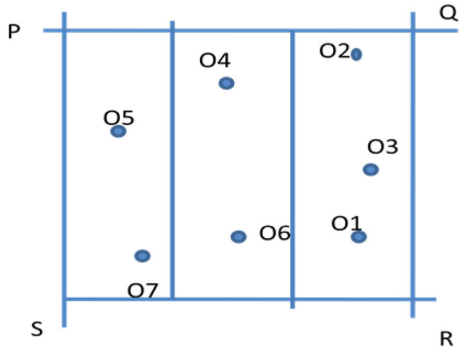


Fig. 6 Distributed strip index

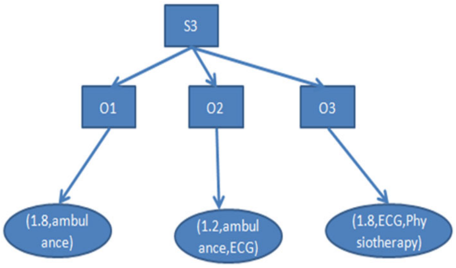


Fig. 7 R tree local index

This transformed data space has taken as the input for distributed strip index.

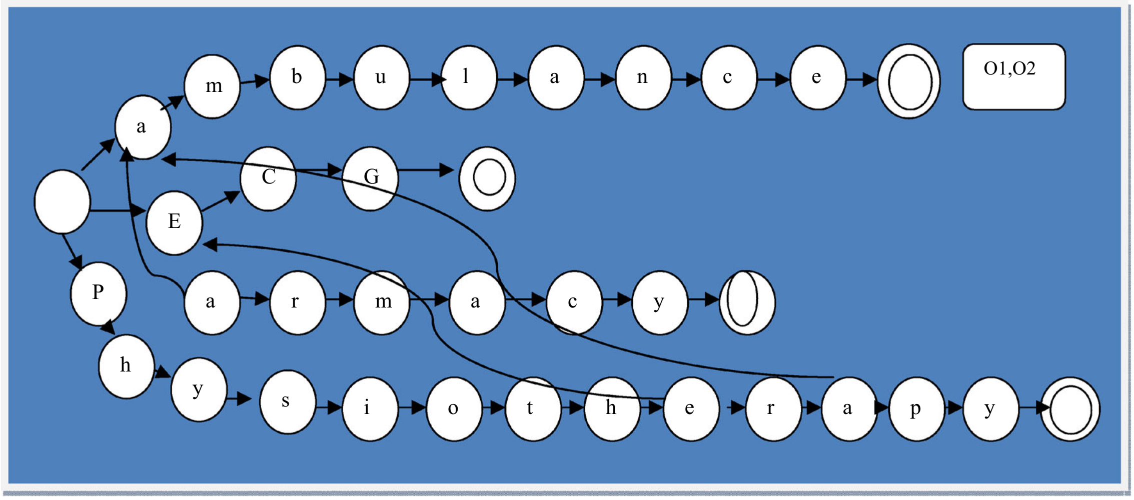
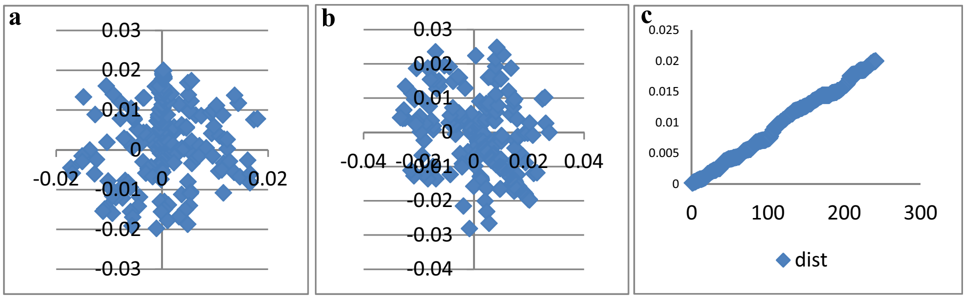


Fig. 8 An aho_corasick tree

Fig. 9 Graphical analysis of data space based on (latitude, longitude) attributes in **a** Original space **b** 2D space **c** Euclidean space



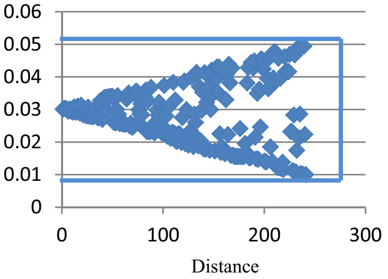


Fig. 10 Vantage point based space transformation

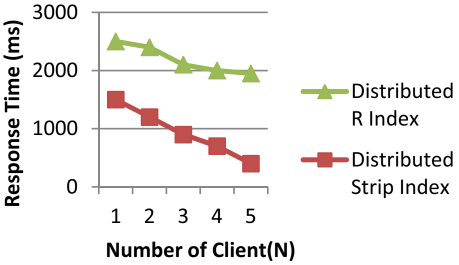


Fig. 12 Performance of distributed strip index

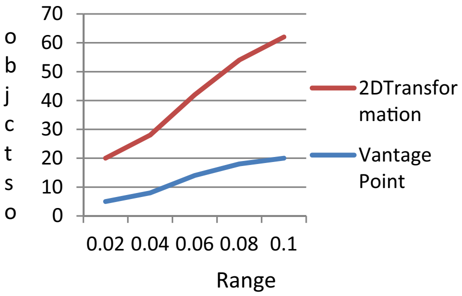


Fig. 11 Comparison based on false positive

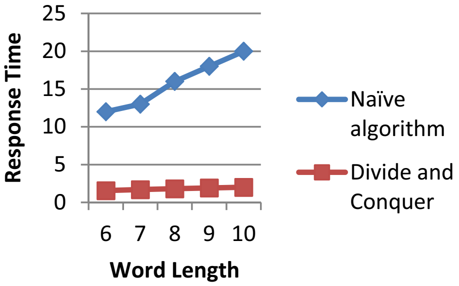


Fig. 13 Time complexity of rule generation

3.2.2 Distributed strip index

Location based queries cannot be processed by a single server since it has to handle large volume of data and requires large preprocessing cost and maintenance cost. In order to handle the large volume of data, two level distributed index structures used. In the global level, distributed strip index used to partition the given box data space into equal partition or the strip of equal size. As shown in Fig. 6, the given data space PQRS is divided into three equal strips called S1, S2, S3 and assigned each strip to a client.

Each client constructs the R tree index as shown in Fig. 7 based on ranges in order to resolve the spatial predicates. The objects that satisfy the spatial condition are retrieved by the clients. In the strip S3 all the objects O1, O2, O3 are within the range of 2 km. Then these objects are checked against the string predicates by constructing an aho_corasick tree.

Aho_Corasick tree is one of the best tree based string matching index. The objects that satisfy the spatial condition is retrieved first. Then using the string attributes associated with that objects, an aho_corasick tree is constructed. Each node in the tree denotes a character. The search for the given query keyword starts from the root of the tree. Each node checks against the query keyword. If the match finds, then follows the corresponding path to retrieve the objects. If the node does not match with the given query keyword character, then the rule from the rule dictionary is applied to continue the process of searching. Suppose the query keyword is “ambulance”, then the search started from the root node and follows the path of keyword “ambulance” and return the objects O1, O2. Figure 8 shows an aho_corasick tree for string attributes. The tree-based index structure is best for string searching since it is possible to prune the search space by cut off the other paths except the search path.

4 Experimental evaluation

The real time medical data set is collected from web site dat.gov. It consists of data about the hospital name, location in terms of the latitude and longitude, address, facilities within the hospital. The input query is “List all hospitals within 2 km with ambulance facility”. The (latitude, longitude) values of the current position are retrieved. The data objects within 2 km based the current position is analyzed. The graphical analysis is made to cover the data space within 2 km. The following Fig. 9a shows the comparison of shape of data in original space with the radius of 0.02, Fig. 9b shows the shape of data after 2d transformation and Fig. 9c shows distribution of data in Euclidean space from the current location of data.

Distribution of data in Euclidean space based on the current location is skew distribution as shown in Fig. 9c. It gives rise to the problem of partitioning the data space in distributed environment. The space transformation based on vantage point converts the range shape into box shape as shown in Fig. 10.

Vantage point based transformation has the advantages of getting box shape, so that the equal space partition technique applied in distributed environment. The shape of distribution can be improved by selecting the vantage point outside the range. The number of false positive is less when compared with the existing 2DTransformation technique (x + y, x − y) shown in Fig. 11.

The distributed index structure with vantage point transformation produces better results than the distributed R tree index structure is shown in Fig. 12. The vantage point transformation reduces the overlapping of nodes in construction of MBR. When the rule dictionary is constructed, two factors such as time complexity and the size of the rule set, have to be considered to improve the query performance. Divide and conquer technique for rule generation takes the time complexity of O(log(st)), but the naïve algorithm for rule generation takes time complexity of O(st) where s, t be the length of the string pair shown in Fig. 13.

5 Conclusion

Nowadays, healthcare information system plays a crucial role in human life. The awareness about the diseases, symptoms, hospitals and its facilities, specialists etc. among the people increases because of mobile devices. The performance of location based query for accessing the medical database has to be improved. This paper discusses two improvements, one is a distributed based index structure proposed to speed up the process of the query as well as to enhance the scalability. A new divide and conquer based learning technology introduced to handle the context sensitive error that occur in the database system.

References

1. Osborn, W., & Hinze, A. (2007). Issues in location based indexing for co-operating mobile information systems. In R. Meersman, Z. Tari, & P. Herrero (Eds.), On the move to meaningful internet system 2007, OTM 2007 workshops, OTM 2007 lecture notes in computer science (vol. 4805). Berlin: Springer.

2. Mouratidis, K., Bakiras, S., & Papadias, D. (2009). Continuous monitoring of spatial queries in wireless broadcast environments. *IEEE Transaction on Mobile Computing, 8*(10), 1297–1311.
3. Naresh, K., Thangakumar, J., & Pannem, D. (2012). Spatial query monitoring in wireless broadcast environment. In *IEEE international conference on internet computing and information communication (ICICI)* (pp. 35–42). Berlin: Springer. ISBN: 978-81-322-1299-7.
4. Lin, L., Cai, Y. Z., & Xu, Z. (2008). Spatial temporal indexing mechanisms based on snapshot increment. *Advanced in spatial temporal analysis*. London: Taylor and Francis group.
5. Zhong, Y., Han, J., Zhang, T., Li, Z., Fang, J., & Chen, G. (2012, May). Towards parallel spatial query processing for big spatial data. In *IEEE 26th international conference on parallel and distributed processing symposium workshops & PhD forum (IPDPSW)* (pp. 2085–2094).
6. Zhang, C., Li, F., & Jestes, J. (2012, March). Efficient parallel kNN joins for large data in MapReduce. In *Proceedings of the 15th international conference on extending database technology* (pp. 38–49). New York City: ACM. ISBN: 978-1-4503-0790-1.
7. Yu, Z., & Liu, Y. (2015). Scalable distributed processing of K nearest neighbor queries over moving objects. *IEEE Transaction on Knowledge and Data Engineering, 7*(5), 1383–1396.
8. Zheng, B., Xu, J., Lee, W. C., & Lee, L. (2006). Grid-partition index: A hybrid method for nearest-neighbour queries in wireless location based services. *VLDB Journal, 15,* 21–39. https://doi.org/10.1007/s00778-004-0146-0.
9. Eldawy, A., & Mokbel, M. F. (2015, April). SpatialHadoop: A MapReduce framework for spatial data. In *IEEE international conference on data engineering* (pp. 1352–1363).
10. Akdogan, A., Demiryurek, U., Banaei-Kashani, F., and Shahabi, C. (2010, November). Voronoi-based geospatial query processing with MapReduce. In *IEEE second conference on cloud computing technology and science* (pp. 9–16).
11. Zhang, J., Chen, X. L., Zhong, C., Wu, H., and Duan, S. (2008, July). Application of geo-spatial information technology in the engineering manage of roller compaction construction. In *IEEE international conference on geoscience and remote sensing symposium* (vol. 3, pp. 1312–1315).
12. Xuan, K., Zhao, G., Taniar, D., Srinivasan, B., Safar, M., and Gavrilova, M. (2009). Network Voronoi diagram based range search. In *IEEE 23rd international conference on advanced information networking and applications* (pp. 741–748).
13. Hariharan, R., Hore, B., Li, C., & Mehrotra, S. (2007, July). Processing spatial keyword queries in geographic information retrieval systems. In *IEEE international conference on scientific and statistical database management* (p. 16). ISSN: 1551-6393.
14. Akiba, T., Iwata, Y., & Yoshida, Y. (2014, April). Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling. In *Proceedings of 23rd international conference on world wide web* (pp. 237–248). New York City: ACM.
15. Suwardi, I. S., Dharma, D., Satya, D. P., & Lestari, D. P. (2015, August). Geohash index based spatial data model for corporate. In *International conference on electrical engineering and informatics* (pp. 478–483). ISSN: 2155-6830.