Efficient Mechanisms for Spatial Approximate String Search for Querying Geographical Databases

S Sahithi[1], M Sheshikala[2], Dr D Rajeshwara Rao[3]
[1]*M.Tech Student, SR Engineering College, Warangal, Telangana, India*
[2]*Assistant Professor, SR Engineering College, Warangal, Telangana, India*
[3]*Professor, SR Engineering College, Warangal, Telangana, India*

Abstract- **Spatial databases have become popular and they are queried frequently in order to obtain required information. In the search process approximate string search plays an important role in spatial databases. Especially, the range queries with approximate string are a challenging job to be achieved. This is the open problem to be solved with respect to spatial approximate string search. Recently Li et al. proposed a solution for the problem. Their solution considers the query as spatial approximate string (SAS) query. The solution was provided both in Euclidean space and road networks using min-wise signatures and MHR-tree. In this paper we proposed a system which is an extension to the existing system which focuses on building a framework that is capable of examining spatial approximate substring queries, designing methods that are more update friendly to solve the selectivity estimation problem for range queries on road networks represented by spatial database. Thus the proposed system reduces the execution cost and space consumption.**

Index Terms **– approximate string search, range query, road network, spatial databases**

I. INTRODUCTION

A spatial database is the database which has been specially optimized to store data pertaining to objects in the real world. In other words spatial data is the data which represents objects in geometric space. The objects are stored in database in the form of lines, points and polygons. A Relational Database Management System (RDBMS) with additional features can support spatial databases which are extensively used in environmental studies, Global Positioning System (GPS), and Geographic Information System (GIS). Spatial Data Mining (SDM) is a process of discovering trends or patterns from large spatial databases that hold geographical data. Objects in space such as roads, rivers, forests, deserts, buildings, cities etc., are stored in spatial database. Spatial databases are so complex and make the SDM more difficult when compared with traditional databases. The major applications of SDM are related to co-location mining, spatial outlier detection and location prediction. PostGIS, Microsoft SQL Server, Oracle Spatial, SpatiaLite etc., are the products available for building spatial databases. Making queries on spatial database has many important real world utilities. For instance it can be used to obtain nearest neighbor data which meets certain spatial and non-spatial criteria. Moreover the spatial approximate string search can be of very much useful in making appropriate queries. This is the motivation behind taking up this project which builds a prototype application for spatial approximate string query processing.

Spatial databases have become popular and they are queried frequently in order to obtain required information. In the search process approximate string search plays an important role in spatial databases. Especially, the range queries with approximate string are a challenging job to be achieved. This is the open problem to be solved with respect to spatial approximate string search. Recently Li et al.

proposed a solution for the problem. Their solution considers the query as spatial approximate string (SAS) query. The solution was provided both in Euclidean space and road networks using min-wise signatures and MHR-tree. In this paper we proposed a system which is an extension to the existing system which focuses on building a framework that is capable of examining spatial approximate substring queries, designing methods that are more update friendly to solve the selectivity estimation problem for range queries on road networks represented by spatial database.

Our contributions in this paper are as given below.

- Spatial approximate string search mechanisms are to be built. This will help making spatial approximate string queries which will help in obtaining geographical information which is very useful in the real world.
- An MHR tree has to be constructed in order to hold spatial data and process spatial queries with good performance.
- Filtering and pruning are to be carried out.
- Finally candidate results are to be processed in order to complete spatial approximate string queries.

The remainder of the paper is structured as follows. Section II provides review of literature. Section III provides details about proposed work. Section IV presents prototype application. Section V presents experimental results while section VI concludes the paper.

II. RELATED WORK

This section throws light into review of literature related to spatial approximate string search. In [1] IR2-tree was introduced for kNN queries on geographical databases. It supports keyword search which is convenient to end user. However, this solution does not support spatial approximate string search. No selectivity estimation concept was used there in. In [2] and [3] m-closest keywords concept was introduced based on IR2-tree. MHR-tree was introduced in [4] for improving spatial approximate string search.

Approximate string search has attracted many researchers and they contributed to its growth in terms of techniques as explored in [5], [6], [7], [8], [9], [10], and [11]. In these functions, the similarity function quantifies the similarity between two strings. They make use of measures like Jaccard and edit distance. The q-grams concept was introduced used in [8] and [10]. In [11] certain improvements are made to q-grams in order to incorporate inverted index as explored in [12]. Since then edit distance function is used by many researchers as explored in [13],[14],[11], [15], [6], and [7].

III. PROPOSED **SOLUTION**

In this paper we proposed a solution for spatial approximate string search. Based on the mechanism found .we proposed the architecture presented in Figure 1. As per this architecture users can make spatial approximate string searches and get query results.

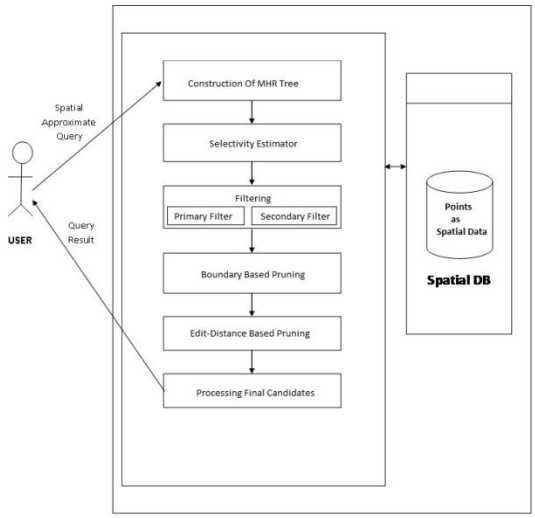

Figure 1- Proposed architecture for spatial approximate string search

As can be seen in Figure 1, it is evident that the functional requirements of the system are logically divided into the modules such as, Construction of MHR Tree, Selectivity Estimation, Filtering, Pruning and Processing Candidate Results.

Construction of MHR Tree

R-tree is a data structure meant for holding spatial data for the ease of processing. This module is responsible to construct MHR Tree which embeds min-wise signatures into the R-tree. This will help in faster processing of query.

Selectivity Estimation

This module is responsible to optimization spatial query processing. The purpose of spatial selectivity estimation is to partition the spatial data into a collection of buckets so that data within each bucket is as close as possible to a uniform distribution (in terms of their geometric coordinates). This is known as spatial uniformity principle. The selectivity estimation in spatial query processing results in reduction of search space and thus cost is reduced.

Filtering

This module is meant for filtering the spatial query given. It is a process of applying given criteria to the processing so as to reduce the number of candidate rows returned by the query. The filter is done in two phases namely primary filter which gives geometrics that has approximate results. The secondary filter applied the predicates in order to filter it further.

Pruning

Pruning is a process of getting rid of unwanted spatial records in order to improve query processing further. This module is responsible for boundary based pruning and edit based pruning.

Processing Candidate Results

The candidate results obtained from the results of the previous module (filtering) are further processed in order to return final results which are accurate.

Algorithm Proposed

Algorithm: Spatial Query Processing Algorithm
Purpose: Selectivity estimation and processing range queries on road networks represented in spatial databases
Inputs : Spatial approximate query *q*, spatial database *SDB*
Outputs: Results

STEP 1: CONSTRUCTION OF MHR TREE
Initialize an R-tree denoted as *rt*
Initialize MHR tree *mt*
Compute q-grams *QG*
Compute min-wise signature *s(QG)* from *QG*
mt = embed min-wise signatures into *rt*

STEP 2: SELECTIVITY ESTIMATION
Convert the *q* into range query *rq*
Divide *mt* into buckets *B*
FOR each bucket *b* in *B* that intersects *rq*
    Compute the area of intersection *aoi*
    Use min-wise signatures of *QG* to build selectivity estimator *se*
END

STEP 3: FILTERING AND PRUNING
Use *se* for primary filtering *pf*
Then use *pf* results for secondary filtering *sf* based on spatial predicates
Apply boundary based pruning *bp* on results of *sf*
Apply edit based pruning on results of *bp* and give candidate results *CR*

STEP 4: PROCESSING CANDIDATE RESULTS
Initialize *CR`*
FOR each candidate result *cr* of *CR*
    Compute distance d of *cr* from query point
    Find *cr`* that satisfies distance d
    Add *cr`* to *CR`*
END
return *CR`*

Figure 2 – Proposed algorithm for spatial query processing

R-tree is a data structure meant for holding spatial data for the ease of processing. This module is responsible to construct MHR Tree which embeds min-wise signatures into the R-tree. This will help in faster processing of query. The selectivity estimation in spatial query processing results in reduction of search space and thus cost is reduced. The filter is done in two phases namely primary filter which gives geometrics that has approximate results. The secondary filter applied the predicates in order to filter it further. Pruning is a process of getting rid of unwanted spatial records in order to improve query processing further. This module is responsible for boundary based pruning and edit based pruning. Finally, the candidate results obtained from the results of the previous module (filtering) are further processed in order to return final results which are accurate.

IV. EXPERIMENTAL RESULTS

We built a prototype application that demonstrates the experimental results. The experiments are made with respect to spatial approximate string search. The environment used to build the application is a PC with 4 GB RAM, core 2 dual processor running Windows 7 operating system. The main search interface is shown in Figure 3.

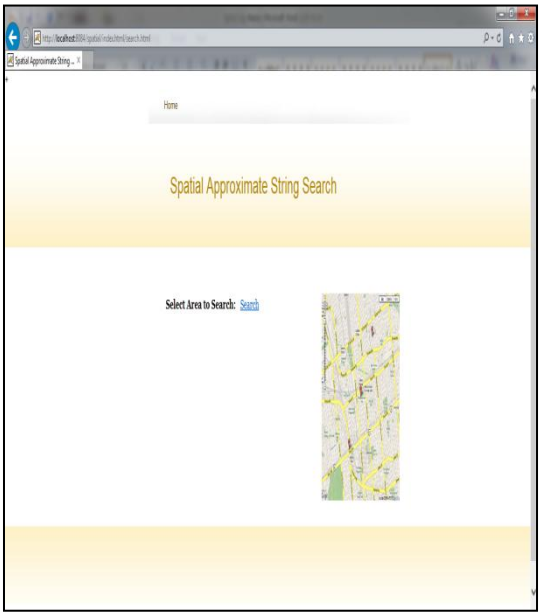

Figure 3 - Screen for Search Page

As can be seen in Figure 3, it is evident that the UI provides search mechanism. User operates on the search interface to have search completed.

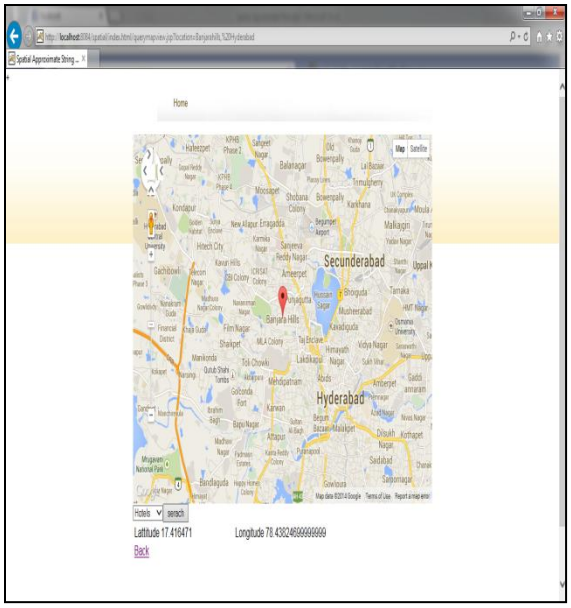

Figure 4 - Screen for displaying the location in the map

As can be seen in Figure 4, it is evident that the search operation is completed and the results are shown in the form of Google Map reflecting the correct location besides showing latitude and longitude of the search result.

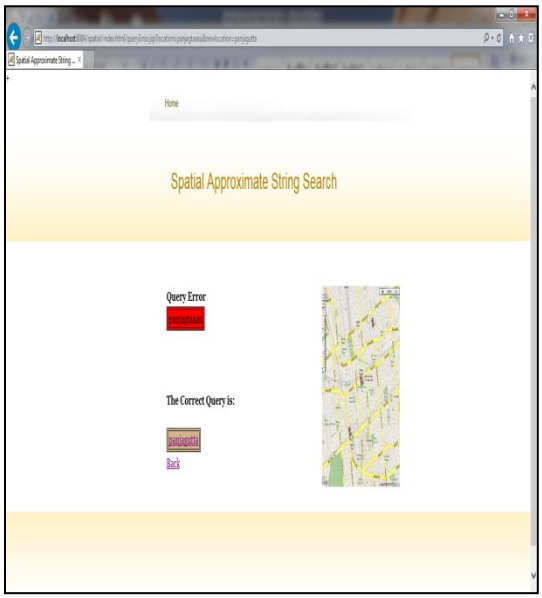

Figure 5- Query correction mechanism

The proposed application has query correction feature that makes use of edit distance mechanism as explored in the algorithm. The solution is more user-friendly with this feature incorporated into the framework.

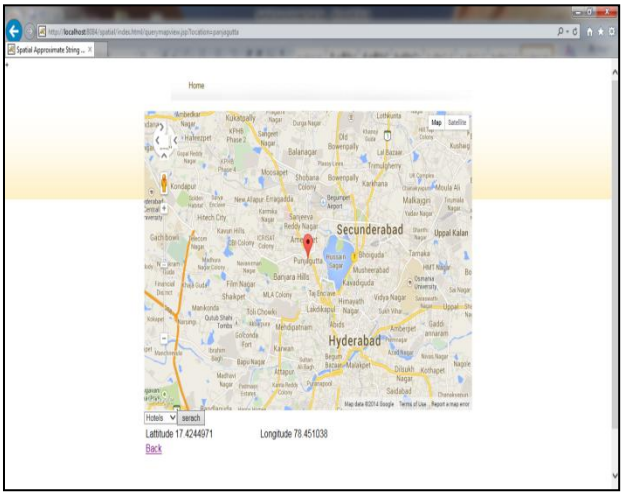

Figure 6 - Screen for displaying the location in the map

As can be seen in Figure 6, it is evident that the search operation is completed and the results are shown in the form of Google Map reflecting the correct location besides showing latitude and longitude of the search result.

V. CONCLUSIONS AND FUTURE WORK

Spatial databases are complex and can hold data about spatial objects and text pertaining to non-spatial. However, making queries in such databases is non-trivial problem that needs to be addressed. Especially the spatial approximate string queries need much more focus. This paper presents a comprehensive study for spatial approximate string queries in both the Euclidean space and road networks. We use the edit distance as the similarity measurement for the string predicate and focus on the

range queries as the spatial predicate. We also address the problem of query selectivity estimation for queries in the Euclidean space. A prototype application was built to demonstrate the spatial approximate string queries. The application is user-friendly and helps in visually presenting results of the spatial approximate string queries. It also helps in transforming search strings appropriately so as to help Selectivity estimation of range queries. In future we intend to improve the spatial approximate string search with different distance measures.

REFERENCES

[1] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, pages 656–665, 2008.

[2] D. Zhang, B. C. Ooi, and A. Tung. Locating mapped resources in web 2.0. In *ICDE*, pages 521–532, 2010.

[3] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *ICDE*, pages 688–699, 2009.

[4] S. Alsubaiee, A. Behm, and C. Li. Supporting location-based approximate-keyword queries. In *GIS*, pages 61–70, 2010.

[5] K. Chakrabarti, S. Chaudhuri, V. Ganti, and D. Xin. An efficient filter for approximate membership checking. In *SIGMOD*, pages 805–818, 2008.

[6] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *SIGMOD*, pages 313– 324, 2003.

[7] S. Chaudhuri, V. Ganti, and R. Kaushik. A primitive operator for similarity joins in data cleaning. In *ICDE*, pages 5–16, 2006.

[8] L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava. Approximate string joins in a database (almost) for free. In *VLDB*, pages 491–500, 2001.

[9] E. Sutinen and J. Tarhio. On using q-gram locations in approximate string matching. In *ESA*, pages 327–340, 1995.

[10] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci.*, 92(1):191–211, 1992.

[11] X. Yang, B. Wang, and C. Li. Cost-based variable-length-gram selection for string collections to support approximate queries efficiently. In *SIGMOD*, pages 353–364, 2008.

[12] M.-S. Kim, K.-Y. Whang, J.-G. Lee, and M.-J. Lee. n-gram/2l: a space and time efficient two-level n-gram inverted index structure. In *VLDB*, pages 325–336, 2005.

[13] G. Li, J. Feng, and C. Li. Supporting search-as-you-type using sql in databases. *TKDE*, To Appear, 2011.

[14] S. Sahinalp, M. Tasan, J. Macker, and Z. Ozsoyoglu. Distance based indexing for string proximity search. In *ICDE*, pages 125–136, 2003.

[15] A. Arasu, S. Chaudhuri, K. Ganjam, and R. Kaushik. Incorporating string transformations in record matching. In *SIGMOD*, pages 1231–1234, 2008.