IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 03 (March. 2014), ||V6|| PP 49-54

A Fast Spatial String Search with Service Composition Method

S.Udhayakumar , D.Sureshkumar
[1,2]*PG Scholar, Department Of CSE,Hindusthan College Of Engineering And Technology,Coimbatore.*

***Abstract:* -** Location-based services and the abundant usage of smart phones and GPS-enabled devices, the necessity of outsourcing spatial data has grown rapidly. This work deals with the fast nearest string search in large spatial databases. Specifically, this investigates spatial associated queries augmented with a string similarity nearest search predicate in both Euclidean space and road networks.Euclidean space is ordinary two- or three-dimensional space.These called as the spatial approximate string (SAS) query. In Euclidean space, this propose an approximate solution, the MHR-tree, which embeds min-wise signatures into an R-tree.Presented a novel adaptive algorithm to find balanced partitions using both the spatial and string information stored in the tree. For queries on road networks, we propose a novel exact method, RSASSOL, which significantly outperforms the baseline algorithm which is serving in basis, as for measurement,calculation or location in practice.The proposed system develops a new access method called Proxy based query answering that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time.For the fast nearest neighbor search additionally the system implements Popularity based answer filtering based on sequence and service composition methods along with the SI scheme. The experiments have been done using C#.net and data set created using SQL server.

General Terms:-Approximate string search, range query, road network, spatial databases.

Keywords: -*Euclidean space,Proxy search,Service composition.*

I. INTRODUCTION

Keyword search over a large amount of data is an important operation in a wide range of domains.Since exact match is a special case of approximate string match, it is clear that keyword search by approximate string matches has a much larger pool of applications. Approximate string search is necessary when users have a fuzzy search condition, or a spelling error when submittingthe query, or the strings in the database contain some degree of uncertainty or error. In the context of spatial databases, approximate string search could be combinedwith any type of spatial queries. In this work, we focus onrange queries and dub such queries as Spatial Approximate String (SAS) queries. An increasing number of applications require the efficient execution of nearest neighbor (NN) queries constrained by the properties of the spatial objects. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the user to provide a list of keywords that the spatial objects (henceforth referred to simply as objects) should contain, in their description or other attribute. A spatial keyword query consists of a query area and a set of keywords. The answer is a list of objects ranked according to a combination of their distance to the query area and the relevance of their text description to the query keywords. The proposed system deals the spatial approximation string search based on the Euclidean space and road space. *K*-Nearest Neighbors Algorithm (*k*-NN) is a non-parametric method for classification and regression, that predicts objects' "values" or class memberships based on the *k*closest training examples in the feature space. *K*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small).

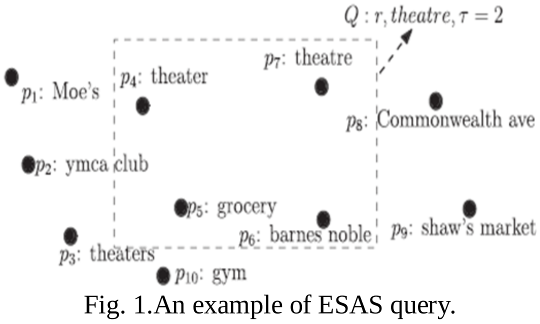


Fig. 1.An example of ESAS query.

An example in the euclidean space isshown in Fig. 1, depicting a common scenario in location based services:find all objects within a spatial range r(specified by a rectangular area) that have a description that is similar to “theatre.” We denote SAS queries in euclidean space as (ESAS) queries. Fig. 2 extends SAS queries to road networks (referred as RSAS queries).

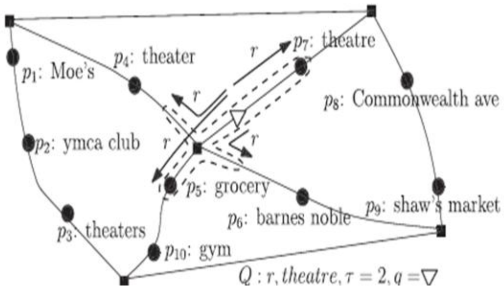


Fig. 2. An example of RSAS query.

II. PROBLEM DESCRIPTION

The problem is want to search in a collection (unordered set) of strings to find those similar to a single query string (“selection query”).Selectivity estimation of range queries on road networks is a much harder problem than its counterpart in the Euclidean space.Different points may contain duplicate strings.

2.1 Existing System

The IR2-tree was proposed to perform exact keyword search with kNN queries in spatial databases. Many existing system studied the m-closest keywords query in Euclidean space. Two other relevant studies concentrates on ranking queries that combine both the spatial and text relevance to the query object.The LBAKtree was proposed to answer location-based approximate keyword queries.There was many similarity functions have been proposed to quantify the closeness between two strings.Several techniques have been proposed for identifying candidate strings within a small edit distance from a query string.

2.1.1 Disadvantages of Existing System

The overall drawbacks of all the existing systems are described in below.

1. The IR2-tree cannot support spatial approximate string searches.
2. IR2-Tree is that it only supports exact keyword search.
3. Existing string solution suffers the same scalability and performance issues.
4. Query optimization problem.
5. R trees suffers from high IO cost and communication overhead.
6. Only considered Euclidean space or road space.

2.2 Proposed System

The proposed system develops a new access method called Proxy based query answering that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time.For the fast nearest neighbor search additionally the system implements Popularity based answer filtering based on sequence and service composition methods along with the SI scheme. The experiments have been done using C#.net and data set created using SQL server.

2.2.1 Advantages of Proposed System

1. Examining spatial approximate substring queries.
2. Introduces a new index for answering ESAS queries efficiently which embeds min-wise signatures of q-grams from subtrees into the R-tree nodes.
3. This examines query selectivity estimation for queries in the Euclidean space.

III. MODULE DESCRIPTION

This project has five main modules to demonstrate the entire process of Spatial String Search. The modules are follows.

3.1Data Set Creation

The server can add service details such as name, location and other contact details. Here the admin can click on the map to get the exact point of the area, which they need to add into database. Creation of spatial datasets dynamically may help the user to get the updated information from the service. The first module contains the appropriate service registration process.

For extracting the clicking points from the map, the following technique has been used.

1. Click point extraction.:
2. LocalMousePosition.X.ToString();
3. LocalMousePosition.Y.ToString();

3.2 User Module

The mobile host is the end user of the proposed system. The user can register themselves and can login into the system. This process helps to track the user interest and searching history from the server. The host id is an unique id, which has been created using random function. The system will automatically provides an id when registration process in progress. This takes the location details while searching. The proposed system tracks the user list based on the registration process.

3.3 Query Processing

This module focuses on range queries and dubs such queries as Spatial queries. The proposed method partitions the road network, adaptively searches relevant subgraphs, and prunes candidate points using both the string matching index and the spatial neighbor reference nodes. Finally, an adapted SI Index method is applied, together with the exact edit distances, to verify the final set of candidates.This module rank tuples using an aggregate score function on their attribute values. In this module we will defines the top-k spatial preference query problem and describes the index structures for the datasets.

3.4Proxy Based Query Answering

The proxy based query answering module maintains the cache and buffer space for frequent query answering. Here the overall area splitted into different regions and every region maintains the cache.

3.5 Service Composition

The final module provides the composite services for the current search. A proxy-based approach to continuous nearest-neighbor (NN) and window queries. The proxy creates estimated valid regions (EVRs) for mobile clients by exploiting spatial and temporal locality of spatial queries. For NN queries, we devise two new algorithms to accelerate EVR growth, leading the proxy to build effective EVRs even when the cache size is small.

IV. ARCHITECTURE

Architecture of the entire system is as follows,

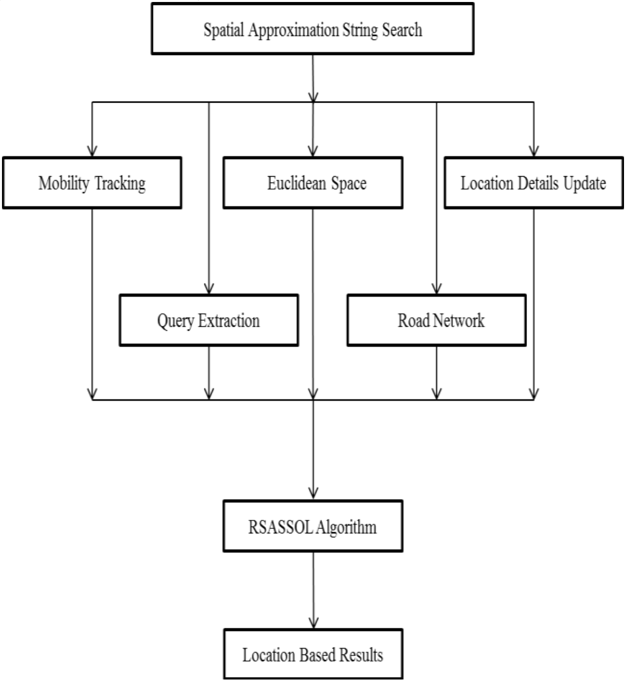


Fig. 4.1 Architecture Diagram

4.1 Flow of Process

The overall process of this spatial string search with service composition method is shown in below.

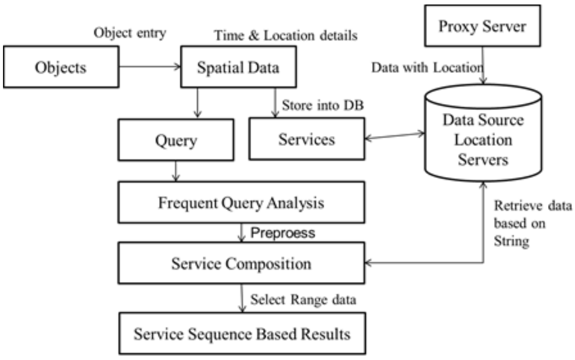


Fig. 4.2 Overall Diagram

Euclidean spaceis a real vector space on which is defined a fixed symmetric bilinear form whose associated quadratic form is positive definite.

V. ALGORITHM DESCRIPTION

The algorithms used in this concept are described in below.

5.1 RSASSOL Algorithm

RSAS query framework consists of five steps . Given a query, we first find all sub graphs that intersect with the query range. Next, we use the Filter Trees of these sub graphs to retrieve the points whose strings are potentially similar to the query string. In the third step, we prune away some of these candidate points by calculating the lower and upper bounds of their distances to the query point, using VR. The fourth step is to further prune away some candidate points using the exact edit distance between the query string and strings of remaining candidates. After this step, the string predicate has been fully explored. In the final step, for the remaining candidate points, we check their exact distances to the query point and return those with distances within r. We dub this algorithm RSASSOL and the rest of this section presents the details of this algorithm.

5.2 Extension to Range Queries

Although we focus on window queries in this paper, the proposed approach is able to support range queries whose query region is circular. To resolve a range query, the main idea is to transform the range query to a window query and then address the window query with the window query processing algorithm. This describes the range query processing procedure as follows:

Step 1. When the proxy receives a range query with query location. The proxy executes the window query processing algorithm to resolve the window query and to compute the corresponding data.

Step 2. Let the answer objects of the window query be fp1; p2; . . . ; png. Without loss of generality, we reorder the answer objects in ascending order based on their distance to.

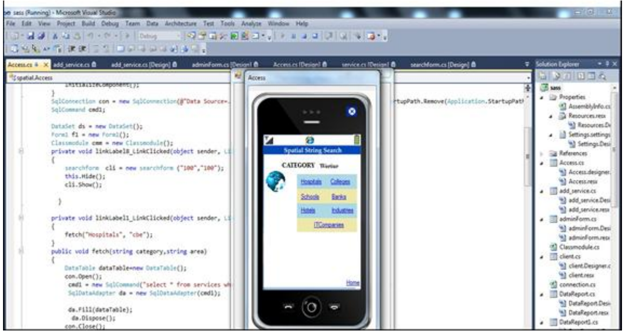
Step 3. This denotes the shortest distance between pi and the range query circle C.

Step 4. The proxy filters out the answer objects which are not within the query circle C and then has the final answer objects.

Step 6. The proxy returns the answer objects.

6.RESULTS

The following screen shots shows the estimated output of this paper.

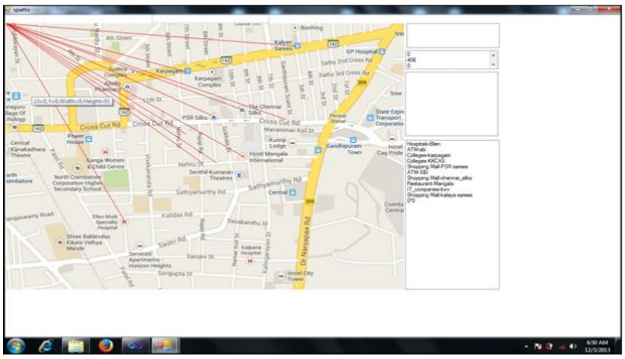


```
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection(@"Data Source=
        SqlCommand cmd1;

        DataSet ds = new DataSet();
        Form1 f1 = new Form1();
        Classmodule cmm = new Classmodule();
        private void linkLabel8_LinkClicked(object sender, Li
        {
            searchform  cli = new searchform ("100","100");
            this.Hide();
            cli.Show();

          }

        private void linkLabel1_LinkClicked(object sender, Li
        {
            fetch("Hospitals", "cbe");
        }
        public void fetch(string category,string area)
        {
            DataTable dataTable=new DataTable();
            con.Open();
             cmd1 = new SqlCommand("select * from services wh
             SqlDataAdapter da = new SqlDataAdapter(cmd1);

             da.Fill(dataTable);
              da.Dispose();
            con.Close();
```

```
tupPath.Remove(Application.StartupPat
```

Fig 6.1 Query Search Home

This contains the several categories which we are going to search mostly in the map. If we choose the particular category means then it will displays all the places related to that category.

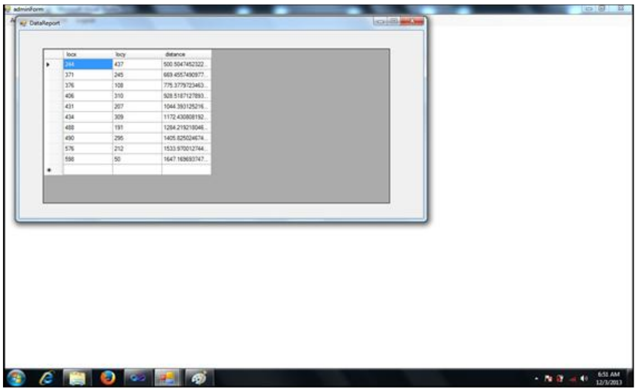

Kalyan Sarees
Karpagam Complex
Cross Cut Rd
Sathyamurthy Rd
Kalidas Rd
Sastri Rd
Sengupta St
Gandhipuram Town
Dr Nanjappa Rd
Hotel City Tower
State Express Transport Corporation
Hotel Mangala International
Senthil Kumaran Theatres
PSR Silks
The Chennai Silks
North Coimbatore Corporation Higher Secondary School
Ellen Multi Specialty Hospital
Central Kanakadhara Theatre
{X=0,Y=0,Width=0,Height=0}

0
406
0

Hospitals-Ellen
ATM-sbi
Colleges-karpagam
Colleges-KKCAS
Shopping Mall-PSR sarees
ATM-SBI
Shopping Mall-chennai_silks
Restaurant-Mangala
IT_companies-bvv
Shopping Mall-kalaya sarees
0\0

Fig 6.2 Map based Search

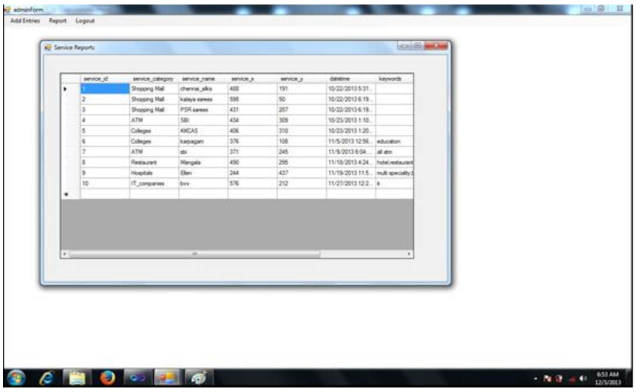
This is the output screen of the entire project. It shows the initial stage which represents the various locations available in the map. If we choose the particular location means it will provide all the informations regarding the area.



locx	locy	distance
244	437	500.5047452322...
371	245	669.4557490977...
376	108	775.3779723463...
406	310	928.5187127893...
431	207	1044.393125216...
434	309	1172.430808192...
488	191	1284.219218046...
490	295	1405.825024674...
576	212	1533.970012744...
598	50	1647.169693747...

Fig 6.3 Data Reports

It represents the Data reports which comes under the Admin side. It displays the Euclidean distance between the string X and Y.



service_id	service_category	service_name	service_x	service_y	datetime	keywords
1	Shopping Mall	chennai_silks	488	191	10/22/2013 5:31...	
2	Shopping Mall	kalaya sarees	598	50	10/22/2013 6:19...	
3	Shopping Mall	PSR sarees	431	207	10/22/2013 6:19...	
4	ATM	SBI	434	309	10/23/2013 1:10...	
5	Colleges	KKCAS	406	310	10/23/2013 1:20...	
6	Colleges	karpagam	376	108	11/5/2013 12:56...	education
7	ATM	sbi	371	245	11/9/2013 6:04...	all atm
8	Restaurant	Mangala	490	295	11/18/2013 4:24...	hotel,restaurant
9	Hospitals	Ellen	244	437	11/19/2013 11:5...	multi speciality,[illegible]
10	IT_companies	bvv	576	212	11/27/2013 12:2...	it

Fig 6.4 Sevice Reports

Service reports are the reports which provide all the sevice details and information about the various services to the administrator. It also contains the various X and Y positions and its Euclidean distances.

VI. CONCLUSION

This paper presents a comprehensive study for spatial approximate string queries in both the euclidean space and networks. We use the edit distance as the similaritymeasurement 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 inthe euclidean space.It also use the concept of proxy based query answering and service composition method,which provides the solution more efficiently.

VII. 8.FUTURE ENHANCEMENT

Future work includes examining spatial approximate substring queries, designing methods that are more update friendly, and solving the selectivity estimation problem for RSAS queries.

REFERENCES

[1] Feifei Li, Member, IEEE, Bin Yao, Mingwang Tang, and Marios Hadjieleftheriou, “Spatial Approximate String Search”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 6, JUNE 2013.

[2] S. Acharya, V. Poosala, and S. Ramaswamy, “Selectivity Estimation in Spatial Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 13-24, 1999.

[3] S. Alsubaiee, A. Behm, and C. Li, “Supporting Location-Based Approximate-Keyword Queries,” Proc. SIGSPATIAL 18th Int’l Conf. Advances in Geographic Information Systems (GIS), pp. 61-70, 2010.

[4] A. Arasu, S. Chaudhuri, K. Ganjam, and R. Kaushik, “Incorporating String Transformations in Record Matching,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 1231-1234, 2008.

[5] A. Arasu, V. Ganti, and R. Kaushik, “Efficient Exact Set-Similarity Joins,” Proc. 32nd Int’l Conf. Very Large Data Bases (VLDB), pp. 918- 929, 2006.