Organizing Duplicate Spatio-Temporal Data in Location Intelligence

Sumeet Gill, Meenakshi

***Abstract:** It has been noticed in research world in previous years that there is a huge explosion in the field of services based on location. These include applications based on location intelligence. All of these applications concentrate around process and management of data based on location and time. The amount of this kind of spatio-temporal data is increasing tremendously day by day and becoming difficult to handle, manage and query a big pile of spatio-temporal data. Because of big jumps in areas of location-aware instruments and applications depending on location, a large number of researchers are devoting their work and time towards indexing, storage and optimized retrieval of spatio-temporal data related to different areas. In this research paper, the researchers are explaining an efficient novel indexing technique to store and retrieve datasets with spatio-temporal attributes.*

***Index Terms:** Duplicate Keys, Hashing, Indexing, k-d Tree, Spatio-temporal.*

I. INTRODUCTION

The period we are living in is the era of spatial, temporal and spatio-temporal data which is exploding continuously. The reason behind it is that a big amount of spatio-temporal data is generated and passed on by satellite images, location-based instruments, metrological departments, urban development research areas, social media evolving at never before speeds and lot many to mention here. This large heap of spatio-temporal data impulses the researchers and application designers to design and develop novel methods to index spatio-temporal data so that the same can be queried efficiently when required. Not only the retrieval of queried spatio-temporal data, but its visualization is also required for effective and timely analysis of output. In this research work, the researchers are explaining a new indexing technique to index spatio-temporal datasets to retrieve data efficiently.

II. RELATED WORK

A lot of new structures based on concurrent and distributed access of spatio-temporal data have been proposed in the recent past. Many of such spatio-temporal indexing and access techniques come under the category of general-purpose large data processing system. Different indexing and accessing techniques can be divided in different categories depending on type, time, purpose of spatio-temporal data and include trajectory, cluster, leadership, migration, crime based datasets and many more with spatio-temporal properties to support different applications.

A. DTNH Indexing Method [1]

This is the indexing structure which is capable to index past, present and predict future location of objects. This structure is composed of three parts, each performing different role to index spatio-temporal data. The first part is D-Tree, also known as Decompose Tree. It is a space partitioning technique to access data related to moving objects, notes spatial points along with temporal values and is linked with a hash table to update on continuous basis. TB*-tree is another part to index history trajectory of moving objects. Another structure named as NT-Tree is the third part and its role is to answer queries related to future location of moving objects.

B. ST-Index[2]

This is a trajectory oriented access method and consists of three components. These three components are spatial index, time list and temporal index. This structure divides a day into different time slots for efficiency purposes and then creates a B-tree on basis of every small time slot. A spatial index is related to every leaf node of the structure and is based on re-segmented network of roads. A time list is maintained for every leaf node of the structure which speeds up the computations related to probable-reachable area, because the application requires finding the trajectories to confirm the probability of reach ability.

III. SPATIO-TEMPORAL SEARCH

There is a variety of queries dependent on spatio-temporal data including the search queries which need to find for either spatio-temporal events or objects which happened or were present at particular spatial location at given time. There might be requirements of different applications in which there can be queries about spatio-temporal objects or events in given range of spatial coordinates at particular time or in given range of spatial coordinates in given time window, and a lot more.

In this research work, the authors have explained the search algorithms which fall under first category which is to find for either spatio-temporal events or objects which happened or were present at particular spatial location at particular queried time.

Revised Manuscript Received on July 20, 2019.

Sumeet Gill, Department of Mathematics, M. D. University, Rohtak, Haryana, India. drsumeetgill@gmail.com

Meenakshi, Department of Mathematics, M. D. University, Rohtak, Haryana, India. mshthebest@gmail.com

DOI: 10.35940/ijitee.J1080.08810S19

The work can be extended for many other related and significant queries including search for N-nearest neighbors, nearest neighbor, objects within given range of particular spatial coordinated and many more depending on requirements of different applications.

A. Organizing Duplicate Spatio-Temporal Data

In past research work proposed by different researchers, it is observed by authors that all the records related to duplicate spatial or spatio-temporal keys are not stored in the indexing structures. Lots of the work either not includes duplicate spatial or spatio-temporal data or keeps quiet on the ways to store and access multiple spatio-temporal records with same spatio-temporal key. [3] manages the duplicate spatial keys is the same way as it handles keys which are smaller to it. In one other work of research proposed in [4], if there is a duplicate spatial key, the address of previously existing node with same key is returned back without any update to the module calling the current module. The researchers in [5] first clean the dataset and delete duplicate spatial records and process rest of the dataset for processing and analysis. Most of the previously proposed work is based on B-tree or R-tree. The authors are explaining a novel indexing structure to store and access data with duplicate spatio-temporal keys. The structure is based on k-d tree and hash table.

B. Searching for objects at particular spatial location at particular time

In this section, the authors explain an algorithm to find the spatio-temporal objects at queried spatial location at queried temporal value in a structure which is capable to store, manage and query spatio-temporal data with duplicate keys.

Algorithm 01: Searching spatio-temporal data with duplicate keys

```
START

IF k-dSTHashRoot IS NULL
THEN
    return NO_SPATIAL_DATA
END IF

Assign currentKDNode ← address of k-dSTHashRoot

IF value of coordinatesToSearch under currentDimension <  spatial value
    of currentKDNode under currentDimension
THEN
    return (CALL  algorithm with update parameters and left pointer of tree)
END IF

IF coordinatesToSearch =  currentKDNode coordinates, for all dimensions
THEN
     epochValue  ← GenerateEpochValue (timeStampToSearch)
     epochHashidSearch←GenerateEpochHashId (timeStampToSearch)

     IF record exist for epochHashidSearch
    THEN
            Traverse the hash table list at epochHashidSearch till end and
            while epochValue is less than equal to
                   epochValue of currentHashNode
            IF epoch value of  currentHashNode = epochValue
                Increment count and add currentHashNode to resultList
            END IF
                     Move to next hashNode
     ELSE
                return NO_TIMEHASH_DATA
     END IF

        return SUCCESS
END IF
return (CALL algorithm with update parameters and right pointer of tree)
END
```

IV. Experimental Analysis

The authors have experimented the algorithm given above for a synthetic datasets with spatio-temporal data records. We have processed a synthetic dataset related to crimes in India downloaded from Internet. The dataset is of 2,18,34,576 spatio-temporal records and modified by the researchers to reflect the requirements of the novel indexing structure and algorithm explained previously. For detailed analysis, the authors divided the main dataset in 08 sub-datasets (sub-dataset 01 – sub-dataset 08) of different sizes and queried for crimes at particular spatial location at particular queried time. The current research paper presents time analysis of queries for all 8 sub-datasets and visual output of queries executed for sub-datasets 01 through 03.

Fig. 1 visualizes the spatial location of crimes happened at queried location (71, 24) on Jan 28, 2019 at 23:10 in sub-dataset 01. The number of records found for queried data is 10 i.e. there were 10 crimes happened at rounded off latitude and longitude at given time. Similarly, Fig. 2 visualizes the crimes location (80, 30) on Feb 28, 2019 at 13:12 and reported 20 crimes as shown in Table 01. On similar basis, Fig. 3 shows the spatial location of 40 crimes which took place at queried location on Mar 03, 2019 at 03:30. Fig. 2 and Fig. 3 show the result of queries performed for sub-dataset 02 and sub-dataset 03 respectively.

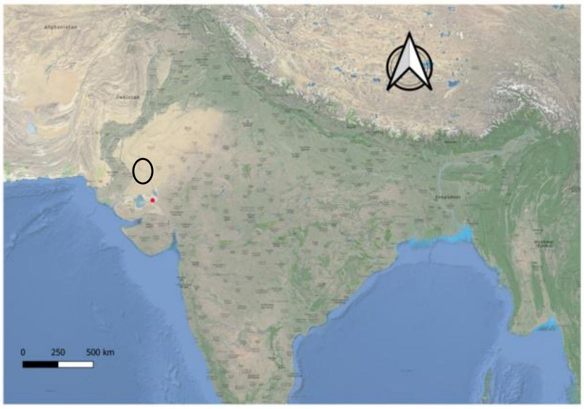

Fig. 1: Crime at Location (71, 24) at Jan 28, 2019 23:10

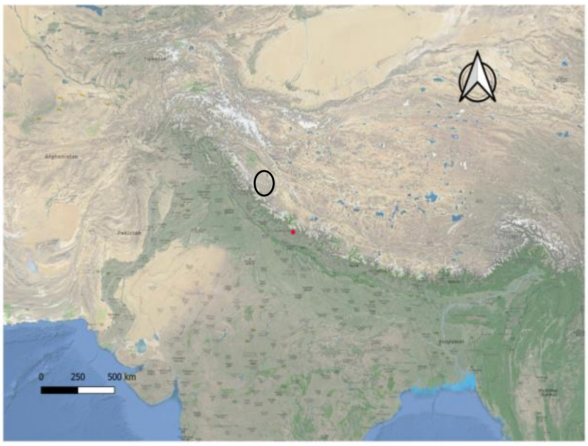

Fig. 2: Crime at Location (80, 30) at Feb 28, 2019 13:12

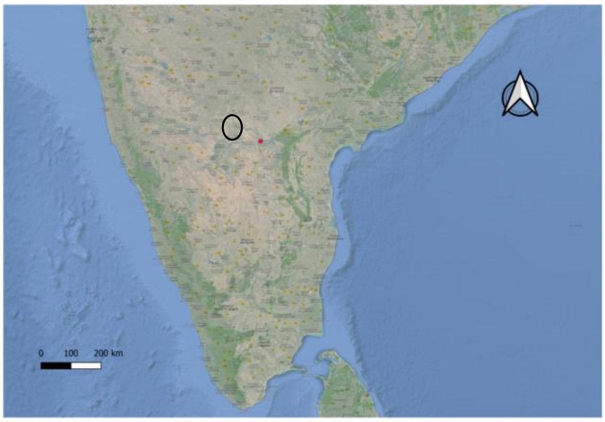

Fig. 3: Crime at Location (78, 16) at Mar 03, 2019 03:30

Table 01 shows the figures related to the performance analysis of search algorithms for different crime sub-datasets 01- 08. It shows the number of records in every sub-dataset, time taken (μs) by two different algorithms, and the number of spatio-temporal records reported as crime locations for queried spatial and temporal values. It is very much clear from the table that algorithm given to organize duplicate spatio-temporal data is many times faster as compared to other brute force method.

Table-I: Performance Analysis of Search Algorithms

Crime Dataset (Number of records: 2,18,41,920) **(Divided into Sub-Datasets of different sizes)**				
Dataset	Number of Records	Brute Force Method Search Time (μs)	Structure to organize Spatio-temporal data Search Time (μs)	Number of Spatio-temporal Records Found
Sub-Dataset 01	75840	3480	11	10
Sub-Dataset 02	151680	7992	19	20
Sub-Dataset 03	303360	16736	53	40
Sub-Dataset 04	606720	34366	129	80
Sub-Dataset 05	1213440	69578	201	160
Sub-Dataset 06	2426880	140092	384	320
Sub-Dataset 07	7280640	421899	1137	960
Sub-Dataset 08	21841920	1267533	3609	2880

The researchers did experimental analysis on 8 sub-datasets of different sizes. The size of sub-datasets is increased at each level and then the searching efficiency of algorithms is compared. Two algorithms, one based on brute force method and other based on the structure to organize spatio-temporal data are being compared. As observed from Table 01, showing performance analysis of search algorithms, the algorithm based on structure to index spatio-temporal data outperforms brute force method for each and every level. For the sub-dataset 01 consisting of 75840 records, both methods reported 10 records for queried time and location, but, where the time taken by algorithm based on brute force method is 3480 μs, it is just 11 μs when method based on spatio-temporal index structure is used. Similarly, for rest of the datasets, as number of records increase, the time taken by brute force method increases exponentially, but it remains very less when other algorithm is tested. Fig. 4 presents the analysis of search algorithms graphically.

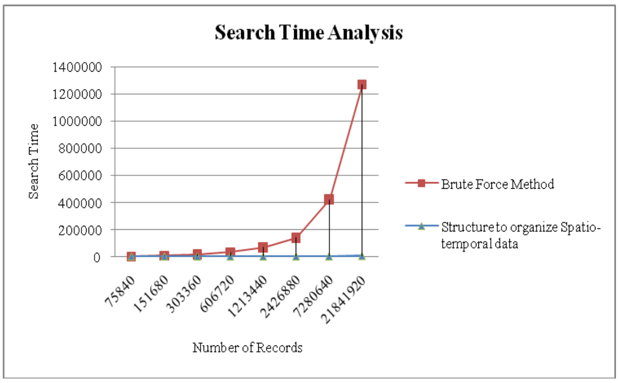

Fig. 4: Performance Analysis of Brute Force Method and Structure to organize duplicate Spatio-temporal Data

When the number of spatio-temporal records increase up to 21841920 records in sub-dataset 08, and the authors queried for some spatio-temporal data, spatio-temporal index structure based algorithm took only 3609 μs as compared to brute force based method, which took 1267533 μs. However, both methods reported that there were 2880 crimes for queried particulars. The results make it very much clear that method based on structure to index spatio-temporal data is far better as compared to methods based on brute force and repots the results in very less time.

V. Conclusion

In this research work, the researchers discussed structure to organize duplicate Spatio-temporal Data. It is based on structures: k-d tree and hash table. The researchers explained the algorithm to search for spatio-temporal objects at queried time and location and conclude that the proposed structure can efficiently organize spatio-temporal data and report the results of spatio-temporal query in very less time as compared to brute force method of searching for spatio-temporal data. The authors used the crime spatio-temporal dataset for this research work and analyzed the results but it can be used for any other spatio-temporal dataset also. Next, the work can be extended to find for spatio-temporal objects in queried range and time period.

REFERENCES

1. John Ayeelyan, Sugumarn Muthukumarasamy, Rengan Sivagurunathan Rajesh (2017). DTNH Indexing Method: Past Present and Future Data Prediction for Spatio-Temporal Data. *International Journal of Intelligent Engineering and Systems*, 10(3), pp.426-434
2. Guojun Wu, Yichen Ding, Yanhua Li, Jie Bao, Yu Zheng, and Jun Luo (2017). Mining Spatio-Temporal Reachable Regions Over Massive Trajectory Data. *IEEE 33rd International Conference on Data Engineering (ICDE)*, pp.1–12
3. Brown R.A.L. (2015). Building a Balanced k-d Tree in O (*kn* log *n*) Time. *Journal of Computer Graphics Techniques*, 4, pp. 50-68.
4. Bentley J. L. (1975). Multidimensional Binary Search Tree used for Associative Searching. *Communications of ACM*, 18, pp 509-517.
5. Friedman, J. H., Bentley J. L. Finkel R.A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions Mathematical Software*, 3(3), pp. 209-226.

AUTHORS PROFILE

Sumeet Gill has done Ph. D in Computer Science. He has taught in many reputed technical institutes and has more than 16 years of experience in the field of System Security and Artificial Intelligence. His research papers have been published in different Journals of International/National repute and the proceedings of the National/International Conferences. He has delivered invited talks and chaired sessions in various conferences. Presently, he is working with Maharshi Dayanand University, Rohtak, Haryana as an Associate Professor.

Meenakshi got Master Degree in Computer Applications and M.Tech in Computer Science. Then she completed her M.Phil in computer science. She has worked with Bharti Telesoft (Comviva Technologies), Okhla, Delhi for approx. 3.5 years as software developer and now working with Maharshi Dayanand University, Rohtak, Haryana as an Assistant Professor. She is also pursuing her Ph. D. in Computer Science from the same university.