American Journal of Applied Sciences, 10 (3): 195-208, 2013
ISSN: 1546-9239

doi:10.3844/ajassp.2013.195.208 Published Online 10 (3) 2013 (http://www.thescipub.com/ajas.toc)

An Efficient Algorithm for Mining Spatially Co-located Moving Objects

[1]Manikandan, G. and [2]S. Srinivasan

[1]Department of Computer Science and Engineering,
Sathyabama University, Chennai, India
[2]Department of Computer Science and Engineering, Anna University of Technology Madurai, India

Received 2012-04-24, Revised 2012-06-15; Accepted 2013-04-13

ABSTRACT

Mining co-location patterns from spatial databases may disclose the types of spatial features which are likely located as neighbors' in space. Accordingly, we present an algorithm previously for mining spatially co-located moving objects using spatial data mining techniques and Prim's Algorithm. In the previous technique, the scanning of database to mine the spatial co-location patterns took much computational cost. In order to reduce the computation time, in this study, we make use of R-tree that is spatial data structure to mine the spatial co-location patterns. The important step presented in the approach is that the transformation of spatial data into the compact format that is well-suitable to mine the patterns. Here, we have adapted the R-tree structure that converts the spatial data with the feature into the transactional data format. Then, the prominent pattern mining algorithm, FP growth is used to mine the spatial co-location patterns from the converted format of data. Finally, the performance of the proposed technique is compared with the previous technique in terms of time and memory usage. From the results, we can ensure that the proposed technique outperformed of about more than 50% of previous algorithm in time and memory usage.

Keywords: Spatial Data Mining, Co-Location Patterns, Minimum Support, Minimum Bounding Rectangle, FP Tree, Vehicle Movement Data

1. INTRODUCTION

An automated discovery of spatial knowledge is required because of the fast expansion of spatial data and extensive use of spatial databases. Nowadays, the spatial data mining turn out to be more eminent and stimulating for the reason that abundant spatial data have been stored in spatial databases. Spatial data mining (Mary and Kumar, 2012) is defined as the process of drawing out the interesting and previously unknown but probably relevant patterns from spatial databases. The mining of meaningful patterns from spatial datasets is more knotty than mining the analogous patterns from conservative numeric and categorical data (Kumar *et al*., 2012a), due to the difficulty of spatial data types, spatial relationships and spatial autocorrelation. In various applications, spatial patterns have excessive demand. To determine the spatial co-location patterns in wide applications (Yoo and Bow, 2011) is the main goal of spatial data mining. For both positive and negative association rules, spatial co-location and de-location patterns are identical. The subsets of Boolean spatial feature types are depicted by spatial co-location patterns and its instances are usually positioned in close geographic proximity Spatial de-location patterns extend the conservative spatial associations in order to include an association rules in the form of A→¬B, which denotes that B will not exist nearby A. In some spatial problems, these association rules are well-organized in discovering useful and previously unknown concealed information and also very advantageous (Saranya and Hemalatha, 2012).

Discovering spatial co-location patterns from respective databases is the primary job of spatial data mining in numerous applications (Kumar *et al*., 2012c) and such co-

Corresponding Author: Manikandan, G., Department of Computer Science and Engineering, Sathyabama University, Chennai, India location patterns depict the subsets of spatial features whose objects are typically located in close geographic proximity. For example, the co-location patterns are drawn in the areas like symbiotic species in ecology such as the Nile crocodile and Egyptian plover, frontage roads and highways in metropolitan road maps and co-located services often requested and located together from mobile devices (e.g., PDAs and cellular phones) in location-based services (Huang and Zhang, 2006; Kim *et al*., 2012). In 2D or 3D metric space, the Boolean spatial events or features indicate the presence or absence of geographic object types at a variety of locations, for e.g., the surface of the Earth. The examples of Boolean spatial events (Shekhar *et al*., 2011) are business types, mobile service request, ailment, offense, weather, plant species and more. A generate-and-test process is carried out in many prior co-location pattern discovery techniques, i.e., generating and testing each candidate to identify whether it is a co-location pattern (Kumar *et al*., 2012b; Xiao *et al*., 2008).

The spatial data mining and spatial analysis methods in spatial database systems are playing a prominent role in mining the important but hidden patterns in spatial datasets of increasing size and complexity. For arise of challenges faced in the co-location rule detection, there are two main reasons (Yoo *et al*., 2005; Shekhar and Huang, 2001) (i) since the instances of spatial features are located in a incessant space and share neighbor relationships, it is difficult to decide co-location instances. Therefore, to discover the co-location instances a large portion of the computation time is consumed. (ii) For co-location pattern mining, it is unable to reprocess association rule mining algorithms since there are no pre-defined transactions in a number of spatial datasets of escalating size and intricacy. On the other hand for spatial datasets, the similar shift of pattern in spatial co-location mining becomes very complicated because of the lack of a transaction concept, which is very important in frequent pattern description and its mining algorithms. Of any co-location mining algorithm neighborhood (co-location row instance) enumeration is a primary challenge and a key element (Yoo and Shekhar, 2006). To deal with this problem a combinatorial approach like apriori or a geometric technique such as spatial-self-join is used. As a smart group enumeration problem from a graph the combinatorial approach depicts the problem, based on the annotation of neighbors. Whereas, a geometric spatial join approach using a plane sweep technique scans the original space and halts at anchor points to accumulate neighborhood particulars. At system level both techniques may utilize optimizations by the spatial database methods such as spatial indexes (Zhang and Gruenwald, 2012; Kumar *et al*., 2012b; Celik *et al*., 2007).

We have proposed an innovative algorithm in this study for efficaciously drawing out the spatially co-located moving objects from the spatial databases. The designed statistic measure supports well to mine the appropriate information rather than the measure represented in the literature namely, popularity measure and conditional threshold. Now, for the proposed algorithm that comprises moving object_id and spatial information (x coordinate value and y coordinate value), spatial database is the input. In order to make the area we split the area by apply the grid and the size of the grid is defined by the user. From particular grid, the co-located patterns are mined, which contains the number of vehicles that should gratify the given Minimum Bounding Rectangle (MBR). The size of the grid is elaborate in all sides in a unit level, if the number of vehicles not supported to the MBR value. Until it is supported by the (MBR) value, this process is repeated up to 3 times. At last, in each grid this process provides the list of vehicles. Then vehicles in the each grid are taken and arranged by a list, namely vehicle list from where we remove the de-location of vehicle by applying the minimum support of vehicle, the value of minimum support should be less than the number of grids in spatial data. Ultimately, by using the FP growth algorithm the co-located patterns are mined.

1.1. Literature Survey

From spatial databases numerous research works exist in the literature focus on the subject of mining spatial co-location patterns. Nowadays, the developing of methods for co-location pattern mining has drawn a great concentration in real life applications. In this topic, some of the most recent literature works are portrayed here. In large spatial datasets the significant co-location and de-location patterns have been mined using a Co-location and De-location patterns Mining algorithm (CODEM), proposed by Wan *et al*. (2008). Here, by means of k-Nearest Features (k-NF) the spatial close/separate relationships of co-location/de-location patterns in spatial datasets have been analyzed. To decide the close/separation relationship between this feature and other features the k-NF set of one feature type's instances have been employed. Subsequently, by applying a correlation checking operation the irrelevant patterns have been filtered. In addition, a grid index method has been engaged to speed up the process of k nearest features query and a Total support tree, called T-tree, structure has also been used to decrease the candidate frequent and infrequent item sets. In the end, the patterns have been generated effectively. Experimental results have revealed that the proposed algorithm was very potent in mining those patterns and its time complexity was O (n).

A technique for discovering the co-location patterns in Sloan Digital Sky Survey (SDSS) data has been proposed by Al-Naymat (2008). 3.6 TB of data was presented in SDSS Data Release 5 (DR5). Due to the presence of such giant amount of useful data, there is a possibility for the application of data mining methods to produce interesting information. The shortage of data in an appropriate format is the main reason for the scarcity of such data mining applications in SDSS. A procedure has been given to acquire more types of galaxy from an available attributes and the data has been converted into maximal cliques of galaxies that has been then used as transactions for data mining applications. The maximal cliques from giant spatial databases have been mined by using the proposed Grid Clique algorithm. NP-Hard represents the general problem of mining a maximal clique from a graph. Experimental results have demonstrated that the proposed Grid Clique algorithm was proficient in generating all maximal cliques in the SDSS data and allows the discovery of relevant co-location patterns.

Wang *et al*. (2008) have investigated the basis of co-location patterns mining and the three different types of co-location patterns mining algorithms developed in recent years. Also, a CPI-tree (Co-location Pattern Instance Tree) structure based join-less scheme has been proposed for drawing out the co-location patterns. The spatial neighbor relationships were described by the CPI-tree. It is possible to create all co-location table instances swiftly with the help of CPI-tree. The accuracy and competence of the proposed scheme have been proved by them. Eventually, experiments conducted on simulated datasets and a real world dataset have exhibited that the proposed algorithm was computationally more competent.

The conservative algorithms used for co-location patterns mining can only determine non-negative co-location patterns. However negative co-location patterns are strong negative associated, its contribution index are below a minimum occurrence threshold and occasionally it holds many important information. Jiang *et al*. (2010) have depicted the concept of the negative co-location patterns. Based on the assessment of the relationship between negative and non-negative participation index, the techniques for negative participation index calculation and negative patterns pruning approaches have been developed. By using the proposed techniques, both positive and negative co-locations have been mined effectively. To study the applications of the proposed algorithm the plant datasets of the “Three Parallel Rivers of Yunnan Protected Areas” have been employed. Finally, by an wide experimental analysis the effectiveness of the algorithm has been exposed.

Yoo and Shekhar (2006) have proposed a join-less scheme for effective co-location pattern mining. Here, as an alternative of using a precious spatial or instance join operation, an instance-lookup approach has been engaged by the join-less co-location mining algorithm for discovering the co-location instances. In discovering co-location rules, the proposed join-less algorithm was precise and robust. Additionally, for spatial data that were clustered in neighborhood areas a partial join technique has been defined. In order to differentiate the performance superiority zones of the join-less technique and the partial join technique with the prior join-based co-location mining approach and also their computational complexities have been compared an algebraic cost model has been developed. It has been found that the proposed scheme was more competent than the join-based approach and was more scalable in dense data, during the experimental assessment using artificial and true datasets.

Wang *et al*. (2009) have proposed a technique based on order-clique for discovering maximum co-locations. At this point, for achieving high effectiveness of the proposed technique two different processes have been carried out. Primarily, to determine candidate maximum co-locations and co-location instances, the spatial neighbor relationships and the size-2 prevalence co-locations have been compressed into extensive prefix-tree structure that facilitates the order-clique-based technique. Secondly, the co-location instances need not to be stored after calculating some features of the respective co-location that considerably diminishes the execution time and space essential for mining maximal co-locations. Experiment results have exposed that the performance of the technique was very effective in extracting both long and short co-location patterns, as well as faster than the join-based technique and the join-less technique.

Qian *et al*. (2009) have discussed that the existing algorithms depend on while making decisions for discovering co-location patterns with dynamic neighborhood constraint. They have considered the mining task as an optimization problem according to this. Furthermore, for discovering the co-location patterns with dynamic neighborhood constraint a greedy algorithm has been developed. Compared to that of the existing algorithm on determining co-location patterns along with the consideration of the distribution of dataset experimental analysis on a valid dataset have exposed that the algorithm was advanced.

Wan and Zhou (2008) have proposed a k-Nearest Feature (k-NF)-based co-location pattern mining algorithm. Now, by means of the k-NF set of a spatial feature's instances, the spatial relationship between this feature and any other feature has been analyzed. In addition, the co-location patterns in huge spatial datasets have been found via the proposed k-NF-based CO-location pattern Mining algorithm in T-tree structure (short for KNFCOM-T). The experimental results have exposed that the KNFCOM-T algorithm was more robust and competent and its complexity was O (n).

1.2. Problem Description and Definition

Given a set of vehicles V = $\{v_i\}$, $1\leq i\leq l$ where l is the maximum number of vehicles in the spatial. Each vehicle having the value of x coordinate value and y coordinate value $v_i = \{x, y\}$. The total spatial area is divided into grids based on the value of P, G = x/p,y/p where P is value which is use for divide the spatial area in x coordinate and y coordinate, here the grids are used to cover the all the places in spatial area. After that the spatial containing the number of grids G = $\{g_i\}$, where $1\leq i\leq k$ here k is the total number of grids in the spatial. Each of the grid contain some number of vehicles if the number of vehicles $[n(v_i)_{gi}]$ is below minimum bounding rectangle $[n(v_i)_{gi}]$<mbr then size of the grid get increase by single unit in all direction g_i = x+1,y+1,-x,-y-1 the increment process is repeated up to three times to satisfy the MBR condition. The MBR function helps to confirm the entire grid g_i with particular amount of vehicle else there is no need to consider the corresponding grid g_i. Find the vehicles v_i in every grid g_i in the spatial data make it as list namely vehicles list V_{hit} then count the number for each of vehicles in the list. In order to find the co-location patterns we need to remove the de-location vehicles from V_{hit} for that we introduce a function minimum support m_s, the value of minimum support m_s is less than the total number of grids in spatial V` $m_{s<}k$. Each vehicle has different count values $[n(v_i)]$ in the V_{list} because of the same vehicle repeat in the different grids due to the expansion of the grid to satisfy the *MBR*. If any vehicle has the count value below minimum support $[n(v_i)]<m_s$then the corresponding vehicles is removed from 0_{list}. At last the modified vehicles list $m[V_{list}]$ is the input for the FP Tree and from the FP Tree the co-location patterns are extracted.

Definition 1 (data). Given a set of vehicles V $=\{v_i\}1\leq i\leq I$ where I is the macimum numberof vehicles in the spatial and the each vehicle having the location of the spatial value such as x coordinate value and y coordinate value $V_i= \{x,y\}$.

Definition 2 (Co-location pattern). A co-location pattern is a set of spatial features with the neighborhood and the node membership function of this set is greater than the user specified minimum conditional threshold.

Definition $3_{mbr.}$ The Minimum Bounding Rectangle (MBR), also known as bounding box or envelope, is defined as that the maximum vehicles vehicle within its 2-D (x, y) coordinate system belonging to defined conditions. Here, the condition to extent the rectangular box is defined based on the minimum number of events bounded within the rectangle and the number of extension carried out to satisfy the minimum bounding condition.

Definition 4 (vehicle list). The vehicle list v_{list} have list of vehicles that are taken from grid g_i after it satisfied the MBR. The modified vehicle list m $[v_{hit}]$ derived from the vehicle list O_{list} from that some of the vehicles are eliminated which vehicles are not satisfied by the minimum support m_s, the value of m_s should be less than the total number of grids in the spatial m_s <k

1.3. Algorithm for Mining Spatially Co-Located Moving Objects

With the wide availability of GPS, wireless, telecommunication and Web technologies, massive amounts of object movement data have been collected from various moving object targets, such as animals, mobile devices, vehicles and climate radars. Analyze such data has deep implications in many applications, e.g., ecological study, traffic control, mobile communication management and climatologically forecast. In this study, we focus of our study on vehicle movement data analysis and examine the mining methods for discovery of various vehicle movement patterns. It is common that objects follow some regular movement patterns. For example, vehicles could have some daily behaviors between one area and its destination. With these types of vehicles, one might want to know the relationships among the individuals in particular area. One of the most useful tasks is to finding the group of objects that move together in particular location. By discover such type of clusters; one can detect the vehicle movements in an area. For finding the co-located patterns there are two major steps given below:

- Conversion of spatial data into vehicle list
- Mining of co-location vehicle from the vehicle list using FP tree (Manikandan and Srinivasan, 2012b)

1.4. Conversion of Spatial Data into Vehicle List

In day today life, vehicles have become important aspects in human life where each vehicle is manufactured for a particular purpose. In order to find the co-location patterns, we need to identify the all vehicles. For our convenience, we classify the vehicles into five classes; each class has some number of vehicles and each vehicle have their unique id. The input for the proposed algorithm is a spatial database that contains three fields such as, instance id (classes), the spatial information (location (x, y)) and moving object id (1, 2 and 3). Here the instances are used to classify the vehicles, the spatial information is used to represent the location of the corresponding vehicle and the object id is used to find out the vehicle.

An illustrated example for each class is shown in **Table 1** that is an input to the proposed algorithm. The ultimate aim is to find the spatially co-located moving objects that will helpful in analyzing the different classes of vehicles moving in an identical locations. Spatial co-location pattern, {(Endeavor and Pajero, A/c Coach)} shows that luxury car and bus are moved in a same location. This will helpful in analyzing the market value of particular vehicles with respect to the location. In order find the mostly moved vehicles in a particular area, first we should identify the each vehicle classes and their types for the here the each class vehicles have the unique id that are describes in **Table 1**. In the spatial temporal data consists of longitude value and latitude value in order to find the location of the object. The following **Table 1** describes example of input representation. Here we have object id and x coordinate value and y coordinate value as given in the following **Table 2**. Based on this **Table 2** the objects are located at the following graph **Fig. 1**.

Here, we use the FP mining algorithm to mine the co-location patterns from the spatial data for that we need to preprocess the input spatial data and convert it into vehicle list. The FP tree processes the data is in the form of a list, with the help of R-Tree data structure we convert the spatial data into the list of vehicle namely vehicle list. In order to convert the spatial data into vehicle list the following steps are required:

- Cover total spatial area by applying grids in spatial
- Incorporate the R-Tree data structure with the help of MBR in every grid
- Construct the modified vehicle list

1.5. Cover Total Spatial Area by Applying Grids in Spatial

The spatial moving objects (vehicles) are placed in a scatter way, in these types of vehicles, one might want to know the relationships among the individuals in particular area. One of the most useful tasks is to find the group of objects that moved together in particular location. In order to find the co-location vehicles we need to check all the locations in the spatial area for which we have adapted the grid function. The size of the grid g is depends on the value p, it can be given by the user. The value of p is use to divide the x coordinate value and y coordinate value of the spatial area, when the value of p get increase then the number of grids in the spatial area also increases and finally the grid g is plotted on the spatial area and it covers all the area of spatial.

1.6. Incorporate the R-Tree Data Structure with the Help of MBR in Every Grid

Each grid g_i in the spatial is treated as area, in that each grid is having some number of vehicles $[n(v_i)_{gi}]$, in order to find the co-location, we need high density area since we assign the minimum bounding rectangle mbr which is minimum number of vehicles in each grid mbr. Here the R-Tree (Francis and Thambidurai, 2007) data structure is use to satisfy the minimum bounding rectangle mbr. If the grid is not satisfy the mbr then the size of the grid is get increase g_i in all side in one unit, this process is repeat up to three times to satisfy the mbr. Here mbr is necessary to find the co-location of vehicles. If the grid g_i has the number of vehicles equals to zero then no need to extend the size of the grid and no need to consider the corresponding grid. We take the vehicles from grid and make the vehicles list V_{list} after the grid satisfied the mbr condition and the vehicles list V_{hits} consist of gird id and vehicles in the corresponding grid.

Example: In the following **Fig. 2** the first cell of the x coordinates has only two vehicles but here the assumption value of minimum bounding rectangle is 3 since we elaborate the space at one unit in both x coordinate and y coordinate to support the minimum bounding rectangle. This elaboration is repeated three times, until the vehicles satisfy the minimum bounding rectangle. After the elaboration of the border of the first cell we get the number of vehicles greater than the minimum bounding rectangle, the obtain vehicles from the first cell are A1, A2, C3, B1, A4, D2.

The following **Table 3** describes the vehicle list of the first row of a above spatial data.

Table 1. Input representation

Car class A	Bus class B	Bike class C	Goods carriers class D	Others class E
1. Maruthi and Omni	1. Govt. bus	1. Apache	1. Three wheeler	1. JCB
2. Swift and Indigo	2. Tour bus	2. Splendor	2. Four wheeler	2. Road roller
3. Accent and Honda city	3. Mini bus	3. Pulsar	3. Lorry	3. Tractor
4. Endeavor and Pajero	4. A/C coach	4. Yamaha	4. Taros	4. Container
5. BMW and Benz	5. Volvo	5. Karizma	5. Tanker	5. Transporter

Table 2. Consist of spatial data information

Classer	Vehicles id and location
Class A	A2 (0, 7, 2.1), A3 (1.3, 5.6), A3 (0.8,5.7) A4 (2.3, 1.5), A2 (6.2, 0.9), A2 (8.9, 7.2), A5 (2.2, 7.4), A2 (3.8, 6.4), A1 (0.2, 0.6), A1 (2.1, 3)
Class B	B1 (0.1, 7.2), B1 (0.7, 8.5), B1 (1.6, 8.8) B1 (2.4, 1.4), B3 (4.7, 0.6), B4 (4.6, 6.3), B3 (5.5, 6.5), B1 (5.5, 7.9), B2 (6.6, 8.0), B3 (9.6, 5.1)
Class C	C1 (0.4, 5.1), C2 (0.0, 5.9), C3 (0.5, 8.8), C3 (1.5, 1.0), C3 (1.0, 7.3), C1 (2.5, 5.5), C3 (2.8, 8.3), C2 (2.5, 2.3), C2 (2.5, 9.5), C3 (3.2, 3.7), C4 (6.6, 6.1), C3 (7.1, 4.4), C4 (8.3, 2.5), C5 (9.4, 1.0), C5 (9.4, 9.6),
Class D	D1 (2.2, 8.4), D2 (3.7, 2.8), D1 (3.8, 3.5),D2 (3.0,1.0), D1 (5.7,3.9), D3 (5.8, 6.9), D1 (6.0, 3.6), D4 (6.9, 1.9), D1 (6.3, 8.0), D5 (8.9, 3.3) D2 (9.9, 1.8), D2 (9.3, 7.9), D1 (9.1, 6.5)
Class E	E4 (5.9, 6.2), E1 (8.3, 7.5)

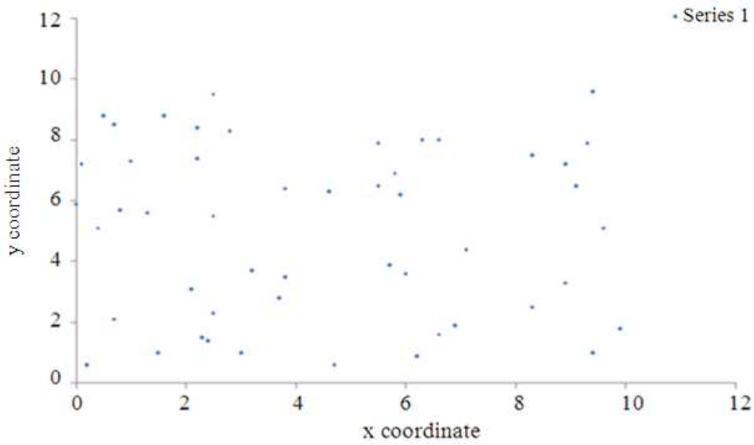

Fig. 1. Describes the location of the objects in the spatial temporal data

1.7. Construct the Modified Vehicle List

The vehicle list consist V_{list} of de-location patterns to remove the de-location pattern and make it as a new list namely modified vehicles list $m[v_{list}]$. In order to make it, here we introduce the function minimum support m_s the value of m_s is depends on the total number of grids k in the spatial data and it should not exceed the total number of grids $m_s<k$ in the spatial data. The modified vehicles list m $[v_{list}]$ is derived from the vehicles list. Each of the vehicles in the vehicle list having the count value of them, the modified vehicle list consists only those vehicles which satisfy the minimum support m_s.

For example with the help of the above **Table 3** we make the modified vehicles list $m[v_{list}]$ for that we count the number for each vehicle in the above **Table 3.** Here the count is A1:1, A2:3, A4:2, B1:2, B3:1, C3:1, C5:1, C4:3, D2:4, D4:2. From the above count value we set the minimum support to remove the de-located objects. Here the minimum support value is three, since here in above **Table 3** we are take the vehicle list only for the five grids, so the value of k is five and we select minimum support if these three the conditions is satisfied $m_s<k$. Those vehicles, whose count value is below the minimum support m_s value, are removed from the list. Removed vehicles are A1, A4, B3, C3 and C5.

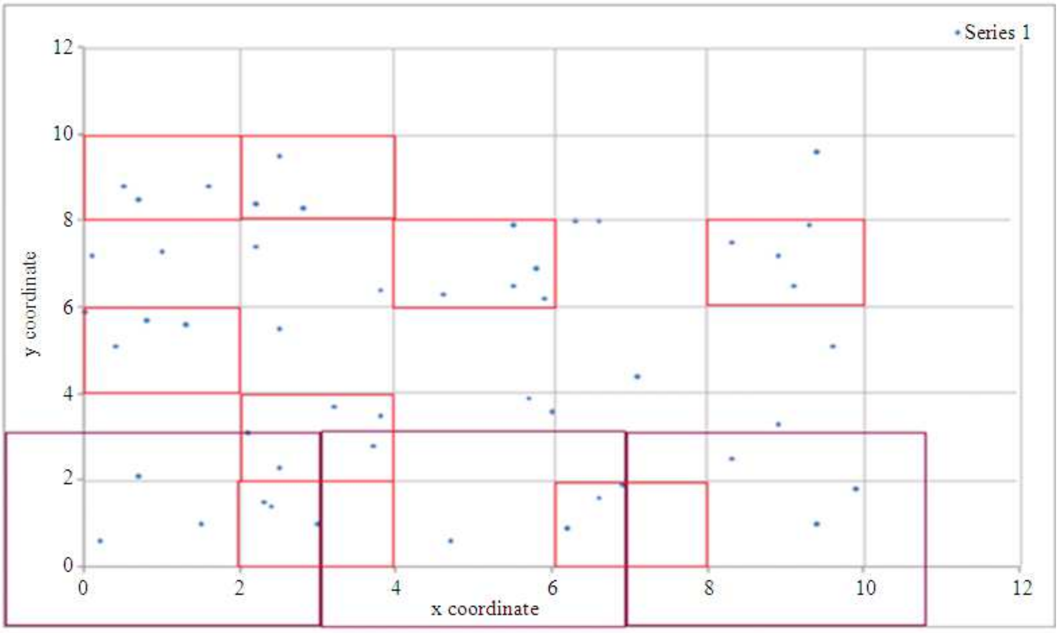

Fig. 2. Mapping of vehicle in each area

Table 3. Consist of list ofvehicles present in the each cell

Row 1	Vehicles in the cell
Cell 1	A1, A2, C3, B1, A4,D2
Cell 2	B1, A4, D2
Cell 3	D2, B3, D4, C4, A2
Cell 4	D4, C4, A2
Cell 5	D2, C4, C5

Table 4. Consist of objects which are support for minimum support value

Cell 1	D2, A2
Cell 2	D2
Cell 3	D2, C4, A2
Cell 4	C4, A2
Cell 5	D2, C4

The modified vehicle list of the **Table 3** is given in following **Table 4** is given below. The vehicles in the modified list are sorted based on the count value.

Pseudo Code

```
    Input: set of spatial objects
    Output: co-location of spatial objects
    Begin
1. Get input object $V = \{v_i\}$
2. Each object has $v_i = \{x,y\}$
3. Get the value of P
4. Apply grid $G = x/p,y/p$
5. Grid has $G = \{g_i\}$ $1 \le I \le k$,
6. Count $[n(v_i)g_i]$
7. Get the value of mbr
8. If $[n\ (v_i)_{gi}] <$ mbr
9. $g_i = x+1, y+1,-x-1,-y-1$
10. Go to step 8
11. If $[n\ (v_i)_{gi}]<0$
12. No need to expand $g_i$
13. Construct $V_{list}$
14. Count $[n(v_i)]$ from $V_{list}$
15. Get value of $m_s$, $m_s < k$
16. If $[n(v_i)]<m_s$
17. Remove corresponding $v_i$
18. Construct modified object list m $[V_{list}]$
19. Construct FP tree
20. Mine co-location pattern from FP tree
End
```

1.8. Mining of Co-Location Vehicle from Vehicle List Using FP Tree

The co-location vehicles are mined by the use of FP mining. The input of the FP mining is modified vehicle list. The co-location vehicles mining using the FP tree consist of two main steps that are:

- Construction of FP tree from the modified vehicles list
- Mining co-location patterns from FP tree
- Construction of FP tree from the modified vehicles list

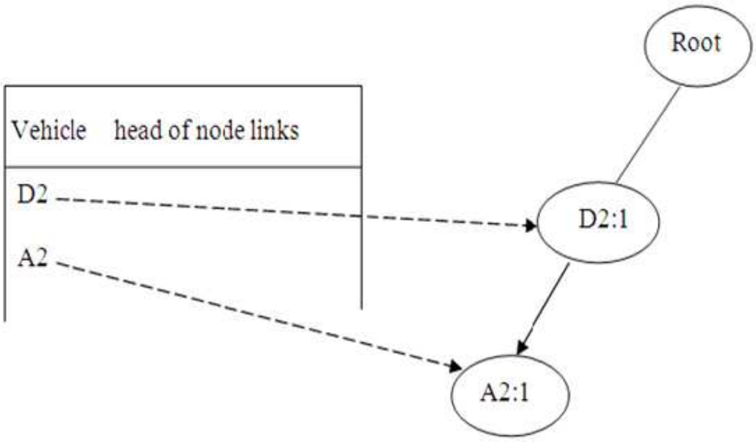

Fig. 3. Give you an idea about the initial step of the FP tree

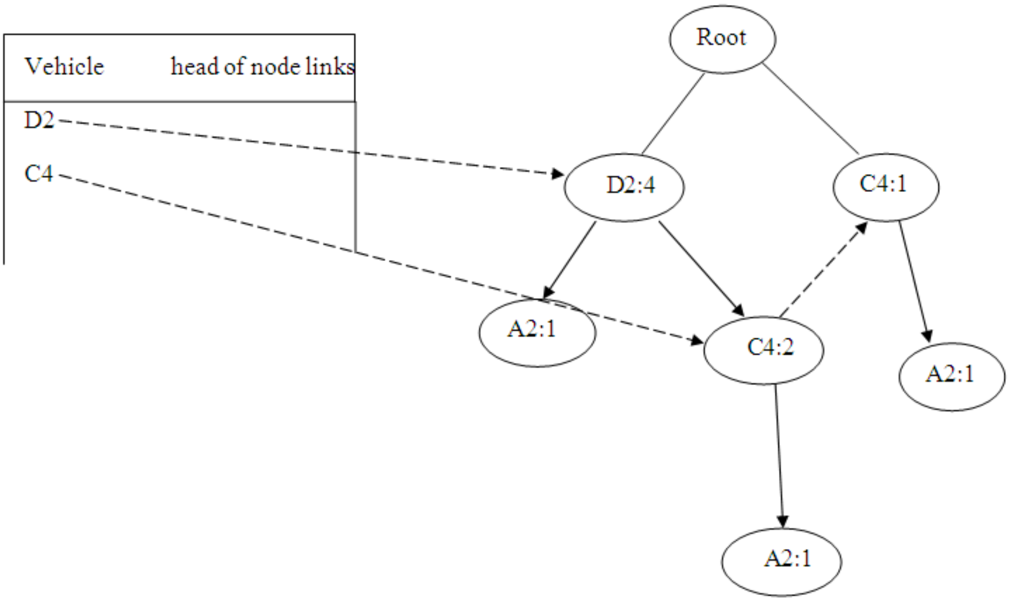

Fig. 4. Give you an idea about the final step of the FP tree

From the modified vehicles list m [v_{list}] we construct the FP tree in step by step procedure. The FP growth algorithm is constructed from this section the above **Table 4** is used for the FP growth algorithm. The following **Fig. 3 and 4** describes the FP growth algorithm.

1.9. Mining Co-Location Patterns from FP Tree

Once we construct the FP tree, the mining of frequent patterns from the compact tree arrangement is done by FP tree mining procedure defined in (Singh and Singh, 2012). FP-growth functions works in divide-and-conquer way. The first scan of the database extracts a list of frequent items in which the items are sorted by frequency- descending order. According to the frequency-descending list, the database is packed into a frequent-pattern tree, or FP-tree, which retrieves the item set association information. The FP-tree is mined by initially from every frequent length-1 pattern (as an initial suffix pattern), building its conditional pattern base (a "sub database", which includes of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then building its conditional FP-tree and executing mining recursively on such a tree. The pattern growth is obtained by the concatenation of the suffix pattern with the frequent patterns obtained from a conditional FP-tree. The FP-growth algorithm converts the problem of finding long frequent patterns to probing for shorter ones recursively and then concatenating the suffix. It uses the slightest frequent items as a suffix, offering good selectivity. Performance studies show that the method substantially decrease search time.

2. MATERIALS AND METHODS

The proposed algorithm for mining of spatially co-located patterns is programmed using Java (jdk 1.6). The experimentation has been carried out on a 2.9 GHz, dual core PC machine with 1 GB main memory running a 32-bit version of Windows XP. The performance of the proposed algorithm has been evaluated using the synthetic datasets. Here, we have generated a set of synthetic spatially co-located vehicle moving data sequence by a data generator similar in spirit to the IBM data generator designed for testing co-location pattern mining algorithms.

The performance of the proposed spatial co-location vehicle mining algorithm from the spatial datasets is evaluated by three standard evaluation measures. They are: (1) Number of generated co-location vehicles, i.e., the significant number of co-location vehicles generated based upon the given minimum support threshold, (2) Execution time, i.e., the time taken to execute the computer program and it characteristically depends with the input size and the 3) Memory usage, i.e., the memory utilized by the current jobs present in the particular system.

3. RESULTS AND DISCUSSION

The experimental results of the proposed algorithm for spatially co-located moving objects from the spatial databases are described here. The experimental results and analysis of the spatially co-located patterns of our proposed algorithm are done here.

3.1. Analysis of the Spatial Co-Location Vehicles Mining Algorithm with Input Sequence = 100

By analyzing the proposed spatial co-located vehicle mining algorithm with the help of the Synthetic datasets, we have utilized three different evaluation measures with diverse minimum support values. We have done the analysis part and plotted as a graph by computing the generated number of co-location vehicles, execution time and the memory usage with different minimum support. We have analyzed the results using synthetic datasets with input sequences = 100 and the plotted graphs is shown in **Fig. 5-7**. In the graph, we have found that the proposed algorithm effectively mines the co-location vehicles with the given input sequences. Here, the better results of our proposed algorithm are shown in **Fig. 5** in which the generated number of co-located vehicles gradually decreased when the minimum support value fell down. But in **Fig. 6** the corresponding execution time of the co-location vehicles mining algorithm gets slightly deviates in particular threshold and remains stable in other cases. The effective usage of the memory for mining the co-location vehicles in the proposed algorithm is shown in **Fig. 7**.

The following **Fig. 5** describes the number of co-location vehicles are occur for the previous algorithm (Manikandan and Srinivasan, 2012a) and proposed algorithm based on the minimum support values. While apply the previous algorithm to find the co-location vehicles, the results are slowly decreased when the minimum support values get increase. At the same time our proposed approach results is same for support value 0.8 and 1.0 afterwards the number of co-located vehicles get decrease for the support value 1.2 and it same for the minimum support value 1.4 and 1.6. The maximum number of co-location vehicles in the previous algorithm is 109 for the minimum support value 0.8 and the maximum number of co- location vehicles in the proposed algorithm is 34 for the minimum support value 0.8. The number of co-location vehicles of the previous algorithm is higher than the proposed algorithm for the input sequences 100.

The above **Fig. 6** describes the execution time taken for the previous algorithm (Manikandan and Srinivasan, 2012a) and proposed algorithm based on the minimum support values. When we use the previous algorithm to find the co-location vehicles, the execution time is fluctuate for each minimum support values. While using our proposed approach to find the co-location vehicles, the execution time is same for the minimum support value 0.8 and 1.0 and the execution time is slightly decreased when the minimum support value get increase. The minimum execution time for the previous algorithm 14281 ms for high minimum support value 1.6 and the minimum execution time for the proposed algorithm is 62ms for the high minimum support values 1.6. The execution time of the previous algorithm is higher than proposed algorithm. Our technique uses less amount of time to find the co-location vehicles when compared with the previous algorithm.

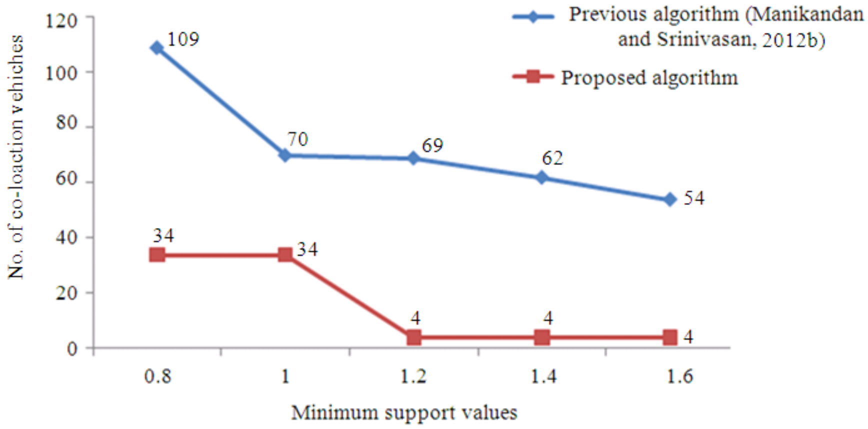

Fig. 5. No. of co-location vehicles of Synthetic dataset with input sequence = 100

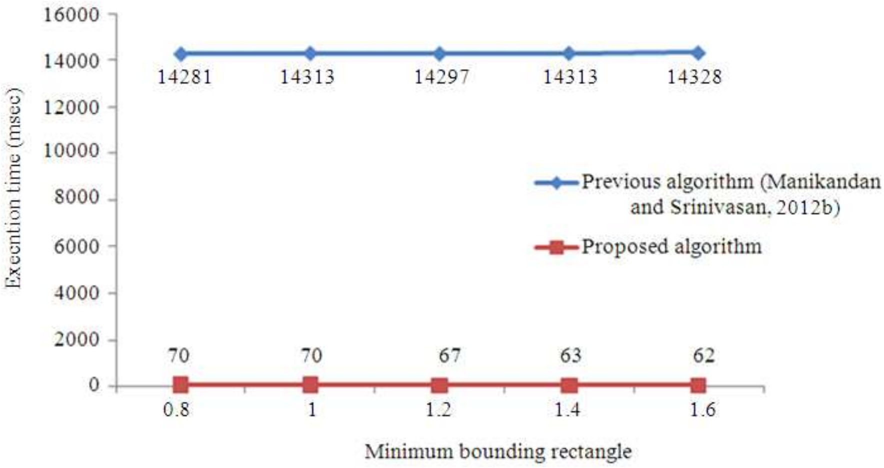

Fig. 6. Execution time of Synthetic dataset with input sequence = 100

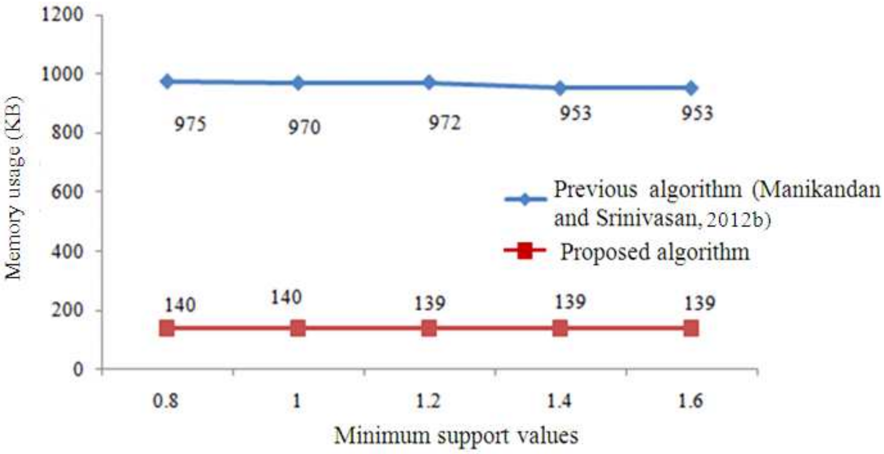

Fig. 7. Memory usage of Synthetic dataset with input sequence = 100

The above **Fig. 7** describes the need of memory for the previous algorithm (Manikandan and Srinivasan, 2012a) and proposed algorithm based on the minimum support values to find co-location vehicles. When we use the previous algorithm to find the co-location vehicles, the memory usage is fluctuate for first three minimum support values and same for next two minimum support values. While using our proposed approach to find the co-location vehicles, the memory usage is same for the first two minimum support value 0.8 and 1.0 and the memory usage slightly decrease and same for other three minimum support values. The maximum value of memory usage for the previous algorithm is 975 kb for the minimum support value 0.8 and the maximum memory usage for the proposed algorithm is 140 kb for the support values 0.8 and 1.0. The previous algorithm needs more memory space to find the co- location vehicles when compared with our proposed approach. When see the graph the memory usage of the previous algorithm and proposed technique is not affect by minimum support value.

3.2. Analysis of the Spatial Co-Location Vehicles Mining Algorithm with Input Sequence = 200

With the help of the Synthetic datasets, we have analyzed the proposed spatial co-located vehicle mining algorithm by three different evaluation measures with diverse minimum support values. We have done the analysis part and plotted as a graph by computing the generated number of co-location vehicles, execution time and the memory usage with different minimum support threshold. We have analyzed the results using synthetic datasets with input sequences = 200 and the plotted graphs is shown in **Fig. 8-10**. In the graph, we have found that the proposed algorithm effectively mines the co-location vehicles with the given input sequences. Here, the better results of our proposed algorithm are shown in **Fig. 8** in which the generated number of co-location vehicles gradually decreased when the support value fell down. But in **Fig. 9** the corresponding execution time of the co-location vehicle mining algorithm has some ups and downs in particular minimum bounding rectangle. The effective usage of the memory for mining the co-location vehicles in the proposed algorithm is shown in **Fig. 10**.

By seeing the following **Fig. 8** number of co-location vehicles occur in the previous algorithm is less than our approach for 200 input sequences. But the number of co-located vehicle result for 100 input sequences is vice versa, from this we conclude one thing when the number of input sequence is directly propos anal to the number of sequences. To find the co-location vehicle using the proposed algorithm the first two result is same for the support value 0.8 and 1.0 then the number of co-location vehicle is decrease massively and it same for the other three support values. The number of co-location vehicle for the previous approach is higher in the first support value and it decrease when the support value is increase. The maximum number of co-location vehicles in the previous approach is 1208 for the support value 0.8 and the maximum number of co-location vehicles for the proposed approach is 20439 for the input values 0.8 and 1.0. When the number of input is high means then the number of co-location vehicle is also high in the proposed approach. If the support value increase then the number of co-located vehicle is decrease.

The above **Fig. 9** describes the execution time taken for the previous algorithm (Manikandan and Srinivasan, 2012a) and proposed algorithm based on the number of support values. When we use the previous algorithm to find the co-location vehicles, the execution time is fluctuate for each support values. While using our proposed approach to find the co-location vehicles, the execution time is fluctuating for each support values. The minimum execution time for the previous algorithm 121062ms for support value 1.4 and the minimum execution time for the proposed algorithm is 248ms for the maximum support value 1.6. The execution time of the previous algorithm is higher than proposed algorithm. Our technique uses less amount of time to find the co-location vehicles when compared with the previous algorithm. When the support value increases consequently the execution time get reduce massively.

The above **Fig. 10** describes the need of memory for the previous algorithm (Manikandan and Srinivasan, 2012a) and proposed algorithm based on the number of support values to find co-location vehicles. When we use the previous algorithm to find the co-location vehicles, the memory usage is fluctuate for the all support values. While using our proposed approach to find the co-location vehicles, the memory usage is same for the first two support value 0.8 and 1.0 and the memory usage slightly increase and same for other three support values. The maximum value of memory usage for the previous algorithm is 2338 kb for the support value 0.8 and the maximum memory usage for the proposed algorithm is 221kb for the support values 0.8 and 1.0. The previous algorithm needs more memory space to find the co- location vehicles when compared with our proposed approach. From the graph, we see infer that the memory usage of the previous algorithm and proposed technique is not affected by minimum support value.

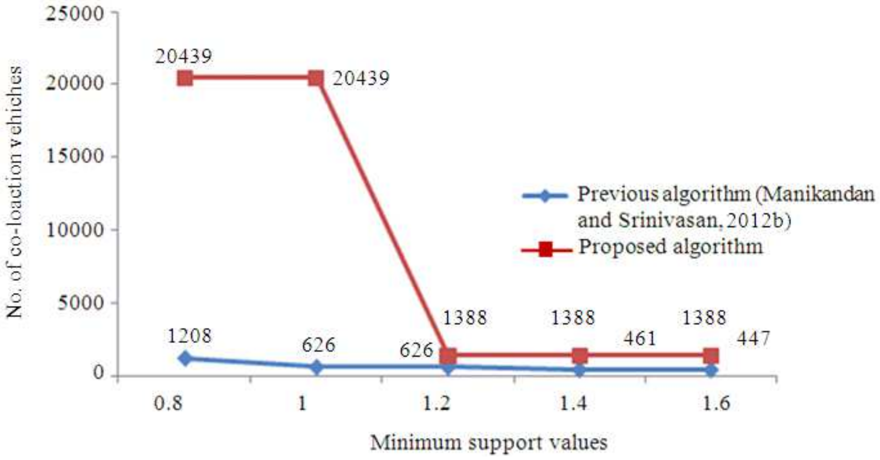

Fig. 8. No. of co-location vehicles of Synthetic dataset with input sequence = 200

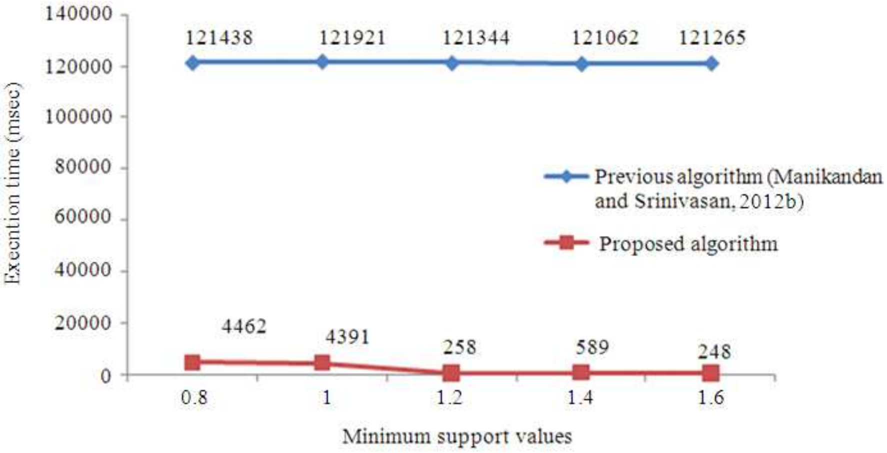

Fig. 9. Execution time of Synthetic dataset with input sequence = 200

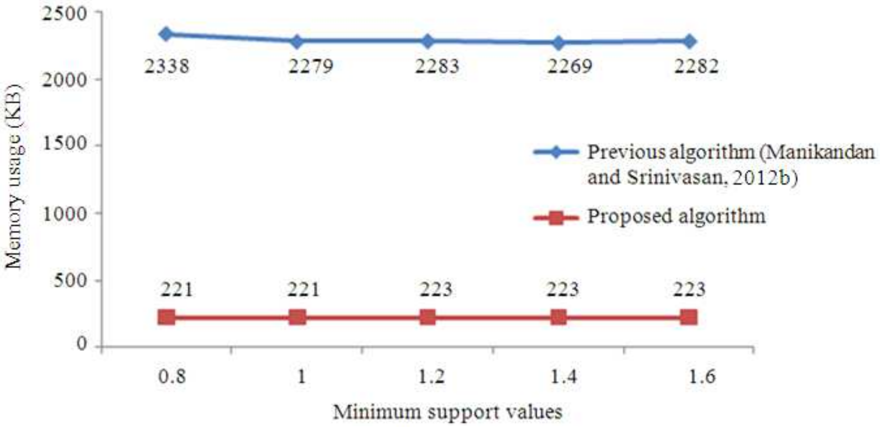

Fig. 10. Memory usage of synthetic dataset with input sequence = 200

4. CONCLUSION

We have presented an efficient algorithm for mining spatially co-located moving objects which materializes spatial neighbor relationship and reduces the computational cost extremely with aid of the well known FP Tree mining algorithm. In first we split the area by applying grids afterwards the minimum bounding rectangle is used to find groups of vehicles by elaborate the size of the grid. Makes the vehicles list of each grid afterwards remove the de-located vehicles with the help of minimum support and named that list as modified vehicles list. FP tree constructed for the modified vehicles list afterwards the co-located vehicles are mined from it. The spatially co-location mining algorithm efficient since the well known frequent vehicle mining is used here. Finally, we have generated the candidate co-location vehicles which satisfy the minimum bounding rectangle. We have carried out the experimental evaluation using the synthetic datasets and obtain the computation time is very less when compared with previous algorithm and our proposed method leads to reduce the memory usage extremely when compared with the previous algorithm. From the results, we ensured that the proposed technique outperformed of about more than 50% of previous algorithm in time and memory usage.

5. REFERENCES

Al-Naymat, G., 2008. Enumeration of maximal clique for mining spatial co-location patterns. Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, Mar. 31-Apr. 4, IEEE Xplore Press, Doha, pp: 126-133. DOI: 10.1109/AICCSA.2008.4493526

Celik, M., J.M. Kang and S. Shekhar, 2007. Zonal co-location pattern discovery with dynamic parameters. Proceedings of the 7th IEEE International Conference on Data Mining, Oct. 28-31, IEEE Xplore Press, Omaha, NE., pp: 433-438. DOI: 10.1109/ICDM.2007.102

Francis, F.S. and P. Thambidurai, 2007. Efficient physical organization of R-trees using node clustering. J. Comput. Sci., 3: 506-514.

Huang, Y. and P. Zhang, 2006. On the Relationships between Clustering and Spatial Co-location Pattern Mining. Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, Nov. 13-15, IEEE Xplore Press, Arlington, VA., pp: 513-522. DOI: 10.1109/ICTAI.2006.91

Jiang, Y., L. Wang, Y. Lu and H. Chen, 2010. Discovering both positive and negative co-location rules from spatial data sets. Proceedings of the 2nd International Conference on Software Engineering and Data Mining, Jun. 23-25, IEEE Xplore Press, Chengdu, pp: 398-403.

Kim, S.K., J.H. Lee, K.H. Ryu and U. Kim, 2012. A framework of spatial co-location pattern mining for ubiquitous GIS. Multimedia Tools Applic. DOI: 10.1007/s11042-012-1007-2

Kumar, G.K., P. Premchand and T.V. Gopal, 2012c. Mining of spatial co-location pattern from spatial datasets. Int. J. Comput. Applic., 42: 25-30. DOI: 10.5120/5836-7994

Kumar, M.R.P., K.S..Ranjith, B.K. Kumar and G.M. Yadav, 2012a. Analysis of Spatial Data Mining and Global Autocorrelation. Int. J. Adv. Res. Comput. Sci. Soft. Eng., 2: 519-523.

Kumar, N.S., V.S. Ramulu, K.S. Reddy, S. Kotha and M. Kumar, 2012b. Spatial data mining using cluster analysis. Int. J. Comput. Sci. Inform. Technol., 4: 71-77.

Manikandan, G. and S. Srinivasan, 2012a. Mining spatially co-located objects from vehicle moving data. Eur. J. Sci. Res., 68: 352-366.

Manikandan, G. and S. Srinivasan, 2012b. Mining of spatial co-location pattern implementation by FP growth. Ind. J. Comput. Sci. Eng., 3: 344-348.

Mary, S.A.L. and K.R.S. Kumar, 2012. A density based dynamic data clustering algorithm based on incremental dataset. J. Comput. Sci., 8: 656-664. DOI: 10.3844/jcssp.2012.656.664

Qian, F., Q. He and J. He, 2009. Mining spatial co-location patterns with dynamic neighborhood constraint. Mach. Learn. Knowl. Discovery Databases, 5782: 238-253. DOI: 10.1007/978-3-642-04174-7_16

Saranya, N.N. and M. Hemalatha, 2012. Estimation of evolutionary optimization algorithm for association rule using spatial data mining. Int. J. Comput. Applic., 51: 1-15. DOI: 10.5120/8019-8204

Shekhar, S. and Y. Huang, 2001. Discovering spatial co-location patterns: A summary of results. Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, (STD' 01), Springer-Verlag London, UK., pp: 236-256.

Shekhar, S., M.R. Evans, J.M. Kang and P. Mohan, 2011. Identifying patterns in spatial information: A survey of methods. Data Min. Knowl. Discovery, 1: 193-214. DOI: 10.1002/widm.25

Singh, S. and J. Singh, 2012. Association rules and mining frequent itemsets using algorithms. Int. J. Comput. Sci. Eng. Technol., 3: 370-373.

Wan, Y. and J. Zhou, 2008. KNFCOM-T: A k-nearest features-based co-location pattern mining algorithm for large spatial data sets by using T-trees. Int. J. Bus. Intell. Data Min., 3: 375-389. DOI: 10.1504/IJBIDM.2008.022735

Wan, Y., J. Zhou. and F. Bian, 2008. CODEM: A novel spatial co-location and de-location patterns mining algorithm. Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, Oct. 18-20, IEEE Xplore Press, Shandong, pp: 576-580. DOI: 10.1109/FSKD.2008.467

Wang, L., L. Zhou, J. Lu and J. Yip, 2009. An order-clique-based approach for mining maximal co-locations. Inform. Sci., 179: 3370-3382. DOI: 10.1016/j.ins.2009.05.023

Wang, L., Y. Bao, J. Lu and J. Yip, 2008. A new join-less approach for co-location pattern mining. Proceedings of the 8th IEEE International Conference on Computer and Information Technology, Jul. 8-11, IEEE Xplroe Press, Sydney, NSW., pp: 197-202. DOI: 10.1109/CIT.2008.4594673

Xiao, X., X. Xie, Q. Luo and W.Y. Ma, 2008. Density based co-location pattern discovery. Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Nov. 05-07, ACM Press, New York, USA., pp: 3-12. DOI: 10.1145/1463434.1463471

Yoo, J.S. and M. Bow, 2011. Mining top-k closed co-location patterns. proceedings of IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services, Jun. 29-Jul. 1, IEEE Xplore Press, Fuzhou, pp: 100-105. DOI: 10.1109/ICSDM.2011.5969013

Yoo, J.S. and S. Shekhar, 2006. A joinless approach for mining spatial colocation patterns. IEEE Trans. Knowl. Data Eng., 18: 1323-1337. DOI: 10.1109/TKDE.2006.150

Yoo, J.S., S. Shekhar and M. Celik, 2005. A join-less approach for co-location pattern mining: A summary of results. Proceedings of 5th IEEE International Conference on Data Mining, Nov. 27-30, IEEE Xplore Press, pp: 813-816. DOI: 10.1109/ICDM.2005.8

Zhang, J. and L. Gruenwald, 2012. Spatial Indexing of large-scale geo-referenced point data on GPGPUs using parallel primitives. Department of Computer Science, The City College of the City University of New York.