

Trajectory Pattern Mining with Multistage Spatial Partitioning

Manta Sanni and Saiful Akbar

School of Electrical Engineering and Informatics
Institut Teknologi Bandung
INDONESIA

Abstract: Most trajectory pattern mining techniques assume that the data to be analyzed contain complete and evenly distributed spatial and temporal information. However in reality, collected data may contain noise, missing or incomplete information, and uneven spatial resolution. In trajectory pattern mining methods, trajectory patterns are extracted by splitting spatial workspace into uniformly tiny sized squares, followed by determining popular cells which contain many data points. Finally, a sequential pattern mining technique, e.g. MiSTA, is used to extract trajectory pattern. This research proposes non-uniform partitioning to handle uneven spatial distribution as modification towards the uniform spatial workspace division process. The proposed approach, named multistage spatial partitioning is developed based on point-region quadtree concept. The new partitioning method is conducted for preprocessing before applying MiSTA. As the result, using multistage spatial partition, MiSTA succeeds in uncovering more detailed and broader coverage patterns compared to uniform partitioning approach through a series of experiments.

Keywords: area partitioning, point-region quadtree, spatio-temporal data mining, trajectory pattern

1. Introduction

Technological advancements in telecommunication and positioning system yield massive amount of spatio-temporal data with great potential to be a source of knowledge in understanding various natural and social phenomena. To extract information and knowledge hidden inside spatio-temporal data, researchers have been developing data mining methods. One of them is aimed to search for trajectory patterns.

Trajectory means path followed by a moving object. One of the main challenges of trajectory pattern mining is spatial bias contained in spatio-temporal data, where semantically similar trajectories might be considered different due to insignificant spatial gaps. To overcome this problem, many methods apply a uniform grid partitioning technique, i.e. splitting the workspace into small cells. After spatial discretization is completed, popular cells are selected to become itemset candidates in the sequential pattern mining algorithm. The detail quality of extracted patterns relies heavily on the grid's cell size. If the spatial distribution of the data is uneven, wrongly selected cell size might obliterate important patterns.

The main contributions of this paper are:

- We propose an algorithm to exploit multistage spatial partitioning concept as spatial partitioning technique in trajectory pattern mining for spatially uneven distributed data.
- We provide experimental evaluation of frequent trajectory pattern mining performance that utilize the proposed algorithm.

2. Related Work

This section contains summarization of several works related to the topic of this paper as an introduction to the basic concept of trajectory pattern mining.

A. Trajectory Pattern Mining

Trajectory pattern refers to popular path that frequently selected by moving objects in similar temporal manner. Patterns can be used as a model to predict object trajectory in the future. Lee

Received: May 27th, 2016. Accepted: June 23rd, 2017

DOI: 10.15676/ijeel.2017.9.2.12

et al. state that there are two main factors in trajectory pattern: geographical location and transition times between locations [1].

Hwang et al. develop mining algorithm to group objects with similar movements [2]. Tsoukatos and Gunopulos propose a method to mine periodical spatial sequence using apriori [3]. Cao et al. suggest a method to search objects with trajectory similarities [4]. Jeung et al., Mamoulis et al., and Verhein et al. exploit association rule mining towards popular areas in trajectory data [5][6][7]. Lee et al., and Li et al. propose process frameworks to perform sub trajectory clustering by dividing trajectory into smaller segments, then grouping the segments based on their geometrical features [8][9]. Cao et al. assume the data in trajectory segments, then search for the patterns [10]. Kalnis et al. harness clustering by assuming trajectory pattern in the form of moving areas in time intervals [11]. Mamoulis et al. search for periodic pattern from the spatial density-based clustering result [6]. Giannotti et al. develop sequential pattern mining algorithm, MiSTA, to search for trajectory pattern from uniform grid spatial discretization result [12].

In order to describe movement sequence of objects, researchers develop various forms of trajectory pattern representation. Yoshida et al. propose *Delta Pattern* which consists of itemset sequence and transition time annotation from one itemset to the next [13]. Vautier et al. develop *Chronicles* to represent a set of temporal requirements among itemset or events [14]. Giannotti et al. suggest the application of *Temporally Annotated Sequence* or *TAS* for short, which consist of spatial sequences along with transition times as temporal annotation [12]. Wang et al. propose *Stay Time Sequence* that exploit the duration of object staying in a location before moving to the next location [15].

B. Regions of Interest

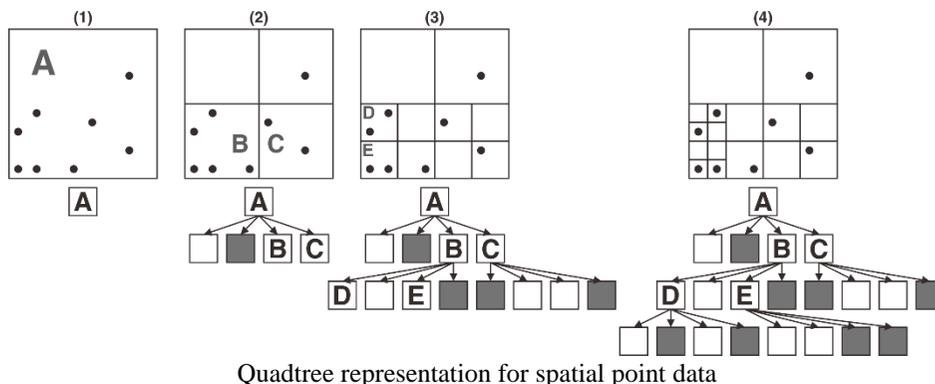
Regions of Interest (RoI) are locations from which trajectory patterns can be extracted. These regions have semantical meaning, e.g. administrative districts, or tourist attraction areas. RoI are resulted from a spatial workspace partition process. Spatial discretization is a way to simplify spatial information within data through a symbolization process. Ashok uses domain expert knowledge regarding area boundaries to manually divide spatial workspace [16]. For efficiency, uniform grid partitioning technique is applied in many RoI generation methods, such as in [12]. Spatial workspace is divided into uniformly sized cells, where the cell size is provided by the user as a parameter of granularity [15].

Kang states that the weakness of uniform grid partitioning is the lack of guide in determining the right size for the grid's cell [17]. If the grid is too coarse, two objects movements with very different trajectories might considered as similar, whereas if the grid is too fine, two objects movements with similar trajectory might be considered as different group. In unevenly distributed data, these problems may occur simultaneously.

As far as our knowledge, all of the trajectory pattern mining techniques proposed in the previous researches assume the spatial data to be evenly distributed across the workspace, making them unable to solve problems addressed in [17]. The motivation of this research is to apply a RoI generation method that can adapt to uneven spatial distribution in the data, and then integrate it to a trajectory pattern mining technique.

C. Quadtree

Finkel and Bantley develop quadtree, a data structure to store information that can be retrieved using composite key [18]. Research conducted by Samet focuses on the representation of quadtree for two dimensional point and region [19]. Figure 1 describes the structure of quadtree to represent spatial points data.



Point-region quadtree divides one cell (node) into four equally sized smaller nodes recursively until every node only contains one data point (or any threshold value). Every node splitting results in four pointers in the tree representation.

D. MiSTA

MiSTA is a TAS mining algorithm that can effectively find temporal patterns from a set of TAS [12]. The algorithm utilize the concept of τ – *CONTAINMENT*, which is a relationship that can be formed between TAS’s with temporal similarity. A TAS T_1 is considered τ – *CONTAINED* within TAS T_2 if every item in T_1 can be mapped into an item in T_2 sequentially, and the differences between transition times for items in T_1 and transition times for corresponding items in T_2 are below or equal to temporal tolerance τ .

τ – *SUPPORT* of T_2 counts the number of TAS in the database that τ – *CONTAINED* to T_2 . *Frequent* TAS is a status given to a particular TAS which has τ – *SUPPORT* more than or equal to minimum support s_{min} .

3. Problem Definition

A. Trajectory

Trajectory is a temporal sequence of spatial location visited by an object to describe its movement. Trajectory can be expressed as triplets of $S = [(x_0, y_0, t_0), \dots, (x_k, y_k, t_k)]$; where t_i ($i = 0, 1, 2, \dots, k$) is timestamp, and (x_i, y_i) represents spatial location in a two dimensional workspace visited by the object. There are other ways to represent trajectory, such as Stay Time Sequence [15] in the form of $(x_i, y_i, \Delta T_i) \rightarrow (x_j, y_j, \Delta T_j)$, where ΔT_i is the duration of object stayed in (x_i, y_i) , and TAS [12] in the form of $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} s_n$, where s_i represents spatial location sequence itemset, and α_i represent temporal annotation, which is basically transition time between s_{i-1} to s_i .

The main purpose of trajectory pattern mining method is to find trajectories that are temporally and spatially similar to a sufficient number of trajectories within the data, and is necessary to obtain useful knowledge regarding moving objects behaviours such as animal migration paths, traffic flows, and mobility pattern of people in urban areas. However, due to temporal and spatial bias contained in the data, finding similar trajectories means introducing some forms of tolerance to allow small differences between trajectories but yet they are still grouped to the same class.

B. Multistage Spatial Partitioning

Area partitioning is a solution to spatial bias problem used in many frequent trajectory pattern mining preprocessing techniques. In this paper we consider two different area partitioning approaches, a simple uniform grid and the proposed point-region quadtree based partitioning. There is no special precautions in dividing workspace using uniform grid into cells, except how to decide the right grid size, which is the main problem addressed by this paper.

In quadtree based partitioning, we want the workspace to be divided into nodes, where the sizes of the nodes are defined according to how populous local zones with data points, making each nodes in quadtree gains about almost equal densities. In this case, the parameter for area partitioning is no longer about the size of cells or nodes, but rather the number of data points allowed for nodes to contain. This threshold for generating quadtree nodes is defined as “bucket”. Quadtree is generated through a series of iterative parent nodes splitting into child nodes according to the bucket. Quadtree nodes are arranged in a tree structure for easy retrieval, started with the root node which covers the workspace entirely. The root node forms the first stage of the tree, and its descendants occupy the next stages of the tree, down to the smallest node, called the leaf node. This is the motive behind the name multistage spatial partitioning.

The resulting quadtree leaf nodes spatially bound certain data points, and by symbolizing each node, the spatial aspect of the spatio-temporal data is removed. The translated data is now in a form of sequential data with temporal annotations, and can be mined using any suitable techniques such as MiSTA to extract the patterns.

C. Popular Area

Discretization of spatial workspace results in a set of areas according to the area partitioning method. In uniform grid, the workspace is divided into n columns \times m rows sized grid cells with equal area coverage for every cells, while in quadtree, the workspace is divided into a set of child nodes, varying in size. The grouping of every data points is done according to the cells or nodes contains the location of the point. Every point corresponds to a cell or node, contributes to the density or popularity value of the cell or node. For simplification, we generalizes uniform grid cells and quadtree child nodes as “areas”.

Some areas will have higher density than the other, and this is the foundation on how mining method extracts the frequent patterns. In order to determine whether a pattern is frequent or not, a density threshold is presented to be used as pruning parameter. One thing for sure, is that frequent trajectory patterns will not be related to any low density areas, therefore it is better to remove them before mining to lower mining cost. The remaining areas are defined as “popular areas” and become item candidates for the sequential pattern mining algorithm.

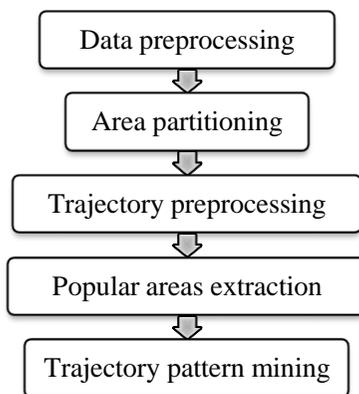
4. Trajectory Pattern Mining with Multistage Spatial Partitioning

A. Process

Our solution to perform trajectory pattern mining with multistage spatial partitioning towards data with uneven spatial distribution starts with preprocessing step. Preprocessing exploits the quadtree partitioning method to construct the RoI, and selects the popular areas to become the candidates for the mining algorithm. As explained in the previous section, area partitioning is used to overcome spatial bias problem, and to deal with uneven spatial distribution, we propose quadtree partition approach.

To extract the patterns, this paper adopts MiSTA algorithm to be performed after preprocessing. MiSTA implements the concept of τ -containment to efficiently overcome temporal bias, which uses temporal tolerance τ as the maximum limit of temporal difference between similar trajectory candidates.

Figure 2 describes the proposed preprocessing and mining steps, start from data preprocessing to perform data acquisition and selection, area partitioning using quadtree to construct RoI, trajectory preprocessing to discretize spatial information in the data, interesting areas extraction to select the candidates for the mining, and lastly the trajectory pattern mining using MiSTA to extract trajectory patterns.



Steps to be performed

B. Algorithm

The main objective of the algorithm presented in this paper is to perform preprocessing and trajectory pattern mining toward trajectory data provided in the database in order to produce output in a form of trajectory patterns visualization on a map.

B.1. Main Function

Figure 3 describes the definition of the trajectory pattern mining with multistage spatial partitioning main function. The inputs are: a set of trajectory data D , bucket integer value b as quadtree node capacity threshold, workspace boundaries coordinates, density threshold δ as minimum frequency threshold for popular area, and temporal tolerance τ for MiSTA grouping temporally similar trajectories (step 5). If the spatial location passed by trajectory A and B is similar, and the transition time difference is less or equal to τ , then trajectory A and B is similar. The last input is a map m for visualization purpose. The output for the main program is visualization V .

Algorithm: MainFunction	
Inputs:	D : set of trajectory data (sensor-based spatio-temporal)
	b : bucket
	a : spatial workspace boundaries
	δ : density threshold
	τ : temporal tolerance
	m : map
Output:	V : visualization of patterns on map
Process:	
$Q = \text{PartitionArea}(D, b, a);$	
$D' = \text{DiscretizeData}(D, Q);$	
$Q' = \text{ExtractPopularArea}(D', Q, \delta);$	
$S = \text{Prune}(Q', D');$	
$F = \text{MiSTA}(S, \delta, \tau);$	
$V = \text{Visualize}(F, m);$	

Figure 3. Main function algorithm

B.2. Area Partitioning

Figure 4 describes area partitioning algorithm which aims to discretize spatial workspace into quadtree nodes with bucket size b . Quadtree’s tree structure consists of square shaped nodes, starting from root node, which is contained by a itself (step 1-2). Each node possess a status as a parent or child according to its position in the tree. Every parent nodes have four children.

Children located at the very end of the tree are given status as leaves. In quadtree discretization algorithm, data points with identical coordinate shall only counted as one point (step 3), hence avoiding infinite node splitting due to multiple identical coordinate point insertions.

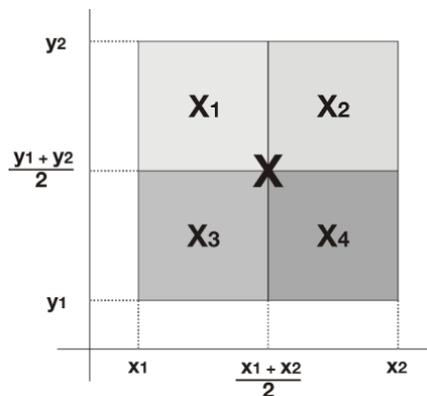
When the number of inserted data points to a node X exceeds the bucket, the program will create four new child nodes (step 5.7), that is X_1, X_2, X_3, X_4 as descendants of node X . The next step is to reclassify data points member of X into one of the four children according to the location (step 5.8).

Algorithm: PartitionArea(D, b, a)	
Inputs:	D : set of trajectory data (sensor-based spatio-temporal)
	b : bucket
	a : spatial workspace boundaries
Output:	Q : quadtree which divides a into nodes
Process: Create node r with spatial boundaries = a , and status = leaf; Add r to Q as root node; For each unique coordinate c within D : --- Create point P with coordinate c ; --- InsertPoint (r, P, b); Output: Q ;	
Procedure: InsertPoint(node X , point P , bucket b) Process: If location P is inside spatial boundary of X : --- Add point P as member of X ; --- X .MemberCount ++; If X .MemberCount > b : --- If X .status == leaf: --- --- X .status = parent; --- --- Create 4 child nodes of X , that is X_1, X_2, X_3, X_4 --- For each point P within X : --- --- For $i = 1$ to 4: --- --- --- If P is inside spatial boundary of X_i : --- --- --- --- InsertPoint(X_i, p, b);	

Figure 4. Area partitioning algorithm

Since node X has four boundaries coordinates, e.g. $(x_1, y_1), (x_2, y_1), (x_2, y_2)$ and (x_1, y_2) , the boundaries coordinates for each of its child nodes is determined based on diagram presented in figure 5.

The core of procedure InsertPoint is that every data point must be a member of a node where it resides (step 5.2). If the particular node's status is parent, then the procedure is repeated to its descendants.



Coordinate for newly created child nodes

B.3. Data Discretization

Figure 6 defines data discretization algorithm to apply quadtree system into the input data. After quadtree Q is formed, along with its child nodes that discretize spatial workspace a into various-sized smaller squares, the next step is to symbolize each child nodes with some sort of indexes (step 3). The node index will be used to replace coordinate-based spatial information in D , so it will be simpler to handle in the mining process (step 5).

Algorithm: DiscretizeData(D, Q)	
Inputs:	D : set of trajectory data (sensor-based spatio-temporal)
	Q : quadtree which divides a into nodes
Output:	D' : set of temporally annotated sequences (discretized data)
Process: $D' = \emptyset$; For every node X in quadtree Q with $X.status == child$: --- $X.index =$ Generated index (symbol) for X ; --- $X.density = 0$; --- For each point P in D which is located within boundaries of X : --- --- Add P and $X.index$ to D' ; --- --- $X.density ++$; Output: D' ;	

Figure 6. Data indexing algorithm

B.4. Popular Area Extraction

Popular area extraction algorithm selects leaf nodes from quadtree Q with sufficient density value based on density threshold δ , then gives them status as popular nodes. Popular node means node with density equal or higher than density threshold δ . User provides the density threshold value to set the minimum number of objects visit an area to make it considered semantically significant. The output of this algorithm is Q' , a quadtree with nodes popularity statuses.

B.5. Pruning

After finding out which nodes in Q' classified as popular, the next step is to perform pruning towards D' to reduce its size. The algorithm removes all points in D' which contained in nodes without popular status. The remaining points are candidate temporally annotated sequences S that can be fed into MiSTA algorithm [12] that will extract frequent sequential patterns with their respective temporal annotations.

5. Experiments

A. Design

The experiments are designed to compare quadtree and uniform partitioning performance by evaluating preprocessing duration, amount of popular areas obtained, mining duration, amount of extracted patterns, and visualization result. The experiments test different parameters values such as the amount of RoI cells/nodes, density threshold, temporal tolerance, and the number of input trajectories to find out whether they affect quadtree and uniform partitioning performance differently.

The experiments use seven days GSM transaction data sample of one mobile carrier in Bali, Indonesia as input spatio-temporal data. In total, there are more than 20,000 trajectories with uneven movement distribution. The experiments are implemented in an Intel® Core i3-3240, 3.4GHz machine, with 16 GB DDR3 PC12800 memory, and 5900 RPM hard disk runs on Microsoft® Windows 7 64-bit operating system. A tool is developed by implementing algorithms specified in chapter 4 to enable us perform the experiments.

To compare uniform grid partition and quadtree techniques in trajectory pattern mining, we expect experiments results such as the number of RoI and popular areas, amount of trajectory patterns found and their visualization, as well as some statistics like preprocessing and mining duration. We hypothesize that for equal number of RoI, the algorithm with quadtree technique will produce more popular areas compared to uniform grid technique, and therefore more trajectory patterns can be found. However, due to higher number of sequences to be processed, quadtree technique will also require more preprocessing and mining time.

B. Result

There is a significant processes performance and result difference between trajectory pattern mining which utilize uniform grid partition and quadtree, specifically for amount of popular areas after preprocessing, preprocessing and mining duration, and amount of patterns extracted. In average, quadtree produce about three times more popular areas than uniform grid, and from this, the average amount of extracted patterns from quadtree is about ten times more than of uniform grid. Table 1 shows one of the experiment results to compare uniform grid and quadtree. Visualization results in Table 1 project all extracted patterns from each approach onto the map which demonstrated that quadtree produces patterns in higher detail than uniform grid, and better represents real mobility patterns. For example, in the uniform grid visualization, there is one line segment (indicated by arrow A) to represent trajectory pattern in the southern area, whereas in the quadtree visualization, there are four line segments (indicated by arrow B1, B2, B3, and B4) to show more diverse trajectories.

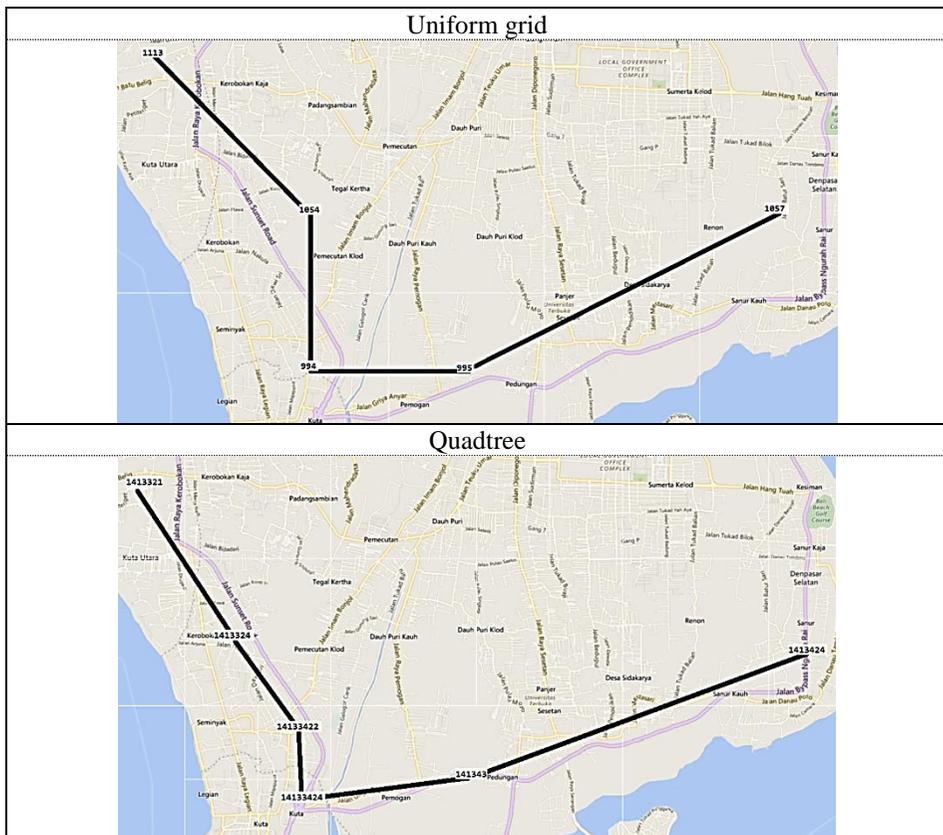
To verify whether both approaches are capable to find trajectory patterns that are relevant to real mobility phenomena, additional experiment is performed, but only selected patterns from the results are visualized. As described in Figure 7, we select the closest patterns (black line) to mimic some of the busiest road segments in the area (purple line). It is shown that both approaches managed to extract patterns from the area. It is also worth noticed that despite of the same mining parameters values, quadtree approach produces higher detail patterns compared to uniform grid in mimicking the shape of the observed road segments.

Diagrams in figure 8-10 show that the overall amount of pattern extracted from quadtree areas are significantly higher than of uniform grid, and in consequence, the mining durations also take much longer time. Figure 10 shows that higher density threshold means lower number of popular areas that leads to lower number of extracted patterns. The contradictions between quadtree and uniform grid are showed in figure 11, where more RoI means more extracted patterns from uniform grid, but less extracted patterns from quadtree. This trend happens because more areas means more popular areas that can be found from uniform grid, but in quadtree, more RoI means smaller bucket, and lower chance for nodes to obtain higher density, hence making them not popular and don't survive the pruning. Figure 12 shows that increasing amount of input data significantly raises preprocessing duration for uniform grid, but this only slightly affect quadtree preprocessing duration.

Table 1. Comparison between Uniform Grid & Quadtree on a selected case

Indicators	Uniform grid ^a	Quadtree ^a
Preprocessing duration	51 seconds	75 seconds
Popular areas	29	84
Mining duration	288 seconds	9852 seconds
Extracted patterns	37	896
Max pattern length	5	6
Visualization		

With trajectory count: 500, number of RoI: 100, temporal pattern: TAS, density threshold: 5%, maximum annotation: 12 hours, temporal tolerance: 1 hour



Comparing level of detail between uniform grid and quadtree

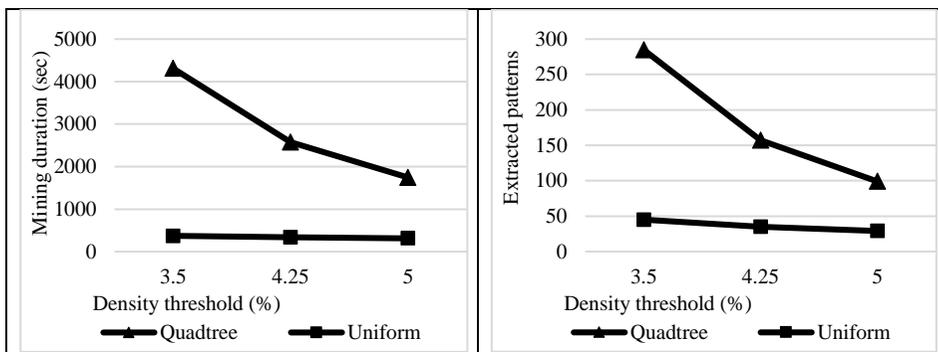


Figure 8. Experiment result in varying density threshold

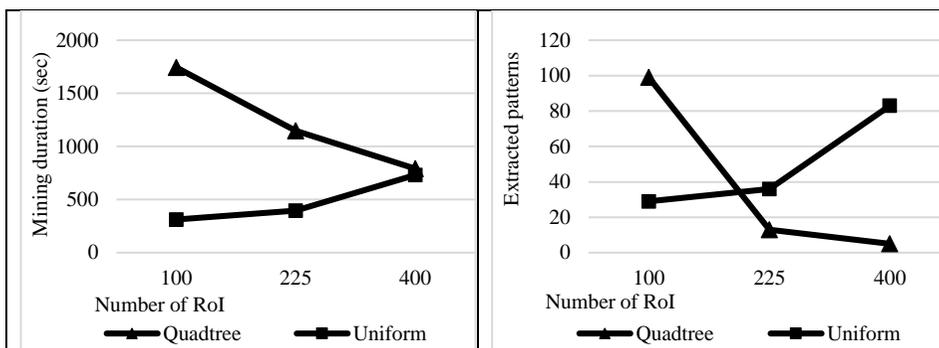


Figure 9. Experiment result in varying number of RoI

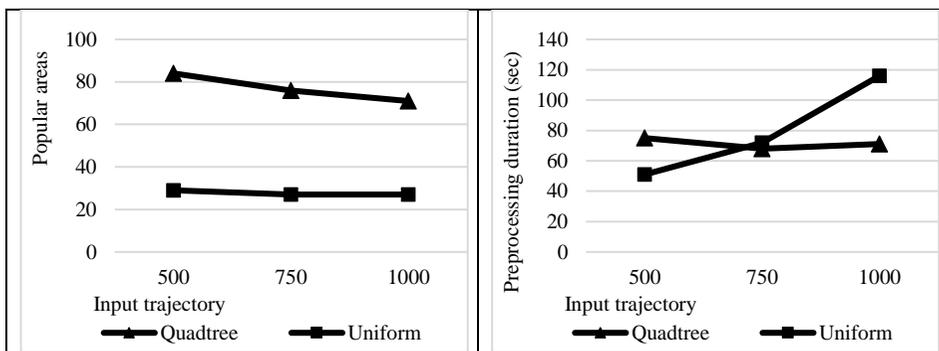


Figure 10. Experiment result in varying input trajectory count

In uniform grid system, area partitioning does not consider spatial distribution of input data at all, therefore, areas with only a few data points will get very low density, while for areas with many data points will get very high density. In contrast to uniform grid, quadtree discretizes spatial workspace based on existing data points. The more an area packed with many points, the higher the chance for that area to be split into smaller nodes, hence making its density spread among its child nodes. Whereas for areas with lower number of points will be allowed to keep their large size. In the end, only small number of areas to be pruned, more popular areas to be mined, and more extracted patterns hidden within spatio-temporal data can be revealed.

Multistage spatial partitioning is effective in handling spatially discrete data, e.g. sensor-based spatio-temporal data, however, not suitable to handle spatially continuous data, like GPS. In this case, uniform partitioning is more appropriate to be used in RoI generation process.

When using uniform partitioning, the optimum number of RoI is determined by how granular the user wish to see the resulting trajectory patterns, i.e. higher number of RoI means more detail patterns. While in multistage spatial partitioning, number of RoI is not predetermined by the user, but is resulted from bucket value for the quadtree generation process. Therefore, the challenge is to set the bucket value so that the generated quadtree divides the workspace proportionally.

6. Conclusion

The development target of this paper is the preprocessing step, that is to integrate point-region quadtree concept as an alternative way for area partitioning. Multistage spatial partitioning is proposed to be a solution for trajectory pattern mining towards unevenly distributed spatio-temporal data. It has the ability to adapt to the data's spatial distribution condition, allowing detailed hidden trajectory patterns between areas with high density and other patterns between lower density areas to be revealed. In comparison to the uniform grid partitioning approach, the experimental evaluation demonstrated that the multistage spatial partitioning deliver higher number of popular areas, which allows the mining algorithm to extract trajectory patters in higher detail and wider coverage from data with uneven spatial distribution.

For the future work, many aspect can be developed, such as expanding the capability of trajectory pattern mining preprocessing to handle spatially continuous data, like GPS. For this matter, preprocessing step is required to discretize infinite spatial locations into finite using techniques such as classification. Future work can also initiate development in the visualization domain, to suggest a better way in visualizing trajectory patterns which consist of two main factors, spatial sequences and temporal annotations.

7. Acknowledgement

This research was supported by School of Electrical Engineering and Informatics, ITB, Bandung. We are immensely grateful to researchers at KDD ISTI-CNR, Dr. Mirco Nanni and Fosca Giannotti for inspiration and assistance given to this work.

8. References

- [1]. J. Lee, J. Han, and K. Whang, "Trajectory clustering: a partition and group framework," in *Proc. 2007 ACM SIGMOD International Conference on Management Data*, 2007, pp. 593-604.
- [2]. S. Hwang, Y. Liu, J. Chiu and E. Lim, "Mining mobile group patterns: a trajectory-based approach," in *Proc. 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2005, pp. 713-718.
- [3]. Tsoukatos, and D. Gunopulos, "Efficient mining of spatiotemporal patterns," in *Proc. Symposium on Advances in Spatial and Temporal Databases*, 2001, pp. 425-442.
- [4]. H. Cao, N. Mamoulis, and D. Cheung, "Discovery of collocation episodes in spatiotemporal data," in *Proc. 6th IEEE International Conference on Data Mining*, 2006, pp. 823-827.
- [5]. H. Jeung, H. Shen, X. Zhou, "Mining trajectory patterns using hidden markov models," in *Proc. DaWaK*, 2007, pp. 47-480.
- [6]. N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao and D. Cheung, "Mining, indexing, and querying historical spatio-temporal data," in *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 236-245.
- [7]. Verhein, and S. Chawla, "Mining spatio-temporal association rules, sources, sinks, stationary regions, and thoroughfares in object mobility databases," in *Proc. Database Systems for Advanced Applications*, 2006, pp. 187-201.
- [8]. Lee, Y. Chen, and W. Ip, "Mining frequent trajectory patterns in spatial-temporal databases," in *Information Sciences*, vol. 179, issue 13, 2009, pp. 2218-2231.
- [9]. Z. Li, "Spatiotemporal pattern mining: algorithms and applications," in *Frequent Pattern Mining*, 2014, pp. 283-306.

- [10]. H. Cao, N. Mamoulis, and D. Cheung, "Mining frequent spatio-temporal sequential patterns," in *Proc. 5th IEEE International Conference on Data Mining*, 2005, pp. 82-89.
- [11]. P. Kalnis, N. Mamoulis, and S. Bakiras, "On Discovering moving cluster ins spatio-temporal data," in *Proc. 9th International Conference on Advances in Spatial and Temporal Databases*, 2005, pp. 364-381.
- [12]. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. 13th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining*, 2007, pp. 330-339.
- [13]. M. Yoshida, T. Lizuka, H. Shiohara, and M. Ishiguro, "Mining sequential patterns including time intervals," in *Proc. SPIE*, 2000, pp. 213-220.
- [14]. Vautier, M. Codier, and R. Quiniou, "An inductive database for mining temporal patterns in event sequences," in *Proc. 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 1640-1641.
- [15]. L. Wang, K. Hu, T. Ku, and J. Wu, "Frequent spatiotemporal trajectory pattern mining based on pheromone concentration," in *Journal of Information & Computational Science* 10:3, 2013, pp. 645-658.
- [16]. K. Ashok, "Estimation and prediction of time dependent origin-destination flows," Ph.D. dissertations, *Dept. Civil and Environmental Eng., Massachusetts Inst. of Technology, Massachusetts*, 1996.
- [17]. J. Kang, and H. Yong, "Mining spatio-temporal patterns in trajectory data," in *Journal of Information Processing Systems*, vol. 6, no. 4, 2010, pp. 521-535.
- [18]. R. Finkel, and J. Bentley, "Quad trees: a data structure for retrieval on composite keys," in *Acta Informatica*, vol. 4 no. 1, 1974, pp. 1-9.
- [19]. Samet, "Using quadtrees to represent spatial data," in *NATO ASI Series*, vol. F18, 1985, pp. 229-247.



Manta Sanni Ibrahim was born on 1990 in Bandung, Indonesia. He received the bachelor degree of Computer Science from Universiti Teknikal Malaysia Melaka, Malaysia in 2012, and master degree in Informatics from Institut Teknologi Bandung. Indonesia in 2016. He is currently working as an software engineer at PT. Netkrom Solusindo, Indonesia. His research interests include artificial intelligence, and business intelligence.



Saiful Akbar received the bachelor and master degrees in Informatics from the Department of Informatics, Institut Teknologi Bandung, Indonesia, in 1997 and 2002, respectively, and the doctorate degree in Engineering Science from Johannes Kepler University of Linz, Austria, in 2007. He was also a Visiting Researcher at Norwegian University of Science and Technology (NTNU) in 2009-2010. He is currently an assistant professor at School of Electrical Engineering and Informatics, Institut Teknologi Bandung. His reaserch interests include data and knowledge engineering, multimedia database and similarity retrieval, information extraction, data analytics and visualization.