International Journal of Database Theory and Application
Vol.8, No.2 (2015), pp.11-24
http://dx.doi.org/10.14257/ijdta.2015.8.2.02

A Spatio-Temporal Simulation Model for Incremental Clustering in Massive Moving Objects Data Set

Dima Alberg

SCE Shamoon College of Engineering
Industrial Engineering and Management Department
dimitria@sce.ac.il

Abstract

The real-world data process of large spatio-temporal data collection presents a very difficult technical problem. Firstly, the given process is very expensive, requiring a lot of various high-technology software instruments and modern hardware infrastructure (sensors, servers, GPS infrastructure etc.) installations; secondly, this process sometimes cannot show special traffic patterns, which we may characterize as patterned traffic trajectories. The Arena simulation framework introduced in this paper uses our suggested random linear interpolation algorithm and spatio-temporal prediction algorithm, which are applicable to visualize, handle and predict movement data with various time resolutions.

Keywords: *Spatio-temporal simulation model, movement data handling, visualization, trajectory, movement patterns*

1. Introduction

1.1. Simulation Modeling

Simulation modeling is an increasingly popular and effective tool for analyzing a wide variety of dynamic problems, not amenable to study by other means. These problems are usually associated with complex processes which cannot readily be described in analytical terms. Usually, these processes are characterized by the interaction of many system components or entities. Often, the behavior of each entity and the interaction of a limited number of entities may be well understood and can be reliably represented logically and mathematically with acceptable confidence. However, the complex, simultaneous interactions of many system components cannot, in general, be adequately described in mathematical or logical forms.

1.2. Problem Definition

The real process of large spatio-temporal data collection presents a very difficult technical problem. Firstly, the given process requires much high-technology software and modern hardware infrastructure (sensors, servers, GPS infrastructure *etc.*); secondly, this process sometimes cannot show special traffic patterns, which we may characterize as traffic trajectories. These traffic trajectories can contain detailed information about personal and vehicular mobile behavior, and therefore offer interesting practical opportunities to find behavioral patterns, useful, for instance, in traffic and sustainable mobility patterns management. Clearly in these applications, privacy is a concern. In particular, how can trajectories of mobile individuals be visualized, stored and analyzed without infringing upon personal privacy rights and expectations? How can patterns be extracted from demonstrably private/sensitive trajectory data, *i.e.*, finding the patterns without disclosing individual's personal information? These questions call for answers

ISSN: 2005-4270 IJDTA

combining technical, legal and social aspects, addressing crucial points regarding both ethics and social acceptance. Solutions that are not fully trustworthy will find insuperable obstacles barring their execution.

On the other hand, demonstrably trustworthy simulation models may open up tremendous opportunities for new knowledge-based applications of public utilities, having a large societal and economic impact. Finally, these challenges are very important for any spatio-temporal data-mining research aimed at implementing, testing and validating a spatio-temporal prediction or classification algorithm (*e.g.,* Gianotti *et. al..*. [28], Elnekave, Last and Maimon [5, 7]). Thirdly, real gathered spatio-temporal data is static and if a researcher wants to repeat his spatio-temporal data collection experiment, it will take a lot of time. Fourthly, in real gathered spatio-temporal data we cannot influence the statistical structure of the data, i.e. the researcher does not have instruments to allow him/her to set up predetermined spatio-temporal data traffic and statistical parameters (number of vehicles on the road, velocity, location, noise distribution, *etc.*).

1.3. Motivation

The proposed simulation model would have to be able to meet the following requirements:

1. To visualize the random vehicle-movement process along predetermined traffic trajectories.
2. To be flexible and scalable, i.e. the user would be able to easily change the input parameters, such as: velocity, quantity of vehicles on the map, time resolution and simulation length. Additionally, the model would have to create large datasets of spatio-temporal data (at least half a million records).
3. To be built as quickly as possible.
4. To be logical and easily interpreted by the user. This requirement is very useful, because it allows the user to change and reconstruct the original simulation model with minimum programming efforts.
5. Finally, the most critical requirement, to create a data file containing spatio-temporal data features about vehicle movement behavior. This file should contain the following current simulation spatio-temporal features: year, month, day, hour, minute, second, trajectory id, (X, Y) vehicle coordinates, vehicle velocity and records all changes in vehicle movement in an especially dedicated text file. Obviously, the fulfillment of this requirement will allow the testing and validation of the spatio-temporal prediction algorithms.

1.4. Proposed Problem Solution

Almost all traffic simulation models describe dynamic systems in which time is the basic, independent variable. Continuous simulation models describe how the elements of a system change state continuously over time in response to continuous signals. Discrete simulation models represent real-world systems (either continuous or discrete) by asserting that their states change abruptly at points in time.

There are generally two types of discrete models: discrete time and discrete event. The first segments time into a succession of known, predetermined time intervals. Within each such interval, the simulation model computes the activities which change the states of selected system elements. This approach is analogous to representing an initial-value differential equation in the form of a finite difference expression with the independent variable, *t*. Some systems are characterized by entities that are "idle" much of the time. For example, the state of a traffic signal indication (say, green) remains constant for many seconds until its state changes instantaneously to yellow. This abrupt change in state is called an event. Since it is possible to accurately describe the operation of the signal by recording its changes in state as a succession of known or computed timed events, considerable savings in computer time can be realized by only executing these events, rather than computing the state of the signal second-by-second.

For systems of limited size or those representing entities whose states change infrequently, discrete event simulations are more appropriate than are discrete time simulation models, and are far more economical in execution time. However, for systems where most entities experience a continuous change in state (*e.g.*, a traffic environment) and where the model objectives require very detailed descriptions, the discrete-time model is likely to be the better choice.

1.5. Arena Simulation Software

For the purpose of our simulation, we chose the Arena® Professional Edition (PE) software, that is one of the most common(about 13.000 installations worldwide) academic, general-purpose, discrete-time simulator capable analyze various types of queuing systems [24-26]. First released in 1993, Arena® employs an object-oriented design for entirely graphic model development. Simulation models are built using graphic objects, called modules, which define the system logic and physical components, such as vehicles, operators, machines, *etc.* Therefore, building a simulation model could be considered an interesting and challenging task, but one which does not demand long-term development.

2. Background

The literature in the field provides several ways for representing spatio-temporal data. Wang, *et. al.* [24] and Mamoulis, *et. al.*[15]established sets of time series for locations, one for each user, where a time series contains triplets *(t, x, y)*. For simplicity, they assume that all user locations are known at every time point and that the time interval between every *t* and *t+1* is fixed. Braz *et. al.* [2], on the other hand, assumed that observations are taken at irregular rates for each object and that there is no temporal alignment between the observations of different objects.

In Nehme and Rundensteiner [16], moving objects are assumed to move in a piecewise linear manner on a road network. Their movements are constrained by roads, which are connected by network nodes. The location updates of moving objects are recorded in the form of the following vector: *{O.ID, O.Location, O.t, O.Speed, O.Destination, O.Attributes},* where *O.Destination* is the connection node's position.

In Ma, *et. al.*[14] and Bakalov, *et. al.*[1], the user-movement history is an ordered*(c, t)* list, where *c* is the cell ID and *t* is the time when the object reaches the cell *c*. Representing spatio-temporal data in a more concise manner can be done by converting it into a trajectory form. In Hwang, *et. al.*[10] and in Pelekis, *et. al.*[18], a trajectory is a function that maps the times to locations. To represent object movement, a trajectory is decomposed into a set of linear functions, one for each disjoint time interval. The derivative of each linear function yields the direction and the speed in the associated time interval. A trajectory is a disjunction of all its linear pieces.

In Braz, *et. al.*[2], Lee, *et. al.* [12], Pfoser, *et. al.* [20], Rasetic, *et. al.* [23], and Güting and Schneider [9], a linear interpolation is used. The sampled positions become the endpoints of line segments of polylines and the movement of an object is represented by an entire polyline in 3D space. A trajectory *T* is a sequence {*(x1, y1, t1), (x1, y1, t1),…, (x1, y1, t1)}*. Objects are assumed to move straight between the observed points with a constant speed. Linear interpolation seems to yield a good tradeoff between flexibility and simplicity.

In Li, *et. al.* [13], objects are also assumed to move in a piecewise linear manner. Namely, an object moves along a straight line with some constant speed until it changes its direction and/or speed. If an object deviates significantly from the expected position, it is responsible for reporting the new velocity. Each moving object *O* (in the 2-D space) is represented by a 5-tuple *(xO, vO, yXo, vYo, tO)*. It is also called the profile of moving object *O*, because it uniquely determines the track of *O*.

D'Auria, *et. al.* [3] use a set of triples *(id; loc; t)*. Starting from the set of triples for a given object *id*, it is, therefore, possible, in principle, to approximate a function: *fid:time→space*, which assigns a location to an object *id* for each moment in a given time interval. Such a function is called a trajectory.

In Niculescu and Nath[17], the trajectory is expressed in the parametric form X(t); Y(t). For example, to move along a line with slope *α* passing through the source with coordinates *(x1, y1)*, the trajectory would be described by:

$$X(t) = x_1 + t \cdot \cos(\alpha), \quad Y(t) = y_1 + t \cdot \sin(\alpha) \tag{1}$$

where α, $x1$, $y1$ are constants and the parameter t actually describes the Euclidean distance traveled along the line.

In Porkaew, *et. al.* [22],the movement data of objects is represented in the database using motion parameters and location functions, which compute the spatial position of an object at any given time. A location function corresponds to a type of motion. Motion parameters specify the instantiation of the location function, *e.g.*, a starting location and speed of motion in the linear translation. As an example, consider an object that translates linearly with constant velocity in two-dimensional space. The motion parameters, in this case, correspond to the object's starting location *(xS, vS)* at some initial time *tS* and its velocity *(vx; vy)* along the spatial dimensions. Using these parameters, the location of the object at any future time *t>tS* can be determined. The authors are primarily concerned with object trajectories that correspond to linear motion at constant speed. As time progresses, the motion of an object may deviate from its representation in the database. When the deviation exceeds a predetermined threshold, the deviating object updates its motion parameters and/or the location function stored in the database to reflect its most current motion information.

In Peng and Chen [19], a movement log contains pairs of *(old VLR, new VLR)*, where *VLR* is a visitor location register representing the time of the visitor's presence within a location. At the beginning of a new path, the *old VLR* is null. For each mobile user, a moving sequence can be obtained from the movement log. Each node in the network topology of a mobile computing system can be viewed as a *VLR* and each link is viewed as a connection between VLRs. A set of maximal moving sequences is represented as a string (*e.g.*, *{ABCDHG})*, where each letter represents a node.

The INFATI dataset [11] is a real-world dataset that contains information about a group of cars and their locations within a period of two months. The INFATI data contains GPS log-data from 11 cars. This data was collected during December 2000 and January 2001. All the cars were driving in the municipality of Aalborg, which includes the City of Aalborg, its suburbs and some neighboring towns. The collected data encompasses a range of 180.8 KM× 379.6 KM (x × y). For more than a month, the movement of each car was registered in the car’s database. When a car was moving, its GPS position was sampled every second. The GPS positions were stored in the Universal Transverse Mercator (UTM 32) format.

Recently, Shahabi, *et. al.* [29] proposed the GeoDec system, which describes an environment that accurately integrates heterogeneous geospatial data sources to create a compelling, realistic and information-rich visualization of a geographic location. Specifically, their current system fuses satellite imagery, 3D models, textures and video streams, road data, raster maps, point data, as well as temporal data. To achieve this, the system utilizes various components such as: object recognition, object tracking, event identification, a 3D building modeler and also includes a glove-based interface to facilitate intuitive user interaction and decision making.

The problem of generating synthetic spatio-temporal data in order to create available datasets for research purposes is discussed by Giannotti, *et. al.* [8], and the CENTRE system is provided (Cellular Network Trajectories Reconstruction Environment). Their aim is to generate benchmark datasets for cellular device positioning data, not publicly available for scientific research due to privacy concerns. Spatio-temporal data is represented as a set of records of the form :*{ object id, antenna id, time, distance from zero point}*. This system aims at simulating semantic-based movement behaviors using a set of user-specified parameters and allows adding user preferences, which may influence random distributions or domain semantics like cartography or geographic constraints. CENTRE is composed of 3 modules. The first, Synthetic Trajectories Generation, is based on the Generator for Spatio-Temporal Data (GSTD) and generates objects simulating human movements. The second, Logs Generation, simulates the cell phone detection by the network and produces the position log. The third, Approximated Trajectories Reconstruction, reconstructs trajectories from the logs, considering the approximation of the data (smaller and denser antennas typically produce a better approximation of original trajectories).

3. Simulation Model

3.1. Overview

The proposed spatio-temporal simulation model is basing on a map segment with 12 bidirectional roads and 10 inter Sections; a more detailed description of this area and inters ection coordinates can be found in chapter 3.2 and 3.3. It is important to emphasize that the proposed spatio-temporal simulation model will simulate a user-defined set of periodic spatio-temporal trajectories, where each trajectory represents a series of data-points traversed by a given moving object during a user-specified period of time (*e.g.*, one day, one week, one month, *etc.*). Since we assume that a moving object behaves according to some periodic spatio-temporal pattern (*e.g.*, a daily pattern of a cell phone user), we have to determine the construction of each spatio-temporal sequence (trajectory).

Thus, in the experimental part of this work, we assume that a moving object repeats its trajectories on a daily basis, meaning that each trajectory describes an object movement during a user-specified period (chapter 3.4 and 3.5) along model predefined traffic roads (chapter 3.6). In a general case, each object should be examined for its periodic behavior atvarious time resolutions (*e.g.*, one day, one week, one month, *etc.*), in order to determine the duration of its periodicity period. Finally, we determined the training data window period, which is used to learn the object's periodic behavior, based on its recorded trajectories during the user-specified simulation-length period.

3.2. Map Segment Representation

The following map (Figure1) presents part of a roadmap located in southern Israel, between the cities of Beer-Sheva and Kiryat Gat. All traffic roads on this map fragment are bidirectional and divided with protective barriers. In real life, this infrastructure is characterized bya very active traffic flow between these two cities and fulfills a very major function in passenger and freight transportation in southern Israel.

Figure 1. Beer-Sheva-Kiryat Gat, Southern Region with Mapped Intersections

3.3. Representation of Intersections

In the previous Figure, we can also see ten mapped intersections, connecting certain roads. The given intersection coordinates play a key role in the simulation-model architecture. The Table (below) demonstrates intersection coordinates, which we used in the Rockwell Arena model, in order to do online calculations of the cars spatio-temporal locations. These coordinates appear in the Arena model as two-dimensional arrays, located in simple, comma- separated text files. Therefore, in order to change or add new intersectionsto this array, we need only edit one text file in the model.

Table 1. Given Intersection Coordinates

Intersection (j)	Intersections	X	Y
1	Beer-Sheva (B7)	1 169	2 880
2	Country Club (C)	8 83	3 019
3	ha-Nasi (H)	4 07	2 435
4	Lehavim (L)	1 157	2 104
5	Kamah (K)	9 31	1 555
6	Plugot (P)	8 62	1 05
7	Shoket (S)	2 001	2 613
8	Tel-Sheva (T7)	1 459	2 978
9	Goral (G)	1 107	2 588
10	Kiryat Gat (KG)	1 030	2 07

3.4 Trajectory Representations

Our simulation model exhibits a two-directional graph (Figure 2), where the nodes represent the intersectionsand the edges represent the roads, and where each road contains a limited quantity of moving vehicles.

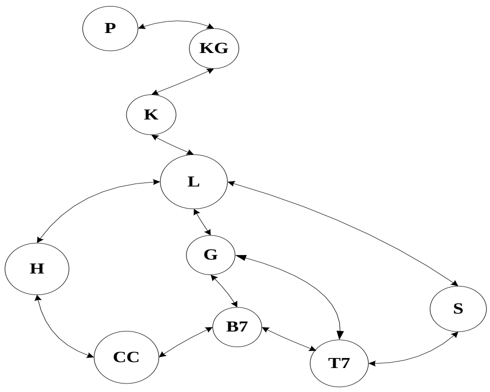

Figure 2. Simulation Model Graph Representation

In the proposed model, we set up four various trajectories, by which we may move vehicles and gather their online spatio-temporal statistics. We show these defined trajectories in Figure 3as color-brushed paths. Each trajectory is determined by means of a beginning point, the sequence of intersections and a final point. When a vehicle enters the system and begins to travel along a trajectory, the simulation model initiates a simulated vehicle entity at the starting point, which then parallels the entire vehicle's movement through the sequence of intersections until arriving at the final point. This vehicle entity is delayed at the final point for a period of [5-9] hours, after which it returns to the starting point along the previously-used trajectory or via another defined trajectory (in the proposed model, we defined delay as uniform distribution with parameters minDelay and maxDelay).

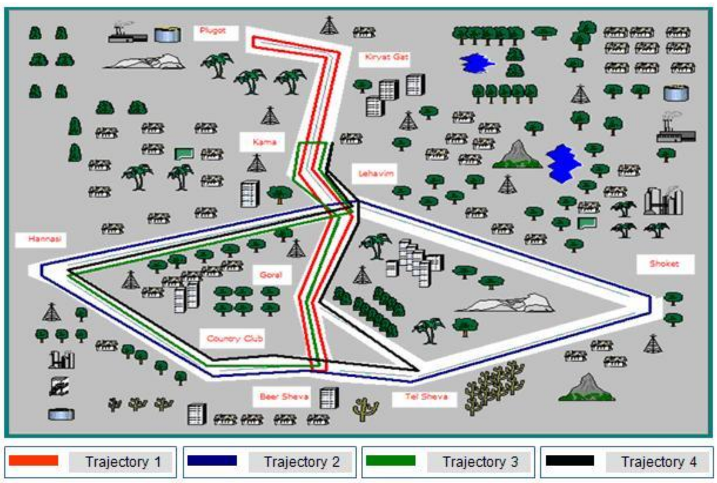

Figure 3: Beer-Sheva-Kiryat Gat, Southern Region, Four Mapped Trajectories

This suggested trajectory-switching procedure is executed by means of a predefined trajectory probability matrix. For example, when we have 4 vehicles, and we want each vehicle to move only through its own defined trajectory, we must identify the trajectory probability matrix as a unit matrix with size 4. Finally, when the vehicle returns to starting point, it will be delayed for [10-12] hours and after that it begins anew, or previous, trajectory loop, conditional to the probability value as predetermined by the user within the trajectory-probability matrix.

3.5. Trajectory Construction

As mentioned above, each simulated trajectory is constructed as a discrete loop sequence of intersections and bounded points. Another important object of the proposed simulation model is the bidirectional road connecting two intersections; thus, we may alternately define a trajectory as a set of connected roads. The following Table 2 represents 4 built trajectories. In other words, we constructed two dependent objects: a pool of trajectories and a prior probability matrix that switches certain vehicles from one trajectory to another. It is important to note that an application user will easily increase/decrease the number of trajectories in each trajectory pool.

Table 2. Trajectory Sequences

Place, ID (j)	Trajectory sequences							
	No. 1		No. 2		No. 3		No. 4	
	.	F .	.	F .	.	I .	.	I .
1		2 1		8				8
2				7				7
3				2 1				6
4		0 1		1 1		4		3
5		9				8		
6		7						
7				0 1				
8				9				9
9		1 1						
10		8						
F.=Forward, B.=Backward								

3.6. Traffic Road Construction

In order to define traffic road logic, we constructed the following block scheme represented in Figure 4. This block scheme represents the two destination movement pattern logic and interpolation logic of current vehicle location defined by car velocity and (X, Y) coordinates.

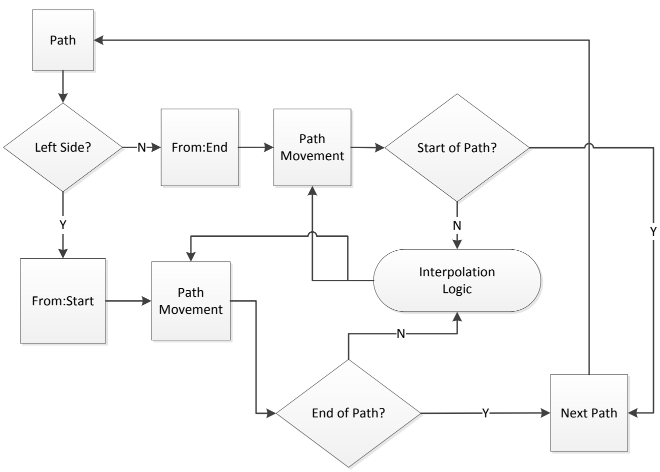

Figure 4. The Path Logic Flow Chart

Finally, the pseudo code represented in Figure 5 defines the macro logic of a specific road. The main idea behind the proposed algorithm is to retrieve and save spatio-temporal features for each vehicle entity participating in this traffic simulation. When a vehicle entity enters a road after an interSection, the procedure determines the moving destination of this vehicle and begins to scan and record the vehicle's (X, Y) coordinates, time stamp, trajectory and velocity into corresponding data log file.

```
Main procedure Path
      GET:
            interSection i, interSection j
            vehicle trajectory
            movement direction

      SET :
            interSection i coordinates,  interSection j coordinates, current vehicle
        coordinates

      IF movement direction=forward THEN
            begin point=interSection i coordinates
            final point=interSection j coordinates
            current point=current vehicle coordinates
                        REPEAT
                        CALL Interpolation
                        UNTILL current point=final point
                        INITIALIZE next trajectory interSection
      ELSE
            begin point=interSection j coordinates
            final point=interSection i coordinates
                        REPEAT
                        CALL Interpolation
                        UNTILL current point=begin point
                        INITIALIZE next trajectory interSection
      END IF
Procedure Interpolation
      GET: current, begin and final points
      SET [formula 2]: vehicle velocity
      COMPUTE [formulas 5 and 6]: vehicle coordinates
      WRITE [ARENA READWRITE MODULE]:
            Current vehicle: ID, timestamp, coordinates, velocity and trajectory to log file.
```

Figure 5. The Path Module Pseudo-Code

Note that the scan frequency will be determined by the user for each simulation run. For example, if the scan frequency is determined as one second, then the spatio-temporal data of each vehicle will be updated each second in the data log file, *etc.* The block scheme in Figure 4and the pseudo-code in Figure 5 provide more detailed explanations about line vehicle feature retrieval and calculation, with the application of the following formulas:

$$\tilde{v}_i \sim Triangular(v(\min)_i, \overline{v}_i, v(\max)_i) \tag{2}$$

where $\tilde{v}_i$ =velocity triangular distribution in trajectory i (i=1, 2, 3, 4) and*v(min)i*, $\overline{v}_i$ and *v(max)i*- predetermined triangular distribution parameters, as defined by the user.

$$\hat{X}(t) = x_0^j + \left(\tilde{v}_i \cdot t\right) \cdot \cos\left(\alpha\right) \tag{3}$$

where x_0^j - the initial x-coordinate of interSection j (see Table 2) and $\hat{X}(t)$ = vehicle X-coordinate in time t. In the same manner, we defined theY-coordinate in time t:

$$\hat{Y}(t) = y_0^j + \left(\tilde{v}_i \cdot t\right) \cdot \sin\left(\alpha\right) \tag{4}$$

Finally, we added a white noise element to each coordinate, in order to bring the gathered spatio- temporal data in line with real vehicle movement data:

$$X(t) = \hat{X}(t) + N(0,1) \tag{5}$$

$$Y(t) = \hat{Y}(t) + N(0,1) \tag{6}$$

Note, that proposed random linear interpolation model improves the previously proposed models of Niculescu and Nath [17] and Rasetic, *et. al.* [23] by means of the two following factors:

1. Velocity within each model road was defined by using certain continuous random distributions. For example, in our proposed model, we supposed that the velocity of each vehicle on all the roads distributes normally with a mean of 70km/h and a standard deviation of 10 (of course, the distribution and its parameter definition we left to the user).

2. In order to bring our collected spatio-temporal data closer to the real data and to increase the heterogeneity between vehicle coordinates, we added the element of noise to our calculated coordinates. Either this element will be eliminated in the final simulation or the user will define this element as known distributed noise or random white noise (which distributes normally with a mean of zero and a standard deviation of one).

4. The Spatio-Temporal Data Simulation and Collection

Our experiments were run on a PC Intel Pentium at 1.86GHz with 1GB RAM and 60GB hard disk. We simulated a "real-life" scenario of four vehicles driving along the main motorways of Beer-Sheva-KiryatGat in the Plugot area (Figure. 1). The cars location was sampled each second for44 days. The car velocities were randomly sampled from a triangular probability distribution (Formula 2). The measured location was sampled with an addition of white noise (Formulas 5 and 6) when the simulation experiments were conducted as follows: each experiment simulated four objects.

One of the purposes of this simulation study was the creation of online spatio-temporal data. The first and the last three rows of this file are shown in Figure 7, whereas the original file contains approximately one million rows of data.

V.ID	Time Stamp	X	Y	Velocity	T.ID
A	A	A	A	A	L
3	1.2.0.0.1	432.8	409.6	80.5	2
2	1.2.0.0.2	431.6	409.3	95.4	1
4	1.2.0.0.2	430.4	408.9	111.2	2
…	…	…	…	…	…
1	2.5.19.20.10	429.2	408.6	87.3	4
3	2.5.19.20.11	359.1	380.1	91.4	2
4	2.5.19.20.12	359.6	378.7	98.7	2

(Where V.ID and T.ID are the vehicles and trajectory's ID andA, L is attribute and label)

Figure 7. The Spatial-Temporal Data Log

Actually, the size of the file was determined by the four following factors:

1. Time resolution, as defined by the user. This resolution determines the time frequency of the spatio-temporal characteristics measurement. In our proposed simulation model, we determined a lower simulation frequency of one second, but the user can choose a higher simulation frequency, for example: 5 seconds, 10 minutes, 1 hour, *etc.*

2. Number of predetermined trajectories (in the proposed simulation, we defined four different trajectories, since these trajectories represent real traffic patterns in a given map segment).

3. Number of created entities of vehicles in the simulation system (in our case, the proposed evaluation model created1000 entities at the start of the simulation).

4. The simulation duration (in our proposed evaluation, the simulation run duration was 44 days with 24 hours per day).

We can also mention that the suggested online spatio-temporal data-collecting mechanism allows us to gather a relatively large quantity of data with predetermined traffic and statistical characteristics over a very short time period.

5. The Spatio-Temporal Prediction Algorithm Evaluation

Finally, we evaluated the spatio-temporal prediction algorithm (Elnekave *et. al.*. [5], [7]) for detecting spatio-temporal outliers by measuring hits and false-alarms of the results. The simulation experiments were conducted as follows: each experiment simulated four objects. For each object, each daily movement was summarized into a MBB based trajectory, and then each object's trajectories were clustered into four periodic movement patterns (four patterns per object). For each object, we detected the outliers (abnormal behavior) in its observed periodic movement patterns and finished the proposed evaluation study by presenting the final, predicted results and validation experiments. We ran our experiments with two iterations amount bounds: 15 and 30, and with the following numbers of clusters (K): 3, 4, 5, 6, 8 and 10. The next Figures will present only five days clustering results since the first clustering of the five is not incremental and thus will act identical in both versions. The results show that the incremental algorithm usually decreases running times. Figure 8(a) presents clustering into 4 clusters with 15 iterations. It demonstrates a decrease in running times in each of the 44 days we clustered. The same is true, yet less significant for K = 30 as shown in Figure 8(b).

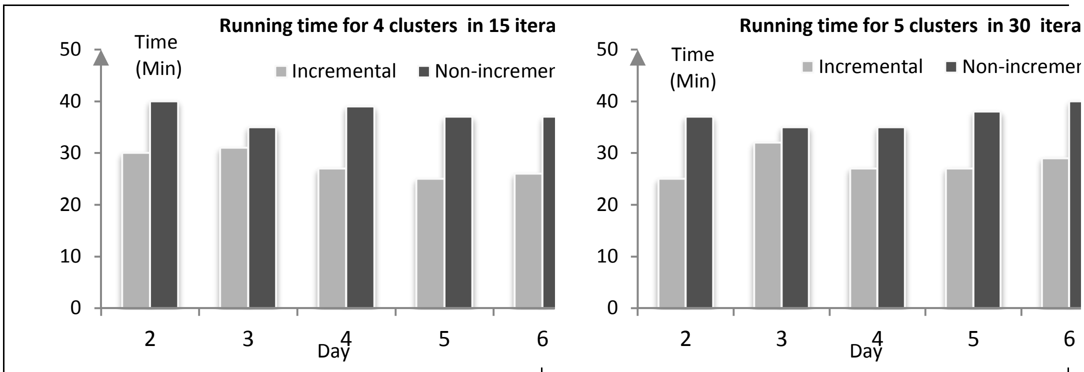

Figure 8(a) Figure 8(b)

Figure 8. Running Times Comparison when Clustering into 4 and 5 Clusters with 15 and 30 Iterations

We can see in Figure 9 (a) that the incremental algorithm usually improves cluster quality according to the Dunn index (the Y axis).When K = 30, the improvement is similar, as shown in Figure 5(b).

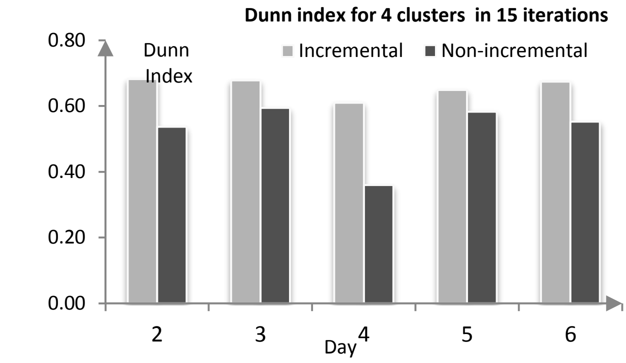

Figure 9(a)

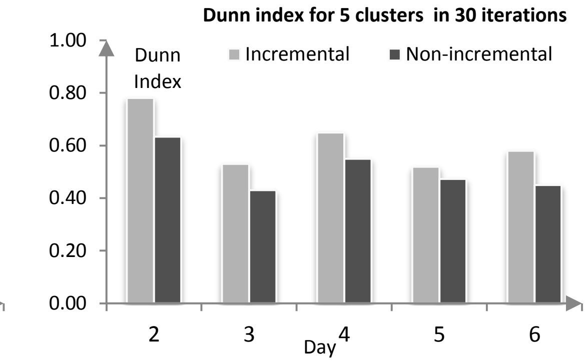

Figure 9(b)

Figure 9. Dunn Index Comparison when Clustering into 4 and 5 Clusters with 15 and 30 Iterations

We can also mention that the suggested online spatio-temporal data collecting mechanism allows us to collect a relatively large quantity of data with predetermined traffic and statistical characteristics over a very short time period and evaluate, in parallel, quality of the predefined path patterns with our proposed incremental clustering algorithm.

6. Conclusions

The purpose of traffic simulation models is to help a human analyst to understand the data and the underlying phenomena. The simulation framework we have introduced in this paper is applicable to diverse movement data. Thus, it can be used in studies of individual movement behaviors, including the behaviors of animals. It can also be used to analyze the movements of multiple entities for the purposes of city planning, traffic management, logistics, the optimization of layouts of public venues and shopping areas, the allocation of facilities or advertisements, *etc.* Finally, the development of a large-scale simulation model, that more closely reflects macro traffic processes, enables the benefits of these processes to be more clearly identified. The real-world applicability of such processes depends upon the quality of the models developed to analyze them. In addition, the iterative process allows the users to implement optimization solutions, guaranteed to be feasible for actual execution. This allows the user to study actual traffic processes in terms of both the simulation and the prediction models.

References

[1] P. Bakalov, M. Hadjieleftheriou, E. J. Keogh and V. J. Tsotras, "Efficient trajectory joins using symbolic representations", Mobile Data Management, **(2005)**, pp. 86-93.

[2] F. Braz, S. Orlando, R. Orsini, A. Raffaeta, A. Roncato and C. Silvestri, "Approximate Aggregations in Trajectory Data Warehouses", ICDE Workshop on Spatio-Temporal Data Mining (STDM07), IEEE, **(2007)**, pp. 536 – 544.

[3] M. D'Auria, M. Nanni and D. Pedreschi, "Time-focused density-based clustering of trajectories of moving objects" JIIS, special issue on "mining spatio-temporal data", vol. 27, no. 3, **(2006)**, pp. 267-289.

[4] S. Elnekave, M. Last, O. Maimon, Y. Ben-Shimol, H. Einsiedler, M. Friedman and M. Siebert, “Discovering Regular Groups of Mobile Objects Using Incremental Clustering”, submitted to the 5th Workshop on Positioning, Navigation and Communication WPNC’08, Hannover, Germany, **(2008)** March 27.

[5] S. Elnekave, M. Last and O. Maimon, (2007) "Incremental Clustering of Mobile Objects". Proceedings of the ICDE Workshop on Spatio-Temporal Data Mining (STDM'07), Istanbul, Turkey, **(2007)** April 20, pp. 585-592.

[6] S. Elnekave, M. Last, O. Maimon, "Measuring Similarity Between Trajectories of Mobile Objects". H. Bunke, A. Kandel, and M. Last (Editors), “Applied Pattern Recognition”. Springer-Verlag Series: Studies in Computational Intelligence, vol. 91, **(2007)**, pp. 101-128.

[7] S. Elnekave, M. Last and O. Maimon, "Predicting Future Locations Using Cluster Cancroids", Proceedings of the 15th ACM International Symposium on Advances in Geographic Information Systems (ACM GIS 2007), Seattle, Washington, USA, **(2007)** November 7-9, pp. 368-371.
[8] F. Giannotti, A. Mazzoni, S. Puntoni and C. Renso, "Synthetic Generation of Cellular Network Positioning Data". GIS'05: Proceedings of the 13th ACM International Workshop on Geographic Information Systems, **(2005)**, pp. 12-20.
[9] R. H. Güting and M. Schneider, "Moving Objects Databases". Morgan Kaufmann Publishers, **(2005)**, p.85.
[10] S. Y. Hwang, Y. H. Liu, J. K. Chiu and F. P. Lim, "Mining Mobile Group Patterns: A Trajectory-based Approach", Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005. Proceedings, vol. 3518, **(2005)** pp. 713-718.
[11] C. S. Jensen, H. Lahrmann, S. Pakalnis, and J. Runge, "The INFATI Data" Aalborg University, Time Center Technical Report. http://www.cits.aau.dk/download/INFATI.pdf. **(2004)**, pp. 1-10.
[12] J. W. Lee, O. H. Paek and K. H. Ryu, "Temporal moving pattern mining for location-based service", Journal of Systems and Software, vol. 73, no. 3, **(2004)**, pp. 481-490.
[13] X. Li, J. Han and S. Kim, "Motion-Alert: Automatic Anomaly Detection in Massive Moving Objects". ISI-2006, **(2006)**, pp. 166-177.
[14] S. Ma, S. Tang, D. Yang, T. Wang and J. Han, "Combining Clustering with Moving Sequential Pattern Mining: A Novel and Efficient Technique", PAKDD, **(2004)**, pp. 419-423.
[15] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao and D. W. Cheung, "Mining, Indexing, and Querying Historical Spatiotemporal Data", Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2004)**, pp. 236-245.
[16] R. Nehme and E. A. Rundensteiner, "SCUBA: Scalable Cluster-Based Algorithm for Evaluating Continuous Spatio-Temporal Queries on Moving Objects", EDBT, **(2006)**, pp. 1001-1019.
[17] D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," MOBICOM, **(2003)**, pp.260-272.
[18] N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos and Y. Theodoridis, "Mining Trajectory Databases via a Suite of Distance Operators", Workshop on Spatio-Temporal Data Mining (STDM07), IEEE, **(2007)**, pp. 575-584.
[19] W. Peng and M. Chen, "Developing Data Allocation Schemes by Incremental Mining of User Moving Patterns in a Mobile Computing System", IEEE Transactions on Knowledge and Data Engineering, , vol.15, no. 1, **(2003)**, pp.70-85.
[20] D. Pfoser, "Indexing the Trajectories of Moving Objects". IEEE Data Engineering Bulletin, vol. 25, no. 2, **(2000)**, pp. 3-9.
[21] D. Pfoser, C. S. Jensen and Y. Theodoridis, , "Novel Approaches in Query Processing for Moving Object Trajectories", Proceedings of the 26th International Conference on Very Large Data Bases, **(2000)**, pp.395-406.
[22] K. Porkaew, I. Lazaridis and S. Mehrotra, "Querying Mobile Objects in Spatio-Temporal Databases", In Proc. of the 7th Int'l Symposium on Advances in Spatial and Temporal Databases, **(2001)**, pp. 55-78.
[23] S. Rasetic, J. Sander, J. Elding and M. A. Nascimento, "A Trajectory Splitting Model for Efficient Spatio-Temporal Indexing", VLDB, **(2005)**, pp. 934-945.
[24] Y. Wang, E. P. Lim and S. Y. Hwang, "On Mining Group Patterns of Mobile Users", Proceedings of the 14th International Conference on Database and Expert Systems Applications, DEXA, **(2003)**, pp. 287-296.
[25] W. D. Kelton, R. P. Sadowski and D. A. Sadowski, "Simulation With Arena", 4nd edition, McGraw-Hill, New York, **(2007)**.
[26] J. Banks, J. Carson, B. Nelson and D. Nıcol, "Discrete Event System Simulation", 4nd edition Prentice Hall, **(2005)**.
[27] C. D. Pegden, R. E. Shannon and R. P. Sadowski, R. "Introduction to Simulation Using SIMAN", 2nd edition, McGraw-Hill, **(1995)**.
[28] F. Gianotti, M. Nanni, D. Pedreschi and C. Renso, "GeoPKDDGeographic Privacy-aware Knowledge Discovery", Mobility, Data Mining and Privacy, Springer, **(2008)**.
[29] C. Shahabi, Y. Y. Chiang, K, Chung, K. C. Huang, J. Khoshgozaran-Haghighi, C. Knoblock, S. C. Lee, U. Neumann, R. Nevatia, A. Rihan, S. Thakkar and S. You, " Geodec: Enabling Geospatial Decision Making", IEEE International Conference on Multimedia & Exp, **(2006)**.