STRAT: Status and Region Aware Taxi Mobility Model for Urban Vehicular Networks

Haiquan Wang
Wenjing Yang
Jingtao Zhang
Jiejie Zhao
School of Software, Beihang University, Beijing, P.R.China
Beijing Key Laboratory of Network Technology, Beijing, P.R.China
Yu Wang*
Department of Computer Science, University of North Carolina at Charlotte, NC, USA

Abstract

Using a realistic mobility model will enhance the validity of simulations. However, the difficulty lies in applying those rules discovered from large amount of data. Researchers have been working on mobility model extracting from real data set, whereas the taxi behavior differences between various taxi statuses have been ignored in previous works. Based on the experience in daily life,There are two assumptions related to the taxi status, one is that the behavior of taxi is influenced by the statuses and the other one is that the macroscopic movement is related with different geographic features in corresponding status, are introduced and estimated by the real data. Based on the two assumptions, a novel taxi mobility model named STRAT is proposed with respect to taxi status. The simulation results illustrate that STRAT has a good reality approximation in trace samples, distribution of nodes and the contact characteristics.

keyword

mobility model, taxi status,region recognition

I. Introduction

In vehicle ad hoc networks (VANETs) [1], realistic mobility model is an important way to improve route planning, control traffic situations, or solve the vehicle-to-vehicle communication problems. However, mobility models might influence simulation performance, since mobility model defines the nodal mobility pattern including speed and direction. However, large amount of data are difficult to be utilized directly. It is necessary to work on realistic mobility models. Some researchers [2], [3] modeled the vehicular mobility, extracting different feathers from real data sets. Nevertheless, taxi status is ignored in the previous works.

In this work, a STatus and Region Aware Taxi mobility model, STRAT, is proposed based on the real taxi GPS data. Two assumptions are introduced in section II. We assume that the taxi behavior and geographic features are related with different status. They are verified to be reasonable by the statistical analysis of the data set. START is modeled based on these assumptions. In the macro scope, instead of simply dividing the area into coarse-grain regions, we classified the area into two set of regions according to the passenger loading or droping event density. When a taxi take a passenger, the current region will be selected from the region set of load-event and the destination region, where the drop-event happens, will be selected from the region set of drop-event. We investigate the relationship between load-event regions and drop-event regions. Paths from the sources to destinations will be found by Dijkstra algorithm. In detailed view, the *speed* for the two taxi statuses are discussed respectively. Simulations are carried out to compare the similarity of node trace characteristics and contact characteristics. The results show that our mobility model has a good approximation with the real scenario in trace samples, distribution of nodes and the contact characteristics.

The rest of our paper is organized as follows: Section II proposes two assumptions which are further validated by statistical results of real data. Section III presents the modeling process. Simulation results are demonstrated in Section IV. Finally, Section V concludes this paper.

This research has been partially supported by the US National Science Foundation (NSF) under Grant No. CNS-1319915 and CNS-1343355, the National Natural Science Foundation of China (NSFC) under Grant No.61300173 and No. 61170295, the Project of Aeronautical Science Foundation of China under Grant No.2013ZC51026 and No.2011ZC51024, the Fundamental Research Funds for the Central Universities under Grant No. YWF-12-LXGY-001, and the State Key Laboratory Software Development Environment and Network Information and Computing Center of Beihang University.

II. Assumptions and statistical analysis of Taxi Trace

In this section, we focus on statistical analysis of the data set. Firstly, the data set is introduced in section II-A. Then, two assumptions are proposed and validated in the following sections.

A. Trace Dataset: Beijing Taxi Traces

A real-world GPS data set, which was generated by $12,455$ taxis in Beijing, China within 5 days from March 3rd,2011 to March 7th,2011, is used in this paper. Each row includes a base station ID, company name, taxi ID (id), timestamp (t), current location (l, including longitude and latitude), speed, event, status, et al. Of all the fields, the taxi ID, time stamp, and current location, status and event are used in this paper. Note that GPS traces from taxis have been used recently for inferring human mobility [4] and modeling city-scale traffics [5]. Therefore, we believe that they are suitable to

characterize the contact patterns among vehicles in large-scale urban scenario.

TABLE I: Explanation of Events and Status

Event	Explanation
0(drop)	A taxi's status change to vacant.
1(load)	A taxi's status change to occupied.
2	Set up defense.
3	Cancel defense.
4	No event happened.
Status	**Explanation**
0(vacant)	A taxi is vacant.
1(occupied)	A taxi is occupied.
2	A taxi is setting up defense.
3	Stop running.

Especially, there are five types of events and four types of statuses, which are explained in table I. We only discuss the vacant/occupied statuses and load/drop events in this paper.

B. Assumptions

According to the experience in daily life, the following two assumptions are given:

- **Assumption 1:**The behavior of a taxi will change when its status changes. When a taxi is occupied, its destination is certain, and the speed of occupied status will accelerate relatively. In contrast, when a taxi is vacant, it will slow down or even stop to search for potential passengers along the road. Thus, taxi behavior characteristics, such as speed and the status duration vary coherently.
- **Assumption 2:** The movement behavior of taxis associates with geographic features. When taxis is occupied, the destination may be inclined to be some places, such as the airport. Meanwhile, when taxis are vacant, drives tend to stay around some hot spots where more people want to take a taxi.
  1) The destination selection is influenced by different regions.
  2) events occurs in different regions un-evenly, passenger drop and load events are distinct.

Therefore, we analyze the speed,duration and passenger load/drop events distribution to estimate our assumptions.

C. Speed analysis

In this section, we investigate the average speed $\overline{speed}$ in each status. For example, taxi i drives in occupied status for a distance d using time t, then the average speed in this status is d/t. From March 3th to 7th, 2011, the $\overline{\overline{speed}_{empty}} = 3.627m/s$, while that for occupied status is $\overline{\overline{speed}_{occupied}} = 7.083m/s$.

To further investigate the cumulative speed distribution, proportion for every $\overline{speed}$ section is calculated. As shown in Fig.1. For example, dot(20,0.0245) means 2.45% records fall in the range $[0, 20)km/h$. We also fit the speed to model the microscope behavior, which will be shown in section III Fig. 1 shows that $\overline{speed}$ distribution differs for each status. For vacant status, the $\overline{speed}$ gather together at 1 also demonstrates that the speed distribution is with strong regularity for each status.

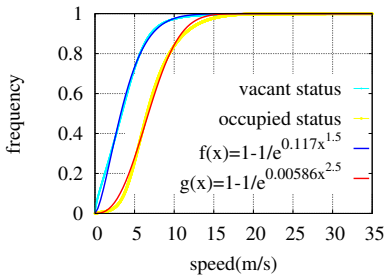


Fig. 1: $\overline{speed}$ distribution for vacant and occupied statuses.

D. Status duration analysis

The duration distribution for each status are shown in fig.2. Status duration represents the time length of a taxi staying in a certain status. The red line represents the duration time distribution for vacant status, and the blue line is for occupied status. A dot of the line means the proportion of the duration. A peak exists in each line, and it is obvious that the peak of the red line is earlier than that of the other line. And the value of duration for vacant status trends to be smaller. It corresponds to the realistic situation, because drivers tend to shorten the waiting time to raise their income.

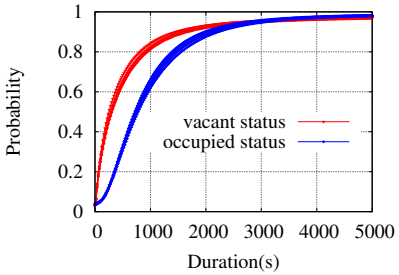


Fig. 2: Status duration distribution.

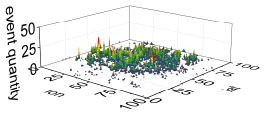
The statistical results are consistent with the *assumption 1*, that is, the behaviors of taxis are similar within each status while differ between the two statuses.

E. Taxi event distribution

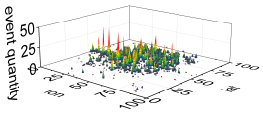
To validate $assumption2$, we quantitatively analyze vehicles density inside one hour. By dividing the whole network into 100×100 grid, taxi density distributions for load-event and drop-event are computed in each cell.

Fig.3 shows the load/drop-events distribution in three time sections. In the morning, people begin to get out. so the load-event distributes more even than that of drop-event. this phenomenon is happened because that the load-event spots are mainly at the homes of the citizens, while the drop-event spots tend to gather together at workplaces, railway stations or scenic spots. Besides, in the evening, the dispersion of load-event is of lower degree than that of drop-event, for people returning home at that time.

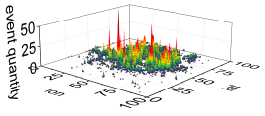
The amount of loading passengers in each cell shows geographic features:the distribution is uneven, and the difference between load/drop-event distributions illustrates that



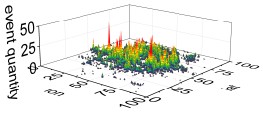
(a) $load-event, 7:00-8:00$



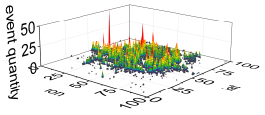
(b) $drop-event, 7:00-8:00$



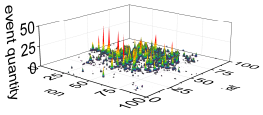
(c) $drop-event, 12:00-13:00$



(d) $load-event, 12:00-13:00$



(e) $load-event, 17:00-18:00$



(f) $drop-event, 17:00-18:00$

Fig. 3: Taxi density for load-event and drop-event in one hour

the load/drop-event regions are different which support the *assumption 2*.

III. Modeling

Movement model defines the mobility pattern of nodes, which can be represented as a collection of pathes, say $Paths :< p_1, p_2, ..., p_n >$, so does the STRAT. A p_i takes two steps to adopt, destination selection and moving process from source location to destination.

Destination selection: In STRAT, to select a destination of a node is closely related to note only its current location but also its current status. Dividing the area into regions by the density of passenger load/drop events or loading passenger events respectively, two transition probability matrices are calculated, one is the probability from passenger drop event regions $\{REGION_{m,drop}\}$ to passenger on events regions $\{REGION_{n,load}\}$. Note that $\bigcup\{REGION_{m,drop}\} = \bigcup\{REGION_{n,load}\} = AREA$. If the status of a taxi changes to vacant, its current location is a $REGION_{i,drop}$. Consequently, a destination region in $\{REGION_{n,load}\}$ will be selected by querying the transition matrix from $\{Region_{m,drop}\}$ to $\{Region_{n,load}\}$. Then, STRAT will randomly select a map node in the region as the destination. The destination selection process is similar for the occupied status transition. During this process, the region transition probability will be utilized according to the current status.

Moving process: When the source location (current location) and destination location is given, next step is to find a path. For simplicity, we adopt the Dijkstra algorithm ,which will find the shortest path from source to the destination, to route on map. The speed of the path is donated as the $\overline{speed}$, which is adopted by the current $\overline{speed}$ distribution of corresponding status introduced in the following section.

Based on the design above, we model the movement on the region transition probability, speed and duration respectively.

A. Region transition probability

A travel path of a taxi can be simplified as a multi-hop process, in which a hop indicates a load/drop event happened. Seeing that, we define a *region transition probability* to figure out the probability of the next hop falling in a different region j from the current region i. Particularly, two successive events are different. Likewise, the region i and j are recognized by different metrics, that is, drop or load event distribution. It is more reasonable. For an instance, if the taxi is occupied, the next hop event is the drop one. Hence, choosing a target region from a region set divided by drop event distribution is more logical.

To calculate the region transition probability, the **region recognition process** should be executed in advance.

Firstly, we divide the area into 100×100 grid, and define cells in it as equation 1. Then, we consider region as adjacent cells as equation 2.

$$CELL_{x,y} ::= \{(lon, lat) | x \leq \frac{lon}{len_x} < x+1, \\ y \leq \frac{lat}{len_y} < y+1\} \tag{1}$$

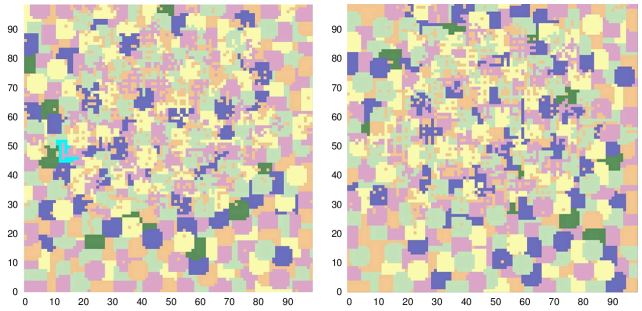
$$REGION_m ::= \{CELL_{x,y} | \exists CELL_{i,j} \in REGION_m \\ \Rightarrow \|x-i\| \leq 1, \|y-j\| \leq 1\} \tag{2}$$

By clustering cells, two region sets, $\{REGION_m^{load}\}$ and $\{REGION_n^{drop}\}$, can be recognized. The main idea of clustering is to put adjacent cells with event density larger than the event threshold η into the same region. To avoid the size of the region become too large or too small, we set a constant $CLUSTERSIZE$ to restrict a region, say $\|REGION_i\| \leq CLUSTERSIZE$, and only limit the top 200 regions, in which $CELL_{x,y}.events \geq \eta$. After that, the other cells not belong to the top 200 regions, will also be classified into regions, while $\|REGION_j\| \leq CLUSTERSIZE$. Consequently, every cells will be classified into regions and the size of each region is not larger than $CLUSTERSIZE$.

We sort the 100×100 cells by event density in descending order, and begin with the first cell to search its neighbors to ask them whether to join the same region using breadth-first traversal. The region recognition results for load/drop events are shown in Fig. 4.

In Fig. 4, every colored block presents a region. In addition, the $CLUSTERSIZE = 200$, $\eta = 121$ for on event and $\eta = 141$ for drop event set by the average event density of the top 5000 cells order by its event density. The detailed clustering algorithm is presented in the appendix.

The calculate process of the region transition probability: After classifying cells into regions, the transition probability from $REGION_i^{load/drop}$ to $REGION_j^{drop/load}$,



(a) passenger drop event regions (b) passenger load event regions

Fig. 4: Region recognition

denoted as $p_{i\to j}^{load\to drop/drop\to load}$. To substatiate, the calculate process of $p_{i\to j}^{load\to drop}$ will be introduced in detail. The records in $REGION_i^{load}$ can be acquired from the data set easily. the record amount is denoted as $\|RECORDS_i^{load} = \{record|record.location \in REGION_i^{load} \cap record.event = load\}\|$. For $record \in RECORD_i^{load}$, the next event and location can be easily required. Therefore, the record can be associated with the its next hop information to $(taxiid, location_{current}, event, event_{next}, location_{next})$. The $RECORDS_{i\to j}^{load\to drop} = \{record|event = load \cap event_{next} = drop \cap location_{current} \in REGION_i^{load} \cap location_{next} \in REGION_j^{drop}\}$ will be obtained.

$$p_{i\to j}^{load\to drop} = \frac{\|RECORDS_{i\to j}^{load\to drop}\|}{\|RECORDS_i^{load}\|} \tag{3}$$

$$P^{load\to drop} = (p_{i\to j}^{load\to drop})_{m\times n} \tag{4}$$

$$P^{drop\to load} = (p_{i\to j}^{drop\to load})_{n\times m} \tag{5}$$

B. Parameter estimation of $\overline{speed}$ distribution

In this section, we modeled the speed distribution. We model the cumulative status average speed distribution to get the cumulative probability distribution function, and then take a derivative with it to obtain the speed probability distribution.

From figure. 1, the $\overline{speed}$ distribution shows exponential law. Given that, we define the function form as follows:

$$\begin{cases} f(x) = 1 - 1/exp(a_1 x^{1.5}) \\ g(x) = 1 - 1/exp(a_2 x^{2.5}) \end{cases} \tag{6}$$

The fit formulas are given as formulas 6. $f(x)$ is the function form for the $\overline{speed}$ distribution of vacant status, and $g(x)$ is for that of occupied status.

TABLE II: The parameter and the rms of residuals of fitting curves

Categories	rms of residuals	
$f(x)$	$a_1 = 0.117$	0.0113159
$g(x)$	$a_2 = 0.0586$	0.0137029

The rms of residuals for each fit are as Table. II. The smaller rms of residuals means better fitting. In the table, the values are all less than 0.02, showing good similarity.

IV. Model Verification

In this section, STRAT mobility model is validated on the aspects of node distribution and contact characteristics. All mobility models are implemented on Opportunistic Networking Environment (ONE)[6].

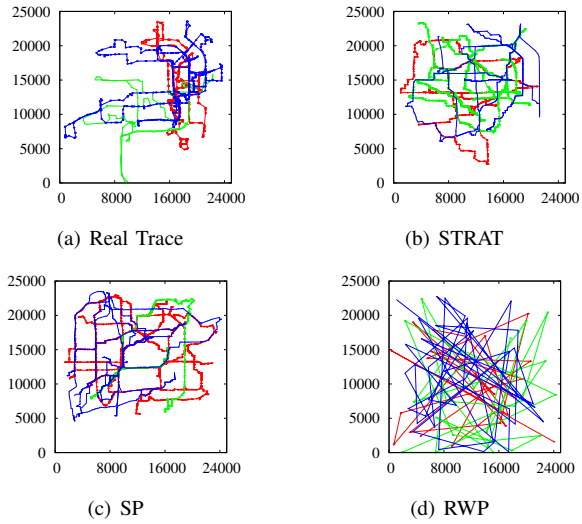
Shortest Path (SP) mobility model based on the map in Beijing is a model for comparison, which is implemented by ONE. A node of SP also moves along the map roads based on Dijkstra algorithm. The RWP model is another comparison model, because it is proved to be an efficient model modeling the nodal movement in VANETs. But it takes no consideration of the node statuses and geographical distribution.

The STRAT, SP and RWP mobility model are compared with the real trace. In simulations, Node number is set as 4000 and the scenario size is $24445 \times 23584m^2$ (a sub-map of the whole area), including fourth ring roads in Beijing. The simulation time is three hours and the warm up time for reports is one hour, so that the nodal movement and position will not be affected by its initial position. The communication range is $200m$.

A. Traces and distribution of nodes

Trace samples and their snapshots are demonstrated in this section, shown in Fig. 5 and Fig.6.

Fig.5 shows the trace in one day. The trace of the real data and STRAT only cover some part of the area, while the trace of SP and RWP almost go through the whole area. Because SP and RWP select destinations randomly, while STRAT takes the associations between current region and destinations into consideration based on the movement rules of taxis.



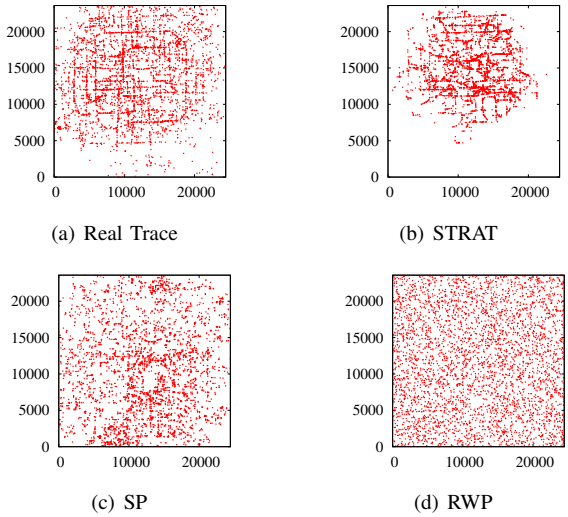
(a) Real Trace (b) STRAT

(c) SP (d) RWP

Fig. 5: Trace samples

From fig.6, Real trace STRAT and SP mobility model exhibit the road structure. However, the node distribution of RWP is much uniform. As to STRAT, the destination section

process decides that it tends to select a destination in the regions with higher load/drop event probability. Therefore, with the decline of the randomness, the snapshot of STRAT become much clearer and more centralized on the main roads.



(a) Real Trace (b) STRAT

(c) SP (d) RWP

Fig. 6: Trace snapshots

B. Contacts characteristics

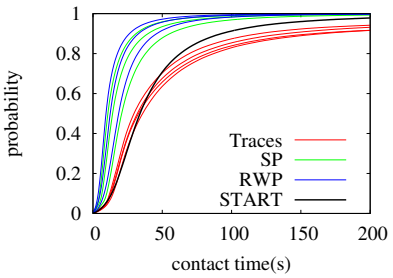
The contact time and inter contact time are evaluated as the indicators to validate the similarity. The speed range of RWP and SP need to be configured. To ensure the accuracy, we choose three speed range $[0, 44.4]m/s$,$[0, 33.3]m/s$ and $[0, 22.2]m/s$, that is, the upper bounds of speed are 80,120,160 km/h.

Fig. 7. (a)(b) is the contact time and inter-contact distribution, which shows the probability of the contact or inter-contact time smaller than certain time length. To substantiate, a point $(200, 0.5)$ in these figures means the probability is 0.5 when contact or inter-contact time is shorter than $200s$. In addition, a point (x, y) in 8.(a) means the sum of contact time is y when the simulation time is x. In this figure, the three green lines of SP and the three blue lines of RWP are coincide with each other. To recognize the differences, we set y axis as log-scale in fig.8.(b). The curves of the total contact time vs. time presents a linear law. For SP and RWP, the differences of speed ranges show little influence on these curves. Clearly, the rank of the contact characteristic similarity with the real data is $STRAT > SP > RWP$.

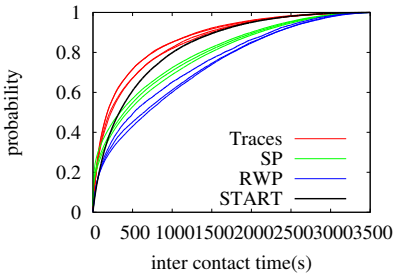
To conclude, by comparing the node distribution and contact characteristics, the STRAT mobility model performs good similarity with the real data. The evaluation results conform to our expectations. Because $STRAT$ takes use of speed and geographic features related to status, however SP employs the map information and RWP is a random model taking use of no realistic data.

V. Conclusion

Since the mobility model is important for mobile network, a novel mobility model STRAT based on real GPS data is proposed. By assuming the taxi behavior is related with its statuses and geographic features, statistical experiments are conducted to verify those assumptions using the real trace data. Further, its parameter–$\overline{speed}$ of each status is estimated respectively, and the region transition probability is calculated. In this case, macroscopic movement, a node moves between load-event regions and drop-event regions, and microscopic movement(speed for each status) can be defined. Finally, the STRAT is implemented on ONE simulator and is estimated by comparing with the real trace, RWP and SP mobility Model. Comparing the node distribution and contact features, STRAT shows better performance. Simulation results demonstrate that STRAT gives a good approximation of reality.

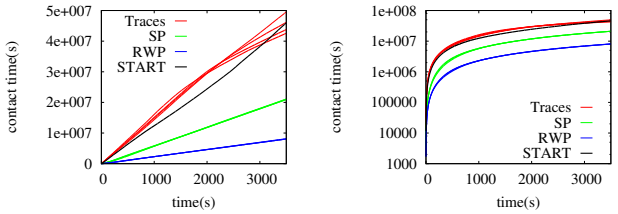


(a) cumulative contact time distribution



(b) cumulative inter contact time distribution

Fig. 7: Contact time, inter contact time distribution



(a) time vs. total contact time (b) time vs. total contact time in logscale

Fig. 8: time vs. total contact time

References

[1] S. Yousefi, M. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): Challenges and perspectives," in *ITS Telecommunications Proceedings, 2006 6th International Conference on*, June 2006, pp. 761–766.

[2] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces." in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, vol. 6, 2006, pp. 1–13.

[3] H. Huang, Y. Zhu, X. Li, M. Li, and M.-Y. Wu, "Meta: A mobility model of metropolitan taxis extracted from gps traces," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010, pp. 1–6.

[4] R. Ganti, M. Srivatsa, A. Ranganathan, and J. Han, "Inferring human mobility patterns from taxicab location traces," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13. New York, NY, USA: ACM, 2013, pp. 459–468.

[5] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 141–154.

[6] A. Keraen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.

APPENDIX

Algorithm 1 Clustering

Require: $Cells = \{CELL\}$
η ##events threshold
$CLUSTERSCALE$
$REGIONSEED = 200$
$ClusterQueue = \varnothing$
$UsedCells = \varnothing$
Sort $Cells$ by events DESC
for $CELL_{x,y} \in Cells$ **do**
  if $CELL_{x,y} \notin UsedCells$ **then**
    $CELL_{x,y}.region = REGIONSEED$
    $size = 1$
    $CLUSTERSEED = CLUSTERSEED - 1$
    $ClusterQueue.enqueue(CELL_{x,y})$
    $UsedCells.add(CELL_{x,y})$
    while $ClusterQueue \neq \varnothing$ **do**
      $CELL_{x,y} = ClusterQueue.dequeue()$
      if $REGIONSEED \geq 0$ **and** $CELL_{x,y}.events \geq \eta$ **then**
        $enqueueNeighbor(CELL_{x-1,y})$
        $enqueueNeighbor(CELL_{x-1,y-1})$
        $enqueueNeighbor(CELL_{x-1,y+1})$
        $enqueueNeighbor(CELL_{x+1,y})$
        $enqueueNeighbor(CELL_{x+1,y-1})$
        $enqueueNeighbor(CELL_{x+1,y+1})$
        $enqueueNeighbor(CELL_{x,y-1})$
        $enqueueNeighbor(CELL_{x,y+1})$
      else
        $enqueueNeighborOthers(CELL_{x-1,y})$
        $enqueueNeighborOthers(CELL_{x-1,y+1})$
        $enqueueNeighborOthers(CELL_{x-1,y-1})$
        $enqueueNeighborOthers(CELL_{x+1,y})$
        $enqueueNeighborOthers(CELL_{x+1,y-1})$
        $enqueueNeighborOthers(CELL_{x+1,y+1})$
        $enqueueNeighborOthers(CELL_{x,y-1})$
        $enqueueNeighborOthers(CELL_{x,y+1})$
      end if
    end while
  end if
end for

Algorithm 2 $enqueueNeighbor(CELL_{x,y})$

if $CELL_{x,y}.events \geq \eta$ **and** $size < CLUSTERSCALE$ **and** $CELL_{x,y} \notin UsedCells$ **then**
  $ClusterQueue.enqueue(CELL_{x,y})$
  $CELL_{x,y}.region = REGIONSEED$
  $UsedCells.add(CELL_{x,y})$
  $size = size + 1$
end if

Algorithm 3 $enqueueNeighborOthers(CELL_{x,y})$

if $size < CLUSTERSCALE$**and** $CELL_{x,y} \notin UsedCells$ **then**
  $ClusterQueue.enqueue(CELL_{x,y})$
  $CELL_{x,y}.region = REGIONSEED$
  $UsedCells.add(CELL_{x,y})$
  $size = size + 1$
end if