# Traffic mining in a road-network: How does the traffic flow?

Irene Ntoutsi*

Department of Informatics,
University of Piraeus, Greece
E-mail: ntoutsi@unipi.gr
*Corresponding author

Nikos Mitsou

School of Electrical and Computer Engineering,
National Technical University of Athens, Greece
E-mail: nmitsou@mail.ntua.gr

Gerasimos Marketos

Department of Informatics,
University of Piraeus, Greece
E-mail: marketos@unipi.gr

**Abstract:** The flow of data coming from modern sensing devices enables the development of novel research techniques related to data management and knowledge extraction. In this work, we undertake the problem of analysing traffic in a road network so as to help the city authorities to optimise traffic flow. A graph based modelling of the network traffic is presented which provides insights on the flow of movements within the network. We exploit this graph in order to analyse the traffic flow in the network and to discover traffic relationships like propagation, split and merge of traffic among the road segments. First experimental results illustrate the applicability and usefulness of our approach.

**Keywords:** traffic mining; traffic flow; traffic relationships; data mining.

**Reference** to this paper should be made as follows: Ntoutsi, I., Mitsou, N. and Marketos, G. (2008) 'Traffic mining in a road-network: How does the traffic flow?', *Int. J. Business Intelligence and Data Mining*, Vol. 3, No. 1, pp.82–98.

**Biographical notes:** Irene Ntoutsi is a PhD candidate at the Department of Informatics, University of Piraeus (UniPi), Greece. She received her Diploma of Computer Engineering and Informatics (2001) and her MSc Degree in Computer Engineering and Informatics (2003) both from the Computer Engineering and Informatics Department of the University of Patras. Her research interests include data mining, pattern management and machine learning.

Nikos Mitsou is a PhD candidate at the School of Electrical and Computer Engineering at National Technical University of Athens (NTUA), Greece.

Copyright © 2008 Inderscience Enterprises Ltd.

He received his Diploma of Computer Engineering and Informatics (2002) from the Computer Engineering and Informatics Department of the University of Patras and his MSc Degree in Automation Systems (2004) from NTUA. His research interests include artificial intelligence on mobile robots, data mining and machine learning.

Gerasimos Marketos is a PhD candidate at the Department of Informatics, University of Piraeus (UniPi), Greece. He received his Bachelor of Science Degree (2003) in Informatics from University of Piraeus and his Master of Science Degree (2004) in Information Systems Engineering from University of Manchester Institute of Science and Technology (UMIST), UK. His research interests include spatiotemporal data warehousing and mining, pattern management and scientific databases. He is member of BCS.

## 1 Introduction

Studying people movements within some road network is both interesting and useful especially if we could understand, manage and predict the traffic phenomenon. In essence, by studying the mass flow of traffic data we can monitor the traffic flow and discover traffic related patterns.

Intelligent Transportation Systems (ITS) have been developed allowing better monitoring and control of traffic in order to optimise traffic flow. ITS collect data utilising various technologies such as sensors, live cameras and cellular phone data and in turn, they propose reroute of traffic to avoid congestion. Beyond transportation perspective, information technology provide a variety of applications and technologies for gathering, storing, analysing and providing access to traffic data. Our aim is to present an intelligent traffic management decision support system that incorporates data mining techniques to extract traffic patterns.

We model the road network as a directed graph. We assume that traffic data is collected through sensors placed along the network and we treat these data as time series that can be further analysed so as to discover relationships among the different edges/road segments of the network. We define various relationships between the edges of the network graph: traffic propagation from one edge to some other edge, split of traffic from one edge into multiple edges and merge of traffic from multiple edges into a single edge. These relationships allow us to combine the temporal information provided by the traffic time series with the spatial semantics inherited in the road network. This way, we are able to discover spatiotemporal patterns and support traffic management decisions. In contrary to existing approaches, we do not rely on the distinct moving objects trajectories but on accumulating data provided by the sensors that monitor the traffic circulation within the network.

In Figures 1 and 2 we present real examples of traffic relationships in the Athens road network (the arrows in the figures show the direction of movement). In Figure 1 an example of traffic propagation for the Sygkrou Str is depicted: objects continue to move along the Sygkrou Str. In the same figure, an example of traffic split is also present: objects at the end of Sygkrou Str have two options to enter Poseidonos Str either by turning to the left for the Kalamaki area, or by turning to the right for the Piraeus area.

**Figure 1** Example of traffic propagation and traffic split in the Athens road network


**Figure 2** Example of traffic merge in the Athens road network


In Figure 2 an example of traffic merge is shown: objects enter Vasilissis Sofias Str from Kifisias Str and Mesogeion Str.

These examples show that detecting traffic relationships between the different road segments of a city network is quite interesting and also that the understanding of such relationships would help the city authorities to improve traffic flow and to react effectively in case of some traffic problem, like car accident or road repairing. Moreover, studying traffic relationships in the current city network, the city authorities could better arrange the construction of new roads, the extension of existing ones, the placement of traffic lights and so on.

The rest of the paper is organised as follows. In Section 2, we present the related work. In Section 3, we describe our model of the traffic network and some interesting relationships between the edges/road segments of the network. In Section 4, we present our approach for clustering traffic edges and detecting traffic relationships; to this end, we introduce different distance measures between the traffic edges that capture different semantics of the traffic flow. In Section 5, we present our first experimental results. Conclusions and open issues are described in Section 6.

## 2 Related work

Recently, there has been considerable research on extending traditional data mining techniques to the context of trajectories (see Kuijpers et al. (2007) for a comprehensive survey).

Closer to our research is the work by Liu et al. (2006), where a distributed traffic stream mining system is proposed: the central server performs the mining tasks and ships the discovered patterns back to the sensors, whereas the sensors monitor whether the incoming traffic violates the patterns extracted from the historical data. All the sensors that have a common pattern are grouped into the same cluster. If some violation occurs at some sensor, the sensor sends an alarm to the server, which in turn notify all the members of the same cluster. This framework has been instantiated for frequent episode patterns. However, this work emphasises on the description of the distributed traffic stream system, rather on the discovery of traffic related patterns.

Also, relative to our approach is the work by Li et al. (2007) for the discovery of hot routes in a road network. Hot routes correspond to sequences of road segments with heavy traffic. The road segments that participate in the formation of a hot route should share some common traffic and should also be nearby. To discover hot routes authors propose a density-based algorithm, called FlowScan, which cluster road segments based on the density of the common traffic they share. The algorithm, however, requires the trajectories of the objects that move within the network, thus cannot be applied in our problem settings.

A line of research relevant to our work is that of *spatiotemporal clustering* or *trajectory clustering* that aims at grouping trajectories of moving objects into groups of similar trajectories.

Lee et al. (2007) propose a partition-and-group framework for trajectory clustering that does not consider clustering of the whole trajectories, but rather it groups together common subtrajectories. In the partition step, a trajectory is split into a set of line segments using the Minimum Description Length (MDL) principle. In the group step, similar line segments are grouped into a cluster using a density based clustering method. For each cluster, the representative trajectory is discovered which is defined

as the trajectory that describes the overall movement of the trajectory partitions that belong to the same cluster. In this work, however, the trajectories of the moving objects are required, thus their solution cannot be transferred to our problem settings. Furthermore, this work concerns free movement and no some predefined network like, in our case, the road network.

Something similar is proposed in Giannotti et al. (2007) where the notion of trajectory patterns (T-patterns) is introduced and appropriate trajectory mining algorithms for their discovery have been proposed. Trajectory patterns do not represent real trajectories but sequences of spatial areas of interest that are temporally related. Such areas of interest can be predefined by the user or they can be discovered in a dynamic way using some density-based algorithm.

Kalnis et al. (2005) introduce the notion of moving clusters for discovering groups of objects that move close to each other for a long time interval. At each timepoint, objects are grouped using a traditional clustering algorithm and then the moving clusters are detected by tracing the common data records between clusters of consecutive timepoints. However their method requires the IDs of the objects, and consequently it does not fit in our case where only the number of objects that passes through some edge/road segment is available. Furthermore, their method considers unconstrained environments, whereas we consider objects moving in a pre-defined road network. Moreover, the authors only consider the discovery of a cluster of objects at the next timepoint whereas we consider further relationships like split and merge.

Also relevant to our work is the work on *change detection*. For example, Spiliopoulou et al. (2006) propose the MONIC framework for modelling and detecting transitions between clusters discovered at consequent time points. MONIC includes both external transitions (e.g., survival, split, absorption) and internal transitions (e.g., change in size, change in location). However, their method relies on cluster members, thus cannot be directly applied to our problem settings, where only the number of objects that passes through some network edge is available and not the IDs of these objects.

Li et al. (2006) deal with the problem of anomaly detection in movements of objects. Instead of focusing on trajectories, they propose some methods to generate motif expressions (in order to smooth out the temporal and spatial granularities) and combine this information with other regular or spatiotemporal features in order to enhance the data mining task. However authors rely on information extracted from the trajectory data, thus the considered problem is different from our problem settings.

Shekhar et al. (2001) discuss the application of traditional data warehousing and data mining techniques on sensor network measurements collected from a freeway system. Furthermore, they propose some research directions on detecting traffic flow outliers, discovering spatiotemporal association rules and sequential patterns. However, this work does not thoroughly examine the traffic problem. Authors just present a case study on how to analyse traffic data using traditional techniques.

Nakata and Takeuchi (2004) employ probe-car data for collecting traffic information concerning much larger areas than by traditional fixed sensors. They model traffic time as time series and they apply the Auto Regression Model after removing periodic patterns. However, in this work spatial information is not taken into consideration.

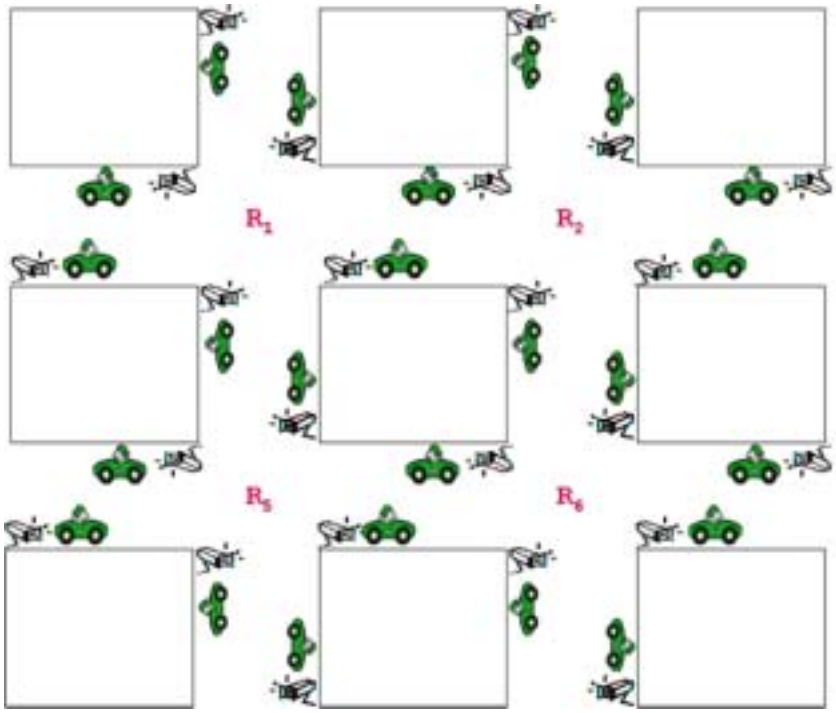
## 3 Traffic modelling

In our model, we consider a predefined network consisting of a set of non-overlapping regions like the one depicted in Figure 3. Regions might be road intersections or landmarks of interest like banks and shopping centres.

**Definition 1:** A traffic network ($TN$) consists of a set of disjoint regions, $\{r_1, r_2, \ldots, r_m\} : (r_i \cap r_j) = \oslash, 1 \leq i, j \leq m$.

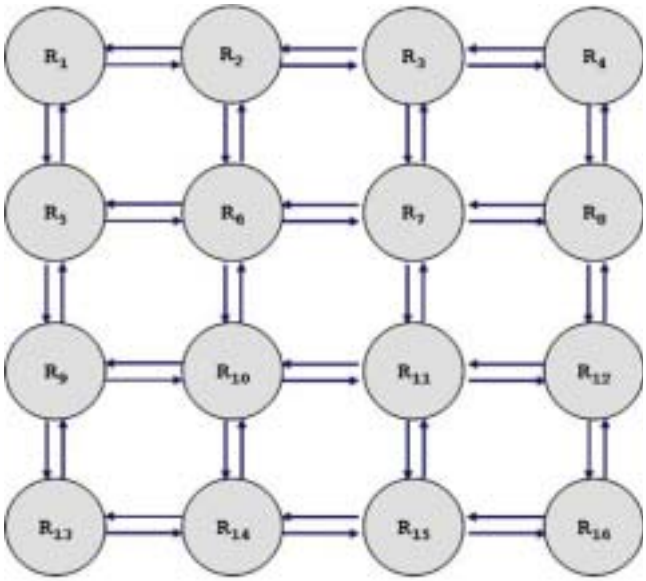
Objects move through these regions according to the network topology. We model the network topology into a directed graph, called *Network Graph* $G = (V, E)$, where $V$ denotes the set of nodes and correspond to the *regions* of the network whereas $E$ denotes the set of edges and correspond to direct connections/road segments between these regions. More specifically, an *edge* $e$ is added between two regions $R_1, R_2$ if there exists some road segment that directly connects $R_1$ and $R_2$. We assume that object movements take place only at the edges and also that as far as an object enters an edge it travels the whole edge. The graph model for the sample network of Figure 3 is depicted in Figure 4.

**Figure 3** A road network example




We denote by $start(e)$ the starting vertex of the edge $e$ and by $end(e)$ the ending vertex of $e$. We call *outgoing edges of* $e$, denoted as $out(e)$, those edges that start from the node $end(e)$. Also, we call *incoming edges of* $e$, denoted as $in(e)$, those edges that end at node $start(e)$.

**Figure 4** A network graph example




In order to collect traffic data, sensors are placed over the network. More specifically, a sensor $s$ monitors the traffic in some edge $e \in E$, thus it transmits information regarding the movement between the corresponding nodes/regions. The traffic of an edge is computed based on the number of cars passed through the corresponding sensor.

Each sensor $s \in S$ transmits the traffic load of its corresponding edge as a series of time stamped measurements $(v, t)$, where $v$ is the number of cars passed through the sensor during the time period $[t, t + \Delta t)$ and $\Delta t$ corresponds to the transmission rate of the sensor, e.g., every 15 min.

**Definition 2:** The **traffic series** of a sensor $s \in S$ during a specific time period of interest $[t_s, t_e]$ is a sequence of the number of cars passed through this sensor during the specific period, i.e., $TS_s = \{v_i, t_i\}$, $t_s \leq t_i \leq t_e$.

An example of a traffic series for a sensor $s$ might be: $TS_s = \{(150, 10:00), (100, 10:15), (80, 10:30), \ldots\}$.

The traffic series of a sensor represents the traffic in the corresponding network edge. The traffic in the whole network is described by a set of time series, each one describing the traffic of a specific network edge.

**Definition 3:** The **traffic** of a network during a specific time period of interest $[t_s, t_e]$ consists of a set of traffic series, $TS = \{TS_s, s \in S\}$, that describe the traffic in the network edges during this period.

We are interested in detecting relationships between the traffic of different road segments of the network. To this end, we rely on the traffic series of the edges in the corresponding network graph.

### *3.1 Traffic relationships*

We propose three different relationships among the traffic of edges/road segments in the network:

- traffic propagation
- traffic split
- traffic merge.

Intuitively, the traffic of an edge might be *propagated* into one of its outgoing edges indicating objects that continue to move within a highway e.g., objects that move along the Sygkrou Str like in Figure 1. In terms of our traffic network model, the traffic propagation would seem as in Figure 5.

Or, the traffic of an edge might *split* into more than one of its outgoing edges indicating objects that leave the highway and follow different directions to their destination, e.g., objects that leave Sygkrou Str and either turn to the left following Poseidonos Str for the Kalamaki area or turn to the right following Poseidonos Str for the Piraeus area (cf. Figure 1). In terms of our traffic network model, the traffic split would seem as in Figure 6.

Inversely, the traffic of an edge might be the result of *merged* traffic from its incoming edges indicating, for example, objects that enter a highway from different directions, e.g., objects that enter Vasilissis Sofias Str from Kifisias Str and Mesogeion Str (cf. Figure 2). In an abstract way, the merge operation would seem as in Figure 7.

**Figure 5** Edge $e_{12}$ propagates its traffic into edge $e_{23}$

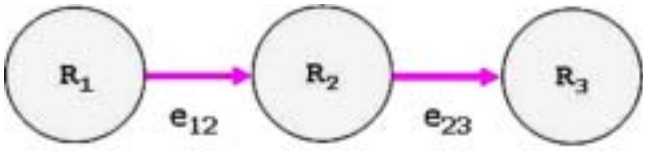
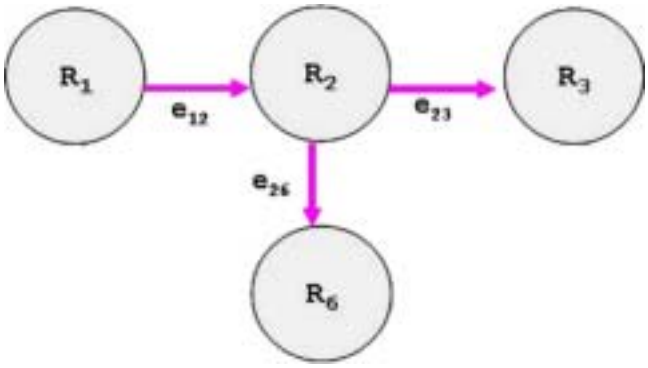
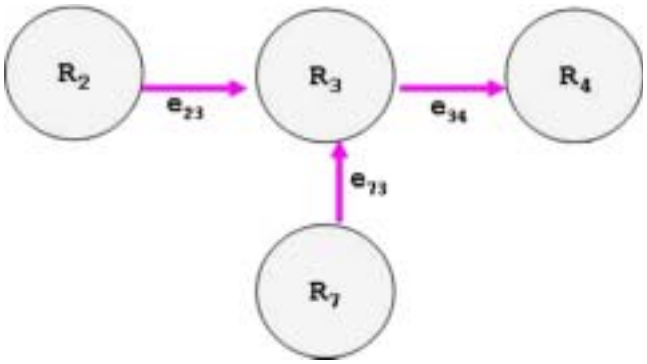



**Figure 6** The traffic of edge $e_{12}$ is split into edges $e_{26}$ and $e_{23}$




So far, we have described three interesting traffic relationships between the edges/road segments of the network. Our approach for detecting such relationships is based on clustering of the network traffic series; we describe our methodology in the next section.

**Figure 7** The traffic of edge $e_{34}$ is the result of incoming traffic from edges $e_{23}$ and $e_{73}$




## 4 Traffic mining

To be able to detect the traffic relationships defined in the previous section, a notion of similarity between the traffic series of the different network edges should be defined. Thus, we first define similarity between the traffic series (Section 4.1) and then, we proceed with the detection of the traffic relationships (Section 4.2).

### *4.1 Similarity between traffic edges*

Let $e_1, e_2$ be two network traffic edges and let $TS_1, TS_2$ be their corresponding traffic series for a specific period of observation $[t_s, t_e]$. In this section, we define distance measures between the two edges based on both their traffic series and network proximity.

**Definition 4:** The **value based distance** between two traffic edges $e_1, e_2$ is given by the Euclidean distance of their corresponding traffic series, $TS_1, TS_2$:

$$dis_{value}(e_1, e_2) = dis_{value}(TS_1, TS_2) = \sqrt{\sum (v_1[t_i] - v_2[t_i])^2}, t_s \leq t_i \leq t_e \quad (1)$$

where $v_1[t_i]$, $v_2[t_i]$ is the value at $t_i$ for $TS_1, TS_2$ respectively.

The value based distance looks for traffic series of (almost) the same values, thus is suitable for detecting whether the traffic of an edge is *propagated* to some of its outgoing edges (cf. Figure 5).

Except for edges of similar traffic values we are also interested in detecting edges of similar traffic shape, i.e., edges whose traffic increases and decreases with the same rate. Euclidean distance is not suitable for this case since it does not allow different baselines in the traffic series (e.g., one series fluctuating around 100 and the other series fluctuating around 30), or different scales (e.g., one series having a small amplitude between 90 and 100 and the other series having a larger amplitude between 20 and 40).

To detect traffic series with similar shape (but not necessary with similar values), we apply normalisation transformations (Das and Gunopulos, 2003) over the original traffic series. Let $\mu(TS), \sigma(TS)$ be the mean and standard deviation of a traffic series

$TS = \{v_i, t_i\}, 1 \leq i \leq n$. The traffic series $TS$ is replaced by the *normalised traffic series* $TS' = \{v'_i, t_i\}, 1 \leq i \leq n$, where:

$$v'_i = \frac{v_i - \mu}{\sigma}$$

and

$$\mu = \frac{1}{n}\sum_{i=1}^{n} v_i$$

$$\sigma = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(v_i - \mu)^2}.$$

**Definition 5:** The **shape based distance** between two traffic edges $e_1, e_2$ is given by the Euclidean distance of their corresponding normalised traffic series $TS'_1, TS'_2$:

$$dis_{shape}(e_1, e_2) = dis_{shape}(TS_1, TS_2) = \sqrt{\sum (v'_1[t_i] - v'_2[t_i])^2}, t_s \leq t_i \leq t_e. \quad (2)$$

Both value based distance and shape based distance rely on the traffic series of the corresponding edges. However, since we know the network topology we can also evaluate the proximity between two edges based on how close these edges are in the traffic network graph. We call this kind of distance structural distance.

**Definition 6:** The **structure based distance** between two traffic edges $e_1, e_2$ equals to the minimum number of edges between $end(e_1)$ and $start(e_2)$. We denote it as $dis_{struct}(e_1, e_2)$.

Value based distance, shape based distance and structure based distance capture different aspects of the distance between two edges in the network graph. We combine all these aspects into a global distance function:

**Definition 7:** The **distance** between two traffic edges $e_1, e_2$ is given as a weighted combination of their corresponding value based, shape based and structure based distance:

$$dis(e_1, e_2) = a * dis_{shape}(e_1, e_2) + b * dis_{struct}(e_1, e_2) + c * dis_{value}(e_1, e_2). \quad (3)$$

The terms $a, b, c$ correspond to the weighting of the value based distance score, shape based distance score and structure based distance score, respectively and indicate how much each of these scores would contribute in the final distance score.

In the next section, we describe our clustering algorithm that exploits the above similarity measures in order to derive a hierarchy of similar traffic edges.

### 4.2 A three level traffic clustering algorithm

We introduced three ways to measure the similarity between two traffic edges based either on their values or on their proximity in the network graph. Each measure reveals different aspects of the traffic problem: the value based measure discovers traffic series of similar values, the shape based measure discovers traffic series of similar shape, whereas the structure based distance discovers edges that are close to each other with respect to the graph topology.

If we consider the three measures separately we realise that each measure further filters the initial set of traffic edges. More specifically, the shape based measure returns groups of edges with similar traffic shape, the structure measure looks further for neighbour edges, and finally the value measure further restricts the result set by searching also for value based similar traffic edges. This rationale provides a hierarchy of traffic flow organised in three levels: the level of *similar traffic shape* edges ($L_1$), the level of *similar traffic shape* edges that are also *nearby* in the graph network ($L_2$) and finally, the level of *similar traffic value* edges ($L_3$). An example of this hierarchy is depicted in Figure 8.

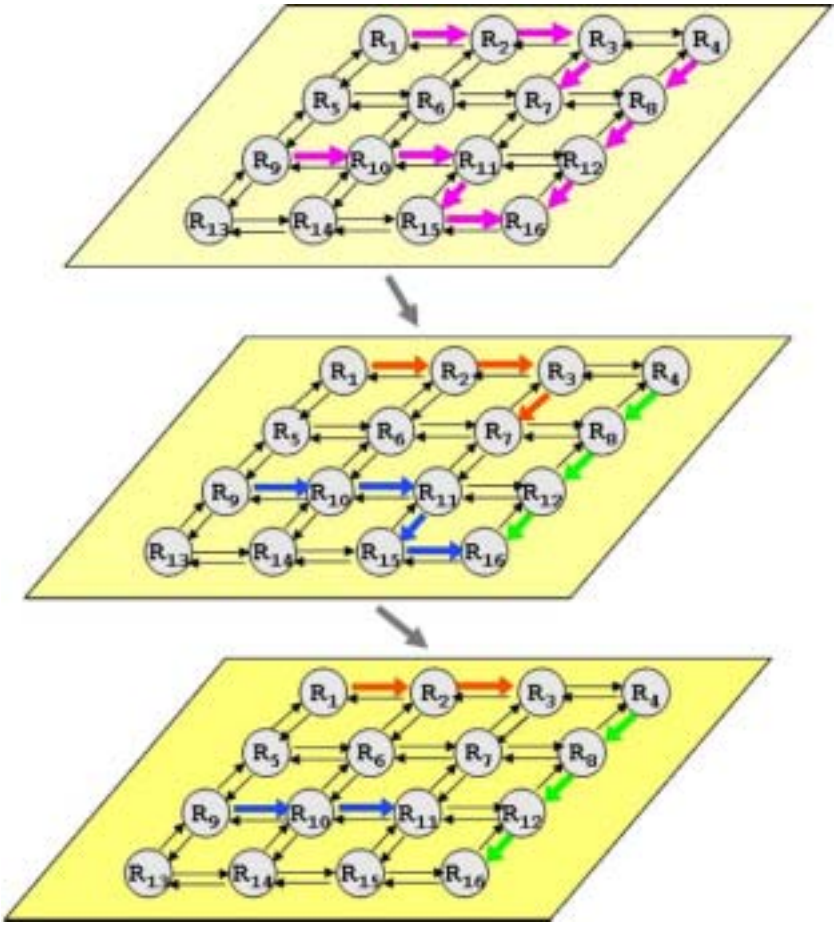
To detect such a hierarchy we employ a *divisive hierarchical clustering algorithm*. The distance measure is that of equation (3), which combines the three notions of distance between traffic edges. Since, as explained before, we are first interested in capturing edges of similar traffic shape, then edges that lie close to each other and finally, edges that have similar values, we instantiate the weights $a, b, c$ of equation (3) according to these requirements, thus: $a >> b >> c$. This way, the algorithm first favours edges of similar traffic shape (weight $a$), then edges that are also nearby (weight $b$) and finally, edges that are also similar with respect to their values (weight $c$).

The algorithm works as follows: Initially all traffic edges are placed in one cluster. At each step of the algorithm, a cluster is further split into subclusters according to the following three steps:

- *Step 1 [Edges of similar shape]*: A cluster is split into subclusters based on the shape similarity of its traffic edge members (cf. Definition 5). This process is continued until a split is caused by the next distance measure, the structure based distance. At the end of this step, the clusters contain edges with similar traffic shape.
- *Step 2 [Nearby edges]*: The clusters generated by the previous step are further split based on the structural distance measure (cf. Definition 6) until a spit is caused by the traffic values distance. At this moment, the clusters contain neighbouring traffic edges with similar shape.
- *Step 3 [Edges of similar values]*: The clusters generated by the previous step are further split based on the similar values distance (cf. Definition 4). At the end of the execution, the clusters contain neighbouring edges with similar values, and similar shape as well.

Discovering such a hierarchy of traffic with respect to the network edges is useful for the domain expert (e.g., city authorities), since she/ he is capable of drilling through the hierarchy from some generic group of edges with similar traffic shape, to groups of nearby edges and then to groups of edges with similar traffic values.

**Figure 8** The traffic edge hierarchy–edges of the same colour (mauve, green, orange, blue) belong to the same cluster (for colours see online version)




## 5 Experimental results

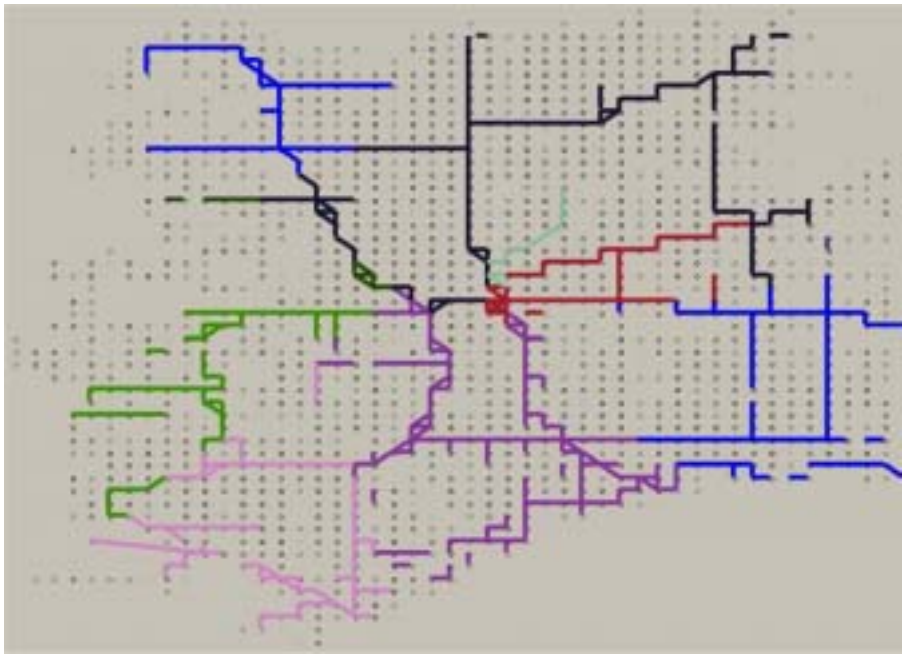
The experiments presented in this section aim at demonstrating the applicability and usefulness of our approach.

Due to the lack of real data, we used synthetic data generated by the network-based data generator developed by Brinkhoff (2007). In particular, we generated 2000 moving objects with 400 sampled positions per each object. The network used in our experiments is depicted in Figure 9.

Before we proceed with the experimental results, we should note at this point that we do not consider the delay of travelling from one edge $e$ to some of its outgoing edges. Rather, we assume that the time required to cross the edge $e$ is smaller that the transmission rate of the sensor that monitors this edge. Also, we applied a threshold value so as to exclude from the clustering process the edges that were rarely visited by moving objects during the observation period.

**Figure 9** The original traffic network

**Figure 10** Results of Layer 1 based on grouping of edges with similar traffic shape (clusters are depicted in different colours) (for colours see online version)



In Figure 10, we present the clusters that were collected at the first step of the clustering algorithm, where the grouping of edges is based on their traffic shape similarity. Edges with the same colour indicate network areas that share similar traffic shape. As we can observe in this figure, there are areas in the network that share the same traffic shape, even if they are non-connected. For example, the cluster depicted with the blue colour is composed of three sub-areas located in different network places. This means that there exist three non-connected areas in the network that share the same traffic shape and thus, all together, they form a cluster (the blue one).

In Figure 11, we present the clusters that were generated by the second step of the clustering algorithm, where the grouping of edges also considers their proximity in the network graph structure. Comparing to the first step of the algorithm (cf. Figure 10), we can observe that old clusters where split into new clusters, because of the proximity based distance measure applied in this step.

For example, the blue cluster described in the first step, was replaced by three new clusters (a yellow one, a blue one and a turquoise blue one). Obviously, this split took place because, as explained earlier, the initial cluster contained disjoint areas, which were placed by the distance measure applied in this step into different clusters.

Finally, in Figure 12, we present the result of the third step of the clustering algorithm, which looks further for traffic edges of similar values. We can see that many new clusters have emerged. The new clusters indicate areas that share the same traffic values. All three relations can be depicted in this figure. The most common relation, however, is the traffic propagation (observed in the previous figures also). Many traffic splits and merges can also be detected.

**Figure 11** Results in Layer 2 based on graph closeness grouping (clusters are depicted in different colours) (for colours see online version)

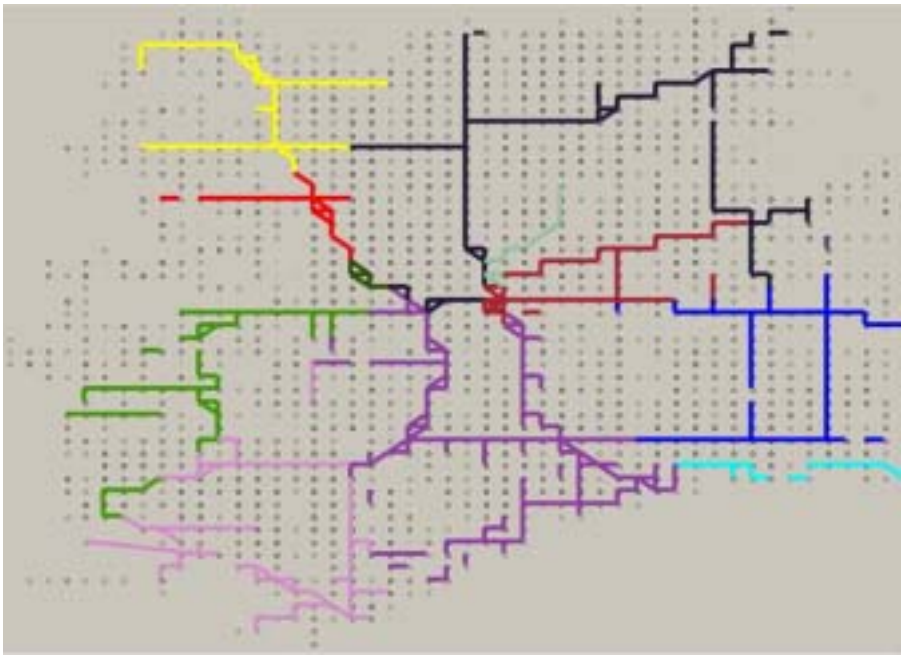
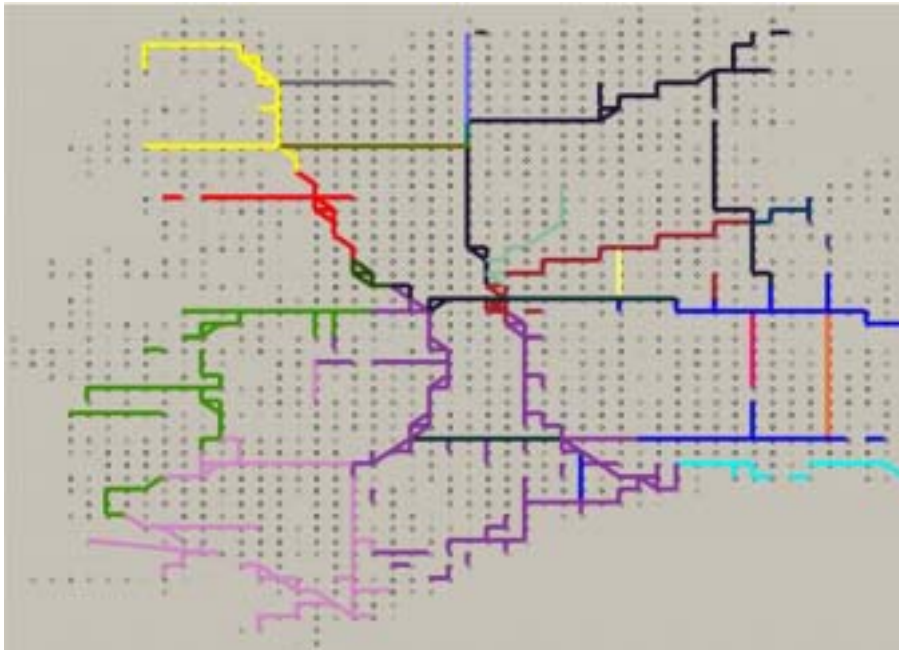


**Figure 12** Results in Layer 3 based on grouping of edges with similar traffic values (clusters are depicted in different colours) (for colours see online version)



As shown from the experiments, our approach can provide useful insights on the traffic problem, whereas the three level architecture facilitates the end user and enables him to monitor the traffic at different levels of abstraction. We are currently running further experiments so as to test the scalability of our method with respect to parameters like the network size, the number of objects moving within the network and the sampling rate.

## 6 Conclusions and outlook

In this paper, we have discussed the problem of traffic management in a road network monitored through sensing devices. We defined different traffic relationships to capture the way that traffic is circulated within the network and proposed a clustering based approach in order to capture these relationships. Our approach provides a hierarchy of groups of similar edges, starting from groups of edges with similar traffic shape, proceeding to groups of nearby edges and finally, resulting into groups of edges with similar traffic values. The first experimental results show that our method can provide useful insights regarding the traffic flow in a road network, e.g., which edges share similar traffic shapes and how these group of edges are modulated when the network proximity or/and the value based similarity is also considered.

Our method is at early stages of development. We are currently enhancing it by considering several issues, like:

- The delay of travelling from one edge to some outgoing edge. So far, we just consider that the transmission rate of the sensor is larger than the time required to cross an edge. However, we plan to consider the shift parameter using some appropriate distance measure like DTW (Das and Gunopulos, 2003) in order the eliminate the shifting effect.
- Discovering more complex relatioships between graph edges. So far we consider propagation, split and merge of traffic. However, more complex cases might appear, e.g., combinations of these primitive relationships like a traffic propagation accompanied by a traffic split and then by a traffic merge, that results after that to the initial traffic series.
- Experimenting on the efficiency of our method under different parameters like the size of the network, the number of moving objects, the length of the consider time period of interest etc. Also, we are investigating appropriate threshold values in order to find when to stop the clustering process at each step.
- Extending our method to continuous monitoring of the traffic network. So far, we analyse the network traffic for a specific period of interest. However, traffic data are dynamic by their nature, so it is also important to online monitor traffic flow in order to detect abnormalities with respect to the expected behaviour (where expected comes from the historical analysis of traffic).

## Acknowledgements

This work is partially supported by FP6/IST Programme of the European Union under the GeoPKDD project (2005–2008). I. Ntoutsi is supported by the Heracletos program co-funded by the European Union- European Social Fund and National Resources- EPEAEK II). G. Marketos is supported by the General Secretariat for Research and Technology of the Greek Ministry of Development under a PENED2003 grant.

## References

Brinkhoff, T. (2007) *Network-based Generator of Moving Objects* (Valid as on 23-06-2007).

Das, G. and Gunopulos, D. (2003) 'Time series similarity and indexing, chapter 11', in Ye, N. (Ed.): *The Handbook of Data Mining*, Lawrence Erlbaum Associates, pp.279–302.

Giannotti, F., Nanni, M., Pedreschi, D. and Pinelli, F. (2007) 'Trajectory pattern mining', *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA, pp.330–339.

Kalnis, P., Mamoulis, N. and Bakiras, S. (2005) 'On discovering moving clusters in spatio-temporal data', *Proc. 9th International Symposium on Spatial and Temporal Databases*, Angra dos Reis, Brazil, pp.364–381.

Kuijpers, B., Nanni, M., Korner, C., May, M. and Pedreschi, D. (2007) 'Spatiotemporal data mining, chapter 10', in Giannotti, F. and Pedreschi, D. (Eds.): *GeoPKDD Book: Geography, Mobility, and Privacy: A Knowledge Discovery Vision*, Springer, pp.275–300.

Lee, J-G., Han, J. and Whang, K-Y. (2007) 'Trajectory clustering: a partition and group framework', *Proc. ACM SIGMOD International Conference on Management of Data*, pp.593–604.

Li, X., Han, J. and Kim, S. (2006) 'Motion-alert: automatic anomaly detection in massive moving objects', *Proc. IEEE International Conference on Intelligence and Security Informatics*, San Diego, CA, USA, pp.166–177.

Li, X., Han, J., Lee, J-G. and Gonzalez, H. (2007) 'Traffic density-based discovery of hot routes in road networks', *Proc. 10th International Symposium on Spatial and Temporal Databases*, Boston, MA, USA, pp.441–459.

Liu, Y., Choudhary, A.N., Zhou, J. and Khokhar, A.A. (2006) 'A scalable distributed stream mining system for highway traffic data', *Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Berlin, Germany, pp.309–321.

Nakata, T. and Takeuchi, J. (2004) 'Mining traffic data from probe-car system for travel time prediction', *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, Washington, USA, pp.817–822.

Shekhar, S., Lu, C-T., Chawla, S. and Zhang, P. (2001) *Data Mining and Visualization of Twin-cities Traffic Data*, Technical Report (TR 01-015), University of Minnesota.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y. and Schult, R. (2006) 'Monic: modeling and monitoring cluster transitions', *Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, pp.706–711.