

ViEWNet: Visual Exploration of Region-Wide Traffic Networks

Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, Matthias Renz

University of Munich, Germany

{kriegel, kunath, pfeifle, renz}@dbs.informatik.uni-muenchen.de

Abstract

Location-based services and data mining algorithms analyzing objects moving on a complex traffic network are becoming increasingly important. In this paper, we introduce a new approach which effectively and efficiently detects dense areas in spatial networks. In an offline phase, we generate a hierarchical partitioning of the traffic network. Thereby, static entities like roads and buildings are likely to be in the same partitioning if they are close to each other according to their network distance. In the online phase, our prototype ViEWNet allows the effective and efficient monitoring of objects moving on a spatial network. Based on a clear visualization of the traffic intensity in each network cell, the user can easily detect hierarchies of dense areas by our powerful prototype ViEWNet.

1. Introduction

The continuous clustering of objects moving on a spatial network is one of the basic operations required for a modern traffic supervision system. Up to now, there exist only snapshot clustering approaches [4] which do not support the continuous detection of dense areas in spatial networks. In this paper, we propose a new approach, called ViEWNet (*Visual Exploration of Region-Wide Traffic Networks*), which generates and visualizes a continuous hierarchical clustering of objects moving on a spatial network. As ViEWNet first generates a hierarchical decomposition of the spatial network in an offline phase, it efficiently supports positional updates of moving objects in the online phase. Thus ViEWNet is able to handle a large amount of moving objects simultaneously. This demonstration provides the following items:

- different modi for generating a hierarchical partitioning of the traffic network
- to augment the static network tree with online information reflecting the traffic intensity
- to visually explore dense network cells by browsing through the hierarchical network (cf. Figure 1)

2. Partitioning a Spatial Network

In this section, we present several methods for the hierarchical partitioning of the traffic network into regions for which the actual volume of traffic is displayed. Thereby the partitions of one hierarchy level are disjoint and cover the entire network graph. An example network partitioning is shown in the lower part of Figure 1. In the following, we present some example methods which are promising for the generation of adequate hierarchical partitions of the traffic network:

- **M-tree partitioning.** The simplest way to get adequate space partitions is to perform a bulkload on all node objects of the traffic network by means of the bulkloading M-tree [1]. This method achieves space partitions which are balanced with respect to the number of node objects (which is nearly similar to the number of streets per chunk).
- **CLARANS partitioning.** This method partitions recursively all node objects by means of the clustering technique CLARANS [2] and stores the partitions within the M-tree. The initial clustering corresponds to the leaf level of the M-tree, where the cluster medoids correspond to the node representatives required for the upper tree level. This method generates a fixed number of partitions per level. It differs from the first one as partitions of nearly similar extensions (according to the network distance) are generated but the number of network nodes per partition can vary arbitrarily.
- **User-defined partitioning.** This partitioning is also performed recursively on the node objects, just as the CLARANS partitioning. However, this method allows the user to define the partition medoids. In each tree level, the node objects are assigned to the nearest medoid based on the network distance. Thereby, in each hierarchy level we achieve a kind of Voronoi diagram on the traffic network.

The quality of a certain partitioning method can be determined by means of the fat-factor and bloat-factor [3] which is a standard method to measure the quality of an M-tree.

3. Interactive Network Browsing

The ViEWNet application is connected to a server which manages the complete spatial cluster hierarchy. The server communicates with a set of clients, each handling approximately the same number of objects. Every client knows only the leaf-level of the roadmap partitioning. The partitioning hierarchy is entirely generated on the server.

We assume that each moving object which is monitored by our application is equipped with a cheap device consisting of a GPS receiver, a mini-computer for storing the roadmap and a transceiver. A moving object is assigned to a certain client all the time. At the beginning of a trip an object sends its position and unique ID to its client. The client replies with a set of road IDs which cross the boundary of the traffic network partition where the object is currently located. By using these IDs and a local copy of the entire traffic network which is e.g. stored on a CD-ROM, the object is aware of the coordinates of the partition boundaries. Until it reaches its destination, the object has to keep track of its current position on the map by using the GPS receiver. As long as the object stays within the boundaries of the current partition, no communication to the client is necessary. Every time it crosses the border, it has to transmit its position to the corresponding client which sends the new partition information as reply. Note that a caching mechanism for the roadmap partitions can be added without much effort to avoid the repeated transmission of the same partitioning informations in case an object repeatedly moves between the same partitions. When an object has reached its destination, it signs off from the client which removes it from the set of currently monitored objects.

ViEWNet demonstrates a continuous hierarchical clustering of thousands of objects moving on a spatial network. Each node of the cluster tree is equipped with a level meter for visualizing the density of the corresponding area as shown in Figure 1. Since the objects are connecting to the clients only when moving from one partition to another, the update cost on the clients can be kept low. Furthermore, the update operation itself is cheap as each client has to decrement the object counter in the old partition and to increment it in the new partition for each border crossing of an object. That way, each client knows how many of its objects are located within a certain partition. Because our partitioning is grid-based and the leaf-level space partitioning is identical on all clients, the server can easily compute the

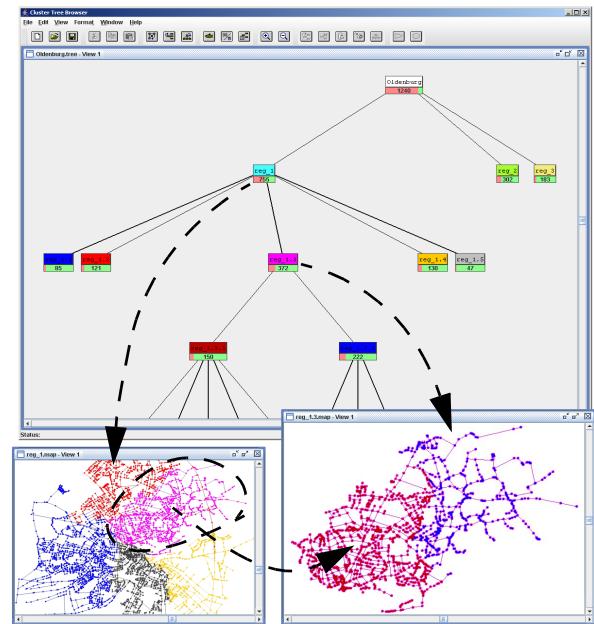


Figure 1: Network exploration with ViEWNet.

overall load of a certain leaf-level partition by summing up the load of that partition on all clients.

At startup of ViEWNet, the user sets the number of moving objects. To simulate a realistic scenario, the user can control the speed and the preferred movement direction of objects by manipulating these parameters. Furthermore, special magnet points can be placed on the road network which either attract or reject the surrounding objects. Our demo visualizes the load of the spatial network based on all these parameters. During the simulation, the user can browse through the cluster hierarchy which is visualized in form of a tree structure as depicted in Figure 1. At each explored node, the aggregated load information of the corresponding partition is visualized as a level meter. Browsing through the cluster tree is supported by different node colors. Each color of a child node corresponds to a different spatial partition, allowing an accurate exploration of the traffic network.

References

- [1] Ciaccia P., Patella M.: *Bulk-loading the M-tree*. ADC'98.
- [2] Kaufmann L., Rousseeuw P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.
- [3] Traina C. Jr., Traina A., Seeger B., Faloutsos C.: *Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes*. EDBT'00.
- [4] Yiu M. L., N. Mamoulis N.: *Clustering Objects on a Spatial Network*. SIGMOD'04.