GENERATING PROGRESSIVELY VECTOR DATA STREAMING FOR ADAPTIVELY MOBILE VISUALIZATION

Bisheng Yang

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China, 430079
bshyang@whu.edu.cn

ThS-4

KEY WORDS: Progressive Transmission, Visualization, Compression, Simplification, Vector Data

ABSTRACT:

This paper proposes and implements a new solution for generating progressive vector lines and polygons data streaming for adaptive visualizations on small mobile devices. The proposed solution firstly recognizes the spatial patterns of vector data via a pattern recognition method, then generates a coarser Level_of_Detail of vector data with the preservation of main characteristics of vector data. Secondly, the solution generates continuous vector data through constrained remove operations of vertices whilst maintaining consistent topology via a set of constraint rules. Finally, the solution generates a compressed vector data streaming via a clustering method. The compressed vector data streaming is decoded for progressive visualization on small mobile devices. Experiments show good qualities of the resulting visualizations. The solution is shown to be scalable to large datasets, to produce graphically acceptable results and to automatically set a viable resolution according to scales.

1. INTRODUCTION

The internet has become the primary tool for both information and data delivery in many fields, ranging from academia through industry to the home user, and to various terminals (e.g., PDA, mobile phones). Contemporaneously geographic information, not just in the form of maps, but also data, has been transformed from the preserve of a small clique of "data keepers" to having widespread availability through web-based geographic information systems (Web-GIS) to an equally wide range of users. Vector map data are widely used in Mobile GIS, Location Based Services (LBS), mobile computing/query, vehicle navigation, web mapping, and so on. The storage of vector data, particularly of large scales, such as are typically appropriate in applications with a very local outlook, has a large impact on both transmission, rendering and response times especially for small devices with low bandwidth and limited processing power. Typical vector data volumes encountered in detailed vector data sets are therefore likely to prove difficult to handle on mobile devices. On the other hand, varying requirements for vector data from different clients (e.g., PDA, mobile phones) have also lead to specific requirements in the rapid delivery of vector data. For example, some clients may require vector data for rapid visualization, while others may require it for spatial analysis. Thus, the representation of vector data should be flexible enough to meet these multiple requirements and be able to dynamically deliver multiple levels of vector data. To overcome this dilemma, this paper proposes an efficient solution to represent vector data as a progressively spatial data streaming. The progressively spatial streaming can thus be adaptively visualized on mobile devices at different Levels of Detail (LoDs) according to displaying scales, data volumes. Thus a good balance between response time and data granularities in Web environments was achieved. The proposed solution was implemented and integrated in a client-server system prototype to explore its feasibilities in terms of adaptive visualizations on mobile devices and progressive transmission. Following the introduction section, related studies are reviewed. Then, the methodology of the proposed solution is elaborated. A set of experiments are presented to illustrate the resulting datasets in terms of their topological consistency, the quality of the resulting visualisations before the conclusions are drawn at the end of this paper.

2. RELATED RESEARCH

Extensive research has focused on the rapid delivery of raster data (e.g. images) over the internet. This has led to many sophisticated algorithms and compression methods, for example, the JEPG 2000 industrial standard, the wavelet transformation method (Martin and Bell 2001), the progressive transmission of images (Rauschenbach and Schumann 1999), and the triangulated irregular network TIN streaming of generating raster DEM (Isenburg *et al.,* 2006). For the delivery of vector data, much existing research has focused on triangle based vector data. Several algorithms and models have been successfully implemented to transmit TIN based vector data over the internet (Park *et al.* 2001). In recent years, much attention has been revolved to the development of data reduction techniques and generalization operators for vector data (e.g., street maps, road networks). Nevertheless, more challenging is still the compression and multiple representations of vector data because of several constraint factors, such as topological consistency, pattern preservation. Topological consistency is an essential property for vector data usability, the answer to a query on the coarser vector data must be consistent with the answer obtained when the same query is applied to the original (fully detailed) map. On the other hand, a low LoD representation of vector data must maintain the main characteristics of the original vector data. For example, the multiple representation of road network should preserve connectivity, density distribution, and street structure. Mackaness and Edwards (2002) suggest that structure and pattern recognition is an essential initial step of multiple representation of vector data.

A lot of algorithms and methods have been developed to generate multiple representation of vector data via generalizations (e.g., simplification, collapse) (e.g., Buttenfield 2000). Two kinds of complementary approaches are mainly used to store and progressively deliver vector data at different LoDs. The first one aims at generating and storing vector data at multiple LoDs and delivering vector data upon user request. But this approach lacks flexibility and it is hard to retain consistent topology within the various levels of vector data. Moreover, it is difficult for this method to dynamically deliver multiple levels of vector data according to varying requirements. The second one is to generate vector data at multiple LoDs and store and transmit them in order of incremental details. Namely, only is the links between multiple LoDs stored and delivered upon request. Han and Bertolotto (2003) reported a prototype system for progressive transmission vector data based on a set of cartographic generalization principles. The performance of the prototype system, however, has not been reported. Yang *et al*. (2004) implemented an algorithm for the progressive transmission of polygon data through a vertex removal method. This approach consists of two key steps: simplification and reconstruction. It extracts continuous levels of vector data through the simplification operator and can dynamically and efficiently transmit vector data with increasing levels of detail. A multi-resolution vector map data model is presented in Yang (2005). This model is efficient for the progressive transmission of polygon data, but does not address the problem of linear vector data. Sester and Brenner (2004) presented a set of elementary generalization operations for visualization on small mobile devices, and two generalization applications, generalization of building ground plans and typification of buildings, are implemented based on a set of elementary generalization operations. The elementary generalization operations are efficient for, but also restricted to the progressive transmission of building ground plans. Ai *et al*. (2004) presented a Changes Accumulation Model for the hierarchical decomposition of polygons into a series of convex hulls or bounding rectangles. The hierarchical decomposition of polygons only considers the topology within one polygon, and it does not consider the topological relationships between the polygon and other objects (e.g. lines, polygons). Therefore, it is hard for the hierarchical decomposition method to maintain consistent topology within a group of polygons (e.g., a polygonal network). Yang *et al* (2007) extended the work of Yang *et al* (2004) and Yang (2005) by including multiple object types, introducing a new set of rules for maintaining topological consistency, which are valid for both line and polygon data and developing a more efficient data structure for the transmission and reinstatement of removed vertices. However, a drawback of Yang *et al* (2007) is that they only adopt vertex removing operator for the multiple representations of vector data. This may limit the simplification of original vector data such as a grid-structured street map as most vertices are not allowed to be removed according to the rules of Yang *et al* (2007).

It is clear from the research reported above that the multiple representation of vector data is still an open problem. However, to the authors' knowledge, a solution has not yet been suggested for realistic map datasets consisting of multiple object types (e.g., lines and polygons) which both conserves topological consistency and has been implemented to demonstrate real (as opposed to conceptual) improvement in efficiency. This paper proposes a solution to generate vector data at multiple representation and store them in order of incremental details, which is compressed as a spatial data streaming according to a clustering approach implemented in Yang *et al* (2008). A big difference from the work of Yang *et al* (2007) is the fact that the proposed solution firstly detects the spatial pattern of vector data according to semantic data and geometric shape of spatial objects before vertices are removed. Thus, this solution is able to overcome the drawback of the work of Yang *et al* (2007).

3. METHODOLOGY

3.1 Framework of the proposed solution

Multiple representations of vector data aims at generating vector data at different LoDs and preserving the main characteristics of vector data whilst maintaining topological consistency. The vertices removing operator of Yang *et al* (2007) is able to preserve the pattern of a single spatial object (e.g., lines, polygons). However, it might have difficulties to maintain the pattern of a group of spatial objects. Therefore, it is necessary to firstly recognize the pattern of vector data before generating multiple representation. This study mainly focuses on the streaming generation of networks (e.g., road network, river network). The proposed solution consists of hierarchical decomposition of vector data, multiple representations of vector data, and spatial streaming compression of vector data. Figure 1 illustrates the framework of the proposed solution.

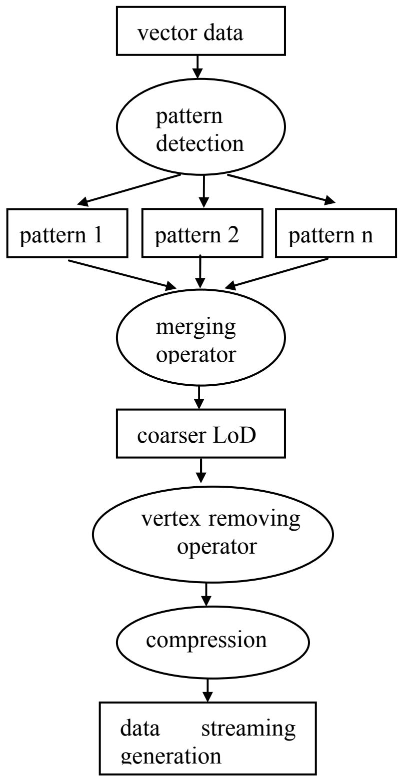


Figure 1. The framework of generating vector data streaming

3.2 Pattern detection of vector data

Patterns can be defined as a property (e.g., shape, orientation, density) within an object or between objects that is repeated regularly (Zhang 2004). In order to maintain the main characteristics of original vector data, implicit information has first to be mined according to pattern recognition or spatial analysis. Semantic data and spatial shapes of spatial objects are the main sources for discovering the pattern or main characteristics of vector data (e.g., road network, street map, administrative regions). Heinzle *et al* (2005, 2006) focus not on the importance of every road in the network, but rather on the identification of specific network patterns such as star-shaped, grid-like, and ring-shaped road patterns.

Inspired by their work, we decompose vector data as a hierarchical group based on semantic data, geometric shapes, and topological relations between spatial objects. The procedure of hierarchical grouping of vector data is shown in Figure 2.

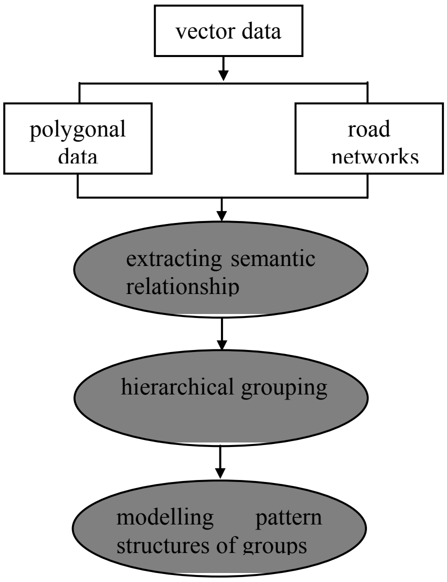


Figure 2. Hierarchical group of vector data based on sematic data

Administrative hierarchies of networks provide a good classification of network that reflects the organization of spatial knowledge. As illustrated in Figure 2, the semantic data of networks is firstly extracted and represented as a hierarchical relation model (e.g., UK->Wales->Cardiff). Then, the spatial objects were classified into different groups according to the hierarchical model. For polygonal networks, the polygons of each group are formalized as a large polygon based on merge operator, as illustrated in Figure 3-a. For road networks, the method of Thomson and Richardson (1999) is utilized to formalize strokes derived from the Gestalt principle of good continuation. Strokes are considered to be a basic pattern of the road network that must be preserved. Then the widths, lengths, types of road segments are extracted. The importance of one stroke within the network can be estimated from their length and connectivity property. The more important of a stroke is; the higher ranking of the stroke has. Therefore, the strokes (in red colour) of a road network subdivides the road network into different parts, as illustrated in Figure 3-b.

3-a. grouping polygons based on semantic data

3-b. subdividing networks based on semantic data

Figure 3. Hierarchical grouping of vector data based on semantic relations

As mentioned in the introduction part, the patterns of networks should be well preserved at different LoDs. As far as road networks are concerned, grid-like patterns are more common consisting of two groups of roughly perpendicular roads. The method of Yang *et al* (2007) has difficulties to represent grid-like network patterns as multiple LoDs. Therefore, the grid-like patterns of networks in each group must be detected before multiple representations. Heinzle *et al* (2005) suggested a method of the edge directions histogram to detect grid-like patterns of road networks. However, the method is more time-consuming. In this study, we presented a seeding incremental method to detect and group grid-like patterns in networks. Moreover, merging the grid-like patterns depends on whether they are at the same level of administrative hierarchies. The following steps depict the seeding incremental method.

1. Transform road networks to nodes, segments, and polygons, build topologies between nodes, segments, and polygons;
2. Rank the polygons of 4 segments in the incremental order of the areas of polygons, and store them in a polygon queue;
3. Initialize a queue for the storing of grid-like patterns polygons;
4. Select one polygon of minimized area and search its neighbour polygons;
5. Suppose that the centroids of neighboured polygons are in roughly identical direction and the areas of polygons are roughly same. The two polygons are in the group of grid-like patterns and removed from the polygon queue. The detection stops until the cetroids or areas of polygons do not meet the condition. These polygons will form a group; and
6. Repeat 4-5 steps until the polygonal queue is empty.

Figure 4-a shows the detection of grid-like patterns. The grouped grid-like patterns are denoted in different colors. The grid-like patterns in the same group will be merged to form a large polygon. Figure 4-b illustrates the merging of grid-like patterns.

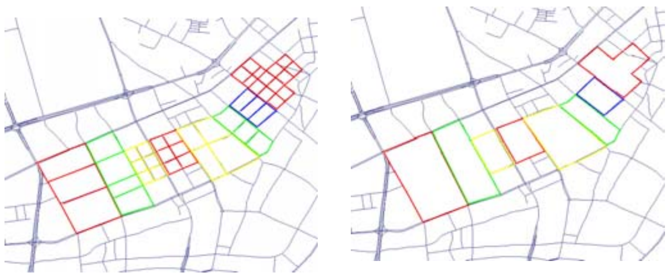


Figure 4. The detection and merging of grid-like patterns

3.3 Multiple representations and compression of vector data

The first step of multiple representations is to merge the polygons or grid-like patterns of the identical group. As illustrated in Figure 4, the merging operator generates a coarser LoD of vector data. It is clear that the merging operator leads to a lot of segments disappearing. Moreover, the segments are the sharing boundary of polygons. To maintain topological consistencies, the topology between disappearing segments and polygons are recorded in a table.

The merging operator generates a coarser LoD of vector data. The main characteristics of vector data are able to be well preserved as the patterns of vector data are detected and maintained after merging operator. To generate coarser LoDs of vector data, we extend our previous work (Yang *et al*, 2007). In our previous work (Yang *et al*, 2007), we propose several rules to maintain topological consistencies. The rules rank vertices as removable vertices and non-removable vertices respectively, and remove vertices in the order of geometric importance. In light of section 3.1 and 3.2, a vector data is represented as a coarser LoD with the maintenance of main characteristics that is modelled as patterns of vector data. As far as polygonal network is concerned, the merging operator generates a coarser LoD with fewer polygons. For road networks, the merging operator generates a coarser LoD with fewer road segments. Then we apply the rules of Yang *et al* (2007) to generate more coarser LoDs. Hence, a vector data is able to be represented as multiple LoDs via the pattern detection and the operator of vertices removing. Moreover, the main characteristics of vector data can be well preserved.

Our real experimental tests show that the removed vertices occupy at least 75% storage space of a vector data. Moreover, the removed vertices are in the format of streaming as they can be stored one by one in a linear queue. The storage of the removed vertices has a large impact on both transmission, rendering and response times especially for small devices with low bandwidth and limited processing power. Data compression is a common approach to both decreasing the storage space required for spatial data and to minimizing the resultant transmission times. We propose a clustering based solution to compress the removed vertices. One advantage of the compressed solution is able to encode the removed vertices as a streaming format. Moreover, the compressed package can be progressively decoded. This is of vital importance for progressive transmission in web environments. For example, the compressed package can be decoded at variable resolutions by fully or partly decoding according to the scales of displaying.

The compression method can be broadly described as follows:

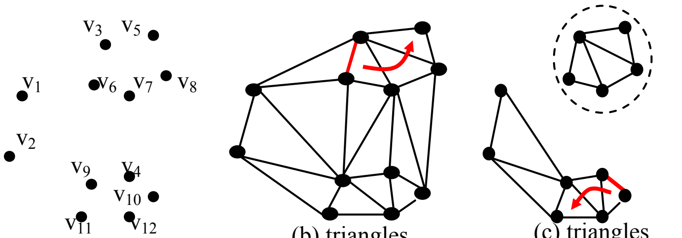
- Identification of candidate vertices for removal by applying the rules of Yang *et al* (2007) – these rules prevent topological inconsistencies from occurring when vertices are removed from the original geometry.

- Calculation of the displacement of removed vertices from simplified geometry and storage in a table together with their associated object ID i and position j in the original object. The identification of candidate vertices for removal and calculation of their displacements is described as:
  $\Delta x_k = x_j - (x_{j-1} + x_{j+1})/2$
  $\Delta y_k = y_j - (y_{j-1} + y_{j+1})/2$
  where $V_j \in O_i$ and $V_j = V_k$

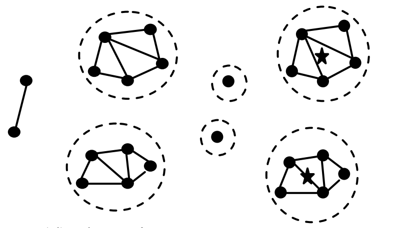
- Based on the displacements calculated, removed vertices are assigned to a TIN and then clustered according to their displacement from the original geometry. Entries in the table describing vertex displacements are replaced with a single reference to the cluster to which they belong.

- Finally, a dictionary is generated to store the mapping the vertices of clusters and the references to the clusters.

The generation of TIN, clustering, and the calculation of the references to the clusters are illustrated in Figure 5.



(a) displacement of removed vertic (b) triangles recursive traverse (c) triangles recursive traverse


(d) clustering generation (e) cluster vertex generation

entry id	entry	mapping
1	w_1	v_1
2	w_2	v_2
3	w_3	v_3, v_5, v_6, v_7, v_8
4	w_4	$v_4, v_9, v_{10}, v_{11}, v_{12}$

(f) dictionary generation

Figure 5. clustering generation and storage from the displacement of removed vertices according to a distance threshold

Figure 5 shows the key steps of the compression method. The compression method generates clusters according to the distances between vertices. Suppose that the distance between vertices is less than a threshold. The vertices form a cluster. Then a new vertex is generated to denote the vertices of a cluster. Thus, the storage space of vertices is reduced. The details of the compression method can be referred in Yang et al (2008). Our previous work in Yang *et al* (2008) shows that the compression method is able to achieve a compression ratio of 60% at least.

3.4 Vector data streaming generation

In light of the depiction of the above sections, a vector data is able to represent as a coarser LoD and a compressed dictionary, which records the removed vertices and topologies between the removed vertices, clusters, and associated spatial objects. To facilitate the progressive transmission and adaptive visualizations, a data structure was implemented to represent the original vector data as a streaming format. The data structure stores 2-level representations of a vector data. The first level is the representation after pattern detection by semantic grouping. It represents a coarser LoD generated by a merging operator. The second level depicts a streaming representation of the coarser LoD generated by a merging operator after removing vertices.

```
Typedef Struct {
   Vector [ ]coarserLoDs; /* record the coarser LoDs generated
                             from a merging operator */
   Dictionary [ ]DicsofLoDs; /* the compressed dictionary of
                             each coarser LoDs */
   Int [ IDofRemovedvertices; /* record the positions of
                             removed vertices at the
                             dictionary */
}STRAMING;
```

where each coarser LoD is represented as a simple spaghetti model.

In light of Section 3.1, a vector data is subdivided as several parts after semantic grouping and pattern detections. Hence, the data dictionary of each part is stored separately. Moreover, the entries of data dictionary are stored as a linear queue, which is easy to represent as a streaming format. When a vector data is delivered or visualized in web environments, the entries of the data dictionary will be delivered one by one after the coarser LoD is transmitted.

4. EXPERIMENTS

(a) LoD1 of polygons (b) LoD2 of polygons

(c) full resolution of polygons (d) LoD1 of road network

(e) LoD2 of road network (f) full resolution of road network

Figure 6. Visualization of polygonal network and road network on mobile device

The three key steps, namely, pattern detection, multiple representations, and streaming compression, of the proposed solution was implemented and tested in a server-client system prototype. Two datasets, a road network and a polygonal network were selected to test the performance of the proposed solution in terms of progressive visualization. The polygonal network dataset was firstly grouped according to administrative hierarchies derived from sematic data. The road network was grouped according to the grid-like patterns detection method and administrative hierarchies. Then the two datasets were simplified by the rules of Yang *et al* (2007) as a coarser LoD and a compressed data dictionary generated by the method of Yang *et al* (2008). Finally, the two datasets were encoded as a steaming format as designed in Section 3.4. We used a mobile device to test the progressive visualizations of the data streaming. To demonstrate the progressive decoding, we implemented a decoder to decode the compressed data streaming. Figure 6 shows the progressive visualizations of polygonal network and road network at a mobile device.

As shown in Figure 6, the compressed data streaming is able to be decoded as a variable resolution representation according to varied requirements. Moreover, the main characteristics of vector data are well preserved at different LoDs. Compared with common visualization strategies, the proposed solution does not load all the data into the main memory for visualization. This is of vital importance for mobile devices with limited memory and weak computing capabilities. As the vector data is represented as a streaming format, the removed vertices can be retrieved one by one and reconstructed to recover to a detailed LoD. The proposed solution is able to maintain topological consistencies as the rules of Yang *et al* (2007) are applied to generate multiple LoDs. Hence, the topology between objects at each LoD can be correctly preserved. Secondly, in light of the vertices removal operator, the reconstruction operator is the reverse procedure, which generates a triangle. The screen space of the generated triangle occupies is calculated as an indicator to control whether the progressive decoding stops. Suppose that the screen space of the generated triangle occupies is less than a threshold (e.g., 3X3 pixels). The progressive decoding will automatically stop. This shows that the proposed method is able to decode the progressive streaming according to displaying scales. That is to say, the proposed method adaptively decodes the progressive streaming according to the screen sizes of mobile devices.

5. CONCLUSIONS

This paper proposes an efficient method to progressively encode vector data as a streaming format, which is of vital importance for small screen devices (e.g., PDA, mobile phones). In light of the framework of the proposed solution, two operators, namely, merging and vertex simplification, are applied to generate the coarser LoDs of vector data. To maintain the main characteristics of vector data, a pattern recognition method was implemented to detect the main patterns of networks (e.g., polygonal networks, road networks), which is beneficial for the preservation of main characteristics of vector data. On the other hand, to further reduce the storage space of vector data, a compression method was implemented to compress vector data streaming, which is necessary for the progressive transmission and visualizations of vector data, particularly in web environments.

Two vector datasets were selected to explore the performance of the proposed method to illustrate its advantages in terms of adaptive visualizations on small mobile devices. Compared with the common visualization strategies, the proposed solution does not load all the vector data into the main memory of mobile devices for visualization. Moreover, it is able to automatically control the resolution of vector data whilst maintaining the main characteristics of vector data, which is of vital importance for interpreting map on small mobile devices. The proposed solution shows a promising approach for visualizing and interpreting large volumes vector data on mobile device, particularly in web environments. Now we are incorporating the proposed solution in car navigation systems and LBS systems. We believe that we will present more interesting results at the 2008 ISPRS conference.

ACKNOWLEDGEMENTS

Work described in this paper was substantially supported by the 863 projects from the Ministry of Science and Technology of the P. R. C (Grant No. 2007AA12Z241, Grant No.2007AA12Z212) and the project from the Outstanding Scholar of the Ministry of Education of the P. R. C (Grant No. NCET-07-0643).

REFERENCES

Ai, T., Li, Z., and Liu Y., 2004. Progressive transmission of vector data based on changes accumulation model. In: *The 11th International Symposium on Spatial Data Handling*, UK, pp.85-96.

Buttenfield, B.P., 2002. Transmitting vector geospatial data across the internet, In: *Proceedings GIScience 2002*, Lecture Notes in Computer Science, Vol. 2748 (Egenhofer and Mark eds), pp.51-64.

Bertolotto, M., and Egenhofer, M.J., 2001. Progressive transmission of vector map data over the World Wide Web. *GeoInformatica*, 5(4), pp. 345-373.

Edwardes, A.J., and Mackaness, W.A., 2000. Intelligent generalisation of urban road networks. In: *Proceedingsof GIS Research UK 2000 Conference (GISRUK 2000)*, University of York, pp. 81－85.

Han, Q., and Betolotto, M., 2003. A prototype for progressive vector transmission with an oracle spatial environment. In: *Proceedings of GIS Research UK 11th Annual Conference*, City of University, London, 9-11 April, 2003, pp.189-194.

Heinzle, F., Anders, K.-H., and Sester, M., 2005. Graph based Approaches for Recognition of Patterns and Implicit Information in Road Networks. *XXII International Cartographic Conference (ICC2005)*(CD-Rom)

Heinzle, F., Anders, K.-H. and Sester, M., 2006. Pattern recognition in road networks on the example of circular road detection. In: *Geographic Information Science - Fourth Int. Conference, GIScience 2006. Lecture Notes in Computer Science 4197*, Berlin, Germany: Springer-Verlag, pp. 153-167.

Isenburg, M., Liu, Y., Shewchuk, J. R., and Snoeyink, J, 2006. Streaming computation of Delaunay triangulations. *ACM Transactions on Graphics*, 25(3), pp.1049-1056.

Martin, M. B., and Bell, A. E., 2001. New image compression techniques using multiwavelets and multiwavelet packets. *IEEE Transactions on Image Processing*, 10(4), pp. 500-511.

Park, D., Cho, H., and Kim, Y., 2001. A TIN compression method using delaunnay triangulation. *International Journal of Geographical Information Science*, 15(3), pp. 255-270.

Rauschenbach, U., and Schumann,H., 1999. Demand-driven image transmission with levels of details and regions of interest. *Computer and Graphics*, 23(6), pp. 857-866.

Sester, M., and Brenner, C, 2004, Continuous generalization for visualization on small mobile devices. In: *The 11th International Symposium on Spatial Data Handling* (Peter Fisher eds), Springer Verlag, 2004, pp. 469-480.

Thomson,R.C.andRichardson,D.E.(1999),The’goodcontinuation ’principleofperceptualorganizationappliedtothegeneralizationofr oadnetworks.In:*Proceedingsofthe19thInternationalCartographi cConference*, ,Ottawa,pp.1215–1223.

Yang, B.S., 2005. A multi-resolution model of vector data for rapid transmission over the internet. *Computers&Geosciences*, 31(5), pp. 569-578.

Yang, B.S., Purves, R.S., and Weibel, R., 2004. Implementation of progressive transmission algorithms for vector map data in web-based visualization. In: *XXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Istanbul, Turkey (CD-ROM).

Yang, B.S., Purves, R, S., and Weibel R., 2007. Efficient transmission of vector data over the internet. *International Journal of Geographical Information Science*, 21(2), pp. 215-237.

Yang, B.S., Purves, R, S., and Weibel R., 2008, Variable-resolution Compression of Vector Data, *GeoInformatica*, DOI 10.1007/s10707-007-0036-x (in press).

Zhang, Q., 2004. Modeling structure and patterns in road network generalization. In: *The 7th ICA Workshop on Generalisation and Multiple representation*, Leicester http://ica.ign.fr (accessed 5 Oct. 2005).