PROGRESSIVE TRANSMISSION OF VECTOR MAP ON THE WEB

AI Bo[a, *], AI Tinghua[b], TANG Xinming[c]

[a]Geomatics College, Shandong University of Science and Technology, Qingdao, China - aibo@sdust.edu.cn
[b]School of Resource and Environment Sciences, Wuhan University, Wuhan, China - tinghuaai@gmail.com
[c]Key Laboratory of Geo-Informatics, State Bureau of Surveying and Mapping, Chinese Academy of Surveying and Mapping, Beijing, China - tang@casm.ac.cn

Commission II, WG II/3

KEY WORDS: Progressive Transmission, Map Generalization, Multi-scale Representation, Internet GIS, Browser/Server

ABSTRACT:

This paper presents a method of progressive transmission of vector data on the web based on the Changes Accumulation Model. The model considers the spatial representation from one scale to another scale as an accumulation of a set of changes, and the change can take place at three levels: layer, object and geometric detail. Taking the example of drainage network, this paper gives a new storage structure without redundancy of geometry for progressive transmission. At the level of object element, the river branches are sorted on descending significance grade which is estimated by drainage area. At the level of geometric detail, a river branch is splitted into sections, and each section is organized by a linear BLG-Tree, enabling selecting the proper representation for a given scale. For a request scale, the optimal number of the needed river branches can be determined by the Topfer rule in map generalization. Furthermore, the good graphic representation of each river branch can be obtained by selecting nodes of the linear BLG-Tree. Finally, this paper presents a browser/server architecture for progressive transmission based on WFS (Web Feature Service), GML (Geography Markup Language) and AJAX (Asynchronous JavaScript and XML). To support the progressive transmission and refinement, some extensions are made to WFS and GML standards.

1. INTRODUCTION

With the development of WebGIS, transmitting large volumes of geospatial data via the Internet has become commonplace. Nevertheless, the low efficiency of geospatial data transmission puts WebGIS in a dilemma. Although the current Internet bandwidth has been expanded many times compared to years ago, it is still unable to meet the increasing requirements of quickly distributing large data sets on the web. Recent research work has proved that the incremental and progressive methods are very useful in quickly transmitting and displaying of spatial raster images, such as satellite data products. For another kind of web data, namely the vector map, how to realize the progressive transmission by this way still presents challenges. Taking the example of drainage network, this paper introduces a method of transmission of vector map data at progressive levels of resolution.

1.1 Requirements for Progressive Transmission

Instead of transmitting vector data by a one-step long process, the progressive transmission first relatively fast delivers and presents a coarser version of a small data volume and then refines it with gradual details. The progressive transmission is:

1. Efficient transmission - By the progressive transmission, the user would avoid waiting to download the complete data and start working with a prompt available coarser version.

2. Self-adaptive transmission - The progressive transmission could present a proper representation according to the resolution or scale of a client map and avoid transmitting over details data.

3. User-controlled transmission – By the progressive transmission, the user could request a transmission equivalent to (e.g.) 30 meter resolution, or appropriate for representation on (e.g.) a 1:25,000 scale display. Further more, in the process of the progressive transmission, if the user satisfies the representation, he (she) could interrupt the transmission at any moment.

As an efficient strategy, the progressive transmission of vector map data becomes an active issue (Bertolotto and Egenhofer 1999, 2001, Buttenfield 1999, 2002, Sester and Brenner 2004, Oosterom 2005, 2006). Nevertheless, due to the intrinsic complexity of vector data structure, to realize progressive transmission still remains challenges.

1.2 Changes Accumulation Model

The progressive transmission of vector map data requires efficient multi-scale data model. This paper pre-organizes the vector map data based on the **Changes Accumulation Model** (Ai, Li and Liu, 2004).

The Changes Accumulation Model considers the spatial representation from one scale to another scale as an accumulation of a set of changes. The initial representation state is S_0. The change between S_i and S_{i+1} is $\triangle S_i$.

$$\triangle S_i = S_{i+1} - S_i$$

Then the ith representation can be derived as the accumulation of a series of changes.

$$S_i = S_{i-1} + \triangle S_{i-1} = S_0 + \triangle S_0 + \triangle S_1 + \cdots + \triangle S_{i-1}$$

---

[*] Corresponding author.

The model indicates that the representation $\{S_i\}$ is composed of the initial representation S_0 and a series of changes. This model, which just records the data S_0 and $\{\triangle S_i\}$, is without redundancy of geometry.

Based on the model, gradual detail additions to the initial representation can realize the progressive transmission from coarse to fine. The change $\triangle S_i$, the difference between consecutive representations, decides the granularity of progressive transmission. **The change can take place at three levels: layer, object and geometric detail.** Only the transmission with granularity at the level of geometric detail is a real progressive transmission.

1.3 Paper Overview

The rest of paper is organized as follows. Section 2 shortly introduces the Binary Line Generalization (BLG)-tree (Oosterom 1990, 1991), which is a famous scaleless data structure of polyline. A method of adapting the BLG-Tree to the progressive transmission is also presented in section 2. Taking the example of drainage network, section 3 presents a new multi-scale storage structure for progressive transmission and the implementation of it. Section 4 offers a browser/server architecture of progressive transmission based on WFS, GML and AJAX. A summary and outlook concludes the paper in Section 5.

2. BINARY LINE GENERALIZATION (BLG)-TREE

The polyline objects are the most important components of a map sheet (The polygon feature can be treated as a closed polyline in some degree). To realize the progressive transmission at the level of geometric detail, the polyline object should be decomposed into details (points). The BLG-Tree (Oosterom, 1990, 1991) is an efficient scaleless data structure for one polyline object. This section introduces the BLG-Tree and presents two steps to adapt it to the progressive transmission.

2.1 BLG-Tree Overview

The BLG-tree stores the result of the Douglas-Peucker algorithm in a binary tree. The original polyline consists of the points P1 through Pn. The most coarse approximation of this polyline is the line segment [P1, Pn]. The point of the original polyline, which has the largest distance to this line segment, determines the error for this approximation. Assume that this is point Pk with distance d, see Figure 1a. Pk and d are stored in the root of the BLG-tree, which represents the line segment [P1, Pn].

The line segments [P1, Pk] and [Pk, Pn] can be treated in the same manner with respect to their part of the original polyline as the line segment [P1, Pn] to the whole polyline. Again, the error of the approximation by a line segment can be determined by the point with the largest distance. And again, this point and distance are stored in a node of the tree which represents a line segment. This process is repeated until the error "distance" is 0. If the original polyline does not contain three or more collinear points, the BLG-tree will contain all points of that polyline. It incorporates an exact representation of the original polyline. The BLG-tree of the polyline of Figure 1a is shown in Figure 1b.

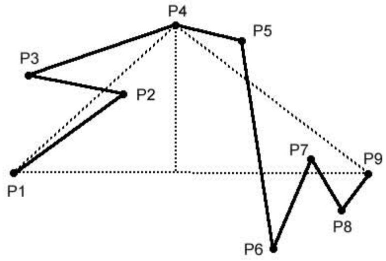

Figure 1a. The original polyline

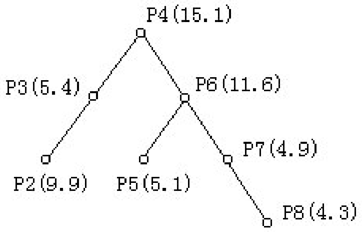

Figure 1b. The BLG-Tree of the polyline

The BLG-Tree is able to export the representation of a polyline at a certain scale. A good graphic representation can be obtained by selecting nodes whose distance values are bigger than the minimum error, and the error is determined by the given scale (e.g. the map size of a pixel on the display screen). Figure 2b shows a representation of the original polyline at the minimum error ‘5’, and the selected nodes of the BLG-Tree are black, see Figure 2a.

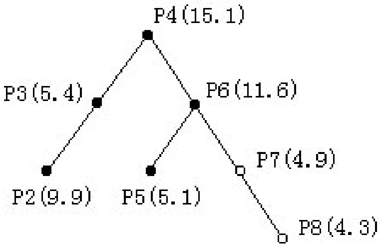

Figure 2a. The black nodes of which the distance values are not smaller than 5

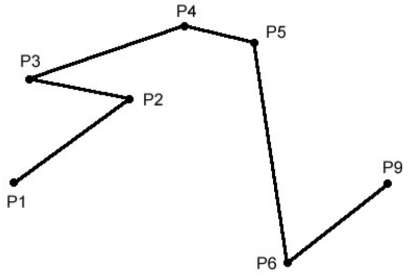

Figure 2b. A presentation of the original polyline at the minimum error ‘5’

2.2 Tolerance Value Promotion

In most cases, the distance values stored in the nodes will become smaller when descending the tree. Unfortunately, this is not always the case as shown in Figure 1b: the distance value ‘9.9’ of P2 is greater than the distance value ‘5.4’ of its parent node P3. It is not a monotonically decreasing series of values.

To build the linear index of points according to the decreasing distance values, the distance values of the BLG-Tree nodes should be revised. Traverse every node of the tree and promote the parent node’s distance value to the child node’s if its distance value is smaller. The distance value of P3 is promoted from ‘5.4’ to ‘9.9’ so that when P2 at a lower level is selected, P3 will also be selected, see Figure 2.

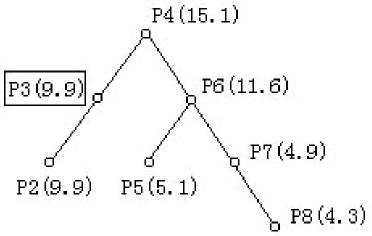

Figure 3. Distance value promotion; the distance value of P3 is promoted from ‘5.4’ to ‘9.9’

2.3 Linear BLG-tree

To improve the efficiency of storage and traversal of nodes, this paper transforms the hierarchical BLG-tree into a linear BLG-tree. The linear BLG-tree is an array of one dimension, in which nodes are sorted on descending distance values. Because the parent node’s distance value is not smaller than its child node’s, the levels of the nodes are implicit in the linear BLG-Tree. Figure 4 gives the linear BLG-Tree of the original polyline, and the selected nodes are grey at the minimum error ‘5’.

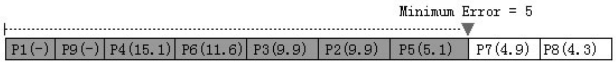

Figure 4. The linear BLG-Tree; the selected nodes are grey at the minimum error ‘5’

3. STORAGE STRUCTURE OF DRAINAGE NETWORK FOR PROGRESSIVE TRANSMISSION

This section presents a new storage structure of drainage network for progressive transmission based on the changes accumulation model. At the level of object element, the river branches are sorted on descending significance grade which is determined by drainage area. At the level of geometric detail, a river branch is splitted into sections by river joints, and each section polyline is organized by the structure of the above mentioned linear BLG-tree.

3.1 River Branches in the Storage Structure

To realize the progressive transmission at the level of object, an ordering of objects within a layer should be established on descending significance grade (Bertolotto and Egenhofer 1999, 2001, Ai, Li and Liu, 2004). In this paper, river branches are organized to a lineal sequence on descending drainage area. The following Table 1 gives the storage structure of the river branches which are shown in the Figure 5.

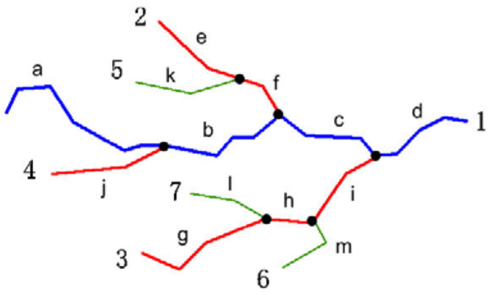

Figure 5. A drainage network; the river branch 1 is splitted into four sections (a, b, c and d) by joints.

ordering	branch_id	drainage_area	MER
1	1	15000	(xl,yl,xh,yh)
2	3	5000	(xl,yl,xh,yh)
3	2	3000	(xl,yl,xh,yh)
4	4	2000	(xl,yl,xh,yh)
5	5	1500	(xl,yl,xh,yh)
6	7	1000	(xl,yl,xh,yh)
7	6	600	(xl,yl,xh,yh)

Table 1. Table ‘river_branches’; the branches are sorted on descending drainage area

Some notes:

- the column ‘branch_id’ is the primary key within this table;
- the column ‘drainage_area’ records the drainage area of the river branches;
- the ‘ordering’ column records the ordering of the branches sorted on descending drainage area, which decides the significance grade of the river branch;
- the ‘ordering’ column is the linear index of this table;
- the column ‘MER’ contains information of the minimum exterior rectangle of the river branches, which is used in the spatial retrieval.

To maintain the topological relationship consistency at different scale, a river branch is splitted into sections by joints. For example, the river branch 1 is splitted into four sections (a, b, c and d), see Figure 5.

3.2 River Sections in the Storage Structure

At the level of geometric detail, the river sections are represented via the linear BLG-trees, enabling to deliver the appropriate number of points per scale level (Oosterom, 2005). The Figure 6 shows the section ‘a’ of our scene with their corresponding linear BLG-tree depicted below (nodes indicate point number and distance values).

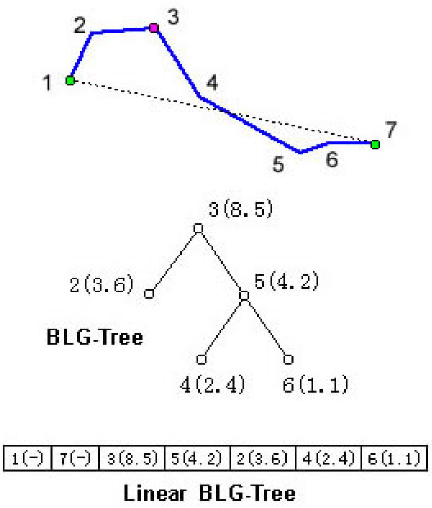

Figure 6. An example linear BLG-tree of section ‘a’; a node contains a point (number) and between brackets a distance value

The Table 2 gives the storage structure of the river sections, and some notes:

- the column ‘branch_id’ is the foreign key referring to the ‘branch_id’ column of table ‘river_branches’;
- in Oracle database, the data type of the column ‘geometry’ is SDO_Geometry, and this column records the original polyline geometry data of the sections;
- the column ‘blg’ contains the linear BLG-tree, and the data type of this column can be explicitly defined by the following SQL:

  ```
  create or replace type floatarray as varray(524288) of float;
  create or replace type intarray as varray(524288) of int;
  create or replace type blgtree as object
  (
  pointid intarray,
  distance floatarray,
  );
  ```

section_id	branch_id	geometry	blg
a	1	(x1,y1,x2,y2…)	blgtree
b	1	(x1,y1,x2,y2…)	blgtree
c	1	(x1,y1,x2,y2…)	blgtree
d	1	(x1,y1,x2,y2…)	blgtree
e	2	(x1,y1,x2,y2…)	blgtree
f	2	(x1,y1,x2,y2…)	blgtree
g	3	(x1,y1,x2,y2…)	blgtree
h	3	(x1,y1,x2,y2…)	blgtree
i	3	(x1,y1,x2,y2…)	blgtree
…	…	…	…

Table 2. Table ‘river_sections’; the branches are sorted on descending drainage area

3.3 Implementation of the Storage Structure for Progressive Transmission

The progressive transmission can be started by receiving a client request including scale and spatial extent.

For a request scale, the data retrieval at the server side includes two steps: retrieving the needed river branches and retrieving the good graphic representations of the branches based on the linear BLG-Tree.

3.3.1 Retrieving the Needed River Branches

The meaningful information density of the client map can be determined by some classic model in map generalization, such as square root model (Topfer and Pillewizer, 1966), see the following formula.

$$n_R = n_A \sqrt{\frac{S_R}{S_A}} \qquad (1)$$

where n_R = optimal number of the river branches at the request scale
n_A = number of river branches at the original map
S_R = the request scale
S_A = the scale of the original map

The optimal number of the river branches at the request scale can be calculated by the above formula (1). For example, the request scale is 1:500,000, and the scale of the original map is 1:100,000, and the number of river branches at the original map is 7, and so:

n_R = 7 * sqrt((1/500,000) / (1/100,000)) = 3.13

The top n_R (3) river branches can be retrieved from the table ‘river_branches’ by following SQL:

```
select top 3 branch_id from river_branches order by drainage_area desc;
```

Furthermore, the needed river branches should overlap the client spatial extent. The spatial overlap retrieval can be realized by comparing the spatial extent with the column ‘MER’.

3.3.2 Retrieving the Good Graphic Representations of the Needed Branches

The request scale determines the minimum error, and the good graphic representation of the needed river branches can be retrieved from the table ‘river_sections’ based on the linear BLG-tree.

The Figure 7 gives the representations of the drainage network at different request scales. With the request scale increasing, more river branches and more geometric details are retrieved and represented.

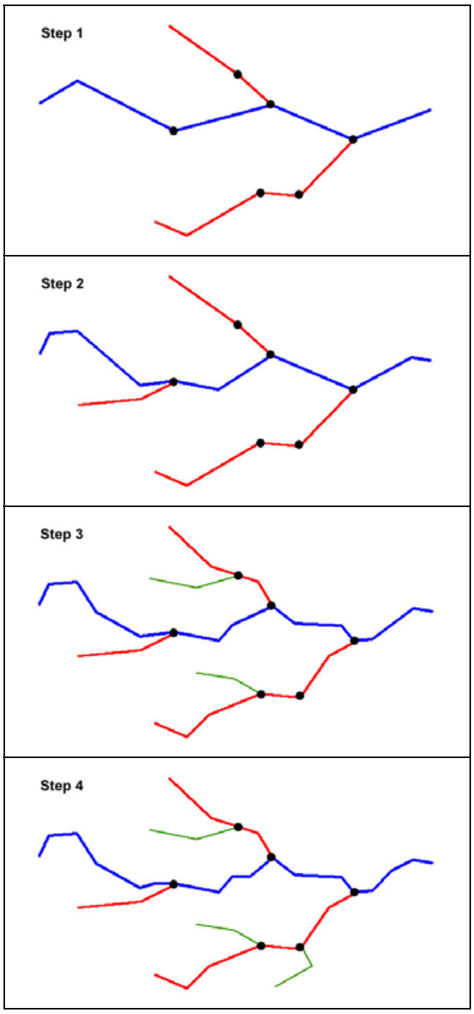

Figure 7. The representations of the drainage network at increasing scales

4. BROWSER-SERVER ARCHITECTURE FOR PROGRESSIVE TRANSMISSION

This section presents a browser/server architecture for progressive transmission based on WFS (Web Feature Service), GML (Geography Markup Language) and AJAX (Asynchronous JavaScript and XML).

4.1 A Browser/Server Architecture for Progressive Transmission

The Figure 8 gives a browser/server architecture for progressive transmission. In the architecture, the AJAX is used to provide quicker and friendlier data transaction and parse the response GML into VML (Vector Markup Language) or SVG (Scalable Vector Graphics) for rendering.

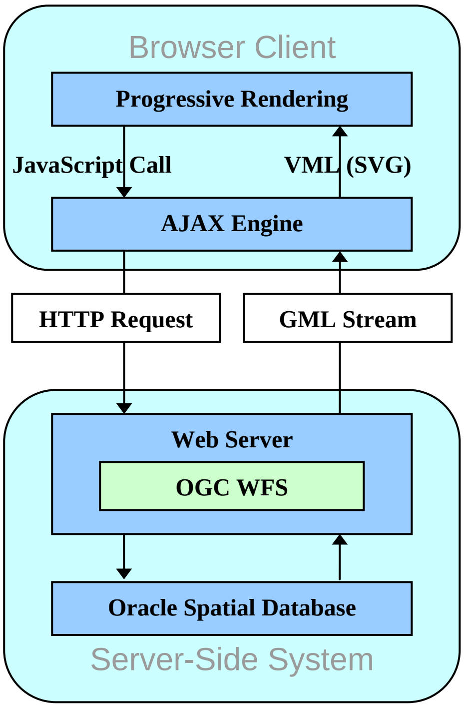

Figure 8. A browser/server architecture for progressive transmission

At the browser client, a partial copy of the storage structure of the drainage network will be built to display the coarse representations of the data. When a user browses the web map, a JavaScript function first retrieves from the local partial copy, and if there is no enough local data, a HTTP request including the current scale and the spatial extent will be sent to the web server.

When a client requests a web map, the web server will create a session object to record the state of the data which have been transmitted to the client. The state can include the transmitted river branches ID and its request error values. Therefore, for a new request, the server-side program only retrieves and transmits the increment data for the client.

The server-side program partitions the increment data into some smaller packets encoding with the GML standard (discussed in section 4.2), and then transmits the GML packets to the browser in the order according to the importance.

The AJAX engine of the browser adds a GML response to the local partial copy. The JavaScript function retrieves the data from the copy and parses them into VML or SVG then display it. With the arrivals of the GML packets, the web map becomes more and more fine.

4.2 Extensions to the GML Standard

To support the progressive transmission and refinement, some extensions should be made to the standard interfaces WFS (Oosterom, Vries and Meijers, 2006) and GML in the architecture.

This paper adds some new tags to the GML for storing the multi-scale representations information of the drainage network, see the following GML fragment.

The tags '<gml:coordinates>', '<gml:pointid>' and '<gml:distance>' store a slice of the linear BLG-Tree of the river section '1'. The starting point has pointid '0' and coordinates '468147,3445249', and its distance value is a given huge value '999999'. The end point has pointid '329' and coordinates '466498,3446529', and its distance value is also '999999'. The point '149' has coordinates '466989,3447416' and distance value '280.575'.

```
<gml:boundedBy>
        <gml:Box srsName="EPSG:4326">
                <gml:coordinates decimal="." cs="," ts="
">465402,3445098 468147,3447761</gml:coordinates>
        </gml:Box>
</gml:boundedBy>
<gml:featureMember>
        <gml:riverbranch id="1" drainagearea="15000">
            <gml:riversections>
                <gml:riversection id="1">
                        <gml:coordinates>
468147,3445249       466498,3446529       466989,3447416
465442,3447254       467071,3445265       466604,3446918
466199,3447761 467383,3445686 ...</gml:coordinates>
                        <gml:pointid>0 329 149 266 45
162 197 233 ...</gml:pointid>
                        <gml:distance>999999    999999
280.575    280.575    207.956    159.233    159.233
159.233  ...</gml:distance>
                </gml:riversection>
            </gml:riversections>
        </gml:riverbranch>
</gml:featureMember>
```

5. CONCLUSION

5.1 Summary of Main Results

This paper presents a method of progressive transmission of vector data on the web based on the Changes Accumulation Model. The model considers the spatial representation from one scale to another scale as an accumulation of a set of changes, and the change can take place at three levels: layer, object and geometric detail.

Taking the example of drainage network, this paper gives a new storage structure without redundancy of geometry for progressive transmission. At the level of object element, the river objects are sorted on descending significance grade which is estimated by drainage area. At the level of geometric detail, a river branch is splitted into sections, and each section is organized by a linear BLG-Tree, enabling selecting the proper representation for a given scale.

For a request scale, the optimal number of the needed river branches can be determined by the Topfer rule in map generalization. Furthermore, the good graphic representation of each river branch can be obtained by selecting nodes of the linear BLG-Tree.

This paper also presents a browser/server architecture for progressive transmission based on WFS, GML and AJAX. To support the progressive transmission and refinement, some extensions should be made to WFS and GML.

5.2 Future Work

Future research of the vector data progressive transmission could include the following topics:

- Including point and polygon objects. Point objects and polygon objects also can be sorted on descending significance grade. The boundary of polygon can be splitted into sections with the linear BLG-Tree.

- Data restoring. If the transmitted vector data is downloaded for the purpose of spatial analysis or imported into other application systems, the decomposed details need to be composed and restored as the original form.

- Maintaining the topological relationship consistency of objects among different layers. The simplification of single polyline (polygon) object possibly destroys the topological relationship with objects of other layers.

REFERENCES:

Ai, T., Li, Z. and Liu, Y., 2004. Progressive transmission of vector data based on changes accumulation model, Developments in Spatial Data Handling, Springer-Verlag, Berlin, pp85-96.

Ai, T. and Oosterom, P. van., 2002. GAP-tree Extensions Based on Skeletons. In: Richardson D and Oosterom P van(eds) Advances in Spatial Data Handling, Springer-Verlag, Berlin, pp501-514.

Bertolotto, M. and Egenhofer, M., 1999. Progressive Vector Transmission. Proceedings, 7th International Symposium on Advances in Geographic Information Systems, Kansas City, MO: p152-157.

Bertolotto, M. and Egenhofer, M., 2001. Progressive Transmission of Vector Map Data over the World Wide Web. GeoInformatica, 5 (4): 345-373.

Buttenfield, B. P., 2002. Transmitting Vector Geospatial Data across the Internet, Egenhofer, M.J. and Mark, D.M (eds.) Proceedings GIScience 2002. Berlin: Springer Verlag, Lecture Notes in Computer Science No 2478: 51-64.

Buttenfield, B. P., 1997. Delivering Maps to the Information Society: A Digital Library for Cartographic Data. InProceedings of the 18th ICA/ACI International Cartographic Conference, vol. 3.

Cecconi, A. 2003. Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping. PhD Thesis, University of Zurich, Zurich.

Douglas, D. H. and Peucker, T.K., 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Line or Its Caricature. The Canadian Cartographer, vol. 10(2): 112-122.

Han, H., Tao, V. and Wu, H., 2003. Progressive Vector Data Transmission, Proceedings of 6th AGILE, Lyon, France.

Mackaness, W. A. and Elizabeth G., 1999. Dynamic Generalisation from Single Detailed Database to Support Web Based Interaction, Proceedings of the 19th ICA,Ottawa.

Muller, J-C., Lagrange, J-P., and Weibel, R., 1995. GIS and Generalization: Methodology and Practice. Taylor & Francis, London.

Oosterom, P. van, 1990. Reactive Data Structures for Geographic Information Systems. PhD-thesis Department of Computer Science, Leiden University.

Oosterom, P. van, 1991. 'The Reactive-tree: A Storage Structure for a Seamless, Scaleless Geographic Database'. In proceedings Auto-Carto 10, Baltimore, Maryland, pages 393-407.

Oosterom, P. van, 1994. Reactive Data Structure for Geographic Information Systems. Oxford University Press, Oxford.

Oosterom, P. van, 2005. Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest, Cartography and Geographic Information Science, 32 (4): 331-346.

Oosterom, P. van, Vries, M. de, and Meijers, M., 2006. Vario-scale data server in a web service context, In: Workshop of the ICA Commission on Map Generalisation and Multiple Representation, Vancouver, United States.

Sester M. and Brenner M, 2004. Continuous Generalization for Visualization on Small Mobile Devices,In: Richardson D and Oosterom P van(eds) Advances in Spatial Data Handling, Springer-Verlag, Berlin, p355-368.

Vermeij, M.J., 2003. Development of a Topological Data Structure for On-the-Fly Map Generalisation, Master's Thesis of TU Delft.