GENERALISATION OF VECTOR DATA SETS BY SIMULTANEOUS LEAST SQUARES ADJUSTMENT

Lars HARRIE [1,2], **Tapani SARJAKOSKI** [1]
[1] Department of Cartography and Geoinformatics
Finnish Geodetic Institute, Finland
Tapani.Sarjakoski@fgi.fi
[2] Department of Real Estate Management
Lund University, Sweden
Lars.Harrie@lantm.lth.se

KEY WORDS: Generalisation, Multi-scale database, Automation, Algorithm, Data Structure, Least Squares Adjustment

ABSTRACT

Manual cartographic generalisation is a holistic process. However, most automatic approaches so far have been sequential; generalisation operators are applied one at a time in a given order. This has been the case both for model generalisation (generalisation of the conceptual model) and the fine-tuning phase (generalisation of the graphics). Our research seeks to demonstrate that fine-tuning phase of cartographic generalisation can be formulated as an optimisation problem and accordingly be solved in a single step. The paper proposes solutions to the two main problems in defining generalisation as an optimisation problem. Firstly, a set of appropriate analytical constraints for the generalisation process is given. In our approach, we are limited to formulating these constraints on point locations. Secondly, the least squares method is proposed to find the optimal solution. To solve the sparse normal equations we use the conjugate-gradient method. A prototype system for simultaneous generalisation has been written in C++. This prototype system communicates with a commercial map production system (LAMPS2 from Laser-Scan). The article concludes with some preliminary test results.

1 INTRODUCTION

Cartographic generalisation is the process of representing cartographic data in smaller scale. Even though automatic cartographic generalisation has been on the research agenda for three decades, most generalisation is still performed manually.

Cartographic generalisation has been divided into about ten operators (Shea and McMaster 1989, Robinson et al. 1995). During recent decades much research has been devoted to finding algorithms for these operators, and most of this effort has been devoted to line simplification (see for example Douglas and Peucker 1973, Plazanet et al. 1995, Wang and Müller 1998). This approach has presupposed a sequential approach to the generalisation process, both to model generalisation (generalisation of the conceptual model) and to fine-tuning (generalisation of the graphics).

Most cartographers are not aware of dividing the process into different operators (Mackaness 1991; Keates 1996), which calls for an optimisation rather than sequential approach to cartographic generalisation. Such approaches have been proposed for parts of the generalisation process, most notably for displacement (Burghardt and Meier 1997, Højholt 1998, Harrie 1999). Other techniques than optimisation have been proposed to overcome the problem of competing goals of the operators in the sequential approaches, for example the agent technique (Baeijs et al. 1996; Lamy et al. 1999).

This paper deals with an optimisation approach to the fine-tuning phase of cartographic generalisation. The approach is based on least squares adjustment. This is a standard method in geodesy (see e.g. Vanicek and Krakiwsky 1986) and photogrammetry (see e.g. Zhizhuo 1990), but it is not common in cartography.

There are several requirements that have to be fulfilled in the fine-tuning phase. Our approach is to formulate those requirements as constraints on point movements. These constraints are then treated as observations in the least squares adjustment. If an ordinary vector data set is generalised, the normal equations will be extensive. To solve the large, sparse normal equations, we have used the conjugate-gradient method (along the line proposed by Sarjakoski and Kilpeläinen 1999).

We have written a prototype of simultaneous generalisation in C++. This prototype system communicates with a commercial map production system (LAMPS2 from Laser-Scan), which enables testing in a suitable environment.

The second section of the article describes the constraints, both their aims and how they are defined analytically. The next section deals with the conjugate-gradient method. Some preliminary results are given in the fourth section. The article concludes with some concluding remarks and suggestions for future work.

2 DEFINITION OF CONSTRAINTS

The section begins with motivating the different constraints by denoting which requirements they aim at fulfilling. Then some comments are given about when the different constraints should be applied. The rest of the section deals with the analytical formulation of the constraints. Firstly, a general formulation is given, and then three of the constraints are described in more detail.

2.1 Description of the different types of constraints

There are several requirements that have to be fulfilled in the fine-tuning phase of cartographic generalisation. These requirements can be divided into two main groups: changing and preserving. The changing requirements seek to make the cartographic data more readable, while the preserving requirements strive to retain the characteristics of the data.

2.1.1 Changing requirements

Several researchers have defined which operators belong to either model generalisation or fine-tuning (see e.g. Kilpeläinen 1997). In this paper we consider fine-tuning to consist basically of the operators *simplification, smoothing*, *exaggeration* and *displacement* (as defined in Shea and McMaster 1989).

Simplification is selecting a subset of the original points, retaining those that are considered most representative. Smoothing is the process of relocating points to smooth away small perturbations and reduce sharp angularities. Some features might be too small for the intended map scale, and must therefore be exaggerated. This is performed by the exaggeration operator. The displacement operator seeks to solve the spatial conflict between cartographic objects that are too close.

In our approach we have used five constraints to fulfil the changing requirements:

1) Smoothing constraints – The aim is to make lines less angular. The constraints are applied on line objects.
2) Exaggeration constraints – These constraints enlarge some bends (depending on the properties of the bends) of line objects.
3) Spatial conflict constraints – Try to solve spatial conflicts between objects.
4) Collinear constraints – The aim is to remove bends and replace them with a straight line.
5) Compress constraints – Make line segments shorter. This type of constraint is applied if and only if collinear constraints are defined.

2.1.2 Preserving requirements

We have defined six constraints for preserving the cartographic data:

1) Movement constraints – The aim is to keep the objects in their original locations.
2) Movement direction constraints – These constraints allow points in line objects to move along the lines, but not in other directions.
3) Stiffness constraints – These constraints seek to preserve the internal geometry of objects.
4) Curvature constraints – Aim at maintaining the angularity of lines.
5) Segment length constraints – These constraints aim at maintaining the length of line segments.
6) Crossing constraints – Aim at preserving the angle between connected line objects.

2.2 When to apply the different constraints

One major question, of course, is when these constraints should be applied. In our tests so far we have grouped the cartographic objects into different generalisation categories. For example, one category is *normal line objects,* a category that contains objects like major and minor roads, streams etc. On objects in this category we define all the constraints above, apart from the stiffness constraints. The bends of the line object to which exaggeration constraints should be applied, for example, is determined using threshold values on bend properties (cf. Figure 2). However, specifying when the constraints should be applied – and the importance of each constraint – is a difficult challenge; more practical experience is required to identify suitable rules.

2.3 General expressions for the constraints

One key issue is to find analytical expressions that model the changing and preserving requirements. In our approach, the generalisation process will not change the number of points, which implies that the constraints are formulated on point locations:

$$f_k(x_1,y_1,x_2,y_2,\ldots,x_n,y_n) = \text{constant}_k \tag{1}$$

where

x_i, y_i the coordinates of the points that build up the objects,
f_k constraint number k, and
n total number of points.

Below we will describe how the smoothing, exaggeration and spatial conflict constraints are defined. (The other constraints are defined similarly.)

2.4 Smoothing constraints

The method is defined on a *link-node* data structure. The basic rules of such a data structure are that each link has exactly one start and one end node; that links may *meet*, but not *intersect*; and that each node must be either the start or end node of at least one link.

2.4.1 Linear expression for smoothing constraint

The smoothing constraints are defined for each point *i* (apart from the end points) in a link that should be smoothed in the following manner:

$$\begin{aligned}&\left.\frac{\partial f_{smoothing}}{\partial x_{i-1}}\right|_{\bar{\mathbf{x}}}\cdot\Delta x_{i-1}+\left.\frac{\partial f_{smoothing}}{\partial y_{i-1}}\right|_{\bar{\mathbf{x}}}\cdot\Delta y_{i-1}+\left.\frac{\partial f_{smoothing}}{\partial x_{i}}\right|_{\bar{\mathbf{x}}}\cdot\Delta x_{i}+\left.\frac{\partial f_{smoothing}}{\partial y_{i}}\right|_{\bar{\mathbf{x}}}\cdot\Delta y_{i}+\left.\frac{\partial f_{smoothing}}{\partial x_{i+1}}\right|_{\bar{\mathbf{x}}}\cdot\Delta x_{i+1}+\\&+\left.\frac{\partial f_{smoothing}}{\partial y_{i+1}}\right|_{\bar{\mathbf{x}}}\cdot\Delta y_{i+1}+\left.\frac{\partial f_{smoothing}}{\partial x_{i+2}}\right|_{\bar{\mathbf{x}}}\cdot\Delta x_{i+2}+\left.\frac{\partial f_{smoothing}}{\partial y_{i+2}}\right|_{\bar{\mathbf{x}}}\cdot\Delta y_{i+2}=(-\text{const})\cdot f_{smoothing}(\bar{\mathbf{x}})\end{aligned} \tag{2}$$

where

$f_{\text{smoothing}}$ the change of direction (equal to b-a in Figure 1),
Δx_i, Δy_i movement of point *i*,
$\bar{\mathbf{x}}$ vector containing the original coordinates, and
const positive constant.

That is, the constraints strive to make all the direction changes as small as possible.

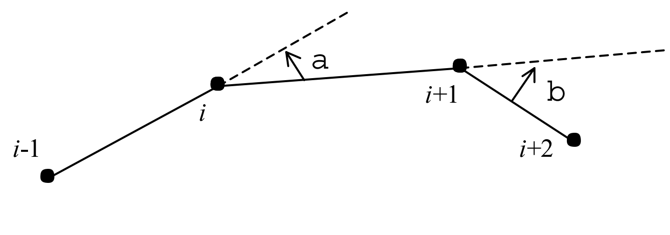

Figure 1. Change of directions of line segments used in smoothing constraint.

2.5 Exaggeration constraints

The lines are generalised based on analyses of bends (an idea borrowed from Plazanet et al. 1995; Lecordix et al. 1997; Wang and Muller 1998). Bends are defined by studying the inflection points of the line. An inflection point is a point where the second derivative of the line changes sign (see Figure 2). In our work we have defined a bend as starting at a point before an inflection point and ending at the next inflection point (see Figure 2). This definition implies that bends partly overlap.

Bends have different characteristics and should be treated differently in the generalisation process. We have used three properties to characterise a bend: *base*, *height* and *length* (see Figure 2). The base is the Euclidean distance between the start and end point of the bend; the height is the distance between the baseline and the point furthest away; and the length is the distance between the start and end point along the line. Depending on the values of these three properties, different types of constraints are set up for the constraints. For example, if the base is short and the height is of medium size, the bend should be exaggerated, i.e. exaggeration constraints are set up.

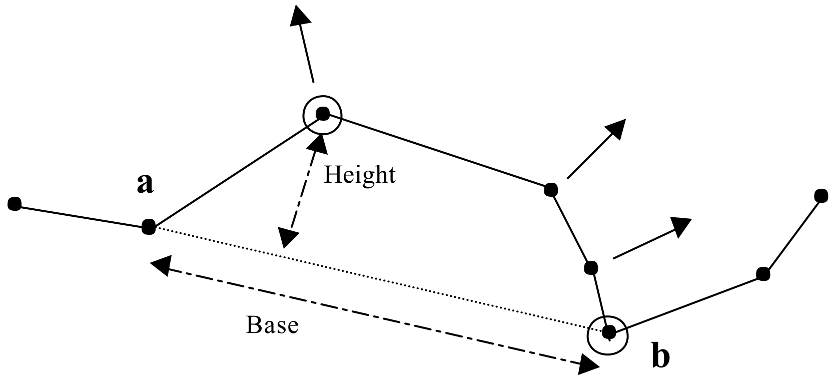

Figure 2. The inflection points are marked with circles, and the bend starts in point **a** and ends in point **b**. When the bend is enlarged, the inner points of the bends are moved in the direction of the arrows.

2.5.1 Linear expression for exaggeration constraints

For each point of a bend (apart from the end points) that is to be exaggerated set up the constraint:

$$\left.\frac{\P f_{exaggeration}}{\P x_i}\right|_{\bar{\mathbf{x}}} \cdot \Delta x_i + \left.\frac{\P f_{exaggeration}}{\P y_i}\right|_{\bar{\mathbf{x}}} \cdot \Delta y_i = \text{const} \cdot f_{exaggeration} \qquad (3)$$

where

$f_{exaggeration}$ the area function of a polygon (area between the base vector and the line; see e.g. Worboys 1995), and

const a positive constant.

2.6 Spatial conflict constraints

The spatial conflict constraints are defined between objects, i.e. we need a spatial data structure that stores information about the spatial relationship between objects. One such data structure frequently used in cartographic generalisation is Delauney triangulation (Bundy et al. 1995; Ruas 1995; Højholt 1998).

2.6.1 Constrained Delauney triangulation and spatial conflicts

Three points constitute a Delauney triangle if and only if the circle through the points contains no other point (see e.g. de Berg et al. 1997). Since we are merely interested in the relationship between objects rather than the relationship between points, we use constrained Delauney triangulation. In this triangulation some edges (in our application: line segments of cartographic objects) are forced to belong to the triangulation.

In the next step we need to define which spatial conflict constraints should be set up using the constrained Delauney triangulation. One possibility would of course be to set up a constraint for each edge in the triangulation, but we have preferred to set up a constraint for the shortest distance between neighbouring objects. This distance is not necessarily along an edge in the triangulation, but still the constrained Delauney triangulation is a computationally fast way to find neighbouring objects.

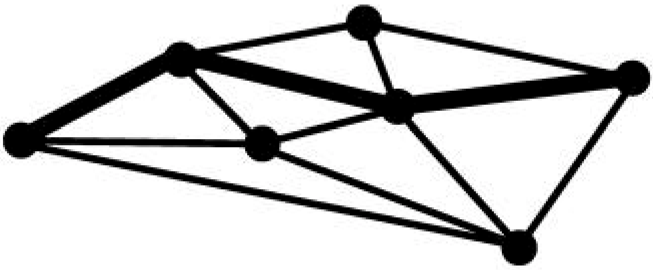

Figure 3. Constrained Delauney triangulation. The input data is a point set (seven points) and a line segment set (three thick lines).

2.6.2 Linear expression for spatial conflict constraint

As mentioned above, a spatial conflict is set up between the shortest distance between objects. This can be either between two points or between one point and a line segment (see Harrie 1999 for details). In equation (4) we give the constraint for a spatial conflict between a point (*i*) and a segment (with the endpoints *j* and *j*+1):

$$\left.\frac{\partial f_{distance}}{\partial x_i}\right|_{\bar{\mathbf{x}}} \cdot \Delta x_i + \left.\frac{\partial f_{distance}}{\partial y_i}\right|_{\bar{\mathbf{x}}} \cdot \Delta y_i + \left.\frac{\partial f_{distance}}{\partial x_j}\right|_{\bar{\mathbf{x}}} \cdot \Delta x_j + \left.\frac{\partial f_{distance}}{\partial y_j}\right|_{\bar{\mathbf{x}}} \cdot \Delta y_j + \\ + \left.\frac{\partial f_{distance}}{\partial x_{j+1}}\right|_{\bar{\mathbf{x}}} \cdot \Delta x_{j+1} + \left.\frac{\partial f_{distance}}{\partial y_{j+1}}\right|_{\bar{\mathbf{x}}} \cdot \Delta y_{j+1} = distance\_value \qquad (4)$$

where

$f_{distance}$ equals the shortest distance between a point and a line segment (see for example Worboys 1995), and
distance_value equal to the desired increase in distance between the objects.

3 CONJUGATE GRADIENT METHOD FOR SOLVING LARGE, SPARSE EQUATION SYSTEMS

3.1 Least squares method

All the constraints are formulated as linear equation of point movements (see equations 2-4). Since the number of constraints exceeds the number of points we get an over-determined linear equation system of the form:

$$\mathbf{Ax} = \mathbf{l} + \mathbf{v} \qquad (5)$$

where

A design matrix,
x vector containing the unknown point movements,
l observation vector (containing the right-hand side of equations 2-4), and
v residual vector.

To solve the equation system we employ the least-squares method where the solution of the normal equation system is given as:

$$\mathbf{Nx} = \mathbf{u}\,. \qquad (6)$$

where

$\mathbf{N} = \mathbf{A}^t\mathbf{WA}$, and
$\mathbf{u} = \mathbf{A}^t\mathbf{Wl}$.

All constraints are considered uncorrelated, which gives a diagonal weighting matrix (**W**) where values of the weighting matrix are functions of user-set attributes.

3.2 Conjugate gradient method

The conjugate-gradient method (CGM) is one of the iterative methods that have been developed for solving systems of linear equations. Its special characteristics are that it is suitable for positive-definite matrices (a property of **N**), and also that it actually produces an algebraically exact solution in *n* iterations, where *n* is the number of unknowns (see e.g. Bazarra et al. 1993). However, in practice the iteration is terminated much earlier when a desired accuracy has been reached.

A central entity in the CGM is the discrepancy vector ($\mathbf{r}_i$):

$$\mathbf{r}_i = \mathbf{u} - \mathbf{N}\mathbf{x}_i \tag{7}$$

which functions as a measure of the numerical accuracy of the solution. The iteration will be started with $\mathbf{x}_0=0$, thus $\mathbf{r}_0=\mathbf{u}$. Then a line search is performed along a search direction ($\mathbf{p}_i$; see equations 8a,b). The first line search is along the discrepancy vector ($\mathbf{p}_0=\mathbf{r}_0$). The last two steps compute a new search direction (8d,e). The iteration is usually terminated when the Euclidean norm of the residual vector is below a predefined threshold value. More precisely, the iteration cycle is:

$$\mathrm{a} = \frac{\mathbf{r}_i{}^t\mathbf{r}_i}{\mathbf{p}_i{}^t\mathbf{N}\mathbf{p}_i} \tag{8a}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathrm{a}\mathbf{p}_i \tag{8b}$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \mathrm{a}\mathbf{N}\mathbf{p}_i \tag{8c}$$

$$\mathrm{b} = \frac{\mathbf{r}_{i+1}{}^t\mathbf{r}_{i+1}}{\mathbf{r}_i{}^t\mathbf{r}_i} \tag{8d}$$

$$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \mathrm{b}\mathbf{p}_i\ . \tag{8e}$$

From the point of implementation it is essential that the coefficient matrix of the normal equation system (**N**) be accessed only in the computation of the matrix-vector product $\mathbf{N}\mathbf{p}_i$. This product can be reformulated as:

$$\mathbf{N}\mathbf{p}_i = \mathbf{A}^t(\mathbf{W}\mathbf{A}\mathbf{p}_i) \quad \text{or} \quad \mathbf{N}\mathbf{p}_i = (\mathbf{A}^t\mathbf{W}^{1/2})(\mathbf{W}^{1/2}\mathbf{A}\mathbf{p}_i) \tag{9}$$

which shows that instead of using **N** we can use the original design and weight matrices, **A** and **W**. This leads to straightforward implementations when the sparsity of the design matrix is utilised (Haljala 1974; Sarjakoski 1982).

In many practical situations the convergence of the CGM can be improved considerably by scaling the normal equation matrix **N** into one having unit diagonal elements (Haljala 1974). This can be realised by defining a diagonal scaling matrix:

$$\mathbf{Q} = diag(\mathbf{N}), \tag{10}$$

and by inserting it into the original linear equation system:

$$\mathbf{A}'\mathbf{x}' = \mathbf{l} + \mathbf{v}\ , \tag{11}$$

where
$\mathbf{A}' = \mathbf{Q}^{-1/2}\mathbf{A}$, and
$\mathbf{x}' = \mathbf{Q}^{1/2}\mathbf{x}$.

This yields a scaled normal equation system that fulfils the criterion of having units on the diagonal:

$$\mathbf{N}'\mathbf{x}' = \mathbf{u}' \tag{12}$$

where

$\mathbf{N}' = \mathbf{A}''\mathbf{W}\mathbf{A}' = \mathbf{Q}^{-1/2}\mathbf{A}'\mathbf{W}\mathbf{A}\mathbf{Q}^{-1/2} = \mathbf{Q}^{-1/2}\mathbf{N}\mathbf{Q}^{-1/2}$,
$\mathbf{u}' = \mathbf{A}''\mathbf{W}\mathbf{l} = \mathbf{Q}^{-1/2}\mathbf{A}'\mathbf{W}\mathbf{l} = \mathbf{Q}^{-1/2}\mathbf{u}$, and
$\mathbf{x}' = \mathbf{Q}^{1/2}\mathbf{x}$.

For implementation purposes it is often desirable to avoid the prescaling of the equation systems and instead carry out the scaling in real time during the iteration (for details, see Sarjakoski 1982).

CGM has several characteristics that make it to be a strong candidate to be used as a tool for solving the linear equation systems related to least squares adjustment for cartographic generalisation. From the implementation point of view, it is rather simple to program and also well suited for sparse equation systems. The convergence of the iteration process is also very rapid in cases when the equation system is well-conditioned.

4. PRELIMINARY RESULT

Ideally, the method should be integrated in a map production system. Due to implementation details we have made a C++ implementation that communicates with LAMPS2 (see Laser-Scan 2000; Hardy 1998) via ASCII-files. Furthermore, the constraint Delauney triangulation is computed by a module written by Jonathan Richard Shewchuk, Pittsburgh PA, USA (Shewchuk 1996, 1997).

So far we have only used the program for minor studies on a few objects. As shown in Figures 4 and 5, the spatial conflicts are nicely solved by the method, but the line generalisation is problematic.

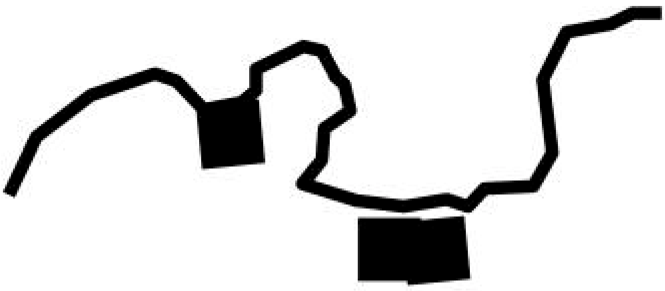

Figure 4. Cartographic data consisting of a shoreline and some houses (point objects). Before generalisation.

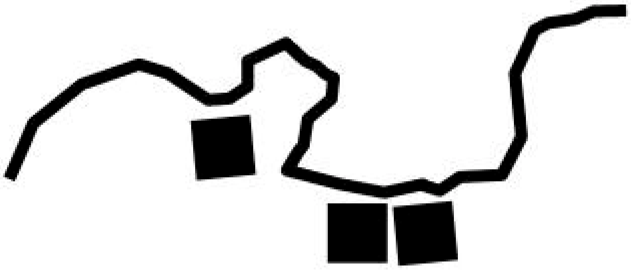

Figure 5. After generalisation.

4 FURTHER WORK AND CONCLUSION

In this paper only a preliminary test is presented. Our aim is, in the near future, to perform more realistic case studies. These case studies will probably also lead to reformulation of some of the constraints. However, we are convinced that the concept described in this paper is a promising approach to simultaneous cartographic generalisation, and that this kind of method is required to avoid the inherent problems of the sequential approach.

ACKNOWLEDGEMENTS

This research is supported by the Finnish Geodetic Institute and Lund University. Thanks to Laser-Scan for implementation guidelines and Jonathan Richard Shewchuk for providing triangulation code.

REFERENCES

Baeijs, C., Y. Demazeau, and L. Alvares, 1996. SIGMA: Application of Multi-Agent Systems to Cartographic Generalization. Proceeding of 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer, pp. 163-176.

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty, 1993. Nonlinear Programming – Theory and Algorithms. John Wiley & Sons, Inc.

de Berg, M., M. Van Kreveld, M. Overmars, and O. Schwarzkopf, 1997. Computational Geometry – Algorithms and Applications, Springer-Verlag, Berlin Heidelberg, Germany.

Bundy, G. Ll., C. B. Jones, and E. Furse, 1995. Holistic generalization of large-scale cartographic data. In Müller, J. -C., J. -P. Lagrange and R. Weibel, (eds.), GIS and Generalization, Gisdata 1, ESF, Taylor & Francis, London, pp. 106-119.

Burghardt, D., and S. Meier, 1997. Cartographic Displacement Using the Snakes Concept. In: Foerstner, W. and L Pluemer, (eds.), Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, Birkhaeuser Verlag, Basel 1997, pp. 59-71.

Douglas, D. H., and T. K. Peucker, 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. The Canadian Cartographer, vol. 10, No. 2, pp 112-122.

Haljala, S., 1974. Method of Conjugate Gradients for the Solution of Normal Equations in the Analytical Block Adjustment. The Photogrammetric Journal of Finland. 6 (2), pp. 160-165.

Hardy, P. G., 1998. Map Production from an Active Object Database, Using Dynamic Representation and Automated Generalisation. The Cartographic Journal, 35 (2), pp. 181-189.

Harrie, L., 1999. The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. Cartography and Geographic Information Science, 26 (1), pp. 55-69.

Højholt, P., 1998. Solving Local and Global Space Conflicts in Map Generalisation Using a Finite Element Method Adapted from Structural Mechanics. In: Proceedings of Spatial Data Handling, Vancouver, Canada, pp. 679-689.

Keates, J. S., 1996. Understanding Maps, 2nd ed. Addison Wesley Longman Limited.

Kilpeläinen, T., 1997. Multiple Representation and Generalization of Geo-Databases for Topographic Maps. Doctorate thesis. Publications of the Finnish Geodetic Institute, no 124.

Lamy S., A. Ruas, Y. Demazeau, M. Jackson, W. A. Mackaness, and R. Weibel, 1999. The Application of Agents in Automated Map Generalisation. In: Proceedings of 19th International Cartographic Conference, Ottawa, Canada, pp. 1225-1234.

Laser-Scan, 2000. LAMPS2 program description. Cambridge, United Kingdom. http://www.laser-scan.com/products/lamps2.htm(20 March 2000).

Lecordix, F., C. Plazanet, and J. -P. Lagrange, 1997. A Platform for Research in Generalization: Application to Caricature. GeoInformatica, 1 (2), pp. 161-182.

Mackaness, W. M., 1991. Integration and Evaluation of Map Generalization. In Buttenfield B. P., and R. B. McMaster, Map Generalization : Making Rules for Knowledge Representation, Longman Group, pp. 217-226.

Plazanet, C., J. -G. Affholder, and E. Fritsch, 1995. The Importance of Geometric Modeling in Linear Feature Generalization. Cartography and Geographic Information Systems, 22 (4), pp. 291-305.

Robinson, H. A., J. L. Morrison, P. C. Muehrcke, A. J. Kimerling, and S. C. Guptill, 1995. Elements of Cartography, 6th ed. Wiley, New York.

Ruas, A., 1995. Multiple Paradigms for Automating Map Generalization: Geometry, Topology, Hierarchical Partitioning and Local Triangulation. In: Proceedings of ACSM/ASPRS, pp. 69-78.

Sarjakoski, T., 1982. Bundle Block Adjustment Based on Conjugate Gradient and Direct Solution Methods. International Archives of Photogrammetry and Remote Sensing, Vol. 24-III, Helsinki, pp. 425-445.

Sarjakoski, T., and T. Kilpeläinen, 1999. Holistic Cartographic Generalization by Least Squares Adjustment for Large Data Sets. In: Proceedings of 19th International Cartographic Conference, Ottawa, Canada, pp. 1091-1098.

Shea, K. S., and R. B. McMaster, 1989. Cartographic Generalization in a Digital Environment: When and How to Generalise. Proceedings of 9th International Symposium on Computer-Assisted Cartography, Baltimore, pp. 56-67.

Shewchuk, J. R., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: First Workshop on Applied Computational Geometry, Philadelphia, Pennsylvania, pp. 124-133.

Shewchuk J. R., 1997. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. Discrete & Computational Geometry
18, pp. 305-363.

Vanicek P., and E. Krakiwsky, 1986. Geodesy: the Concepts. Elsevier, Amsterdam, the Netherlands.

Wang, Z., and J. -C. Müller, 1998. Line Generalisation Based on Analysis of Shape Characteristics. Cartography and Geographic Information Systems, Vol. 25, No. 1, pp 3-15.

Zhizhuo, W., 1990. Principles of Photogrammetry. Publishing House of Surveying and Mapping, Beijing, China.

Worboys, M. F., 1995. GIS - A Computing Perspective. Taylor & Francis, London.