Weight-Setting and Quality Assessment in Simultaneous Graphic Generalisation

Lars Harrie

Department of Real Estate Management
Lund University
PO Box 118
SE-221 00 Lund
Sweden
E-mail: Lars.Harrie@lantm.lth.se

Abstract. Cartographic generalisation aims at simplifying the representation of data to suit the scale and purpose of the map. This paper deals with an algorithm that implements the whole graphic generalisation process (roughly defined as the operators simplification, smoothing, exaggeration and displacement) called *simultaneous graphic generalisation*. This method is based on constraints, i.e. requirements that must be fulfilled in the generalisation process. The constraints strive to make the map readable while preserving the characteristics of the data, which implies that all constraints cannot be completely satisfied. This study was concentrated on finding the optimal compromise between the constraints in simultaneous graphic generalisation by setting weights for the constraints. Four strategies for determining the weights are described and their advantages and disadvantages are discussed. The discussion is based on the following assumptions: the constraints are independent, and the weights are only dependent on constraint type and object type. A comparison of the strategies reveals that *constraint violation* is the most promising strategy. One advantage of this strategy is that it provides a numerical measure for quality assessment. The article concludes with a case study of the constraint violation strategy, in which visualisation of the numerical quality measure is used. The case study shows that the constraint violation strategy gives a sound compromise between the constraints.

Keywords: cartographic generalisation, graphic generalisation, optimisation, least-squares method, quality assessment, weight-setting

1 Introduction

Cartographic generalisation aims at simplifying the representation of cartographic data to suit the scale and purpose of the map. Much research during recent decades has been devoted to automation of the generalisation process (Müller et al., 1995). The process has been divided into about ten operators, and several algorithms have been proposed to implement these operators. More than one operator is required to solve most generalisation problems and, accordingly, the optimal sequence of operators must be determined (see e.g. Nickerson, 1988; Mackaness, 1994b; Regnauld et al., 1999; Ruas, 1999). One problem of the sequential approach is that the operators have different goals. When an algorithm is applied to solve a conflict, the algorithm may create other conflicts that have to be solved by subsequent algorithms. Another common problem with the sequential approach is that objects are (normally) treated in isolation. That is, if the geometry of an object is generalised there might be new (spatial) conflicts with neighbouring objects that must be identified and solved. To circumvent the problems of the sequential approach, algorithms should implement several operators in a single step, and they should also model the relationship between objects. This strategy has been suggested by several authors (Ware and Jones, 1998; Sarjakoski and Kilpeläinen, 1999; Sester, 2000a), but no solution has been presented for implementation of the complete generalisation process in a single step. The development of such an algorithm is difficult, in particular for model generalisation (generalisation due to changes in the conceptual model).

A good map must satisfy several requirements: the map must be visually legible and the cartographic objects must be a good representation of reality. These requirements can act as constraints in the generalisation process (Beard, 1991). Recently several generalisation methods have been developed based on analytical constraints (Ruas and Plazanet, 1996; Harrie, 1999; Sester, 2000a; Ruas, 2000). Constraints are interesting both for model generalisation and in graphic generalisation (moving and/or distorting objects to make the data visually legible; roughly the same as the operators: simplification, smoothing, exaggeration and displacement). However, this paper concentrates on constraints for graphic generalisation.

Harrie and Sarjakoski (2001) have described an optimisation method of graphic generalisation called *simultaneous graphic generalisation*. The method is based on a number of analytical constraints, where some are defined for single objects and some for groups of objects. Ideally, all these constraints should be fulfilled. However, the constraints are contradictory, some of them strive to change the data to make the map readable, while others strive to maintain the characteristics of the data. The solution will be a compromise between the constraints and, depending on the weighting of the constraints, different solutions are obtained. The idea of computing a solution as a compromise between constraints has recently been explored for different generalisation problems (Burghardt and Meier, 1997; Harrie, 1999; Højholt, 2000; Sester, 2000a; Harrie and Sarjakoski, 2001) but none of these studies has fully addressed the weighting problem.

This paper deals with the determination of the weights of the constraints in simultaneous graphic generalisation. Four strategies for weight determination are presented and their advantages and disadvantages are discussed. The strategy that is recommended – constraint violation – is tested in a case study. The major advantage of this strategy is that it gives a sound compromise between the constraints. Another advantage is that it provides a numerical measure of the quality of the graphic generalisation process based on constraint violations. That is, if a constraint is severely violated in the generalisation process the user is warned. It is important to warn the user of potential problems, particularly in an interactive generalisation environment. This information helps the user to identify the parts of the map that were not solved properly by the automatic routine and which require editing by the user.

The next section presents a general introduction to constraints. Then follows a brief description of simultaneous graphic generalisation. Section 4 contains a discussion of strategies for setting weights, and the following section is devoted to a quality measure. The recommended strategy for setting the weights – constraint violation – and the quality measure are evaluated in a case study, which is described in Section 6. The paper is concluded with a discussion and conclusions.

2 Constraints

Several requirements must be fulfilled in the generalisation process. A possible framework for automatic generalisation is to formulate these requirements as constraints and let them control the process (Beard, 1991). The major difference between rules and constraints is that rules state *what is to be done* and constraints state *what results should be obtained*. Since it is difficult to formalise the generalisation process in form of rules (e.g. condition-action rules), several authors have proposed and used constraints in the generalisation process (e.g. Brassel and Weibel, 1988; Ruas and Plazanet, 1996; Harrie, 1999; Ruas, 2000). In this paper, three main categories of constraints are presented: *legibility*, *characteristic* and *position* (a modification of the typologies given in Ruas and Plazanet, 1996 and Weibel and Dutton, 1998).

Legibility constraints

The visual representation of cartographic objects is important. The data must not contain any spatial conflict, objects (and features within objects) must be large enough and not too detailed, and the chosen symbolisation must conform to graphic limits.

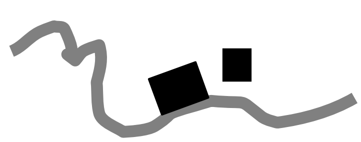


Figure 1: Two legibility constraints are violated. Some of the bends in the road object are too narrow, and there is a spatial conflict between one of the building objects and the road object.

Characteristic constraints

It is essential that the characteristics of single objects, as well as of groups of objects, are maintained in the generalisation process. Several constraints have been proposed for single objects, such as preservation of area and angularity. Defining constraints on groups of objects is more difficult, and often requires the preservation of object patterns, which is particularly difficult if objects are removed. Examples of constraints on groups of objects are: alignment of objects (Regnauld, 1996), mean distance between objects (Anders and Sester, 2000) and size distribution (Ruas, 2000).

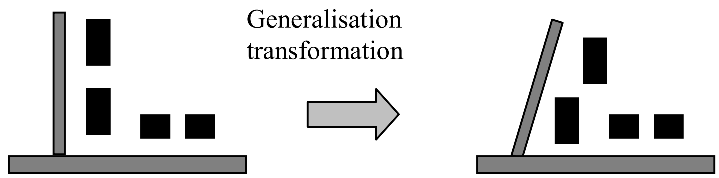


Figure 2: Two characteristic constraints are violated in the generalisation transformation. The angle between the road objects at the junction has changed, and the large building objects are no longer aligned.

Position constraints

Position constraints are concerned with the movement of objects in the generalisation process. There are two types of position constraints: *absolute* and *relative*. Absolute position constraints state that objects should not move in relation to the geodetic reference system, while relative position constraints dictate that distances between objects, as well as topological relationships, must be maintained.

3 Simultaneous graphic generalisation

Simultaneous graphic generalisation aims at computing the optimal solution according to a set of analytical constraints. The constraints are expressed as linear equations of point movements (of all points that make up the objects included in the generalisation process). This method implies that the constraints will constitute an equation system in which the unknowns are the point movements. Solving this equation system, which is performed by the least-squares method, gives the "ideal" point movements required to distort and move objects in the graphic generalisation transformation.

This section starts by presenting the constraints used in simultaneous graphic generalisation. Then, the general form of these constraints is stated, as well as the equation system formed by the constraints. The least-squares method is briefly described, a method that is used for solving the equation system. The section ends with a description of the geometrical meaning of constraint residuals.

The description of simultaneous graphic generalisation here is brief. Definitions of the analytical form of the constraints, rules for when the constraints are set up, computational details, etc., are given by Harrie and Sarjakoski (2001).

Constraints in simultaneous graphic generalisation

Simultaneous graphic generalisation has ten constraint types (as presented in Harrie and Sarjakoski, 2001) but more constraint types can be added. These constraint types are described below using the typology presented above. One should be aware that the displacement and exaggeration constraints may have different goals, depending on the context in which they are used. If there is a spatial conflict, the displacement constraints strive to increase the distance between the objects, otherwise the constraints strive to maintain the distance between the objects. Exaggeration constraints can be used to increase the size of an object and to maintain the shape.

Legibility constraints

- Displacement: Spatial conflicts are not allowed. These constraints are set up when the distance between objects is shorter than a predefined minimum distance.
- Simplification: Line and area objects should not contain more points than necessary to represent their characteristics. The simplification constraints force an unnecessary point to lie on the straight line between the two neighbouring points; the points that are unnecessary is determined by the area-based algorithm in Visvalingam and Whyatt (1993). The reason for not simply removing the unnecessary points is to control the spatial relationships to other objects; however, the unnecessary points are removed during a post-processing in our application, using the Douglas-Peucker algorithm (Douglas and Peucker, 1973).
- Smoothing: Line and area objects should not be too angular.
- Exaggeration: Objects, and features within objects, should be large enough to be clearly visible.

Characteristic constraints

- Curvature and Segment length: The characteristics of line and area objects must be maintained.
- Stiffness: The internal geometry of some objects must be invariant.
- Crossing: The angle between line objects in junctions must not change.
- Exaggeration: The shape of some objects must be maintained.

Position constraints

- Movement: Points should not move (absolute).
- Movement direction: Points on a line should not move in any direction across the line (absolute).
- Displacement: The distance between close objects that are not in spatial conflict, should not change (relative). These constraints are set up when the distance between objects is greater than the predefined minimum distance, but shorter than 1.5 times the minimum distance.

General analytical form of the constraints

One key issue is to find analytical expressions for the constraints. In simultaneous graphic generalisation the number of points is invariant, which enables the formulation of the constraints on point movements. For the sake of computational simplicity, we restrict ourselves to linear equations; that is, all the constraints are of the form:

$$const_{x1} \cdot \Delta x_1 + const_{y1} \cdot \Delta y_1 + \ldots + const_{xn} \cdot \Delta x_n + const_{yn} \cdot \Delta y_n = const_{obs} \quad (1)$$

where

Δx_i , Δy_i	are point movements,
$const_{xx}$	are constant values, and
n	is the total number of points.

All the constraints together constitute an equation system in which the point movements are the unknowns. In matrix form this can be written as:

$$\mathbf{Ax} = \mathbf{l} + \mathbf{v} \quad (2)$$

where

A	is the design matrix,
x	is a vector containing the unknown point movements,
l	is the observation vector (containing the right-hand side of Equation (1)), and
v	is the residual vector.

The residual vector has to be introduced since the Equation system (2) is over-determined. The method guarantees that either a movement or a simplification constraint is set up for each *x*- and *y*-coordinate.

That is, there are always at least as many constraints as unknowns, and in realistic applications there are about twice as many constraints as unknowns.

Least-squares method

The “best solution” of Equation system (2) is the one that agrees as far as possible with the constraints, i.e. we face a minimisation problem of a function of the residual vector. To solve the equation system the least-squares method is used, which minimises a weighted l_2 norm:

$$\mathbf{v}^{\mathrm{T}}\mathbf{P}\ \mathbf{v} \tag{3}$$

where

$\mathbf{P}$	is the weighting matrix, and
$\mathbf{v}^{\mathrm{T}}$	is the residual vector transposed.

The least-squares solution is given by:

$$(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})\cdot\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{P}\ \mathbf{l} \tag{4}$$

A normal-sized graphical generalisation application contains thousands of points, and Equation system (4) will contain twice as many unknowns as points. To solve these large equation systems we use the conjugate gradient method (as proposed by Sarjakoski and Kilpeläinen, 1999), which is a computationally efficient method for this kind of application. To solve the graphic generalisation problem in Figure 6, containing almost 1000 points, took a few seconds on a PC with Pentium II, 266 MHz processor. The processing time is, however, highly dependent on the application and the input parameters.

The whole process of simultaneous graphic generalisation can be viewed as follows. In the initial state, the legibility constraints are severely violated and the characteristic and position constraints are not violated at all. The generalisation process then distributes the violations more evenly over all the constraints; the degree to which the violations are spread is dependent on the norm (of the residual vector) being minimised. Minimising the l_2 norm has the property of quite evenly distributing the violations over the constraints (in comparison with minimising the l_1 norm). This is in accordance with the compromise sought between the constraints in simultaneous graphic generalisation. A severe violation of any constraint is problematic, but minor violations of several constraints are acceptable. Another advantage of minimising the l_2 norm is that it is computationally simple. Furthermore, as can be seen from Equation system (4), the compromise between the constraints is dependent on the weights stored in matrix **P**. In Section 4, strategies for determining the weights are discussed.

Geometrical meaning of constraint residuals

The constraint residuals are important quantities in setting the weights of the constraints and in quality assessment. In quality assessment, a large residual indicates that the constraint, i.e. the sought requirement, has not been fulfilled. Such measures can be relative (comparison of the relative size of the residuals; one such measure is given in Harrie and Sarjakoski, 2001) or absolute (control of the absolute size of each residual, Equation (7)). The geometrical meanings of the residuals are listed below. The list should be read as: this is the situation of constraint i if residual v_i is equal to σ.

- Displacement: The separation between two objects is σ metres shorter than the predefined minimum distance between the objects. (This predefined minimum distance is dependent on the symbol sizes of the objects and the minimum separation between symbols.) In the case when the distance between the objects was originally large enough, the displacement constraints aim at maintaining the distance between the objects. In such cases, a residual of value σ means that the distance has changed by σ metres.
- Simplification: The unnecessary point is σ metres from its ideal position on the straight line between neighbouring points (along one of the coordinate axes). However, if σ is sufficiently small, the point will be removed anyhow by the Douglas-Peucker algorithm in the post-processing.

- Smoothing: The point is σ metres along one of the coordinate axes, from its ideal relative position computed by a Gauss smoothing algorithm.
- Exaggeration: The point is σ metres along one of the coordinate axes, from the ideal relative position to satisfy the exaggeration and shape preserving requirements.
- Curvature: The angle between two consecutive line segments has changed by σ radians.
- Segment length: The length of a line segment has been changed σ metres.
- Stiffness: The point has changed its relative position σ metres from other points in a rigid object (along one of the coordinate axes).
- Crossing: The point is σ metres from the ideal position, along one of the coordinate axes, to satisfy the condition that the angles at a junction are invariant.
- Movement: The point has moved σ metres along either of the coordinate axes.
- Movement direction: The area between the original line and the new line is "locally" σ square-metres.

4 Weight-setting in simultaneous graphic generalisation

This section is devoted to strategies for setting weights in simultaneous graphic generalisation. The discussion is based on the following assumptions:

- The weights are functions of the type of constraints and type of objects only. That is, the weights are not dependent on the shape or other individual properties of the objects.
- All the constraints are considered to be independent.

These assumptions may be questioned. For some type of constraints it is natural that the weights are independent of the shape of the object. For other constraints, such as maintaining the curvature, it can be motivated by using different weights for objects of the same type. However, establishing individual weights for objects is beyond the scope of this investigation, and is a subject for future studies. (In Harrie and Sarjakoski, 2001 the movement and displacement constraints have somewhat individual weights, but in this study these weights are redefined to be the same for all the constraints.)

The second assumption means that the residuals of two constraints (A and B) are a priori independent (which implies that P_{AB} and P_{BA} are equal to zero; i.e. $\mathbf{P}$ is a diagonal matrix). A model without any dependencies between the constraints is a simplified model. There may be correlations between the constraints that could change the relative importance of the constraints from the desired relative importance set by the weights.

In this section, four strategies for setting the weights in simultaneous graphic generalisation are described: *empiricism*, *machine learning*, *constraint violation* and *variance component estimation*. The basic concepts behind these strategies are quite different. Constraint violation is an a priori strategy, empiricism a trial-and-error strategy, machine learning aims at creating relationships between measures of the data and weights, and variance component estimation is an a posteriori statistical method. The section concludes with a discussion of the most appropriate strategy (consistent with the two assumptions stated above).

None of the strategies gives weights that can be used for all type of applications, but they should be capable of determining weights that can be used for all applications of a certain type. The variance component estimation strategy is, however, an a posteriori method that uses the properties of the constraints; i.e. new weights are set for each application.

Empiricism

This is basically a trial-and-error strategy. The user has a rough conception of the importance of the constraints and defines a preliminary set of weights for each type of constraint and object. The weights are then improved by iterative testing. The disadvantage of this strategy is that new applications require laborious tuning of the weights. Furthermore, since the number of weights may be very large, it is difficult to determine the relative importance of each weight by pure testing. Without any theory of the importance of each constraint type, there is a risk that the testing procedure may give too low values of some weights, and thus the corresponding constraints will be almost neglected in the generalisation process. An empirical strategy for setting weights was used in Harrie and Sarjakoski (2001).

Machine learning

Machine learning, in this context, requires a training set of good computer-generalised maps, together with information about the original cartographic data and the algorithms/parameters used. This information is then used by a machine-learning algorithm to establish relationships between the cartographic data (e.g. shape measures) and the recommended algorithms/parameters. This strategy has recently been used for parameter-setting in line generalisation (Lagrange et al., 2000; using a neural network).

As long as the weights only are dependent on the type of constraints and objects, there is no need for machine learning. However, under the circumstance that the weights also should be dependent on the shape or other characteristics of objects (or group of objects) machine learning may be a suitable strategy for determining the relationships between the measure (of the characteristics) and the weights. The main problem with this strategy would probably be to establish useful measures of the characteristics (cf. Lagrange et al., 2000) and the fact that an extensive training set, for each type of application, would be required.

Constraint violation

This strategy for setting the weights is based on an a priori estimate of acceptable violations to the constraints, which are called *allowed violation values*. Ideally, the *allowed violation values* for the constraints would be related to the quality requirements on the map, in e.g. the metadata. Regarding Equation (3), the “best” relationship between the weight *i* ($P_{i,i}$) and the *allowed violation value* for constraint *i* is:

$$P_{i,i} = \frac{1}{\left(allowed\ \ violation_i\right)^2} . \quad (5)$$

Equation (5) is “best” in the sense that it provides the following property: if two constraints are contradictory (and neither of them is affected by other constraints), the violations of the two constraints would be equal when measured in relation to the *allowed violation values*. In a real application, the dependencies between the constraints are much more complex, but Equation (5) still gives, theoretically, a sound compromise between the constraints.

A minor test is applied below to demonstrate the use of the constraint violation strategy. In Figure 3 an example of simultaneous graphic generalisation is shown, where generalisation is performed by the Matlab program described in Harrie (1999). The map was generalised by applying movement, stiffness, curvature, segment length and displacement constraints, where the weights were dependent only on the type of constraints. The weights were set using Equation (5), with the *allowed violation values* in Table 1.

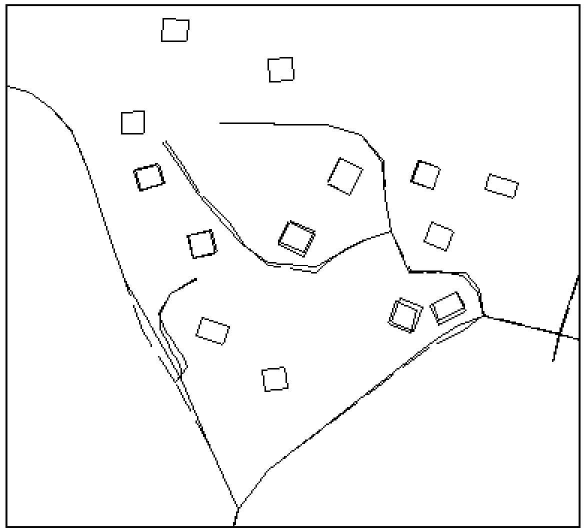


Figure 3: Simultaneous graphic generalisation of a map using constraint violation strategy for weight-setting. Movement and stiffness constraints were used for the building objects, and movement, curvature and segment length constraints were used for the road objects. Displacement constraints were set up between close-lying objects. The original map is drawn with solid lines and the generalised map with dashed lines.

Movement	Stiffness	Curvature	Segment length	Displacement
1.0 m	0.05 m	0.05 rad	0.5 m	0.5 m

Table 1: *Allowed violation values* for simultaneous graphic generalisation of the map in Figure 3.

To investigate whether the intended requirements on the map were fulfilled, a study of the residuals (in vector **v**) was performed (Figure 4). As shown in the histograms, mainly the curvature and the displacement constraints were violated. The problems associated with the curvature constraints are also seen in Figure 3, where some road objects have lost their characteristics. The residuals of the movement constraints were generally small, since constraints of this type are set up irrespective of whether any conflicts are present. This differs from the case of displacement constraints which are only set up when there is a (spatial) conflict. Furthermore, the mean value of the violations of the displacement constraints is not zero. This stems from the fact that the displacement constraints are, mainly, set up if the distance between objects is too short. To fulfil the constraints, the distance must be increased so as to be equal to the specified minimum distance; while at the same time, the characteristic and position constraints strive to maintain the distance. Hence, the violations of the displacement constraints are most often negative.

There are some disadvantages of the constraint violation strategy for weight-setting. The strategy requires that *allowed violation values* of the constraints have a clear meaning, and that the values are related to the quality of the map (ideally, to the quality specification in the metadata). In simultaneous graphic generalisation, violations of most types of constraints have a clear geometrical meaning, but there are exceptions. For example, violations of the movement direction constraints are related to the "local area" between the original and generalised line, a quantity that is seldom used when stating the quality of a map. Another disadvantage of the constraint violation strategy is that it may be too rigid. For example, dependencies between the constraints may change the relative importance of the constraints from the original importance set by the weights, and therefore, the weights should be somewhat changed. However, a pure constraint violation strategy does not include any possibility of such changes.

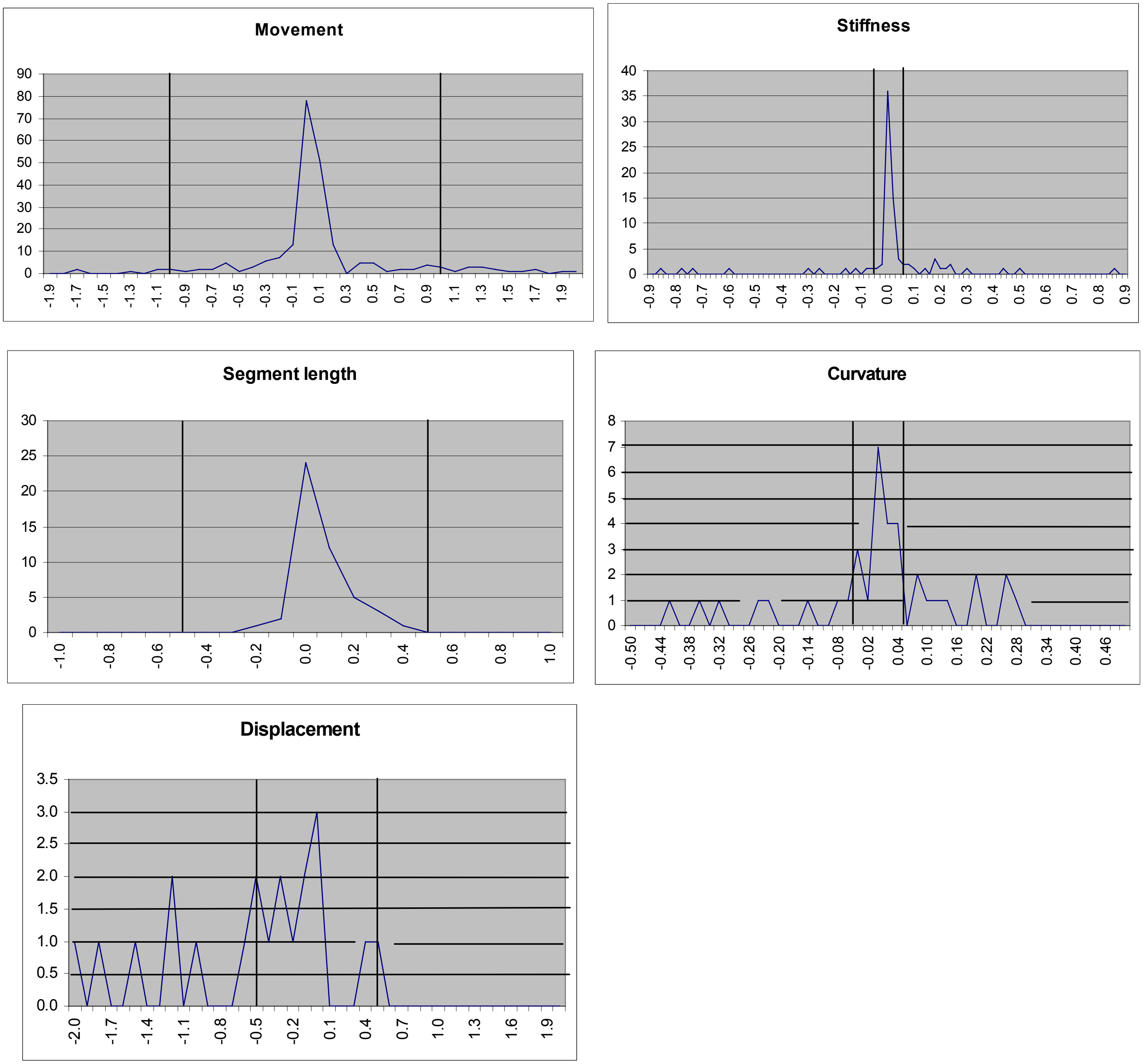


Figure 4: Histograms of the residuals of the constraints in the simultaneous graphic generalisation of the map in Figure 3. The vertical lines indicate the *allowed violation* values.

Variance component estimation

This is an a posteriori strategy for determining the weights, based on a statistical approach. From a statistical point of view, the optimal weighting matrix is equal to the inverse of the variance-covariance matrix of the constraints (the absolute values are not interesting, it is only required that the relative size of the variances and covariances be correct). Several techniques have been developed to estimate the variance-covariance components from the original linear equations (see, for example, Koch, 1987 and Persson, 1981). The question is whether these techniques can be used to improve the weights in simultaneous graphic generalisation or, in other words, if a statistical optimal solution is a cartographically good solution.

In this study, the constraints are considered to be a priori independent which, in statistical terms, implies that the constraints are a priori uncorrelated. This assumption is also used below in the a posteriori analyses, i.e. only the variances need to be estimated. Furthermore, the weights are only dependent on the type of objects and constraints. That is, it is only necessary to estimate a variance for each type of constraint and object; and for the sake of simplicity, the variance is only dependent on constraint type in the discussion below. The a posteriori variances (σ_j^2) for constraint type j are estimated as (cf. Persson, 1981):

$$\sigma_j^2 = \frac{\underline{\mathbf{v}}_j^{\mathrm{T}} \underline{\mathbf{v}}_j}{r_j} \tag{6}$$

where

$\underline{\mathbf{v}}_j$ is a vector containing the residuals for constraint type j, and

r_j is the sum of the diagonal elements of the matrix $\mathbf{I}\text{-}\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{P}$ corresponding to constraint type j (r_j is related to the contribution of the constraint type j to the over-determination of the equation system), where

I is the identity matrix of dimensions n x n, where n is the number of constraints.

It should be noted that the terms *variance* and *covariance* are not used in a true statistical sense here, since the expectation values of the residuals are not always zero.

In Figure 5 the same map as in Figure 3 has been generalised with the same constraints. In this test, however, the weight matrix for the constraints in the simultaneous graphic generalisation was set to the inverse of the a posteriori variance matrix, where the a posteriori variances were determined by Equation (6), using the residuals given in Figure 4. The map in Figure 5 was hardly generalised at all. The reason for this is that the displacement constraints had a low weight, which, in turn, was caused by the large residuals of the displacement constraints in the first generalisation (see Figure 4). This change in weights for the displacement was comparable to setting the *allowed violation values* three times larger than those in Table 1. Another reason for the low level of generalisation in Figure 5 was that the weights for movement constraints had increased (since there were many movement constraints with low residuals in the first generalisation). This result stresses an important property of the variance component estimation strategy: if a type of observation (here: constraint) has large residuals the weights will decrease. Variance component estimation techniques have been developed for applications where the equations in the least-squares method (cf. Equation (1)) are linearised observation equations with random errors (mostly considered to be normally distributed). In such applications it is, of course, statistically correct to decrease the weight of an observation type with large random errors. However, in simultaneous graphic generalisation it is not the case that a constraint is less important just because it is often violated. To conclude, in simultaneous graphic generalisation the constraints, as such, do not contain any information about the *importance* of the constraints.

The estimate of the variances from Equation (6) is dependent on the a priori weights (or the a priori variance-covariance matrix). To estimate the statistically optimal weights, i.e. to find the relative variances of the constraints, an iterative method is required. One such method, the Förstner estimator (Förstner, 1979), was tested for estimating the weights in simultaneous graphic generalisation, but the method never converged. The lack of convergence may be due to: too few over-determinations, the fact that the correlation of the constraints was badly modelled, or the distribution of the residuals (in particular the fact that the expectation value of the displacement constraint residuals is not zero, see Figure 4). However, as argued above, the statistical optimal weights are not the best weights from a cartographic point of view, and therefore a discussion of the convergence problem is outside the scope of this paper.

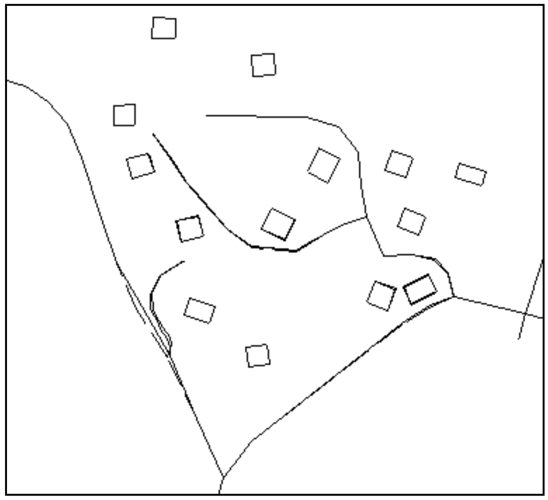


Figure 5: Simultaneous graphic generalisation of a map using variance component estimation strategy for weight-setting. Movement and stiffness constraints were used for the building objects, and movement, curvature and segment length constraints were used for the road objects. Displacement constraints were set up between close-lying objects. The original map is drawn with solid lines and the generalised map with dashed lines.

Choice of strategy for the case study

The constraint violation strategy seems to be the most promising of the four strategies discussed above for determining the weights in simultaneous graphic generalisation. This statement is based on the assumptions regarding the weights and constraints stated at the beginning of this section. The main advantages with the constraint violation strategy are that it, theoretically, gives a sound compromise between the constraints and that the weights are determined in connection with the quality requirements of the map. The constraint violation strategy was therefore chosen to be used in the case study described in Section 6.

5 A quality assessment measure

Ideally, the user of a generalisation system should be informed about the quality of the automatic routines; i.e. a quality measure is required. In this section, a quantitative measure provided by the constraint violation strategy for setting the weights is presented. The measure is based on the residuals of the constraints, which is logical in the sense that the constraints are the desired requirements on the map. The measure is called *quality level* and is defined as:

$$quality\ level_i = \sqrt{v_i^2 \cdot P_{i,i}}\ . \tag{7}$$

This measure has the following two attractive properties:

- If the absolute value of the residual (v_i) is equal to the *allowed violation value* then the *quality level* will be equal to one.
- The *quality level* is proportional to the absolute value of the residual.

It is, however, not sufficient to have a measure of quality; a way of informing the user of the quality is also necessary. Ideally, the user of a generalisation system should be visually warned about potential problems in the generalisation process. In our implementation of simultaneous graphic generalisation, a warning object is created for each constraint whose *quality level* is greater than a threshold value (see Figure 6d). These warning objects have the same position as the points of the violated constraints and, in addition, they have two attributes: type of constraint and *quality level*. The idea is that the user immediately notices which warning objects were created in the generalisation process (in our implementation they are symbolised by red and lie on top of all other objects). The user can then select the warning objects and check which type of constraints have been violated and the severity of the violation. From this information, and by visual assessment, the user can then decide if any editing of the generalised data is necessary.

6 A case study using the constraint violation strategy for weight-setting

In this case study, a cartographic data set on the scale of 1:10,000 was generalised to the scale of roughly
1 : 50,000 (see Figure 6). The aim of the case study was to evaluate the constraint violation strategy for setting weights and the proposed quality measure.

The data were generalised as follows. Firstly, a model generalisation step was performed interactively using built-in functionalities in the map production software LAMPS2 (Laser-Scan, 1999) and some of our own routines. In this case study, the model generalisation step was only a preparatory step for graphic generalisation, and therefore, the model generalisation was not evaluated. Secondly, graphic generalisation was performed by a C++ implementation of simultaneous graphic generalisation which communicates with LAMPS2 via ASCII files (see Harrie and Sarjakoski, 2001 for details). The quality assessment of the graphic generalisation was performed both visually and quantitatively.

Model generalisation

The original cartographic data set consisted of the object types: *building-residence*, *building-other*, *major road*, *minor road*, *dirt road*, *field* and *power line*. The new data set had almost the same object types. The major difference was that the building objects in this data set were of type *building-point* (point objects) or *building-area* (area objects). The following rules were applied in the model generalisation step (see Figures 6a and 6b).

Building objects

- *Building-residence* and *building-other* objects larger than 250 m^2 were represented as *building-area* objects with the same geometry as the original objects.
- *Building-residence* objects smaller than 250 m^2 were represented as *building-point* objects, where the direction of the symbol corresponded to the longest side of the original *building-residence* object.
- *Buildings-other* objects smaller than 250 m^2 were not represented.

Road objects

- *Minor road* objects shorter than 100 metres and not leading to a *building-residence* object were omitted.
- *Dirt road* objects were not represented.

Field objects

- Small islands were removed.

Graphic generalisation

This step was performed by simultaneous graphic generalisation, where the weights of the constraints were set by the constraint violation strategy. In Table 2, the *allowed violation values* for each type of constraints and type of objects are given. The weights were computed from these values using Equation (5).

In the new data set, six object types were used. Objects of these types should behave differently in the graphic generalisation process. The reasons for choosing those constraints and *allowed violation values* in Table 2 are briefly as follows. *Building-area* objects are regarded as being rigid objects. In this case study, the rigid property was implemented by only allowing small deviations from the exaggeration constraints. The reason to use exaggeration constraints, rather than stiffness constraints, was that the *building-area* objects should be enlarged at the same time as the shape should be maintained. For *building-point* objects only the position of the object was of interest; hence, only displacement and movement constraints were used. *Major* and *minor road* objects should be plastic in the graphic generalisation process, and the objects should be simplified and smoothed. Therefore, simplification, smoothing, movement, curvature, segment length and movement direction constraints were used as individual constraints for these objects, and displacement and crossing constraints for relationships to other objects. The *allowed violation values* were generally smaller for the *major road* objects than for *minor road* objects. *Power line*, and *field* objects should also be plastic. However, these objects represent man-made entities which often have an angular impression; therefore, smoothing constraints were not set up for these objects.

	Building-area	*Building-point*	*Major road*	*Minor road*	*Field*	*Power line*
Simplification (m)	*	*	1.0	1.0	1.0	1.0
Smoothing (m)	*	*	1.0	1.0	*	*
Exaggeration (m)	0.2	*	*	*	*	*
Displacement (m)	1.0	1.0	1.0	1.0	1.0	1.0
Movement (m)	8.0	8.0	4.0	8.0	8.0	8.0
Stiffness (m)	*	*	*	*	*	*
Curvature (rad)	*	*	0.05	0.1	0.1	0.05
Segment length (m)	*	*	1.0	2.0	2.0	2.0
Movement direction (m^2)	*	*	10.0	20.0	20.0	10.0
Crossing (m)	*	*	1.0	1.0	1.0	1.0

Table 2: *Allowed violation values* for the constraint types and object types in the case study. The geometrical meaning of the residuals of the constraints are given in Section 3. An asterisk indicates that the constraint type was not applicable to that object type. The displacement and crossing constraints may contain objects of more than one object type. For these constraints, the program uses the lowest *allowed violation value* (in this case study, the *allowed violation values* are equal for all object types).

Assessment of the result of the graphic generalisation step

The results of the simultaneous graphic generalisation were assessed both visually and quantitatively. The visual assessment was performed by studying Figures 6b and 6c, and the quantitative assessment was performed by studying the *quality levels* (Figures 6d and 7).

In the model-generalised map (Figure 6b) there were some conflicts that had to be solved by graphic generalisation. These conflicts were mainly spatial conflicts between *road* and *building* objects, *field* objects that were too detailed, and *building-area* objects that were too small in relation to *building-point* objects. However, there was no need for major simplification or smoothing of the *road* objects

By studying Figure 6c we can see that most conflicts were solved. For example, there are some spatial conflicts and a size conflict (too small *building-area* object) in the top left circle in Figure 6b. As shown in Figure 6d, the simultaneous graphic generalisation process was able to solve these conflicts. There were basically just two circumstances under which the spatial conflicts could not be solved. The first case was related to a lack of free map space and the second was dependent on the rules for setting up displacement constraints; the latter is illustrated here by an example. In the bottom right circle in Figure 6b, the *building-area* object was too small and was too close to the *road* object just left of the *building-area* object. Between the *building-area* object and the left part of the *minor road* object a displacement constraint was set up. However, when the *building-area* object was enlarged and moved away from the *road* object a new spatial conflict was introduced to the same *road* object below the *building-area* object. Due to the rules for setting up displacement constraints, no constraint was set up between the *building-area* object and that part of the *road* object. See Harrie and Sarjakoski (2001) for definitions of the rules for setting up displacement constraints and a discussion about their applicability.

(a)

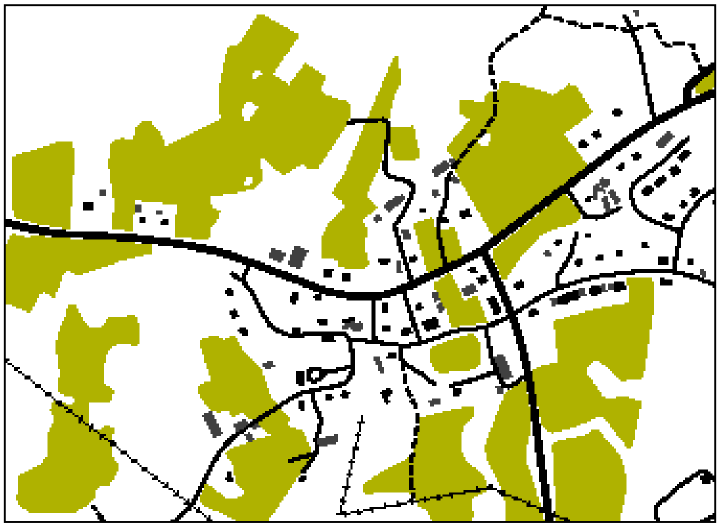


Figure 6: Case study of constraint violation strategy for weight-setting in simultaneous graphic generalisation.

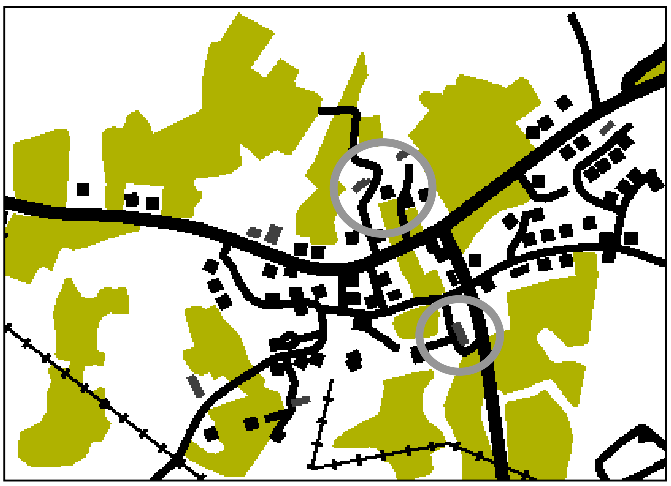
(a) the original map, (b) the model-generalised map, (c) the model- and graphic-generalised map, (d) the warning objects created in the graphic generalisation, and (e) an explanation of the geometry of the warning objects.

In (a) the *building-residence* objects are represented in black and other building objects in grey. In Figures (b), (c) and (d) *building-area* objects are represented in grey and *building-point* objects are represented by black squares. The warning threshold in (d) is equal to 2.

Model generalisation

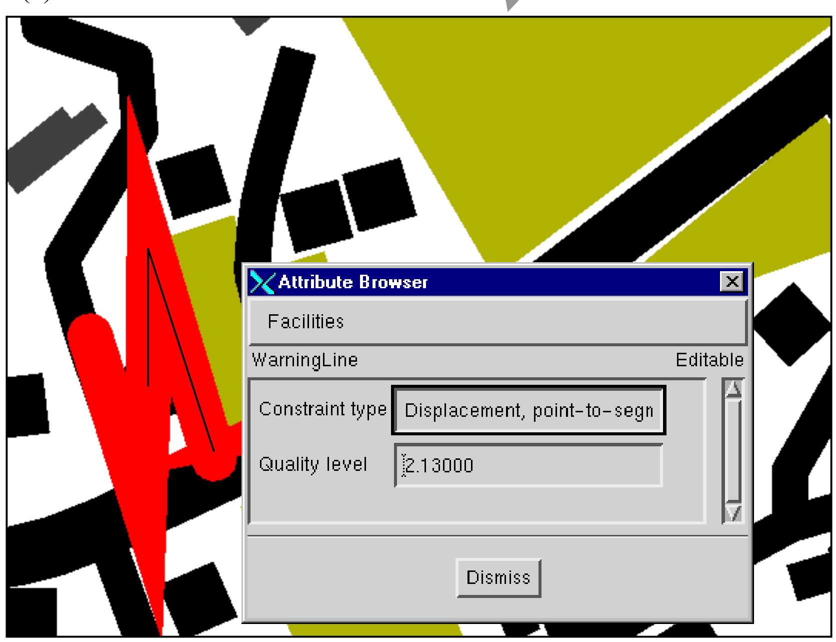
Graphic generalisation

(b)



(c)

(d)



The original cartographic data was provided by the National Land Survey of Sweden.
Printed by permission 507-98-4091.

The quantitative assessment was mainly performed by creating warning objects, as described in Section 5 above. In, for example, Figure 6d two warning objects were created due to violations of displacement constraints (between a point and a segment, see Figure 6e). The violations were caused by a lack of available map space for the *building-point* object; it was squeezed between a *minor road* object and a *field* object (see Figures 6b and 6c). One of the warning objects was selected and in an attribute browser it is stated that the *quality level* is roughly two (Figure 6d). With regard to Table 2, which states that the *allowed violation* was 1 metre, and by considering Equations (5) and (7), we can conclude that the *building-point* object was about two metres closer to the field object than the predefined minimum distance.

The number of warning objects created is dependent on the warning threshold set by the user. In this case study, there were 2 *quality levels* larger than 3, 22 larger than 2, and 121 larger than 1 (of the total of around 4000 constraints). That so many constraints had been violated might be surprising based on a visual assessment of Figure 6c. However, the *allowed violation values* given in Table 2 are quite low which implies that a *quality level* of about 1 is not always detected visually. Based on the resulting *quality levels*, a threshold value between 2 and 3 can be recommended for quality assessment in this application.

Regards Figure 7, the displacement, movement and simplification constraints are violated to a similar degree in relation to their *allowed violation values*. A study of the spatial distribution of the *quality levels* gave that values of the same size level are often found in a neighbourhood. For example, in Figure 6e the *quality level* of the "other" warning object is roughly equal to 2 similar to the selected warning object in Figure 6d. The interpretation of this is that one or more initial severe violations of legibility constraints has been spread to similar size violations to several constraints in the neighbourhood. That is, if the *allowed violation values* really reflects the relative importance of the constraints, the constraint violation strategy gives a sound compromise in the neighbourhood.

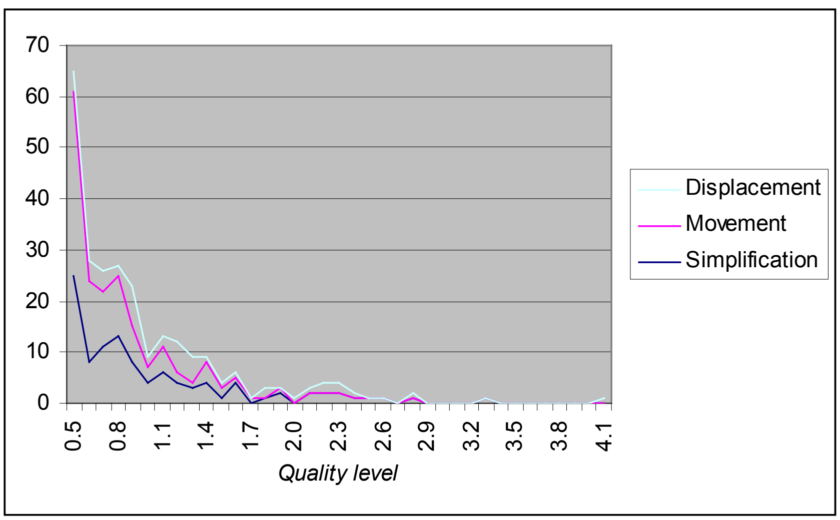


Figure 7: Histogram of the *quality levels* of the displacement, movement and simplification constraints in the graphic generalisation step illustrated in Figure 6.

7 Discussion

Graphic generalisation is required both in map production and in real-time navigation systems. Today, most automatic routines for graphic generalisation are based on sequentially applying algorithms for the graphic generalisation operators, where the objects are treated in isolation. This paper is, however, concerned with simultaneous graphic generalisation; a method that solves the graphic generalisation in a single step, where relationships between objects are also modelled. Simultaneous graphic generalisation is based on fulfilling analytical constraints. In Harrie and Sarjakoski (2001) ten constraint types are described; applying these constraints enables modelling of quite different behaviour in the generalisation process. The final solution is a compromise between all the constraints, where the compromise is dependent on the weights of the constraints. This paper is concerned with finding the optimal weights of the constraints.

The most interesting parts of this paper are the discussion and analysis of different strategies for setting the weights. This discussion led to the recommendation of using a constraint violation strategy for setting the weights in simultaneous graphic generalisation (and in similar methods based on analytical constraints and the least-squares method). The main advantages of the constraint violation strategy are:

- The violations of constraints in the same neighbourhood will be of similar size in relation to the *allowed violations*, which makes the solution a sound compromise between the constraints in the neighbourhood.
- It is a simple method.
- It is related to the quality measures of maps.
- It provides a clear measure of quality (*quality level*, see Equation (7)).

The recommendation of constraint violation strategy is based on the assumptions that: the weights are only dependent on the types of constraints and objects, and all the constraints are independent. If, for example, the weights were to be dependent on the shape of the objects, modification of the constraint violation strategy would be necessary. This modification could probably be realised by defining the quantity *allowed violation* as a function of shape measures. To which degree this is possible, and if a machine-learning approach could be used for this purpose, is an open question.

In the case study, the *allowed violation values* were set a priori. However, under certain circumstances a combination of a priori and empiricism strategies may be necessary. The constraint violation strategy could be seen as a theory of the relative importance of the constraints, but one should not be afraid to change the weights if the results are not satisfactory. In connection with setting the weights, an analysis of the residuals of the constraints should be performed, as well as a visual assessment, to evaluate whether the sought requirements have been obtained. If problems arise, for example, with the legibility constraints, it should be considered whether some of the characteristic and/or position constraints have lower *allowed violation values* than necessary, etc. Changes in *allowed violation values* are most likely to be made for those constraint types whose residuals do not have any clear geometrical meaning (e.g. movement direction constraints).

Quality assessment is often a neglected issue in cartographic generalisation. Most commercial and research systems lack the ability to warn the user about potential problems. The assessment technique proposed in this paper is based on creating a warning object for each constraint that is violated (see Figure 6d). This is a clear and informative method. The user can choose the warning level according to his preferences, and may also choose a form of visualisation of the warning objects that is dependent on the type and severity of the violation (the latter has not yet been implemented). Furthermore, the user is given a numerical value that describes the severity of the violation which is easy to interpret, for example, this building object lies σ metres too close to the road object.

The main drawback of the proposed quality assessment method is that the same constraints are used for both controlling and assessing the generalisation process. This might create a too optimistic view of the overall quality of the map. For example, under certain circumstances, some displacement constraints are not set up although they would improve the results (due to the difficulty of providing a definition of a spatial conflict that is valid in all cartographic situations). This implies not only that the spatial relations between the objects will be badly handled in the generalisation process, but also that the quality assessment technique will be unable to identify this failure (since no constraints are violated). It would be feasible to use different sets of constraints to control the procedure and assess the results. However, an argument against this approach is as follows. If we regard a requirement of the generalised map as being important, we should strive to fulfil this requirement in the process (i.e. setting up a constraint), and not only check if the requirement is fulfilled afterwards.

The quantitative assessment of the case study (e.g. Figure 7) indicates that the graphic generalisation process has distributed the violations evenly within a neighbourhood. The question is whether it is possible to distribute the violations over a larger region, thus decreasing the highest *quality level*. This is highly dependent on the geometrical situation; for isolated objects (e.g. to remote building objects) that are in conflict, it is not possible to spread the violations to any other constraints. However, these situations are uninteresting from a simultaneous graphic generalisation perspective. The main advantage of this method is that it can solve situations where there are several related conflicts in a region. To solve these situations, techniques other than the least-squares method and constraint violation strategy should perhaps be used. It might be possible to improve the solution by using a post-processing strategy for weight-setting that increases the weight of violated constraints. Theoretically, the solution might be improved by minimising another norm of the residual vector (e.g. the l_3 or l_4 norm), but this is complicated. Another solution might be to integrate the graphic generalisation process into a conceptual framework that can navigate back if an acceptable solution is not found, where the conceptual framework should enable objects to be removed before the process is run again (cf. Ruas and Plazanet, 1996).

8 Conclusions

During recent years, a number of generalisation methods based on a compromise between constraints, have been developed. These methods require that the relative importance of the constraints be stated. In this paper, strategies for determining the importance of the constraints in simultaneous graphic generalisation are discussed. A strategy based on constraint violations is recommended, under the given assumptions. Visual assessment of the results shows that this strategy gives a sound compromise between the constraints. Furthermore, the strategy also provides a clear and informative quality measure. The paper presents a visual method based on this measure to help the user in assessing the generalisation quality.

Acknowledgements

This research was financially supported by the National Land Survey of Sweden and Lund University. I would like to thank the National Land Survey of Sweden for providing cartographic data sets, and especially Anders Rosén for practical help with the transfer of the data. Comments on draft versions from Leif Svensson (Lund University), Tapani Sarjakoski (the Finnish Geodetic Institute), Jonas Ågren (the Royal Institute of Technology, Stockholm) and Bengt Rystedt (the National Land Survey of Sweden) are appreciated. Finally, I would like to thank Helen Sheppard for correcting and improving my English.

References

Anders, K.-H., and M. Sester, 2000. Parameter-Free Cluster Detection in Spatial Databases and its Application to Typification. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B4, Amsterdam, pp. 75-82.

Beard, M. K., 1991. Constraints on Rule Formation. In Buttenfield, B. P., and R. B. McMaster (eds), *Map Generalization: Making Rules for Knowledge Representation*, Longman Group, pp. 121-135.

Brassel, K. E., and R. Weibel, 1988. A Review and Conceptual Framework of Automated Map Generalization. *International Journal of Geographical Information Systems*, Vol. 2, No. 3, pp. 229-244.

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 Topografic Information from Images and Maps*, Birkhaeuser Verlag, 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.

Förstner, W., 1979. Konvergenzbeschleunigung bei der a posteriori varianzschätzung. *Zeitschrift für Vermessungswesen*, No. 4, pp. 149-156.

Harrie, L., 1999. The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science*, Vol. 26, No. 1, pp. 55-69.

Harrie, L., and T. Sarjakoski, 2001. Simultaneous Graphic Generalization of Vector Data Sets. Submitted to *GeoInformatica*.

Højholt, P., 2000. Solving Space Conflicts in Map Generalization: Using a Finite Element Method. *Cartography and Geographic Information Science*, Vol. 27, No. 1, pp. 65-73.

Koch, K.-R., 1987. Parameter Estimation and Hypothesis Testing in Linear Models, Springer.

Lagrange, F., B. Landras, and S. Mustière, 2000. Machine Learning Techniques for Determining Parameters of Cartographic Generalisation Algorithms. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B4, Amsterdam, pp. 718-725.

Laser-Scan, 1999. *Gothic LAMPS2 v3.2, Online Documentation*, Cambridge, UK.

Mackaness, W. A., 1994. Issues in Resolving Visual Spatial Conflicts in Automated Map Design. *Proceedings of the* 6^{th} *Spatial Data Handling Symposium*, Edinburgh, pp. 325-340.

Müller, J.-C., J.-P. Lagrange, and R. Weibel (eds), 1995. *GIS and Generalization*, Gisdata 1, Taylor & Francis.

Mustière, S., J.-D. Zucker, and L. Saitta, 2000. An Abstraction-Based Machine Learning Approach to Cartographic Generalisation. *Proceedings of the* 9^{th} *Spatial Data Handling Symposium*, Beijing, pp. 1a.50-1a.63.

Nickerson, B. G., 1988. Automated Cartographic Generalization for Linear Features. *Cartographica*, Vol. 25, No. 3, pp.15-66.

Persson, C.-G., 1981. *On the Estimation of Variance Components in Linear Models and Related Problems*, Doctoral Thesis, Department of Geodesy and Photogrammetry, Royal Institute of Technology, Stockholm.

Regnauld, N., 1996. Recognition of Building Clusters for Generalization. *Proceedings of the* 7^{th} *Spatial Data Handling Symposium*, Delft, the Netherlands, pp. 185-198.

Regnauld, N., A. Edwardes, and M. Barrault, 1999. Strategies in Building Generalisation: Modelling the Sequence, Constraining the Choice. *Third Workshop on Progress in Automated Map Generalization*, Ottawa.

Ruas, A., and C. Plazanet, 1996. Strategies for Automated Generalization. *Proceedings of the* 7^{th} *Spatial Data Handling Symposium*, Delft, the Netherlands, pp. 319-336.

Ruas, A., 1999. *Synthesis of a Test on Generalisation*, Report from OEEPE Working Group on Generalization, Draft version, OEEPE Publication in 2001.

Ruas, A., 2000. The Roles of Meso Objects for Generalisation. *Proceedings of the* 9^{th} *Spatial Data Handling Symposium*, Beijing, pp. 3b.50-3b.63.

Sarjakoski, T., and T. Kilpeläinen, 1999. Holistic Cartographic Generalization by Least Squares Adjustment for Large Data Sets. *Proceedings of the* 19^{th} *International Cartographic Conference*, Ottawa, pp. 1091-1098.

Sester, M., 2000. Generalization Based on Least Squares Adjustment. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B4, Amsterdam, pp. 931-938.

Ware, J. M., and C. B. Jones, 1998. Conflict Reduction in Map Generalization Using Iterative Improvement. *GeoInformatica*, Vol. 2, No. 4, pp. 383-407.

Weibel, R., and G. Dutton, 1998. Constraint-Based Automated Map Generalization. *Proceedings of the* 8^{th} *Spatial Data Handling Symposium*, Vancouver, pp. 214-224.

Visvalingam, M., and J. D. Whyatt, 1993. Line Generalisation by Repeated Elimination of Points. *The Cartographic Journal*, Vol. 30, No. 1, pp. 46-51.