Project Number	**ESPRIT / LTR /24 939**
Project Title	**AGENT**
Deliverable Type	**Report**

Deliverable Number	**D D2**
Internal ID	**AG-99-05**
Contractual Date of Delivery	**01/03/99 (Technical Annex)**
Actual Date of Delivery	**23/02/99**
Version	**2.3**
Workpackage/Task contributing	**D1**
Authors	**UNI-ZH**
Confidentiality	**Public**

Title of Deliverable	**Selection of Basic Algorithms**

Abstract

This report provides a state of the art in algorithms for cartographic generalisation. It provides an operator classification and points out algorithms commonly in use, as well as new approaches. After an assessment, a selection of algorithms for the project prototype is presented.

Keyword List

generalisation algorithms, feature characterisation, algorithm hierarchy, measures

Executive Summary

The AGENT Technical Annex describes the objectives of Task D1 as "to identify a set of generalisation algorithms and their conditions of use according to the nature and geometric characteristics of an agent". While the identification of generalisation algorithms is described in the "D1-Report on algorithm specifications", this report represents the state of the art and is extended by an assessment and some recommendations for the prototype.

The aims of this report are, in part:

- to build a common basis for the choice of algorithms to be used in the prototype
- to present a list of the most useful algorithms available to help partners stay current with generalisation research
- to identify missing algorithm classes, so that deficiencies may be filled during D1
- to define a common classification of operators

Driven by these aims, we present a classification of the operators underlying cartographic generalisation. The individual operators are explained and partly illustrated. Both accepted algorithms already commonly in use and newer, experimental algorithms and approaches for additional functionality are described in the Appendix.

The report closes with an assessment describing the deficiencies of simplification algorithms as the primary tools for cartographic generalisation. The generalisation of complex objects such as roads, not just simple lines, creates the need for more specialised algorithms. The extensive research by the IGN supporting more sophisticated treatment of geographic features allows us to integrate new, powerful routines. When adopting such specialised algorithms, it is essential that we are able to characterise and delineate important phenomena with strong and robust measures.

Table of content

1 Introduction

1.1 Purpose

The AGENT Technical Annex describes the objectives of Task D1 as “to identify a set of generalisation algorithms and their conditions of use according to the nature and geometric characteristics of an agent”.

While the identification of generalisation algorithms is described in the “D1-Report on algorithm specification”, this report is a state of the art in the field of cartographic generalisation. As such the aims of this report are:

- to build a common basis for the choice of algorithms to be used in the prototype;
- to present a list of the most useful algorithms available to help partners stay current with generalisation research;
- to identify missing algorithm classes, so that deficiencies may be filled during D1;
- to define a common classification of operators;

1.2 Structure of this report

Driven by the purpose stated in 1.1, the report is divided into 3 main parts which cross-reference each another.

Chapter 2 deals with the identification and classification of the operators underlying cartographic generalisation. A classification or typology of operators is presented; the individual operators are explained and partly illustrated.

A review of existing algorithms is presented in Chapter 3. The list is structured using the classification developed in Chapter 2. A short assessment of D1-algorithms, algorithms on individual objects, is made which concludes with a recommendation for the AGENT project, Chapter 4.

After an extensive bibliography, some of the algorithms are described in the Appendix. Note that the Appendix assumes a significant familiarisation with domain knowledge in generalisation — the descriptions are meant for the partners to brush up their know-how or to communicate the idea of less known algorithms.

1.3 Terms

The term “algorithm” is universally used in computer science to describe problem-solving methods suitable for implementation as computer programs (Sedgewick, 1984). Throughout this report we will take this definition literally: *A generalisation algorithm is a formal mathematical construct that solves a generalisation problem by changing an object’s geometry or attribute* (transformation). Contrary, a measure is a method that does not change the state of map objects, but is used to characterise it.

Using this definition, the same piece of implemented code can be used as an algorithm or as a measure. For example, the mathematical construct, the point of gravity, can be used either as a measure, when describing the position of an object or as an algorithm, when collapsing an area object to a point. On the boundary between measures and algorithms are routines for segmentation; they not only characterise objects but also split (transform) their geometry.

"Auxiliary data structure" is a complex data structure that provides assistance when performing assessments (measures) or transformations (generalisation algorithms).

Refer also to the A2 report, where the relations between measures (assessment tools) and generalisation algorithms (transformation tools) are described and the role of constraints in defining measures and generalisation algorithms is given. For completion, note also the definitions used in the C1 report. Throughout this report, the term "algorithms" always implies generalisation algorithms.

The concept of "operators" arises from manual cartographic generalisation. Operators are an abstract identification of the procedures performed when generalising maps manually. Algorithms, on the other hand, can be viewed as the computer-based implementation of a generalisation step. "Algorithm classification" classifies algorithms based on their mathematical approach and does not take generalisation concerns into account. "Operator decomposition" focuses on the procedures needed during generalisation.

2 Typology of operators

The following typology of operators respects ideas developed by McMaster and Shea (1991), Hake and Grünreich (1994), Peng and Tempfli (1996) and the COGIT Lab (Ruas, 1995).

As seen in the literature there are dozens of ways to subdivide the generalisation process into disjoint operators. Depending on the point of view, some classifications are more suitable than others. Even though not all of the partner's positions could be fully integrated (as they occasionally contradicted), the following classification should be acceptable to everybody. Because the aim of this report is to help with algorithm selection and development, effort should not be spent herein debating terminology.

2.1 Typology

Operator hierarchy

As shown in Figure 1, a hierarchical decomposition of generalisation operators is presented. The two relevant levels are “Traditional Operators” and “Digital Operators”.

- Traditional Operators: Traditional operators describe procedures usually performed in manual generalisation. They describe procedures identified as similar by conventional cartographers.
- Digital Operators: Because most traditional operators can not be easily transformed one-to-one to algorithms, researchers in the field of digital cartography provide additional classes to group operators by similar functionality. This computation oriented classification might be useful to detect holes in the algorithm construction, but is practically too detailed to discuss basic cartographic concepts.

A description of algorithms and their classes is postponed to Chapter 3, where the operator containers are filled.

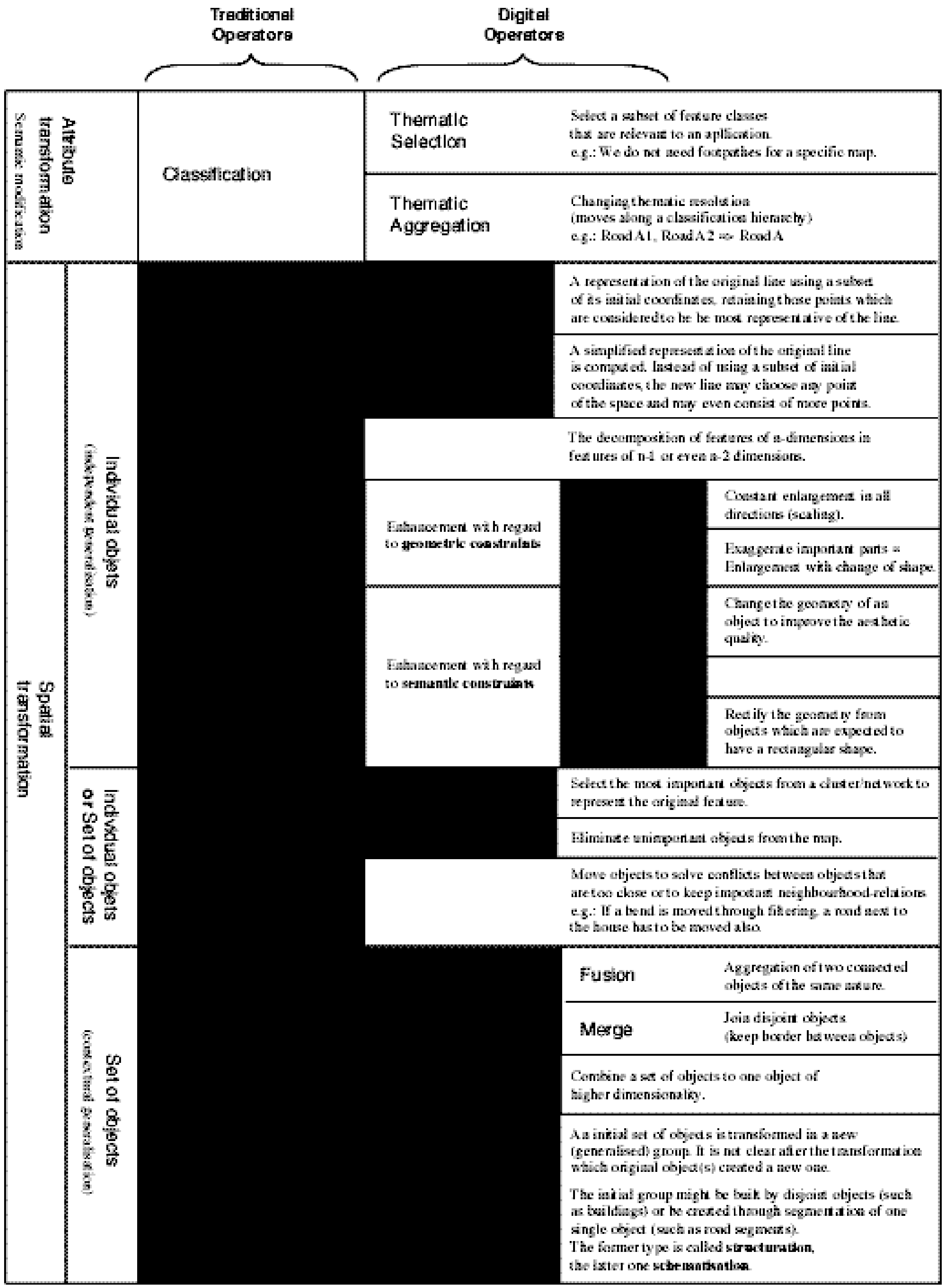

Figure 1 : Typology of operators

2.2 Attribute transformation operators

The operations for the transformation of attributes are not relevant to this report. Attribute transformations perform operations on the schema, which do not change geometry directly. However, they are still listed here because there is still the possibility of employing attribute transformations when generalising a map.

2.2.1 Classification

Thematic Selection

The process of thematic selection extracts application-relevant feature classes for a specific map-purpose, such as choosing roads for a highway map.

Thematic Aggregation

This operator changes the thematic resolution of a map. It can be applied to both classes and attributes. Applied on classes using Gothic, this process is mainly carried out by aggregating classes to their common parent-class.

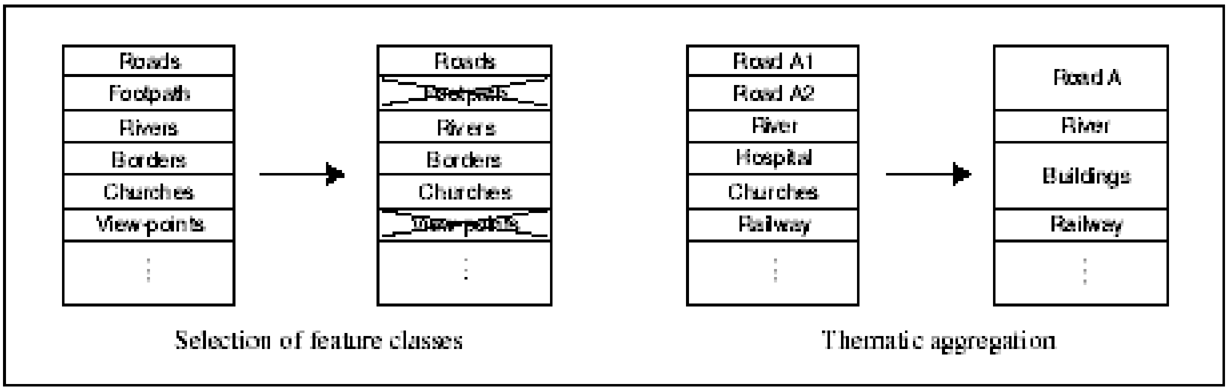

Figure 2: Classification

While a distinction is drawn only between thematic selection and thematic aggregation in this report, other authors identified additional attribute transformations, such as reclassification and thematic simplification.

Contrary to thematic aggregation, which moves along a hierarchy, Peng and Tempfli (1996) describe reclassification as the creation of new feature classes by changing themes of existing classes. While thematic aggregation moves along a hierarchy to build new feature classes, reclassification does not. Thematic simplification reduces the number of attributes of a class by taking out some attributes, leaving the theme and resolution unchanged. Although these classes are important for semantic integration of data into a GIS, they are not relevant to solve the generalisation tasks outlined by the AGENT project.

Symbolisation is an important step in the generalisation process, however (unlike McMaster and Shea, 1991) we do not consider symbolisation as an operator. The symbolisation of a map is defined by the user at the beginning of the generalisation process, herewith introducing additional constraints into the system, and is therefore a map control (as defined in the A2 report) and not an operator.

We are aware that this is also true for classification, as the represented classes are mainly predefined and do not underlie a change to solve specific conflicts. Nevertheless we list classification here, because the classification process must be coded. This means that for an abstraction of the database, a program must explicitly ‘do’ something. This is not true for symbolisation.

2.3 Spatial transformation operators

We distinguish between three classes of operators. The first group of operators changes only the geometry of **individual objects**, while the last group of operators transforms a **group of objects**. The middle class contains displacement and selection/elimination, which connects the previous classes. In this report, an object implies a simple geometric primitive.

Before describing the operators in greater detail, we want to clarify that even if the first group of operators is applied on individual objects, such algorithms may work in a context-dependent manner. The most powerful algorithms take care of their environment even if they only process an individual object.

2.3.1 Simplification

Simplification eliminates detail. In the literature, simplification is often termed “filtering”, to express that a simplified line consists of a subset of the original co-ordinates. However, this restriction is too narrow as there are also algorithms that represent the simplified line, calculating points not used in the original line. These algorithms relocate or shift co-ordinates to eliminate detail and are therefore presented as *simplification*-algorithms and not as *smoothing* algorithms.

Weeding

Representation of the simplified line using a subset of the original co-ordinates. Algorithms either select the shape-describing points or reject points considered to be unnecessary to display a line’s character.

If used with “hard” parameters, these algorithms can also be used to eliminate redundant points (like alined points) in order to retain a lower number of data points without loss of accuracy.

Unrestricted Simplification

Unrestricted Simplification algorithms compute a simplified line by reallocating points. Even if most of these algorithms may also be used as smoothing algorithms, the principle aim of such algorithms lies not in an attempt to capture only the trend of a line, but in the elimination of unwanted detail while retaining significant aspects of the line character (e.g. Li and Openshaw (1992)).

2.3.2 Collapse

Collapse describes the reduction of line or area features to point features, or area features to line features. While CUSP (Change under Scale Progression) describes any change of dimensionality, collapse focuses on the reduction of dimensionality: features of dimension n are reduced to dimension n-1 or even (if possible) to n-2.

We do not list all CUSP-operators here, because operators that increase the dimensionality always act on a group of objects. Furthermore we think that the change of dimensionality is not an operator *per se*. The collapse operator however does group actions that are similar in nature.

For the collapse operator, it is more difficult to define the thing being collapsed, rather than the result. The collapsing operation becomes much easier if the original objects are structured to make the operation involve less decision making. Therefore things like estuaries and cloverleaves should be identified and structured beforehand.

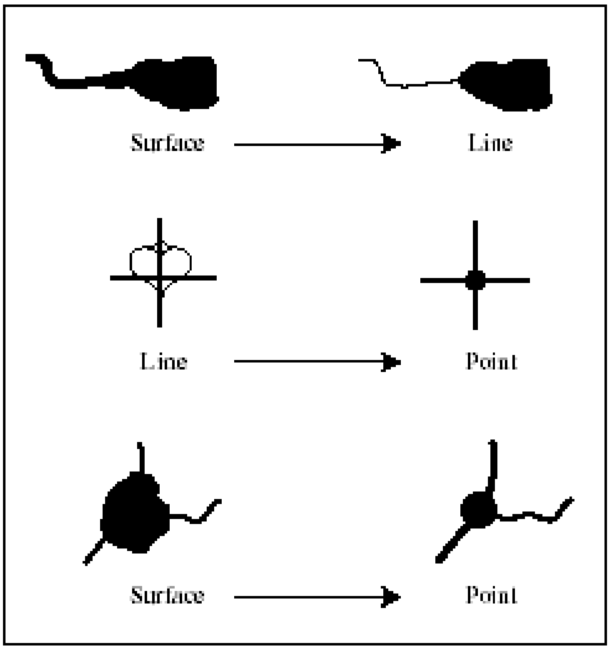

Figure 3: Collapse

2.3.3 Enhancement

Enhancement is the operation where an object - or parts of it - are enhanced to meet geometric or semantic constraints.

We differentiate enlargement and exaggeration, which are both operations used to satisfy **geometric requirements** from smoothing, fractalization and rectification, which are used to **magnify the semantics** or meaning of an object.

While this distinction does help understanding the classification, it is however slightly vague, as the exaggeration and enlargement operators are not only called with regard to geometric constraints. Sometimes the object semantics contribute to the decision to exaggerate or enlarge.

Enlargement

This operator is used to enlarge objects equally in each direction. The result is an object with the same shape, but scaled by a magnitude. Enlargement is mainly applied to objects to reach a minimum size, but can also be used to preserve differences in size between several objects.

Exaggeration

Exaggeration is used to enlarge parts of objects, either because they do not satisfy the geometric constraints or because such parts are of special interest.

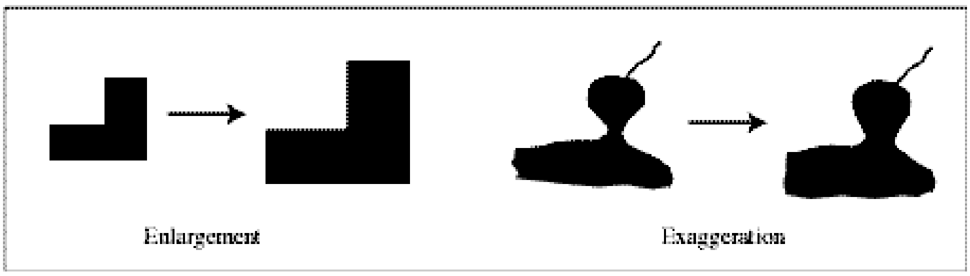

Figure 4: Enhancement with regard to geometric constraints

There is a lot of confusion about the terms “enlargement”, “exaggeration” and “caricature”. While “enlargement” is generally accepted as a simple scaling, the terms “exaggeration” and “caricature” are used differently by different partners. While some researchers use exaggeration synonymously for enlargement, other researchers use exaggeration like caricature. Even though we are aware that exaggeration is somehow weaker then caricature, we have merged them into one group. We believe this is reasonable because:

- The term “exaggeration” (as understood in natural, not generalisation influenced English) does not mean enlargement. Therefore it confuses native English speakers when treating these words similarly.
- The operations of exaggeration and caricature are based on the same principle (exaggeration of important parts). Therefore they are combined for this typology.
- “Caricature”, when following definitions in dictionaries, is rather an aspect of rendering that is involved in many operators, including simplification, smoothing, enhancement and typification that causes distortions of shape to a small or large degree. The term “caricature” might therefore be misunderstood when used as a single group.

Smoothing, Rectification and Fractalization

These operators are applied on objects to enhance the object shape according to their semantics. Important parts, or the whole object itself, are therefore enhanced to support the underlying, natural character thus helping the reader to recognise an object’s type.

- **Smoothing**: Reduces sharp angularity from objects having smooth shapes.
- **Rectification**: Rectifies the geometry from objects, which are expected to have a rectangular shape.
- **Fractalization**: Add self-similar details to objects that might be expected to possess them.

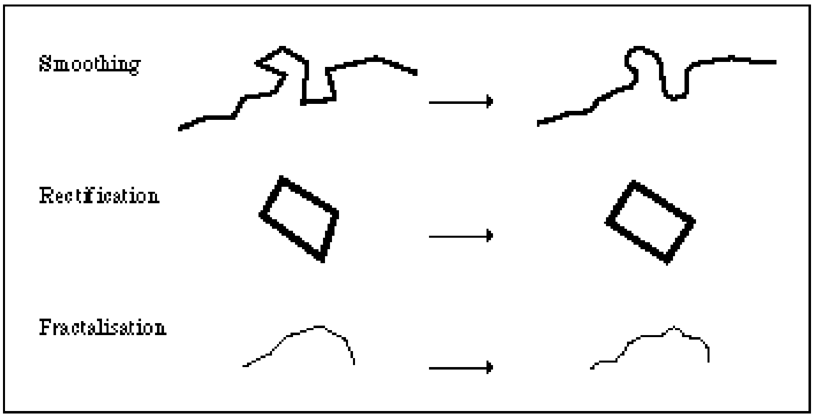

Figure 5: Enhancement with regard to semantic constraints

2.3.4 Selection / Elimination

Selection / Elimination is the process of reducing the number of objects within a class. Elimination is hereby just the antonym of object selection. Nevertheless we separate these two operators, as the underlying concepts of algorithms for selection and elimination are quite different (when not applied on disjoint objects), even if they have the same target.

2.3.5 Displacement

This operator displaces objects to meet map requirements. This operation might be important to solve conflicts between objects that are too close or to preserve important neighbourhood relations.

2.3.6 Aggregation

Aggregation is the process of joining features, or more generally: *to represent a group of objects with another representation*.

We distinguish the aggregation of several objects into one object and the aggregation of several objects into a new group of objects. In other words, the first group builds a single resulting geometry, while the second operator creates a group of primitives.

The first group is subdivided into amalgamation and combining; the second group is called typification.

Amalgamation

A group of objects is amalgamated into one geometry without change in dimensionality. We can split this operator in 2 sub-operators:

- **Fusion**: Aggregation of two connected objects of the same nature (class). The operator only needs to dissolve the intermediate border.
- **Merge**: Disjoint objects of different classes are aggregated/ merged to one 'block'.

The classical amalgamation is a combination of these two basic operations (merging and fusion).

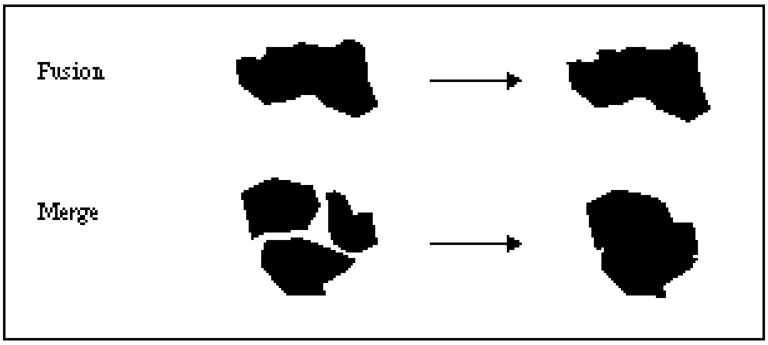

Figure 6: Amalgamation

Combine

A group of objects of the same class are combined to one object with higher dimensionality.

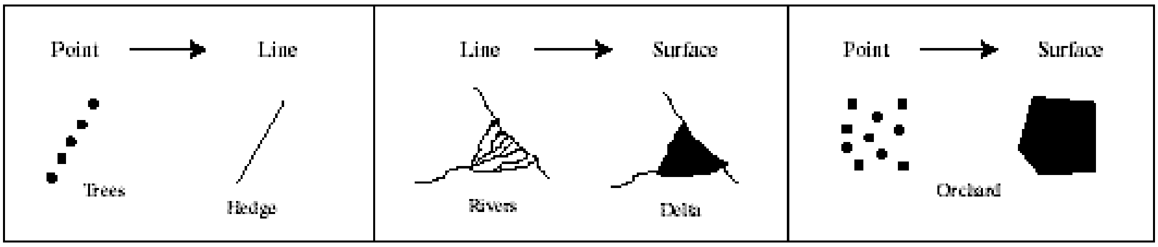

Figure 7: Combine

Typification

A group of objects is represented by a new, reduced set of objects. The new group has to show similar characteristics concerning density, orientation and so on. While comparing the original with the derived map, it does not have to be obvious, which objects turned into the new ones.

The structuring process is often the combination of several basic algorithms (selection, aggregation, simplification, etc.) into one algorithm. Nevertheless it is important to have these combined algorithms, as they allow imitating more complex and holistic transformations.

Typification can be processed on a group of isolated objects as well as on a group of object-parts. Split objects usually emerge through segmentation of liner features; for typification the parts of an object are usually treated like individual objects even though they need additional handling due to requirements such as second order continuity at connections.

Two types of typification are specially named by the IGN: the typification of buildings is called *sic* structuration (structuring) (a), the reduction of bends in line generalisation is called schematisation (b).

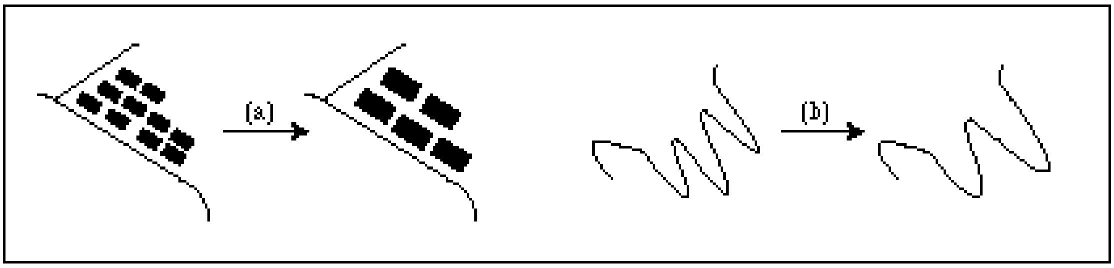

Figure 8: Typification

3 Overview of algorithms

3.1 List of Algorithms sorted by operator

	Independant point algorithms	line	Random choice of points n-th point
	Local processing algorithms	line	McMaster's local processing routines (McMaster, 1987) Walking algorithm (Müller, 1987) Reumann-Witkam (Reumann and Witkam, 1974) Lang (Lang, 1969) Jenks (Jenks, 1989) Sinuosity-guided Point-Selection (Dutton, 1998) Visvalingam & Whyatt (Visvalingam and Whyatt, 1993)
		urban blocks	Visvalingam & Whyatt (Visvalingam and Whyatt, 1993)
		DTM	ATM filtering (Heller, 1990)
	Global processing algorithms	line	Cromley & Campbell (Cromley and Campbell, 1992) Douglas & Peucker (Douglas and Peucker, 1973) deBerg et al. (deBerg et al., 1995)
		line	VanHorn-grid (Van Horn, 1986) Li & Openshaw (Li and Openshaw, 1992) QTM-based line simplification (Dutton, 1998) Whirlpool (Dougenik, 1980) Line gen. based on high level characteristic (Wang and Müller, 1998)
		coastlines	Complex coastline generalization (Wang and Müller, 1993)
		area / line	Epsilon-generalization (Perkal, 1966)
		building	Same-sized best-oriented rectangle (Hangouët, 1996) SIMP-BATI (Ruas, 1988, and Damour, 1994) Manmade (LSL, 1998)
		road junctions	Simplification of road junction (Mackaness and Mackechnie, 1997)
		DTM	Heuristic generalisation of DTM's (Weibel, 1992)
	area to line		LSL-collapse (LSL, 1998) Collapsing using the skeleton
	area to point		LSL-typification (LSL, 1998) Center of gravity
	line to point		
		area	LSL-area-enlargement (LSL, 1998) Enlargement/Shrinking using the skeleton (Chithambaram et al., 1991)
		buildings	Enlargement using the sum of normals (Bundy et al., 1995)
		line	Accordion (Plazanet, 1996)
		line	Fourier line enhancement (Clarke et al., 1993) Smoothing using spectral filtering (Fritsch and Lagrange, 1995) LSL-exaggeration (LSL, 1998) Brophy-Dutton-exaggeration (Brophy, 1973)(Dutton, 1981)
		roads	Lowe-Barillot (Lowe, 1988)(Barillot, 1996) Galbe (Mustière, 1998) Plaster (Fritsch, 1997) Mechansitic approach (Fritsch, 1997)
		bends	Balloon (Lecordix, Plazanet, 1996) Maximal Break (Mustière, 1998) Minimal Break (Mustière, 1998)

Point averaging routines	line	Weighted-moving-average (McMaster, 1989) Sliding-moving-average (McMaster, 1989) Distance-weighted-average (McMaster, 1989) Gaussian Smoothing algorithm (Badaud, 1986)
	line / polygon	Brophy smoothing (Brophy, 1973)
Tolerancing routines	DTM	Global filtering (Weibel, 1992)
	line / polygon (bathymetric)	Epsilon-generalisation (Perkal, 1966)
Mathematical curve fitting routines	line	Spline (McMaster, 1989) Bezier (McMaster, 1989) Chaikin's smoothing algorithm (Chaikin, 1974) AKIMA DTM reconstruction using Fourier transformation (Gallant, 1996)
	line	Carpenter (Carpenter, 1981) Dutton (Dutton, 1981)
	buildings	Global approach (energy minimization) for rectification (Airault, 1996)
	all objects	Radical law (Töpfer, 1974)
	rivers	Horton-order (Horton, 1945)
	roads	Road-selection constrained by quickest path and attractive points (Reyner, 1997)
	polygons	Elimination of polygons by using its skeleton (Bader, 1997)
Anamorphosis	all objects	Focus-line displacement (Michel, 1997)
	clusters of points	Propotional Radial Enlargement (Mackaness and Fisher, 1987)
	polygons	Displacement based on push-forces (Bader, 1997)
	point-point line-point	Müller-displacement (Müller, 1990)
Displacement	all objects	Lichtner-displacement (Lichtner, 1979) Nickerson (Nickerson, 1988) LSL-Displacement (LSL, 1998)
	line vs. line	Cartographic Displacement using the snakes concept (Burghardt and Meier, 1997)
	buildings vs. roads	Displacement using the Finite Element Method (Hojholt, 1998) Displacement using a local triangulation (Ruas, 1995)
	area	LSL-merging (LSL, 1998) Merging using the SDS (Jones et al., 1995)
	buildings	Agreg-disp (Regnauld, 1997)
points to area		Delaunay trinagulation for point aggregation (DeLucia and Black, 1987) Graph-theoret. methods for detecting and describing gestalt clusters (Zahn, 1971) LSL-aggregation with Convex-hull and inner rings (LSL, 1998) Shrink Wrapped Hull (LSL, 1998)
points to line		-
lines to area		-
	buildings	Interpolation-based typification (Hangouët, 1996) Lichtner-structuration (Lichtner, 1979) Regnauld-structuration (Regnauld, 1997)
	areas (lakes)	Area-patch generalisation (Müller and Wang, 1992)
	bends	IGN-schematisation (Lecordix and Plazanet, 1996)

Figure 9 : Algorithms sorted by operator and object classes. Algorithms that can be applied to linear features can also sometimes be applied to the border of polygons.

3.2 Comparison of algorithms

In this chapter we intend to classify algorithms and to compare different approaches to the same task. The conclusions of this evaluation are found in Chapter 4 in a condensed form. Note that this evaluation is restricted to algorithms on individual objects (D1), even though the list of algorithms does not stop at D1. As there are very few algorithms for collapsing objects, the comparison focuses primarily on simplification and enhancement.

3.2.1 Algorithms for Simplification

Most algorithms in generalisation deal with simplification, where again the major part is restricted to line simplification. It is thus not surprising that there are various classifications of simplification algorithms. We adopt a classification presented by McMaster (1987) to group these algorithms.

- Independent point algorithms: Such algorithms do not account for the relationships with the neighbouring co-ordinate pairs and operate independently of topology. Examples are the n-th point algorithm or the random selection of points. It is obvious that such procedures cannot satisfy cartographic requirements.
- Local processing algorithms: These utilise the characteristics of the neighbouring points to judge the importance of a point. Such a characterisation can be restricted to the immediate neighbours (e.g. Jenks (1989), or McMaster's local processing algorithms (McMaster, 1987)), or go beyond the immediate neighbours and evaluate short sections of lines (e.g. Lang, 1969, or Reumann-Witkam, 1974).
- Global routines: Consider the entire line, or specified line segment, during processing. The Douglas-Peucker algorithm (Douglas and Peucker, 1973) is the famous representative of this class.

Instead of judging the advantages of these categories one should rather question their applicability to generalisation in general. Here we address the topic of algorithm intention and object characterisation.

Since the mid-1960s techniques for weeding polylines have been developed. McMaster and Shea (1992) provide four justifications for the use of simplification algorithms in information theory in general:

- Reduced plotting time
- Reduced storage space
- Faster vector to raster conversion
- Faster vector processing

Note that none of these purposes are pursued in cartographic generalisation. Nevertheless, as seen above, these methods were translated one to one into the cartographic generalisation domain. It is then no surprise that these methods cannot fulfill all the demands we state for a good simplification algorithm.

With becoming more important, the field of cartographic generalisation attracted researchers to find algorithms which satisfy the purposes of generalisation more specifically. This change of view resulted in two different kinds of approaches.

On one side, the weeding was further constrained in cartographic terms. DeBerg et al. (1995) introduced topological constraints to their weeding method, Cromley and Campbell (1992) improved weeding algorithms in a way that they optimise a geometrical characteristic which is thought to be relevant for generalisation.

On the other hand, algorithms explicitly designed for cartographic generalisation have no need to stay restricted to a simple weeding of point strings. Unrestricted simplification algorithms allow the geometry of objects to change entirely as long as they represent the original object adequately. Li and Openshaw (1992) describe an algorithm that is oriented on generalisation principles like viewing things from increasing distances.

Most of these line simplification algorithms still suffer from their lack of specialisation. In generalisation a line is not simply a line but the representation of a road, a coastline or the outline of a building. These lines have to be treated in a different way, as they have different underlying character. Wang and Muller (1993) present an algorithm that does simplify a coastline explicitly, by taking the semantics of the mouths of rivers into account. Modern software packages such as the Generalizer by LaserScan respect the semantics of objects by providing algorithms for specific features. LaserScan provides an algorithm called "manmade" that respects the squared character of human buildings. Extended work on building simplification was also performed in Hannover by Lichtner (1979), Meyer (1986) and Staufenbiel (1973)

On objects more complex than buildings (e.g. roads), a simplification can not lead to a good generalisation anymore, as the huge amount of semantic information is not processed. ESRI tried to solve this problem by integrating a routine called BENDSIMPLIFY (see Wang and Muller, 1998) which simplifies roads by characterising each bend. Yet a good generalisation is not possible using simplifying alone. The extensive research by IGN resulted in the production of far more complex algorithms to generalise roads, acting as enhancement operators on details like isolated bends. When accepting that generalisation does not merely mean simplification, we hope that further research is directed towards other domains than simplification.

Another field of algorithms is opened by the simplification of non-linear objects. Whether simplifying a network of junctions (Mackaness and Mackechnie, 1997) or simplifying a digital terrain model (Heller, 1990), the approaches are so different and there are so few algorithms in the respective domain, that there is no use yet for a classification.

3.2.2 Collapsing algorithms

While the collapsing of a polygon to a point is trivial (the point location needs only to be defined), collapsing a polygon to a line is more complex. This operation is important mainly for broad rivers because they are usually digitised by their bank lines, but need to be reduced to a centreline under scale reduction. The task is therefore well specified, even though the approaches for constructing a skeleton-like structure differ.

3.2.3 Enhancement algorithms

Realising that line simplification algorithms cannot generalise complex objects in an adequate way, the combined use of different enhancement methods becomes necessary. Most efforts in this domain were investigated by the IGN who developed an entire array of enhancement methods for roads in order to replace line weeding algorithms.

Exaggeration

While constant enlargement in all directions, or scaling, is a purely mathematical operation, the exaggeration operation, also called caricature, is more complex. The IGN developed a series of caricature algorithms— Balloon, Maximal Break, Minimal Break, and Accordion, which all deal with very specific problems. These algorithms work locally on one road bend pursuing a strict aim. It is therefore possible for an algorithm to fulfill its task with high reliability under the assumption that the problem with the line is precisely detected. When adopting such specialised generalisation algorithms, it is therefore fundamental to have measures available that are strong and robust in characterising and delimiting important phenomena. Hence the quality of the solution is dependent on both the transforming algorithm and the segmentation routines.

Another set of caricature algorithms approaches the problem from a frequency domain based perspective (e.g. Clarke et al., 1993, Fritsch and Lagrange, 1995). With this perspective, detecting important structures in the frequency domain becomes easier. Most of these approaches do not go beyond an experimental stage however. Especially, the specification of control parameters is not solved satisfactorily.

Smoothing

The smoothing operator, as defined in this report, is understood as a cosmetic operation (see 2.4.3). In our project we will use this operation for two purposes: in its classical use, smoothing is used to improve the aesthetic appearance of overly angular lines. Additionally, we use smoothing to remove small crenulations, which appear as noise in the map.

McMaster (1989) as well as Lewis (1990) presented a three part classification of smoothing methods, both following similar ideas.

Point averaging routines compute for each point of a line a new position based on their neighbouring points. Such averaging methods differ in the number of neighbours taken into consideration as well as the different weighting function used to adjust the importance of a neighbouring point.

The second important class consists of mathematical curve fitting routines. These routines try to approximate a sequence of points, or even the whole line, with a mathematical function with continuous character (e.g. splines, Bézier curves).

The third class is built by tolerance routines, which use a user-defined tolerance as an accuracy threshold. A trend is then computed, by adding successively points or rolling a ball along a line for instance, and all details with a deviation smaller than the magnitude of the tolerance distance are ignored. This type of smoothing is normally applied frequently to simplify a geometry and is therefore placed in our unrestricted simplification category.

The advantage of point averaging routines over mathematical curve-fitting routines lies in their ease of use and their predictability. Even with complex weighting functions (such as the Gaussian smoothing used by Badaud, 1986) the algorithms are easy to comprehend and their implementation and data organisation usually proceed without problems if appropriately constrained. Over-oscillation can be avoided which is only partly true for mathematical approximations. While the quality of trivial averaging functions, such as simple moving averages, can be poor, the appearance of a Gaussian smoothed line is generally adequate. The problems with using weighting functions often arise due to the lack of finding appropriate parameters to guide the method.

An advantage of curve fitting techniques, however, is that the shape of a curve can be easily deformed using few vertices. Thus, mathematical curve fitting techniques are predominately useful as representations to model the geometry of smooth linear objects, such as roads (see Affholder, 1993, and Plazanet et al., 1995).

Fractalization

Dutton (1981) introduced the fractal dimension in cartographic generalisation. Its use for generalisation is limited to objects with fractal characteristics, mainly rivers and coastlines. With introducing fractality, readibility is won while loosing accuracy. Dutton’s method is straightforward to use and very few other methods are available to perform fractalization.

3.3 Further relevant concepts

Generalisation algorithms are supported and extended by various techniques in order to enhance their operation. Such extensions mainly relate to auxiliary data structures, specialisations of generalisation algorithms for particular applications, and segmentation methods. Many of these techniques (e.g., auxiliary data structures and segmentation techniques) are used to capture a contextual notion in generalisation algorithms. They are associated with algorithms for contextual generalisation rather than independent generalisation. As a consequence, we restrict the description in this section to approaches that assist or improve algorithms for independent generalisation (task D1). A more detailed discussion of auxiliary techniques will be presented in the report for task D2 (contextual generalisation).

Auxiliary data structures

Some basic algorithms on individual objects profit from advanced data structures to keep track of topological relations.

Data structure	References	Use	Remarks
Triangulation	Ruas (1995) Bundy et al. (1995) Jones et al. (1995) Bader (1997)	• Explicit expression of the neighbour relations; • *Ad hoc* computation of neighbourhood relations. • Basis for Finite Element methods.	Depending on the purpose of the triangulation, all vertices of an object are triangulated (Bundy, Jones, Bader) or only the centroids (Ruas).
Skeleton	Chithambaram et al. (1991), Brassel et al. (1984), Lee (1982), Jones et al. (1995). DeLucia and Black (1987)	• Area to line collapsing; • Orientation of objects (main axis).	Various mathematical solutions possible (e.g. bisector vs. bimedial skeleton, TIN-based skeleton).
Convex-Hull	e.g. O'Rourke (1994)	• Combine points to an area (DeLucia and Black, 1987); • Mathematically correct minimum bounding rectangles (MBR); • Computation with polygons.	Algorithms well-known, fast and easy to handle.

Concepts to improve algorithms

When some algorithms are applied to special cases, weaknesses become evident. To overcome these problems, researchers extend the algorithm abilities. The Douglas-Peucker algorithm especially was extended to preserve areas of polygons and topological relations.

Algorithm	References	Use	Remarks
Preserve area	Williams (1987)	This algorithm can be used in conjunction with line-simplification-algorithms. It resizes a polygon after simplification to its initial size.	Algorithm can also be used for enlargement.
Side-constrained-simplification	Zhang and Tian (1997)	This algorithm ensures that segments are only moved to one side from the simplified line.	
Topologically consistent line simplification (no self-intersections)	Saalfeld (1998)	This algorithm insures a topologically correct line after the use of the Douglas-Peucker-algorithm.	See also de Berg et al. (1995).

Segmentation methods

Such functionality is needed to split up complex objects into homogeneous parts. This segmentation is a prerequisite for many locally applied algorithms. The report of task C1 (measures for agents, Deliverable D C1) explains the necessity for segmentation in more detail. See also the specifications of task C1 (Deliverable D C2) for details of segmentation functions for roads.

Algorithm	Reference	Description
Inflection points / Segmentation	Plazanet et al. (1995), Plazanet (1995) Dutton (1998)	Inflection points are detected using a Gaussian smoothing of the angular changes at each vertex. A line is split at inflection points and the resulting segments are analysed to detect homogeneous parts.
Coalescence detection	Mustière (1998)	Mustière's algorithm detects problems that arise due to the increased symbol width of lines. The line (usually a road) is segmented into parts where the symbol overlaps (in bends) and parts without coalescence.

4 Conclusions and Recommendations

The following list summarizes the recommendations that can be made for the choice of algorithms to be implemented in the prototype. It builds on the ideas and concepts discussed in section 3 as well as on the comprehensive survey of algorithms for independent generalisation presented in the Appendix.

Recommendation	Consequences for AGENT project
If simplification algorithms are used, they have to respect the feature class they are applied upon. As such, the building simplification algorithm must be different from a river simplification algorithm.	Integration of an improved building simplification algorithm. Use method by LaserScan (1998, manmade) and Ruas and Damour (1995) to start the research. Simplification algorithms are used when a more detailed analysis of a feature class is not yet available.
Simplification algorithms (weeding and unrestricted simplification) do not allow a good generalisation of complex features such as roads. The generalisation of such objects must be done by characterising parts locally. However, unspecific weeding is adequate for minor generalisation. A weeding method is needed to eliminate (nearly) redundant points. When we use such a weeding method, we normally use it for the ease of further computation steps and for the reduction of data, but not for cartographic generalisation.	Integrate Douglas-Peucker (1973). Check Visvalingam and Whyatt (1993).
The use of enhancement methods to fulfill generalisation prompts the need for various, locally effective and highly specialised methods.	Integrate IGN's caricature algorithms for roads: Accordion (Plazanet, 1996), Maximal Break (Mustière, 1998), Minimal Break (IGN, 1998), Plaster (Fritsch, 1997). Research on specialised enhancement methods for buildings, such as the enlargement of narrow parts (UNI-ZH).
To control and orchestrate these enhancement methods we depend on expressive measures to identify which algorithm to invoke where and when. To apply locally effective algorithms on lines (roads), the lines have to be segmented first into homogeneous sections with respect to one characteristic.	Coalescence detection (Mustière, 1998), Noise detection Inflection points (see Plazanet and Dutton)
Aesthetic smoothing is used for eliminating noise and to support an object's character. Such a method has to be found for each feature class with different appearance.	Gaussian smoothing (Badaud, 1986) for lines. Improved squaring methods for buildings (start from Airault, 1996).
Due to the automatic approach chosen, some standard GIS functionality has to be integrated in the generalisation system, as automation can lead to errors, prompting for corrections that are not necessary in manual cartography.	Scale, Rotation, Displacement (translation) by an offset.
Functionality that is used as basis for other algorithms should be made available individually, instead of integrating them into the method.	Skeletonisation, Delaunay Triangulation, Convex Hull.

	These methods can be extracted from standard GIS and Computational Geometry books.

The assessment and recommendations made in this report were mainly based on evaluations and assertions conveyed in the literature, partly based on empirical evidence by previous research by AGENT partners, and to a good deal dictated by the pragmatics of the AGENT project, such as the choice of feature classes which in turn defined the operators that needed to be available. It is clear that the existence of algorithms itself is not enough to make sure that a generalisation operator is reliably translated into an algorithm. It is one of the next steps to empirically assess the actual performance of these algorithms, once the algorithms defined in the specification document (D D1) are available in the E2 prototype.

Further empirical work can be conducted in parallel to the ongoing efforts on task E2 and D2 (algorithms for contextual generalisation). Empirical assessment and detailed specification of the conditions of use (i.e., procedural knowledge acquisition) will primarily focus on the algorithms that have been implemented in the first version of the E2 prototype so far. When further algorithms are added, the overall goal of AGENT to develop a fully automatic system must be kept in mind. The chance for an automatic selection of appropriate generalisation algorithms and determination of adequate values of control parameters increases if

- the algorithm has only few control parameters;
- each parameter is independent (orthogonal) in its effects from the other parameters;
- the meaning and function of each parameter can be conceptually understood by the user (i.e., cause and consequence are clear);
- and the parameters can be directly linked to cartographic constraints (e.g., symbol size, maximum displacement) and associated with measures that assess the potential violation of constraints.

These criteria are particularly important for algorithms for contextual generalisation (task C2), as they tend to be more complex than those for independent generalisation. Unfortunately the above criteria cannot always be met, particularly the second one. However, they can be optimized. And they can also be used as a guideline for the development and selection of generalisation algorithms in future tasks: Algorithms that violate more than one of the above criteria are bound not to be useful for our purposes.

5 References

Airault, S. (1996): De la base données à la carte: une approche globale pour l'équarrissage de bâtiments. *Revue internationale de géomatique*, Vol.6, Num.2-3, 203-217.

Badaud, et al (1986): Uniqueness of the Gaussian Kernel for scale-space filtering. *IEEE Transactions on pattern analysis and image processing*, 8 (1), 183-193.

Bader, M. and R. Weibel (1997): Detecting and Resolving Proximity Conflicts in the Generalization of Polygonal Maps. *Proceedings 18th International Cartographic Conference*, Stockholm, 1525-1532.

Barillot, X. (1996): *Correction de Lowe*. Rapport interne IGN.

Beard, M.K. (1991): *Theory of the cartographic line revisited / Implications for automated generalization*. Unpublished.

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

Bern, M. and D. Epstein (1992): Mesh generation and optimal triangulation. In: *Computing in Euclidean geometry*, D.-Z. Du and F. Hwang (eds.). World Scientific, Lecture Notes Series on Computing, 1, 23-90.

Brassel, K., M. Heller and P.L. Jones (1984): The construction of bisector skeletons for polygonal networks. *Proceedings of the 4th International Symposium on Spatial Data Handling*, 1, Zürich, 117-128.

Brophy, D.N. (1973): An Automated Methodology for Linear Generalization in Thematic Cartography. *Proceeding of the ACSM Conference*, 33rd Annual Meeting, Washington D.C., 300-314.

Bandy, G.L., C.B. Jones and E. Furse (1995): Holistic approach of large-scale cartographic data. In: Muller, J-C., Lagrange, J.P. and Weibel, R. (eds.). *GIS and Generalization*, GISDATA 1, London, Taylor & Francis, 106-119.

Burghardt, D. (1997): Cartographic Displacement using the Snakes Concept. In: Foerstner, W. and L. Pluemer (eds.): *Semantic Modeling for the acquisition of Topografic {sic} Information from Images and Maps*, Birkhäuser-Verlag, Basel.

Buttenfield, B. (1985): Treatment of the cartographic line. *Cartographica*, 22 (2), 1-26.

Buttenfield, B. (1989): Scale dependence and self-similarity in cartographic lines. *Cartographica*, 26 (1), 79-99.

Buttenfield, B. (1991): A rule for describing line feature geometry. In: Buttenfield, B.P. und McMaster, R.B. (eds.). *Map Generalization: Making Rules for Knowledge Representation*. London, Longman, 150-171.

Carpenter, L.C. (1981): Computer rendering of fractal curves and surfaces. *Proceeding of ACM SIGGRAPH Conference*, Seattle, Washington.

Chaiken, G.M. (1974): An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3 (1), 346-349.

Chithambaram, R., K. Beard and R. Barrera (1991): Skeletonizing polygons for map generalization. *Cartography and GIS/LIS*, ACSM-ASPRS, 2, Baltimore, 61-72.

Clarke, K.C., P. Cippoletti and G. Olsen (1993): Empirical comparison of two line enhancement methods. *Proceedings of AUTO-CARTO 11*.

Cromley, R. (1991): Hierarchical methods of line simplification. *Cartography and Geographic Information Systems*, 18 (2), 125-131.

Cromley, R.G. (1992): *Digital Cartography*. Prentice Hall, Engelwood Cliffs (New Jersey), 317 p.

Cromley, R.G. and G.M. Campbell (1990): A geometrically efficient bandwidth line simplification algorithm. *Proceedings of the 4th International Symposium on Spatial Data Handling*, Vol.1. Zurich, Switzerland, 77-84.

Cromley, R.G. and G.M. Campbell (1992): Integrating quantitative and qualitative aspects of digital line simplification. *The Cartographic Journal*, 29 (1), 25-30.

Damour, S. (1994): *Généralisation du bâti: la simplification des contours*. DEA-SIG, ENSG:

de Berg, M., M. van Kreveld and S. Schirra (1995): A new approach to subdivision simplification. *Proceedings of AUTO-CARTO 12*, 4, 79-88.

DeLucia, A.A. and R.T. Black (1987): A Comprehensive Approach to Automatic Feature Generalization. *Proceedings of the International Cartographic Conference*, Morelia, Mexico, 169-192.

Dougenik, J. (1980): Whirlpool: A geometric processor for polygon coverage data. *Proceedings of AUTO-CARTO 4*, 304-311.

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 character. *The Canadian Cartographer*, 10 (2), 112-123.

Dutton, G. (1998): *Scale, Sinuosity and Point Selection in digital generalization*. to be published.

Dutton, G.H. (1981): Fractal enhancement of cartographic line detail. *The American Cartographer*, 8 (1), 23-40.

Dutton, G.H. (1998): *Important compared to what?* To be published.

Dutton, G.H. (1998): *A Hierarchical Coordinate System for Geoprocessing and Cartography*. Lecture Notes in Earth Science, Springer, 228 pages.

Franklin, A. (1992): Faster line segment intersection. *Graphics Gems III*, edited by D. Kirk, Academic Press Inc., San Diego, 199-203.

Fritsch, E. (1997): Utilisation de la coubure pour la généralisation du linéaire routier. *Bulletin d'Information n.66*, IGN, France.

Fritsch, E. and J.-P. Lagrange (1995): Spectral Representations of Linear Features for Generalisation. In: Frank, A.U. and W. Kuhn (eds.): *Spatial Information Theory*, Lecture Notes in Computer Science 988, Springer, 157-171.

Gallant, J.C. and M.F. Hutchinson (1996): Towards an Understanding of Landscape, Scale and Structure. *3rd conference on integrating GIS and Environmental Modeling*, Santa Fé.

Hake, G. and D. Grünreich (1994): *Kartographie*. 7.Auflage, de Gruyter, Berlin, 599 p.

Hangouët, J.-F. (1996): Automated Generalization Fed on latent Geographical Phenomena. *Proceedings of InterCarto 2*, Irkutsk, 125-128.

Hangouët, J.F. (1995): Computation of the Hausdorff distance between plane vector polylines. *Proceedings of AUTO-CARTO 12*, 4, 1-10.

Heller, M. (1990): Triangulation algorithms for adaptive terrain modeling. *Proceedings of the 4th International Symposium on Spatial Data Handling*, 1, Zürich, 163-174.

Højholt, P. (1998): Solving Local and Global Space Conflicts in Map Generalisation Using a Finite Element Method Adapted from Structural Mechanics. *Proceedings of 8th International Symposium on Spatial Data Handling*, Vancouver, Canada. 679-689.

Horton, R.E. (1945): Erosional Development of Streams and their Drainage Basins - Hydrophysical Approach to Quantitative Morphology. *Geological Society of America, Bulletin 56*, 275-370.

IGN (1997-1998): *Description and Assessment of Generalisation Algorithms available at the COGIT Lab (IGN).* Internal Report of the IGN.

Imai, H. and M. Iri (1988): Polygonal approximation of a curve - formulation and algorithms. In: Toussaint, G.T. : *Computational Morphology (Machine Intelligence and Pattern Recognition 6)*, Elsevier, Holland, 71-86.

Jansen, M. and M. van Kreveld (1998): Evaluating the consistency of cartographic generalization. *Proceedings of 8th International Symposium on Spatial Data Handling*, Vancouver, Canada.

Jenks, G.F. (1989): Geographic logic in line generalization. *Cartographica*, 26 (1), 27-43.

João, E.M. (1995): The importance of quantifying the effects of generalization. In: Muller, J-C., Lagrange, J.P. and Weibel, R. (eds.). *GIS and Generalization*, GISDATA 1, London, Taylor & Francis, 183-193.

Jones, C.B., G.L. Bandy and J.M. Ware (1995): Map generalization with a triangulated data structure. *Cartography and Geographic Information Systems*, 22 (4), 317-331.

Lam, N.S.-N, and L. DeCola (1993): Fractal simulation and interpolation. In: Lam, N.S.-N. and L. DeCola (eds.): *Fractals in Geography*, Prentice Hall, Englewood Cliffs, New Jersey, 56-74.

Lam, N.S.-N. and L. DeCola (1993): *Fractals in Geography*. Prentice Hall, Engelwood Cliffs (New Jersey), 308 p.

Lang, T. (1969): Rules for the Robot Droughtsmen. *The Geographical Magazine*, 42 (1), 50-51.

Lecordix, F. and C. Plazanet (1996): Plateform de generalisation PlaGe et application a la caricature. *Bulletin d'Information n.65*, IGN, France.

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

Lee, D.T. (1982): Medial axis transformation of a planar shape. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 4 (4), 363-369.

Li, Z. (1995): From phenomena to essence: envisioning the nature of digital map generalization. *The Cartographic Journal*, 32 (1), 45-47.

Li, Z. and S. Openshaw (1992): Algorithms for automated line generalization based on a natural principle of objective generalization. *IJGIS*, 6 (5), 373-389.

Lichtner, W. (1979): Computer-assisted processes of cartographic generalization in topographic maps. *Geo-Processing* 1 (1), 183-198.

Longley, P.A. and M. Batty (1989): On the fractal measurement of geographical boundaries. *Geographical Analysis*, 21 (1), 47-67.

Lowe, D.G. (1988): Organization of Smooth Image Curve at Multiple Scales. *International Conference on Computer Vision*, 558-567.

LSL (1998): *Documentation of the Gen-Lib (for the AGENT-project)*. Unpublished.

Mackaness, W.A. (1995): A constraint based approach to human computer interaction in automated cartography. *Proceeding ICA/ACI*, 2, Barcelona, 1423-1433.

Mackaness, W.A. and G.A. Mackechnie (1997): Detection and simplification of road junctions in automated map generalisation. *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden, 1013-1021.

Mackaness, W.A. and P.F. Fisher (1987): Automatic recognition and resolution of spatial conflicts in cartographic symbolisation. *Proceedings of AUTO-CARTO 8*, 709-718.

McMaster, R.B. (1986): A statistical analysis of mathematical measures of linear simplification. *The American Geographer*, 13 (2), 103-116.

McMaster, R.B. (1987): Automated line generalization. *Cartographica*, 24 (2), 74-111.

McMaster, R.B. (1989): The integration of simplification and smoothing algorithms in line generalization. *Cartographica*, 26 (1), Monograph 40, 101-121.

McMaster, R.B. and K.S. Shea (1992): Generalization in digital cartography. *Association of American Geographers*, Washington D.C., 134 p.

Meyer, U. (1986): Software-Developments for Computer-Assisted Generalization. *Auto-Carto 2*, London (GB), 247-256.

Michel, R. (1997): Computer-supported symbol displacement. *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden, 1795-1803.

Moellering, H. and J.N. Rayner (1982): The dual axis Fourier shape analysis of closed cartographic forms. *The Cartographic Journal*, 19 (1), 53-59.

Muller, J.C. (1986): Fractal dimension and inconsistencies in cartographic line representations. *The Cartographic Journal*, 23 (4), 123-130.

Muller, J.C. (1987): Optimum point density and compaction rates for the representation of geographic lines. *Proceedings of AUTO-CARTO 8*, 221-230.

Muller, J.C. (1987): Fractal and automated line generalization. *The Cartographic Journal*, 24 (1), 27-34.

Muller, J.C. (1990): The removal of spatial conflicts in line generalization. *Cartography and GIS*, 17 (2), 141-149.

Müller, J.C. and Z. Wang (1992): Area-patch generalization. *The Cartographic Journal*, 29 (2), 137-144.

Mustière, S. (1998): An algorithm for legibility evaluation of symbolized lines. *Intercarto 4*, Barnaul (Russia), 43-47.

Mustière, S. (1998): GALBE: Adaptive Generalisation. *GIS PlaNet 1*. Lisbon (Portugal).

Nickerson, B.G. (1988): Automatic cartographic generalization for linear features. *Cartographica*, 25 (3), 15-66.

Nickerson, B.G. and H. Freeman (1986): Development of a rule-based system for automatic map generalization. *Proceedings of the 2nd International Symposium on Spatial Data Handling*, Seattle (Washington), 537-556.

O'Neil, M.P. and D.M. Mark (1987): The psi-s Plot: A useful representation for digital cartographic lines. *Proceedings of AUTO-CARTO 8*, 231-240.

O'Rourke, J. (1993): *Computational Geometry in C*. Cambridge University Press, 346 p.

Peng, W. and K. Tempfli (1996): An object-oriented design for automated database generalization. *Proceeding of the 7th International Symposium on Spatial Data Handling*, Delft (NL), 199-213.

Perkal, J. (1966): An Attempt at Objective Generalization. In: *Michigan Inter-University Community of Mathematical Geographers*, Discussion Paper 10, University of Michigan.

Plazanet, C. (1995): Measurement, characterization and classification for automated line feature generalization. *Proceedings of AUTO-CARTO 12*, 4, 59-68.

Plazanet, C. (1996): *Enrichissement des bases de données géographiques: analyse de la géometrie des objets linéaires pour la généralisation cartographique (application aux routes)*. PHd Thesis of Marne la Vallee University.

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), 291-305.

Preparata, F.P. and M.I. Shamos (1985): *Computational Geometry*. Springer-Verlag, New York, 390 p.

Pullar, D. (1990): Comparative study of algorithms for reporting geometrical intersections. *Proceedings of the 4th International Symposium on Spatial Data Handling*, 2, Zürich, 66-76.

Regnauld, N. (1996): Recognition of building clusters for generalization. *Proceeding of the 7th International Symposium on Spatial Data Handling*, Delft (NL), 185-198.

Regnauld, N. (1997): Structuration du bâti pour la généralisation. *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden.

Reumann, K. and A.P.M. Witkam: Optimizing Curve Segmentation in Computer Graphics. *International Computing Symposium*, 467-472.

Reynes J.L. (1997): *Sélection du réseau routier en vue de la sélection*. DESS Report, University Paris VI, COGIT.

Ruas, A. (1995): Multiple Representation and Generalization. *IGN Internal Report* DT-95. 1037.

Ruas, A. (1997): Formalism pour l'automatisation de la généralisation: métrique, topologie, partitionnement hiérarchique et triangulation locale. *Proceedings 17th International Cartographic Conference*, Barcelona, Spain, 950-959.

Ruas, A. and W. Mackaness (1997): Strategies for urban map generalization. *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden, 1387-1394.

Ruas, A. (1988): *Généralisation d'imeubles*. Ordnance Survey - ENSG.

Saalfeld, A. (1999): Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and GIS*, 26 (1).

Saigal, R. (1968): A constrained shortest route problem. *Operations Research*, 16 (1), 205-209.

Sedgewick, R. (1984): Algorithms. Addison-Wesley Publishing Company, Inc. (USA), 552 p.

Shapiro, B., J. Pisa and J. Sklansky (1981): Skeleton generalization from x,y boundary sequences. *Computer Graphics and Image Processing*, 15 (2), 136-151.

Shea, K.S. (1991): Design consideration for artificially intelligent system. In: Buttenfield, B.P. and McMaster, R.B. (eds.). *Map Generalization: Making Rules for Knowledge Representation*. London, Longman, 3-20.

Staufenbiel, W. (1973): *Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung grossmasstäbiger Gebäudedarstellungen*. WissArbUH Nr.51, Hannover (D).

Thapa, K. (1989): Data compression and critical points detection using normalized symmetric scattered matrix. *Proceedings of AUTO-CARTO 9*, 78-89.

Töpfer, F. (1974): *Kartographische Generalisierung*. VEB Hermann Haack, Geographisch-Kartographische Anstalt Gotha/Leipzig, 336 p.

Töpfer, F. and W. Pillewizer (1974): The principles of selection. *The Cartographic Journal*, 3 (1), 10-16.

Van Horn, E. (1986): Generalizing Cartographic Data Base. *Proceedings of AUTO-CARTO 7*, 532-540.

van Oosterom, P. (1995): The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning. In: Muller, J-C., Lagrange, J.P. and Weibel, R. (eds.). *GIS and Generalization*, GISDATA 1, London, Taylor & Francis, 120-125.

Visvalingam, M. and J.D. Whyatt (1990): The Douglas-Peucker algorithm for line simplification. *Computer Graphics Forum*, 9 (3), 213-228.

Visvalingam, M. and J.D. Whyatt (1993): Line generalization by repeated elimination of points. *The Cartographic Journal*, 30 (1), 46-51.

Wang, Z. and J.C. Muller (1993): Complex coastline generalization. *Cartography and GIS*, 20 (2), 96-106.

Wang, Z. and J.C. Muller (1998): Line gneralization based on high-level characteristics. *Cartography and GIS*, 25 (1).

Weibel, R. (1992): Models and Experiments for Adaptive Computer-Assisted Terrain Generalization. *Cartography and Geographic Information Systems*, 19 (3), 133-153.

Weibel, R. (1997): Generalization of Spatial Data: Principles and Selected Algorithms. In: van Kreveld, M. et al (eds.): *Algorithmic Foundations of Geographic Information Systems, Lecture Notes in Computer Science 1340*, Springer, 99-152.

Weibel, R. and G. Dutton (1999): Generalizing Spatial Data and Dealing with Multiple Representations. In: Longley et al. (eds.): *Geographic Information Systems*, John Wiley & Sons. 125-155.

Williams, R. (1987): Preserving the area of regions. *Computer Graphics Forum*, 6 (1), 43-48.

Zahn, Ch.T. (1971): Graph-theoretical methods for detecting and describing Gestalt clusters. *IEEE Transactions on Computers*, C-20 (1), 68-86.

Zhang, L. and Z. Tian (1997): Refinement of Douglas-Peucker algorithm to move the segments toward only one line. *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden, 830-835.

6 Appendix

This Appendix presents a detailed survey of algorithms for generalisation, with a focus on techniques for independent generalisation. Note that the survey is broader than needed for the specific purposes of the prototype. The concluding section of report D D2 provides a summary of the recommendations that can be drawn from this comprehensive survey. The specifications presented in D D1 give details of the generalisation algorithms that were selected for implementation in the first version of the E2 prototype.

Simplification: Weeding

Algorithm	Description	Assessment, Comments and Recommendations
Random-point selection	Choose points of simplified line randomly.	No cartographic use.
n^{th} point algorithm	Every n^{th} vertex of a polyline is selected; all others are eliminated.	No cartographic use.
McMaster's local processing routines **(McMaster, 1987)**	These algorithms work very locally, this means they only take the two neighboring points into consideration to decide if a point should be eliminated or not. McMaster describes various measures, which allow defining if a point is negligible, or not, e.g.: angular change, distance to neighbours or area between consecutive points.	Fast, but cartographically inferior (McMaster, 1987).
Walking algorithm **(Müller, 1987)**	Define a step-length and walk along a line marking all points that are visited using this step-length. The simplified line is built of the marked points. [parameter: size of walking step]	• The Walking algorithm has the advantage that it automatically preserves fractal dimension for any level of reduction. All other algorithms have the tendency to produce generalized lines with lower fractal dimension (Müller, 1987). • The algorithm can be used also for measuring the fractality of a line (Müller, 1987). • But: what is the meaning of self-similarity in cartographic data? • The algorithm does not detect features that are smaller than twice the size of the walking step (sampling theorem). Improvements see (Müller, 1987).
Reumann-Witkam Corridor Search **(Reumann, and Witkam, 1974)**	This algorithm uses a corridor of two parallel lines with width ε. This corridor is placed in direction of the first two vertices. The intersection of the corridor boundaries with the line is calculated and the last vertex of the line before the intersection was reported is stored. The corridor is reoriented using the direction of the stored vertices and its consecutive to calculate the next intersection. The simplified line is constructed by connecting the stored vertices. [parameter: ε = corridor width]	• Used in the MGE Map Generalizer. • Can essentially be compared to local processing routines based on angular deviation. • Not useful, because better algorithms exist.

Lang algorithm (Lang, 1969)	A 'look-ahead' digit and a distance tolerance control the execution of this algorithm. A vector connecting the start-point and the floating end-point (=start-point + 'look-ahead') is constructed. The perpendicular distances to all intermediate points are computed. If any of the intermediate distances is greater than a tolerance, the floating-end-point is withdrawn one point. This is done until all distances are smaller than the distance tolerance. The floating end-point is marked and the algorithm continues with this point as new start-point. [parameters: • 'look-ahead' defines the maximum number of points that can be deleted between two points that go into the simplified line. • tolerance distance: Specifies the maximal deviation of a point of the generalized line from the initial line.]	For assessment see the IGN (1997-1998) and McMaster (1987). Performance similar but slightly inferior to Dougals-Peucker according to McMaster.
Jenks algorithm **(Jenks, 1989)**	(For description see also explanations for parameters) If (1) the distance from *point1* to *point2* is less than *min1*, or (2) the distance from *point1* to *point3* is less than *min2*, *point2* is rejected. If both are larger, the angular check is calculated using *ang*. An angle smaller than *ang* will result in the removal of point2. [Parameters: • • *min1*: minimum allowable distance from point 1 to point 2; • • *min2*: minimum allowable distance from point 1 to point 3; • • *ang*: maximum allowable angle of change between the vectors connecting the three points;]	For assessment see McMaster (1987). Slightly more sophisticated variant of angular local processing routines. Inferior to Douglas-Peucker algorithm.
Visvaligam and Whyatt (1993)	For each vertex of a line compute the area of the triangle built with its two neighboring points. Then iteratively drop the points, which results in least areal displacement from the current partly simplified line. Recalculate the area for the neighbors next to the eliminated point. [parameter: max. area between 3 consecutive points so that a point is still allowed to be removed.]	• The algorithm results in 'least' areal displacement (Visvalingam, Whyatt, 1993). Because of that it performs better than Douglas-Peucker where minimisation of areal displacement is sought. • Simple and fast. • Method powerful for generalisation of buildings.
Sinuositiy-guided Point Selection (Dutton, 1998)	Calculate a sinuosity value for each point of a line. This is computed by constructing the ratio of distance to ±k vertices along the line to the length of an anchor centred at the given vertex. For each point compute then the difference between the local sinuosity (sinuosity computed with the neighboring points) and the regional sinuosity (sinuosity computed with points further away). Then drop points having minimum absolute distances between local and regional sinuosity (or contrary select important points, which have maximum differences between local and regional sinuosity).	• There are several ways presented how to calculate the sinuosity and how to compute the local and regional sinuosity. • Does not need a QTM data structure to compute sinuosity, but needs a QTM for selecting vertices based on QTM-levels. • Mainly useful for hierarchical generalisation.
ATM filtering (Heller, 1990)	ATM filtering is based on a coherent approach of successive construction of Delaunay triangulation: Starting with a triangulation of selected points from the Convex Hull, points are successively added to the triangulation, until vertical distances between the initial set and the triangle-faces are less than a tolerance ε. [parameter: tolerance ε = maximum vertical deviation from the initial set of DTM-points to the triangles.]	DTM simplification only! Can be linked to a 3-D variant of the Douglas-Peucker algorithm. Used for DTM data. Not required for the purposes of this project.
Cromley and Campbell (1992)	Simplification as an optimisation problem: The problem of line simplification becomes one of minimising (or maximising) a particular numerical property (see comments) of a line, subject to a constraint on the number of individual segments retained in the simplified line. [parameters: Number of line-segments to keep (or percentage) and choice of geometric criteria to use for optimisation.]	In a model, where the bandwidth constraint is allowed to dominate any ancillary objective, the optimal value of the second objective will be less than in a solution with a heuristic algorithm (such as the Douglas-Peucker-algorithm) (Cromley and Campbell, 1990). Subject of maximising/minimising can be: ▪ - total line length ▪ - angular change ▪ - perpendicular distance ▪ - areal displacement For easy, but slow implementation see Saigal, (1968).

Douglas-Peucker-Algorithm **(Douglas, and Peucker, 1973)**	The algorithm starts by connecting the two end-points of the original line with a straight line (termed the base line or anchor line). If the perpendicular distances of all intermediate vertices are within the tolerance ε from the baseline, these vertices may be eliminated and the original line can be represented by the base line. If any of the intermediate vertices falls outside ε, however, the line is split into two parts at the furthest vertex and the process is repeated recursively on the two parts. [parameter: maximal perpendicular distance of the simplified line from the original line.]	For literature assessment see Bader (1997), Beard (1991) or Visvalingam and Whyatt (1990) for a more critical appraisal. Improvements: • If the line represents a border of a closed polygon: see Williams (1987) for an algorithm to keep area of polygon while using the Douglas-algorithm; • Tree-structures for on-the-fly generalisation are described by vanOosterom (1995), Cromley (1991) and Buttenfield (1985); • In LSL-software the algorithm is followed by a geometry-cleaning-algorithm (see comments 'manmade'); • See Zhang and Tian (1997) for an improvement to move the line-segments to only one side. • See Saalfeld (1998) and deBerg et al. (1995) for topologically consistent line simplification using the Douglas-algorithm.
deBerg et al. (1995) **[see also Imai and Iri, 1988]**	Approach: Avoid conflicts resulting through generalisation from the beginning by using a geometric algorithm. Algorithm: See the chain as a graph. Add new links to the graph that represent valid shortcuts through the graph under a certain criteria. For the following criterias these shortcuts are computed: ▪ shortcuts that guarantee a maximum deviation of intermediate points; ▪ shortcuts that leave a given set of points P on the same side of the line; ▪ shortcuts that do not result in self-intersections and intersections with other chains. Compute the shortest path through the chain using only shortcuts that are valid under all the above criterias. [parameter: maximal perpendicular distance of the simplified line from the original line.]	Algorithm insures: ▪ no point on the chain *C* has distance more than a pre-specified error tolerance to its simplification *C′*; ▪ the simplification *C′* is a chain with no self-intersections; ▪ the simplification *C′* may not intersect other chains of the subdivision; ▪ all points of *P* lie to the same side of *C′* as of *C*. Similar algorithm: This algorithm is an adaptation of the algorithm presented by Imai and Iri (1988); Improvements: deBerg et al. (1995) provide certain information about how to increase speed (e.g. reducing the critical points to the convex hull of the chain). A more detailed description can be found in deBerg et al. (1998).

Simplification: Unrestricted Simplification

Algorithm	**Description**	**Assessment, Comments and Recommendations**
Van Horn grid **(Van Horn, 1986)**	Given a grid, displace each point of the line to the nearest node of the grid. [parameter: grid size]	'Step like'- results; Creates often self-intersections (Mustière, 1998). Compare to algorithm by Li and Openshaw (1992).
Algorithm based on a natural principle of objective generalization **(Li and Openshaw, 1992)**	3 similar algorithms are proposed. At this place the Raster-Vector-Algorithm is described: ▪ Define a raster size. The raster is positioned over the original line, so that the starting point is centred in a raster-cell. ▪ Compute the intersections between the line and the raster. ▪ The midpoints between two consecutive intersection-points are connected to the simplified line. [parameter: smallest visible object (SVO). From this value the raster-size/step-length is derived.]	The algorithm works in raster mode, vector mode, and raster vector mode. (Li and Openshaw, 1992) See also Bader (1997) for an evaluation. Quite similar to the algorithm working in raster-mode is the van Horn algorithm.
Class of algorithms for QTM-based line simplification **(Dutton, 1998)**	Compute Quaternary triangular mesh (QTM) of the original line; The line enters and exits several mesh elements (*mel*). Such runs are identified by inspecting the sequence of QTM ID's along the line at the target resolution; All points of a run are replaced by the median point of a run. [parameter: QTM-level]	Various improvements described by Dutton (1998). Most important: Use a 'lookahead' parameter to identify *mel's* where the line enters and exists several times (several runs per *mel*); Quality and appearance similar to Li and Openshaw (1992).
Whirlpool **(Dougenik, 1980)**	The principle of the algorithm is to replace a cluster of points on a line by one representative point: ▪ Two points are neighbours if their distance is under ε; ▪ A set of points form a cluster whenever they are linked to one another by neighbourhood relations;	• Evaluation must be made to avoid that important shape 'between' two cluster-points is removed. • "At the end, the distance between remaining points is always bigger than ε which can be the line width. In such a case whirlpool can be used to avoid line

	▪ Whenever a cluster of points is detected, the average position is chosen (and the points between the points forming the cluster are deleted); [parameter: ε: Distance for defining neighboring points. This ε corresponds to the parameter used by Perkal (1966).]	symbol overlappings" (IGN, 1997-1998). • A major drawback lies in the lack of capacity to detect conflicts produced by points being very close to line segments. • See also IGN (1997-1998). • Personal note: The Li and Openshaw (1992) algorithm results in similar output.
Epsilon-Generalization **(Perkal, 1966)**	Place a circle of diameter ε inside a region. The circle is then rotated in such a way that it remains completely inside the area. The ε -region of M is then the set of all points p having the property that they are contained within the circle of diameter ε, which can be completely included within the region M. [parameter: radius of circle (ε)]	• Problems (Mustière, 1998): brutal smoothing of sinuous lines; • The algorithm produces different results depending on which side the algorithm is applied. • See also Beard (1991). • Not implemented by Perkal. For code implementing the rolling ball principle see Brophy (1973), Deugenik's Whirlpool (1980), Mustière (1998).
Line generalization based on high-level characteristics **(Wang and Müller, 1998)**	• Detection of bends by partitioning a line in positive and negative bends (a positive bend is a series of nodes, where the incoming segments form angles > 180º); • Computation of a compactness index for each bend in order to judge importance of a bend; • Iterative elimination of smallest bends; [parameter: threshold for cutting bends (by specifying degree of compactness)]	Important bends stay, small bends are eliminated (Wang and Müller, 1998); The same underlying principle may be used for bend combining and exaggeration. An empirical (and critical) assessment of this procedure was done by Visvalingam (accepted for publication in Cartography and GIS, 1999). The algorithm is implemented in the `BENDSIMPLIFY` command used in ARC/INFO.
Complex coastline generalization **(Wang and Müller, 1993)**	▪ Searching rivers; Identification of necks; ▪ River hierarchy; ▪ Selection and Elimination of rivers; ▪ River widening and simplification; [parameters: various]	Very specific solution for estuaries of rivers and coastlines. The algorithm can also be used for selection.
Same-sized best-oriented rectangle **(Hangouët, 1996)**	A building's main orientation, based on the direction of its walls, is computed (see measure 'approximate mean orthogonal directions'). With this orientation the building is rotated to the horizontal. The bounding-rectangle is computed. A similar rectangle is calculated with the same size as the original building. This rectangle is rotated back.	See very similar algorithm distributed by LaserScan (LaserScan uses a different criteria to calculate the main direction of the building).
Manmade **(LSL, 1998)**	The following steps are processed only if 3 consecutive points form approx. a right angle (arbitrary angles are not supported by the algorithm): ▪ A triangulation is accomplished by forming triangles using 3 adjacent points; ▪ Decide on triangles which are considered for removal based on a min. size for non-hypotenuse triangle-sides; ▪ Remove triangles. Order is important: Start with small ones, then take exterior triangles. [parameters: minimum small length and minimum long length	• As the title expresses: this algorithm handles only objects with rectangular shape (buildings); • The order in which the triangles are processed is important. There is no simple rule for deciding this order (LSL, 1998). • The manmade algorithm is followed by a geometry cleaning algorithm: • Points are removed from the area boundaries if the angle between the incoming and outgoing segments incident on a point is outwidth a certain tolerance (close to 0º or 180º)
Detection and simplification of road junctions in automated map generalization **(Mackaness and Mackechnie, 1997)**	Detect junctions through cluster analysis. Results are stored in dendrograms and as graphs (modeling the connectivity of roads). Simplify junctions: Determine subgraphs in the cluster. For each subgraph determine a new centroid, determine junction in cluster and connect incoming roads to new centroid-junction. [parameters are needed for identification of clusters]	Problems arise when using big clusters (e.g. 'collapsing star' effect). (Mackaness and Mackechnie, 1997)

Collapsing

Algorithm	Description	Assessment, Comments and Recommendations
LSL collapse **(LSL, 1998)**	This algorithm uses Scan-lines to determine too narrow parts. The object is scanned in regular distance (*scan_pitch*) with a user-defined ray-length (*scan_width*). If the scan-line hits the other side of the object, the midpoint on the scan-line is computed. [parameters: ▪ *scan_width*: The scan-line length; ▪ *scan_pitch*: distance between two scan lines;]	This algorithm is mainly used to collapse close 'parallel' lines to single lines (e.g. the algorithm collapses a road represented by two parallel lines to its centreline).
Collapsing using the skeleton	Compute the skeleton of a polygon and collapse the borders to this structure. See Chithambaram and Beard (1991), Bader (1997), Jones et al. (1995).	Depends on the robustness of the algorithm for skeleton computation and (if implemented using triangles) on the arrangement of triangles. See Bader (1997) for a description of collapsing an area in a polygonal network.
LSL typification **(LSL, 1998)**	Reduce the complexity of a group of point objects by removing all points except one. The representative pattern is kept by clustering the overall point-distribution in adequate groups. The representative point of a group is set at one of the following positions: ▪ at the centre of gravity (or nearest point to the centre of gravity); ▪ at the midpoint of the MBR; ▪ at the simple arithmetic mean of all the points in group (Mean Point); ▪ at the Seed point (as calculated by GOTHIC) [parameter: the algorithm need a definition of the group to collapse]	How are the clusters / groups found? If this algorithm is used in conjunction with a clustering algorithm to determine groups, the algorithm performs the task of the structuration operator.

Enhancement: Enlargement

Algorithm	Description	Assessment, Comments and Recommendations
LSL-area-enlargement **(LSL, 1998)**	Scale the object around the area's seed point by a parameterised amount. [parameter: scale-parameter]	Also lines can be scaled with this algorithm if they build 'nearly' a closed loop.
Exaggeration / Shrinking using the skeleton **(Chithambaram et al., 1991)**	Compute the skeleton of a polygon; Exaggerate using the branches of the skeleton. [parameters: Distance to exaggerate along the skeleton-branches]	Only vague description by Chithambaram et al. (1991).
Enlargement using the sum of normals **(Bundy, Lee, Jones, 1995)**	The vector of displacement is calculated for each point in the object's outline as the vector sum of the normals (in the outward direction and of length delta). [parameters: delta: buffer width around building]	no evaluation.
Accordion **(Plazanet, 1996)**	v.1: This algorithm aims to enlarge a bend or bend-series in order to remove coalescencing bends. The central inflexion point of the line does not move and all other points are moved away from it by a value ε in direction of the regression line of the inflexion points. v.2: The points are moved away perpendicular to the direction of each bend axis. [parameter: • ε: Enlargement; • (σ: Gaussian parameter for detection of inflexion points.)]	The detection of inflection-points is critical and hard to do automatically (IGN, 1997-1998)

Enhancement: Caricature

Algorithm	Description	Assessment, Comments and Recommendations
Fourier line enhancement **(Clarke et al., 1993)**	The method uses (1) an equidistant resampling of the source line to yield independent x and y series, (2) a discrete Fourier transformation of these two series, (3) modification in the Fourier domain by either extraction of significant harmonics or deliberate amplification of the higher frequency harmonics, and (4) the equivalent inverse Fourier transform back to the spatial domain. [parameters: various, e.g. step-length for sampling; or filter for enhancement of high frequencies]	It has been empirically shown that the results are similar to those achieved with Dutton's technique (Clarke et al., 1993). Independent parameterisation of the *x* and *y* coordinates is not useful (Werschlein, 1996). Filtering via the Fourier transform affects the line globally; no localised enhancement possible.
Smoothing using spectral filtering **(Fritsch and Lagrange, 1995)**	Transform a line using Fourier series. Represent the line using less frequencies to open/enlarge bends (the lack of high frequencies avoids sharp bends).	A transformation using Fourier series can also be used to smooth a line. The same applies as for Clarke et al. (1993): Fourier based filtering can only be applied at the global level. Wavelet filtering can be used for localised enhancement (see also Werschlein, 1996) but it is difficult to handle and would need more research.
LSL-line-exaggeration **(LSL, 1998)**	Simplify line; Construction of 'base lines': Base-lines are drawn between the turning points (inflection points) of the simplified lines; The line is divided into disjoint curves. The dividing points are the points on the original line closest to the midpoints of each segment of the simplified line; The exaggeration is now performed with all points on the original curve being displaced against the midpoints of the baselines of their particular line curve segment (amount of displacement weighted by distance from midpoint);	Check if this algorithm also be used for the detection of bends.
Brophy-Dutton-exaggeration **(Brophy, 1973)** **(Dutton, 1981)**	Same as Brophy-algorithm for smoothing (see description there), but instead of moving points towards the centre of their inscribed circle, the point are shifted in the opposite direction. [parameters: see Brophy-smoothing algorithm]	Enhancement of this 'fractalisation' type may not make much sense in cartographic terms.
Lowe - Barillot **(Lowe, 1988)** **(Barillot, 1996)**	Problem to solve: Through a Gaussian smoothing, bends are usually moved in direction of the curvature center. Algorithm: After the Gaussian smoothing, the Lowe correction is applied, computing the curvature and the curvature center in each point of the line. This result is finally smoothed again gently. [parameters: ▪ σ (Gauss): to describe the Gaussian curve; ▪ σ (Lowe): parameter of the Lowe-algorithm; ▪ enhancement coefficient]	Results are very different depending on which side of the line the ball is rolled (this effect can be either positive or negative!). Using the same algorithmic principle with a strong enhancement coefficient an exaggeration of bends can be done (IGN-templates, 1998).
Plaster **(Fritsch, 1997)**	The smoothing of curvature (e.g. Gaussian) insures bend enlargement, with warranted minimal curvature radius. The extremities of the line and the summits of the principal bends are reallocated (to the original position) using different planar transformations. [parameters: σ: Gauss-parameter, used for smoothing as well as for detection of hairpin bends (places with high values of curvature after smoothing)].	See also IGN (1997-1998).
Mechanistic approach **(Fritsch, 1997)**	The algorithm works with repulsion forces between objects: 1) the 'objects' are small pieces of a road (and no 'atomic' database objects, such as 'lake perimeter'...) so the algorithm needs to cut the line in a lot of short segments. 2) concurrently with repulsion, the algorithm computes other forces, in order to model various cartographic requirements (e.g.: geometric precision is modelled by attraction, regularity of road direction is modelled with rigidity, and so on).	See also Højholt (1998) or Bader (1997) for similar approaches.

Balloon **(Lecordix and Plazanet, 1996)**	This algorithm aims to enlarge bend-summits in order to remove bend coalescence. It moves each point of the line in the local perpendicular direction of the line by a value ε (enlargement). The inflexion points of the line do not move and the maximum displacement is realised at the bend's summit. [parameters: ε: Enlargement; (σ: Gaussian parameter for the detection of inflexion points.)]	See also IGN (1997-1998) and Lecordix et al., (1997).
Maximal Break **(Mustière, 1998)**	Derives a new line that is at a distance d to the left (or right) from the original line. Therefore the line-segments are moved by *d*, while a circle of radius *d* is calculated around vertices. The shifted line-segments and the circles are connected to one line. The circular parts are finally approximated by line-segments. [parameters: the side, to which the line should be shifted and the shifting distance *d*]	Algorithm useful for widening bends where coalescence was previously detected (IGN, 1997-1998). This algorithm may also be used for an implementation of the rolling-ball principle.
Minimal Break **(IGN-templates, 1998)**	• Compute the skeleton (using a triangulation) of a bend; • Compute a new bend that has distance *d* to the skeleton; • The result is a line that can be symbolized with width 2**d*, so that no coalescence in the bend is produced. [parameter: distance *d* (= half of line-symbol-width)]	See also IGN (1997-1998)

Enhancement: Smoothing

Algorithm	Description	Assessment, Comments and Recommendations
Weighted moving average **(McMaster, 1989)**	A new position for each point of a line is computed. Therefore the new position of a point n is computed based on the coordinates of the *n-k* and *n+k* surrounding points. The neighboring points are weighted without taking their distance to the actual point into consideration (for example *k*=2: Weights: 0.1; 0.2; 0.4; 0.2; 0.1). [parameters: • number of neighboring points *k* which have to be respected during computation; • weighting of points;]	Simplistic approach: Gaussian smoothing is similar but seems better.
Sliding moving average **(McMaster, 1989)**	First compute a simple arithmetic average (e.g. Weighted-moving-average). In a second step, the actual point is displaced towards the calculated coordinates. [parameters: In addition to the parameters used for a simple arithmetic averaging, the amount of displacement has to be specified.]	Simplistic approach: Gaussian smoothing is similar but seems better.
Distance weighted average **(McMaster, 1989)**	In addition to the Weighted-moving-average and the Sliding-moving-average this technique uses the actual distances between the points to define a weighting function. [parameters: • number of neighboring points k which have to be respected during computation; • distance weighting formulae (e.g. 1/d^2)]	The Gaussian smoothing algorithm is similar (only using a Gaussian weighting-factor distribution).
Gaussian smoothing **(Badaud et al, 1986)**	For computing the position of each point its own and the coordinates of the neighbours are taken into account using a distance weighted average. The weighting function is described by the Gauss function. [parameters: • σ (for definition of the Gaussian function); • number of neighboring points which have to be respected during computation; eventually, this may be derived from σ;]	• See IGN (1997-1998) for detailed assessment. • The Gaussian smoothing moves bends in the direction of their curvature centre. The bends can be enhanced again using the Lowe-algorithm.

Brophy smoothing **(Brophy, 1973)**	The following method is computed for every point of the initial line: ▪ Build a triangle with the current point p and the points *p+k* and *p-k*. ▪ Inscribe each triangle a circle. ▪ The current point is moved a specific distance towards the center of the circle. [parameters: ▪ 'look-ahead' *k* to build a triangle. ▪ smoothing factor: How far should the point be moved along the line connecting the initial point and the center of the circle.]	• "Unfortunately, this procedure requires that points are equally spaced along the curve" (McMaster, 1987) • "Results usually not very good if the line is too heterogeneous" (Mustière, 1998). • Used in the MGE Map Generalizer
Global filtering **(Weibel, 1992)**	Global filtering consists of a variety of smoothing filters (in the spatial and frequency domain), combined with filters for enhancement.	2-D variants of filters for lines. These filters can be used for smoothing or surfaces (e.g. DTM's). Not needed for the purposes of AGENT.
Epsilon generalization **(Perkal, 1966) (McMaster, 1989)**	Line smoothing is achieved by rolling a circle with a predefined radius (ε) along a line. As the circle is rolled indentations not covered or touched by the circle are eliminated. [parameter: radius of circle (ε)]	• Problems (Mustière, 1998): brutal smoothing of sinuous lines; • The algorithm produces different results depending on the side of the line the algorithm is applied to. • See also Beard (1991a) • Not implemented by Perkal. For code implementing the rolling ball principle see Brophy (1973), Dougenik (1980), Mustière (1998).
Splines **(McMaster, 1989)**	Mathematical approx. by a *k-th* order polynom using n consecutive points. (usually computing a polynomial function of degree 4, using 5 points) [parameters: order of polynom (*k*) and step-length (n)]	Consists of a family of mathematical blending functions with different characteristics, depending on the properties and constraints of the spline function used. Splines may be useful for modelling and representation of cartographic lines, but less so as generalisation algorithms.
Bézier **(McMaster, 1989)**	The Bézier-curve retains only the end points of a line and smoothes it by developing an n-1 polynomial equation (where n equals number of coordinate pairs)	As the Bézier curve represents a variant of splines, the same statement applies as above.
Chaikin's smoothing algorithm **(Chaikin, 1974)**	Consider a curve described by 4 points (*p1*, *p2*, *p3*, *p4*). The smoothed curve begins at p1 tangent to line *p1p2*, intersects the midpoint of line *p2p3* tangent to *p2p3* and ends at *p4*, tangent to line *p3p4*. This is achieved by iteratively dividing the quadriliteral at the midpoint of *p2p3* in two triangles. These triangles are again divided to build a total of 4 new quadriliterals. All midpoints of the new line segment are taken for the smoothed line. The iteration stops using a tolerance distance. [parameter: tolerance distance]	When the tolerance value is infinitely small, the smoothed curve approximates a B-spline. (McMaster and Shea, 1992)

Enhancement: Fractalization and Squaring

Algorithm	Description	Assessment, Comments and Recommendations
Carpenter Midpoint Subdivision Method **(Carpenter, 1981)** **Fournier et al. (1982)**	The approach is considered exact in that a set of existing points is retained throughout the process. The technique then adds randomly placed midpoints (= computed midpoints + stochastic component controlled by the length of the segment, a scale factor, a roughness factor and a Gaussian random number) to the line segments, connecting the points from a previous iteration.	The incorporation of an aspect of randomness in Carpenter's method is considered desirable for simulating natural boundaries (Lam and DeCola, 1993). Dutton (1981) suggested a similar algorithm.
Fractal enhancement of cartographic line detail **(Dutton, 1981)**	Introduction of fractality into a generalized line by adding non-collinear vertices in each line segment. [parameters: • Sinuosity dimension; • Uniformity Coefficient (regularity of introduced fractality/distortion); • Straightness and smoothing tolerances (preserve straight features, smallest segment to modify)]	• "The ability to manipulate the fractal dimensionality of cartographic objects is perhaps more useful for thematic mapping than for other cartographic applications" (Dutton, 1981) • For an evaluation see also Clarke et al., (1993) and Lam and DeCola (1993). • Carpenter (1981) suggested a similar algorithm.
Airault (1996)	This algorithm uses a global optimisation regarding several constraints. Each constraint is expressed as energy function for each point of the building. This results in a potential energy, which has to be minimised over the entire building. The final energy function takes the following constraints into account: ▪ Try to make right angles; ▪ Keep points close to their initial position; ▪ Preserve parallelism and alignment between buildings; ▪ Do not change initial topology. The method is implemented by moving the points iterative towards a state of lesser energy. Therefore, for each point the potential energy of all 8 neighbouring points is computed and the direction of least energy is chosen for displacement.	

Selection / Elimination

Algorithm	Description	Assessment, Comments and Recommendations
Radical law **(Töpfer, 1974)**	An empirical formula, which allows to compute the number of objects that should be maintained in the target map. *n.t* = *n.s* * sqrt(*s.s* / *s.t*) [parameters: ▪ *n.s* = number of objects in source map ▪ *s.s* = source scale ▪ *s.t* = target scale]	The radical law only expresses the number of objects to keep, but provides no information about the choice of objects. However, if linked to cartographic or geometric attributes ordered by relative importance, it can provide guidance also for the selection of individual objects or parts of objects.
Horton order **(Horton, 1945)**	Stream ordering schema: This schema reflects the topological order of edges in a river-tree from the sources to the outlet. The order combines topological order and metric properties (the longest branches in the tree are assigned the highest order). [Parameters to weight the importance of branches and a cut-off-parameter for the decision if branches should be eliminated.]	The Horton order is considered to be the most suitable order for cartographic generalisation of river networks. Additional orderings were presented by Strahler and by Shreve (see Weibel, 1991). It is easy to add more constraints to the ordering-process besides the length of branches (e.g. A river is more important, when there is a bridge over it)
Road selection **(Reynes, 1997)**	Road selection constrained by quickest path and attractive points.	Experimental. A similar approach using shortest path algorithms was presented by Thompson and Richardson (1995).
Elimination of polygons by using its skeleton **(Bader, 1997)**	Compute skeleton of polygon using a Delaunay triangulation; Connect neighboring polygon to skeleton; Skeleton is new border between polygons (update topology).	Algorithm works only for the elimination of polygons from a polygon-mosaic.

Displacement

Algorithm	Description	Assessment, Comments and Recommendations
Focus line displacement **(Michel, 1997)**	Important lines (mainly long straight roads) are identified and used as anchors for displacement (focus lines). Displacement is done perpendicular to these important lines. The induced displacement runs non-linearly to zero with decreasing distance to the map edge.	Only vague description available. No assessment possible.
Proportional radial enlargement **(Mackaness, 1994)**	• Detect a cluster of points; • Select a centre of the cluster; • Move all points away from the center by a distance d, such that d is proportional to the original distance from the center to that point; • If algorithm produces topological errors (some points lie on the other side of a linear feature), the position of the center has to be altered. [parameters for the identification of clusters, the definition of the cluster center and for displacement are needed]	Only useful for point features. Assumes that the displacement takes place in a radial arrangement about a single displacement centre (centre of gravity).
Displacement based on repelling forces **(Bader, 1997)**	Compute buffer around areas and their resulting intersections. The overlapping buffers build overlay-polygons. Triangulate overlay-polygons and determine their skeleton. For every triangle a repelling force is computed proportional to its area in orthogonal direction to the skeleton; For all points of the polygons, which created the overlaps, a final displacement vector is calculated by taking all previously computed repelling forces into account. [parameters: a buffer-size and a model to compute final displacement of area-points from vectors on the skeleton.]	• It is not easy to find an appropriate model to compute the resulting displacement. Good parameters must be determined for every situation. • Compare with similar approach used by Fritsch (1997).
Müller displacement **(Müller, 1990)**	Displacement using an empirical model for displacement. (point-to-point-conflict and point-to-line-conflict). [parameters: one parameter for defining the minimal distance between two points, and a second parameter for defining the search area for further critical points (propagation)]	The algorithm may result in subsequent collisions and may introduce new kinds of spatial interferences. Hence, the process may turn into an endless cycle (Müller, 1990). See also Mackeness and Fisher (1987).
Lichtner displacement **(Lichtner, 1979)**	Equation for computing displacement-vectors perpendicular to a straight-line object.	This is the displacement method used in CHANGE. Can be used for points and lines. It provided the basis for Nickerson and Freeman (1986).
Nickerson **(Nickerson, 1988)** **(Nickerson and Freeman, 1986)**	Algorithm using several steps (rather complex), including ▪ Accommodating end-node displacement; ▪ Detection of interfering lines; ▪ Ranking of interfering lines (computing priorities based on shared nodes, distance from nodes and esp. relative directions) ▪ Resolving interference: first compute single vector-displacements, then calculate final vectors by using a triangle-filter; ▪ Propagation of node displacement;	Improved version of the method by Lichtner (1979). Has been implemented in Plage.
LSL-Displacement **(LSL, 1998)**	The algorithm loops through all points. For each point the following steps are executed: • Find all points influencing displacement. (For lines too close to the current point, it is the point on the line closest to that point) • Calculate the displacement effect. This is derived by calculating the minimum distance, which the objects must be apart, and weighting the distance by the ratio of the object-priorities. • The result is a set of displacement vectors expressing the total amount of displacement. This displacement however is limited on any axis by the longest single displacement vector along that axis. [parameters: classes influenced by the displacement; priorities of objects (to derive amount of displacement); minimum distance between objects]	The same algorithm is used for line and area displacement. A line is treated as a set of points (special cases at end-points), an area as set of lines. Therefore the algorithm makes no attempt to displace either lines or area objects against each other.

Displacement using the FEM **(Højholt, 1998)**	This hollistic approach intends to solve cpnflict problems of an entire map partition simultaneously. The following steps are performed: 1. Delaunay triangulation of the area; 2. Allocation of stiffnesses and boundary conditions to triangles on the basis of the requirement for the solution (e.g. no deformation of building); 3. Solution of displacement problem with the Finite Element Method.	This method could be very useful for the AGENT project, as it is based on useful partitions (making displacement for a Meso agnet).
Displacement using the snakes concept **(Burghardt and Meier, 1997)**	Burghardt describes a method for displacing linear features using the snakes concept. Snakes are described and used in computer vision for the identification of pattern contours using energy minimising attraction. The displacement of lines is its reversal; the problem can thus also solved using a energy minimising function taking geometric and semantic constraints into account.	Restricted to solve conflicts between a pair (or a group) of lines.

Aggregation

Algorithm	**Description**	**Assessment, Comments and Recommendations**
LSL-merging **(LSL, 1998)**	If two areas already overlap: use geom_simp_combine (from GEOMLIB) to maintain the outline of the overlapping objects; If objects are in close proximity: areas are processed vertex by vertex and if vertices can be moved in order to create an overlap within the specified gap-tolerance, this displacement is carried out. Additional controls: Only move when more than one vertex triggers an overlap / only merge when moving does not create self-intersection. [parameter: gap-tolerance]	
Merging using the SDS **(Jones et al., 1995)**	Merging using a triangulation between objects.	See also Bader (1997) for quite similar algorithm.
Agreg-disp (Regnauld)	This algorithm aggregates two buildings by displacing them until they overlap sufficiently. A perceptual threshold that defines the amount (length) of overlap specifies the amount of overlap.	
Delaunay triangulation for point aggregation **(DeLucia and Black, 1987)**	Generate a network of Delaunay triangulation with the point features as nodes. This network consists of a set of interior and border Voronoi-cells. Clusters of cells are built using a threshold. The border cells are used to define the new area outline. [parameter: tolerance for defining clusters.]	See also McMaster and Shea (1991).
Graph-theoretical methods for detecting and describing gestalt clusters **(Zahn, 1971)**	Various methods exist for describing clusters using graph-theoretical methods (esp. MST). The methods are not developed in the environment of cartographic generalisation, but may be adopted.	No evaluation in the area of cartographic generalisation
LSL-Aggrgation with Convex-hull and inner rings **(LSL, 1998)**	• Compute Convex-Hull of group of objects to aggregate; • Determine points that will form the inner ring: Place a circular buffer that is centered on the center of the MBR of the original hole and use several criterias (see LSL) to define the inner ring.	• Improved methods of the DeLucia and Black algorithm. • What is the ‘original hole’? • Has the hole to be set by a user?

Shrink Wrapped Hull **(LSL, 1998)**	Compute the Convex-Hull of a group of objects to aggregate; Scan around the hull and calculate the distance between two consecutive points. If the distance is greater than a tolerance, the algorithm looks for inner points; If an inner point is found that satisfies certain criterias (distance to outer point, no intersection with other lines, ...) the point is added to the current hull; The scan continues from this point and new neighbours have to be determined for this point.	
Interpolation-based typification **(Hangouët, 1996)**	• Select and re-allocate buildings belonging to a known row so that rendering at the new resolution retains the main characteristics of the original group. • Decide how many buildings should be represented; • Select buildings to be represented; • Re-allocate the selected buildings. [parameters are various, e.g. which characteristics define an important building, …]	Requires that groups of buildings are identified and qualified (linear or clutter); See also IGN (1997-1998).
Lichtner-structuration **(Lichtner, 1979)**	Generalisation of buildings using elimination, enlargement, merging (smaller buildings to larger ones) and simplification of the outline.	Only vague algorithm description, therefore number of user-specified parameters not known.
Regnauld-structuration **(Regnauld, 1997)**	Recognition and qualification of house-structure: Analysing groups using area, orientation and elongation and identifying the homogeneity of a group. Decomposition in homogenous groups. Structurization: For each group a new representation is derived. There is a distinction between internal space (density of the group) and external space (distances to other groups). [various parameters]	
Area-patch generalization **(Müller and Wang, 1992)**	Sequential use of different basic operations: Preprocessing data, expand or contract, eliminate, reselect, merge, displace, verify topological integrity, smooth contours of patches. [Each step might be changed individually, so a lot of parameters required]	▪ Use is limited to area-patches ▪ No underlying structure/order in group of areas is maintained.
IGN-Schematization **(Lecordix and Plazanet, 1996)**	This algorithm removes the 2m last bends of a bend series and enlarges the remaining ones. Each remaining vertex is displaced in the main direction of the bends of such a value that bends are reconnected to the original extremities of the bend series. [parameters: 2m: Even number of bends to remove; (σ: Gaussian parameter for the detection of inflexion points.)]	See also IGN (1997-1998).
GALBE **(Mustière, 1998)**	GALBE is the integration of various algorithms described above: Decompose line in homogenous parts regarding coalescence; Process each part individually: ▪ - if no legibility conflict: Gaussian smoothing ▪ - if simple legibility conflict (hair-pin-bend): Maximal Break ▪ - if complex legibility conflict: Accordion Local refinement: Minimal Break; Global refinement: gentle smoothing.	GALBE is the integration of several algorithms developed at IGN. It specifies the order, in which the algorithms are processed.