Chapter 5. Generalization of Spatial Data: Principles and Selected Algorithms

Robert Weibel

Dept. of Geography
University of Zürich
Switzerland
weibel@geo.unizh.ch

1 Introduction

In recent years, applications of geographic information systems (GIS) have matured, and spatial databases of considerable size have been built, and are being built and maintained continuously. Large amounts of money and time are invested into ambitious projects to build so-called spatial data infrastructures at the national level (e.g., MSC 1993, Clinton 1994, Grünreich 1992) and at the international level (e.g., EUROGI 1996). To enable the creation, maintenance, and use of such vast repositories of spatial information, a variety of methodological issues must be addressed. Besides methods for data modeling, data management and retrieval, as well as data distribution – including, for instance, standards for data documentation and exchange – techniques for automated generalization of spatial data (short: generalization) are of premier importance in this context. In GIS, generalization functions are needed for a variety of purposes, including the creation and maintenance of spatial databases at multiple scales, cartographic visualization at variable scales, and data reduction, to name just a few.

It is generally acknowledged that generalization is a complex process with ill-defined objectives, involving a good deal of subjective decisions. In order to solve the problem comprehensively, a variety of techniques including non-algorithmic solutions such knowledge-based methods and decision support systems approaches are needed. Clearly, however, data structures and algorithms form an indispensable foundation on which other approaches can build.

This survey presents an overview of principles, algorithms, and data structures for the generalization of spatial data. Sections 2 to 9 describe basic techniques, including an introduction to the principles and concepts underlying generalization, for those who are less familiar with the topic. Sections 10 , then, attempts to analyze what is wrong with basic generalization methods, and sections 11 to 13 discuss methods that extend the basic algorithms described in the first part and overcome some of their functional weaknesses. Other compilations of recent research in generalization can be found in Buttenfield and McMaster (1991), Müller et al. (1995a) and Weibel (1995a), and Molenaar (1996a).

2 What is Generalization and Why Does It Matter?

2.1 The Issue of Scale Change

Map generalization is a key element of cartography. Traditionally, spatial phenomena are cartographically portrayed on maps at different scales and for different purposes (e.g., topographic maps, geological maps, hiking maps, road maps). National topographic maps, for instance, are commonly produced at a series of scales [1], such as 1:25,000, 1:50,000, 1:100,000, 1:250,000, 1:500,000, and 1:1,000,000. The map scale is typically halved at each step in such a series (e.g., from 1:25,000 to 1:50,000). At the same time, the space available for drawing on the target map is divided by four, meaning that there is only a quarter of the space left to present the same amount of information as on the source map.

At the same time, as the *map scale* is reduced, small map objects may approach the limits of visual perceptibility. These perceptibility limits are termed *minimum dimensions* in cartography and are said to be, for instance, 0.35 mm for the length of sides of a black square (e.g., used to symbolize a building), or 0.25 mm for the distance between double lines which are often used to symbolize roads (SSC 1977). So, any map objects that would fall below these thresholds, but which the cartographer would still like to display on a map, would need to be enlarged accordingly in order to be clearly visible and discernible on the resulting map image. For example, on a map of 1:100,000, all buildings which have sidelengths smaller than 35 m – the vast majority of single family homes – would need to be enlarged to that minimum size. The same problem occurs with road objects; most roads are narrower than 25 m on the ground. So, to summarize, when reducing the scale of a map we are facing two problems which have a cumulative effect: available physical space on the map is reduced, and many objects may need to be enlarged in order to still remain visible. Both problems lead to a competition for available space among map objects. This situation is illustrated in Figure 1, which also depicts the necessary consequences. Only a subset of the original objects of the source map can be displayed on the target map and some objects may need to be displaced in order to avoid overlaps. This illustration, although schematic, also clearly shows why a mere photographic reduction would not be sufficient.

[1] Map *scale* is defined as the size ratio between an object (feature) in reality and its graphical representation on the map. Map scales with larger scale denominators (e.g., 1:500,000) are called 'small scales' in cartography, because they map everything to a small display area. Conversely, scales with smaller denominators (e.g., 1:10,000) are termed 'large scales'.

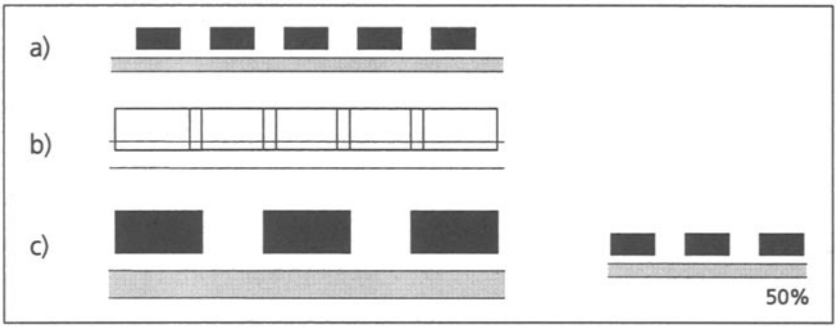

Fig. 1. Competition for space among map objects as a consequence of scale reduction. Reduction of available map space and enlargement of symbol sizes leads to overlaps and other spatial conflicts. These can be resolved by selecting only a subset of the objects of the source scale and by displacing some objects away from others (i.e., the buildings are displaced from the street).

2.2 Defining Generalization

In cartography, the process which is responsible for cartographic scale reduction is termed *generalization* (or map generalization, or cartographic generalization). It encompasses a reduction of the complexity in a map, emphasizing the essential while suppressing the unimportant, maintaining logical and unambiguous relations between map objects, and preserving aesthetic quality. The main objective then is to create maps of high graphical clarity, so that the map image is easily perceived and the message that the map intends to deliver can be readily understood. This position is expressed by the following concise definition by the International Cartographic Association:

> Selection and simplified representation of detail appropriate to the scale and/or the purpose of the map (ICA 1973: 173).

Note that map scale is not the only factor that influences generalization. Map purpose is equally important. A good map should portray the information that is essential to its intended audience. Thus, a map for bicyclists will emphasize a different selection of roads than a map targeted at car-drivers. Other factors that control traditional map generalization are the quality of the source material, the symbol specifications (e.g., the thickness and color of line symbols for roads, political boundaries, etc.), and technical reproduction capabilities (SSC 1977). The combination of these factors are called the *controls* of generalization (cf. Section 4.1).

In the context of digital cartographic systems and GIS, generalization has obtained an even wider meaning. This statement may at first sound surprising. One might expect that the transition from static paper maps to digital maps on computer screens, with the possibility to flexibly select and compose feature classes and features (objects) through queries, and the capability to interactively zoom and inspect the data at any desired scale (i.e., magnification) factor would have overcome the need for generalization. The answer clearly is negative, which is mainly due to two fundamental facts. First, spatial phenomena and processes are usually scale-dependent. Ideally, spatial data should be analyzed and viewed at the scale at which the modeled phenomena and processes are meaningful and best understood (Müller et al. 1995b). Second, generalization – which is essentially a process of abstraction and reduction of complexity – is a fundamental human intellectual activity and part of the general scientific process as well as everyday behavior and decision-making (Brassel and Weibel 1988). Without concentration on the essential aspects of a given problem we are soon lost in irrelevant details and unable to understand overriding patterns, let alone communicate them to outsiders. Thus, generalization also is of fundamental importance as a process of maximizing information content in building, maintaining, and communicating the content of spatial databases (Müller 1991). The definition of generalization in the digital context used by McMaster and Shea (1992) exhibits this close relation to the traditional process:

> Digital generalization can be defined as the process of deriving, from a data source, a symbolically or digitally-encoded cartographic data set through the application of spatial and attribute transformations. Objectives of this derivation are: to reduce in scope the amount, type, and cartographic portrayal of the mapped or encoded data consistent with the chosen map purpose and intended audience; and to maintain clarity of presentation at the target scale (McMaster and Shea 1992: 3-4).

Note that the same major generalization *controls* – scale, map purpose, intended audience – are mentioned as in the traditional view of map generalization. The 'spatial and attribute transformations' needed to realize the actual generalization process form the focus of Sections 7 to 13.

Viewed from yet another perspective, digital generalization can be understood as a process of ***resolution reduction*** (Ruas 1995a), affecting both the thematic and geometric domain[2]. In the thematic domain generalization implies a change of the database schema; the number of entities is reduced, attributes are eliminated, and attribute values are made less accurate (e.g., averaged). In the geometric domain generalization the resolution is reduced by eliminating objects or parts of objects, simplifying shapes, or displacing objects from one another in order to maintain good separability (cf. Fig. 1).

2.3 Motivations of Generalization

We have already alluded to some of the motivations of generalization above, but it is worthwhile making the list complete. Extending on Muller's (1991)

[2] Note that if temporal aspects are also modeled, generalization can be applied similarly to the temporal domain (e.g., reducing the resolution of a time series from daily to monthly averages).

discussion of requirements for generalization, we can develop a more detailed list of motivations:

1. **Develop primary database:** Build a digital model of the real world, with the resolution and content appropriate to the intended application(s), and populate it (object generalization; cf. Subsection 3.1).
   - Select objects
   - Approximate objects
2. **Use resources economically:** Minimize use of computing resources by filtering and selection within tolerable (and controllable) accuracy limits.
   - Save storage space
   - Save processing time
3. **Increase/ensure data robustness:** Build clean, lean and consistent spatial databases by reducing spurious and/or unnecessary detail.
   - Suppress unneeded high-frequency detail
   - Detect and suppress errors and random variations of data capture
   - Homogenize (standardize) resolution and accuracy of heterogeneous data for data integration
4. **Derive data and maps for multiple purposes:** From a detailed multi-purpose database, derive data and map products according to specific requirements.
   - Derive secondary scale and/or theme-specific datasets
   - Compose special-purpose maps (i.e. all new maps)
   - Avoid redundancy, increase consistency
5. **Optimize visual communication:** Develop meaningful and legible visualizations.
   - Maintain legibility of cartographic visualizations of a database
   - Convey an unambiguous message by focusing on main theme
   - Adapt to properties of varying output media

Examination of the above list reveals that classical cartographic generalization mainly relates to task 5 (visual communication) and to a lesser extent also to task 4, while tasks 1 to 3 are more specific to the digital domain (object generalization, model generalization). In task 5, an aspect of cartographic generalization germane to a GIS environment is that output may be generated for media of varying specifications, such as high-resolution plotted maps or low-resolution CRT views, requiring consideration of the resolution of the output media when composing maps for display (Spiess 1995).

3 Different Views of Digital Generalization

The reasons and motivations of *digital* generalization listed above show quite clearly how diverse the requirements are towards procedures implementing this process. Different views of the overall process are thus possible.

3.1 Generalization as a Sequence of Modeling Operations

A first view understands generalization as a process which realizes transitions between different models representing a portion of the real world at decreasing detail, while maximizing information content with respect to a given application. Figure 2 shows how transitions take place in three different areas along the database and map production workflow. The terminology used here was originally developed for the German ATKIS project (Grünreich 1992), but has since been adopted by other authors:

- as part of building a primary model of the real world (a so-called digital landscape model = DLM) – also known as *object generalization*
- as part of the derivation of special-purpose secondary models of reduced contents and/or resolution from the primary model – also known as *model generalization* (Also termed model-oriented, or statistical (database) generalization by different authors; cf. Weibel 1995b)
- as part of the derivation of cartographic visualizations (digital cartographic models = DCM) from either primary or secondary models – commonly known as *cartographic generalization*

Let us take a closer look at the scope and the objectives of these three generalization types.

Object generalization takes place at the time of defining and building the original database, called 'primary model' in Figure 2. Since databases are abstract representations of a portion of the real world, a certain degree of generalization (in the sense of abstraction, selection, and simplification) must take place, as only the subset of information relevant for the intended use(s) is represented in this database. Although seen from the perspective of generalization here, this operation is sufficiently explained by methods of semantic and geometric data modeling (define the relevant object classes and their attributes), as well as sampling methods (define the sampling strategy and its resolution), combined with human interpretation skills (e.g., if photogrammetric data capture is used). This survey will therefore not go to any further detail regarding object generalization.

While the process of object generalization had to be carried out in much the same way when preparing data for a traditional map, *model generalization* is new and specific to the digital domain. In digital systems, generalization can affect directly the map data, and not the map graphics alone. The main objective of model generalization is *controlled data reduction* for various purposes. Data reduction may be desirable for reasons of computational or storage efficiency in analysis functions, but also in light of data transfer via communication networks. It may further serve the purpose of *deriving datasets of reduced accuracy and/or resolution.* This capability is particularly useful in the integration of data sets of heterogeneous resolution and accuracy as well as in the context of multi-resolution databases. While model generalization may also be used as a *preprocessing* step to cartographic generalization, it is important to note that it is not oriented towards graphical depiction, and thus involves no artistic, intuitive components. Instead, it encompasses processes which can be modelled completely formally (Weibel 1995b); these may, however, have aestethic consequences on subsequent cartographic generalization. Model generalization is discussed further in Molenaar (1996a) and Weibel (1995b); an example of an algorithm for this class of generalization functions is presented in Section 9.

Cartographic generalization is the term commonly used to describe the generalization of spatial data for cartographic visualization. It is this the process that most people typically think of when they hear the term 'generalization'. The difference to model generalization is that it is aimed at generating visualizations, and brings about graphical symbolization of data objects. Therefore, cartographic generalization must also encompass operations to deal with problems created by symbology (cf. Fig. 1) such as object displacement, which model generalization does not. The objectives of digital cartographic generalization remain basically the same as in conventional cartography (cf. Subsection 2.1). However, technological change has also brought along new tasks with new requirements such as interactive zooming, visualization for exploratory data analysis, or progressively adapting the level of detail of 3-D perspective views to the viewing depth. The concept of cartographic generalization thus needs to be extended. On the other hand, typical maps generated in information systems are no longer complex multi-purpose maps with a multitude of feature classes involved, but rather single-purpose maps consisting of few layers. Furthermore, maps and other forms of visualizations are often presented by means of a series of different partial views in a multi-window arrangement, particularly in exploratory data analysis. Together with the capabilities of interactive direct manipulation these new forms of cartographic presentations may partially alleviate (but by no means eliminate) some of the generalization problems, or at least make them less salient for many GIS users.

3.2 Generalization Strategies

A second distinction of different views of generalization can be made with respect to the strategy used for developing digital generalization capabilities.

Process-oriented view – Deriving generalizations from a detailed database. A *process-oriented view* understands generalization as the *process* of obtaining through a series of scale and purpose-dependent transformations a database or map of reduced complexity at *arbitrary* scale or resolution, starting from a *detailed* database. As was already mentioned, generalization is a complex process, and indeed, complete solutions for all the transformation operations necessary to achieve comprehensive automated generalization largely remain to be developed.

Representation-oriented view – Multi-scale databases. A more pragmatic approach is to develop multi-scale databases in analogy to the scale series

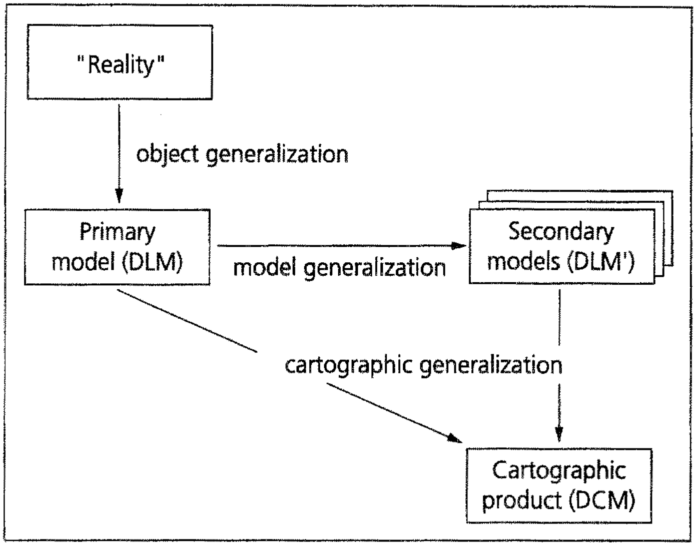

Fig. 2. Generalization as a sequence of modeling operations (modified after Grünreich 1985).

used in national topographic maps. We term this approach the *representation-oriented view*, because it attempts to develop databases that integrate single representations at different scales into a consistent multi-scale representation. Instead of devising the methods necessary to achieve the processes for transforming one level of scale into the next smaller one, scale transitions between different levels are formally coded. As one might expect, techniques of multiple representation spatial databases are needed to develop this strategy. Examples of this approach include van Oosterom and Schenkelaars (1995), Kidner and Jones (1994) and Devogele et al. (1996), to name but a few. While the representation-oriented strategy certainly overcomes the problems of missing generalization methods, it poses maintenance problems. If updates (e.g., insertions or deletions of objects) take place at a certain scale level, inconsistencies are easily introduced if they cannot be automatically propagated to all other levels – which in turn would require generalization functionality. Since both approaches have their advantages and disadvantages, it is probably safe to say that they should be exploited in conjunction during the next few years, with a gradual shift towards process-oriented generalization, since it offers more flexibility.

To summarize, this survey will mainly focus on cartographic generalization (with one exception in Section 9) and on methods to implement a process-oriented strategy towards automated generalization. *In the remainder of this survey, 'generalization' will therefore denote process-oriented cartographic generalization.*

4 Conceptual Frameworks of Generalization

In order to render a complex and holistic process such as cartographic generalization amenable to automation, conceptual frameworks need to be developed. Such theoretical models must be capable of describing the overall process and must at the same time allow to identify essential process components and steps. Of the many conceptual frameworks proposed in the literature, we briefly describe the models by Brassel and Weibel (1988) and by McMaster and Shea (1992). The latter authors discuss further models.

4.1 The Model by Brassel and Weibel

Brassel and Weibel (1988) proposed a conceptual framework of cartographic generalization which attempts to identify the major steps of the manual generalization process and transpose these concepts into the digital realm. The model departs from a view of generalization as an intellectual process which explicitly structures experienced reality into a number of individual entities, and which then selects important entities and represents them in a new form.

Figure 3 shows a schematic outline of the model. The source database is first subjected to a process termed *structure recognition* (a). This step aims at the identification of objects or aggregates, their spatial and semantic relations and the establishment of measures of relative importance (i.e., a priority order). Structure recognition is of overriding importance to the entire map generalization process, as it establishes the foundation upon which all other steps build. It is governed by the generalization controls (map purpose, source and target scale, quality of the source database, symbol specifications, minimal dimensions, etc.). Traditionally, this step is performed by visual inspection and intellectual evaluation of the map by an experienced cartographer. In the digital domain, it is possible to use methods for cartometric analysis (i.e., automated measurement in maps), but a comprehensive automation of structure recognition is non-trivial. Once the prevalent structures of the source database are known, the relevant generalization processes can be defined in *process recognition* (b). This involves the identification of both the types of data modifications (process types) and the parameters controlling these procedures (process parameters) necessary to yield the desired target map. This step too is influenced by the generalization controls. Next is *process modeling* (c), which compiles rules and procedures from a process library. Digital generalization takes place with *process execution* (d), where the rules and procedures are applied to the source database in order to create the generalized target database. As a last process, *data display* (e) converts the target database into a fully symbolized target map.

The main contribution of this model at the time of its publication was the distinction of steps leading to a characterization of the contents and structure of the source database (steps a, b, c) from operational, mechanical steps (d, e). The analysis of the shape and structure of map elements is thus made explicit. As we will see in the remainder of this survey, a weakness found in most of the basic algorithms discussed in Sections 7 to 9 is that they take the opposite approach,

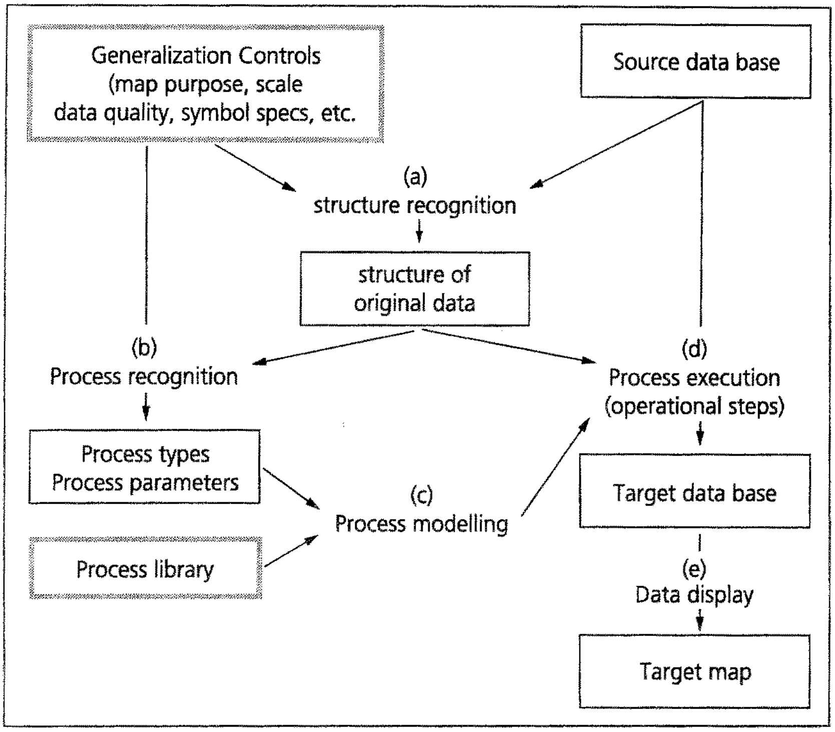

Fig. 3. The conceptual framework by Brassel and Weibel. Slightly modified after Brassel and Weibel (1988). Note that the term 'process' is equivalent to 'operator' as defined by McMaster and Shea (1992).

'hiding' shape characterization in built-in, implicit heuristics of generalization algorithms. As a result, the effectiveness and flexibility of such algorithms is limited.

4.2 The Model by McMaster and Shea

The model by Brassel and Weibel (1988) was extended by McMaster and Shea (1992), who added some missing parts and specified details for model components which were previously defined only in general terms. The resulting conceptual framework was therefore termed a comprehensive one by the authors. It decomposes the generalization process into three operational areas: (1) a consideration of the philosophical objectives of *why* to generalize; (2) a cartometric evaluation of the conditions which indicates *when* to generalize; and (3) the selection of appropriate spatial and attribute transformations which provide the techniques on *how* to generalize (McMaster and Shea 1992: 27).

Figure 4 gives a complete overview of the components of this conceptual model. McMaster and Shea (1992) first present a detailed treatment of the *philo-*

sophical objectives (why to generalize) which have been discussed here in a similar way in Subsection 2.3.

The second area of the McMaster and Shea model, *cartometric evaluation* (when to generalize) is essentially equivalent in scope to the key steps in the framework by Brassel and Weibel (1988) called structure recognition and process recognition. 'Spatial and holistic measures' are made available to characterize the source data by quantifying the density of object clustering, the spatial distribution arrangement, the length, sinuosity, and shape of map objects, and so on. These measures then serve to evaluate whether critical 'geometric conditions' are reached which trigger generalization, such as congestion (crowding) of map objects, coalescence of adjacent objects, conflicts (e.g., overlap), imperceptible objects (e.g., objects that are too small to be clearly visible), etc. Process recognition, as specified in the Brassel and Weibel model, is covered by 'transformation controls' in order to select appropriate operators, algorithms, and parameters to resolve the critical geometric conditions.

Finally in the third area, *spatial and attribute transformations* (how to generalize), a list of twelve 'generalization operators' (cf. 4.3) is proposed, sub-divided into ten operators performing spatial transformations – simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, exaggeration, enhancement and displacement – and two operators for attribute transformations – classification and symbolization. This area may be thought of as being equivalent to the 'process library' in the Brassel and Weibel model. The definition of a useful set of operators is of particular interest in the conceptual modelling of generalization, and deserves further discussion in the next paragraph.

4.3 A Closer Look at Generalization Operators

The overall process of generalization is often decomposed into individual subprocesses. Depending on the author, 'operator' may be used, or other terms such as 'operation' or 'process'. Cartographers have traditionally used terms such as 'selection', 'simplification', 'combination' or 'displacement' to describe the various facets of generalization, examples of which are the definitions given in Subsection 2.2. In the digital context, however, a functional breakdown into operators has obviously become even more important, as it clarifies identification of constituents of generalization and informs the development of specific solutions to implement these sub-problems. Naturally, given the holistic nature of the generalization process, this reductionist approach is too simple, as the whole can be expected to be more than just the sum of its parts, but it provides a useful starting point for understanding a complex of diffuse and challenging problems.

Owing to the importance of the functional decomposition of generalization various authors (e.g., Hake 1975, McMaster and Shea 1992, Ruas and Lagrange 1995, Ruas 1995a) have proposed typologies of generalization operators, each of them intended to comprehensively define the overall process. Unfortunately, no consensus has yet been reached on an all-encompassing set of operators. Even worse, authors may use different definitions for the same term or use different

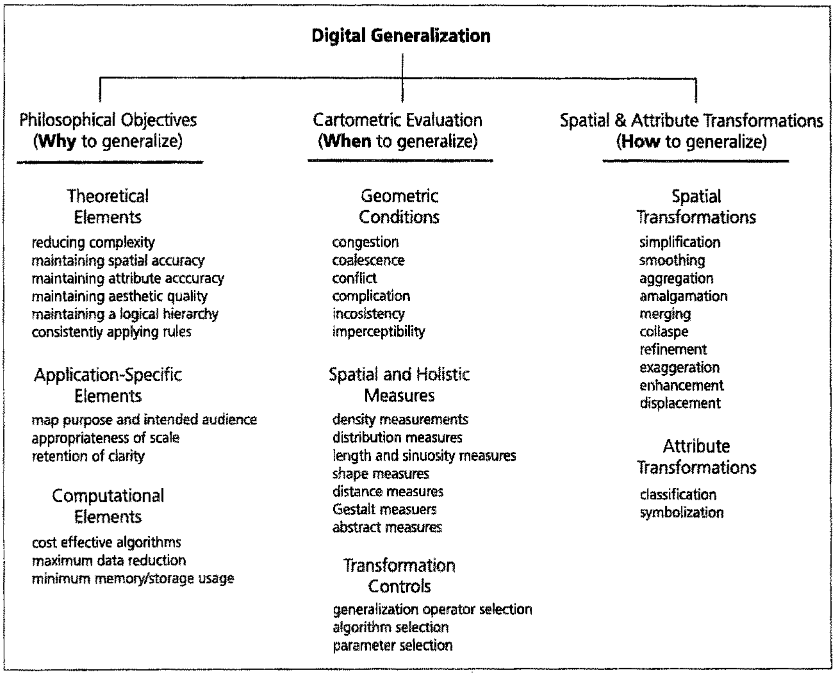

Fig. 4. The conceptual framework of digital generalization by McMaster and Shea (1992).

terms for the same definition, as a recent study by Rieger and Coulson (1993) has shown.

McMaster and Shea's (1992) typology is the first detailed one which also attempts to accommodate the requirements of digital generalization, spans a variety of data types including point, line, area and volume data. Still, closer inspection of this set of operators reveals that some fundamental operators are missing (e.g. selection/elimination) and that the definitions of some operators are perhaps not sufficiently clear (e.g. refinement) or overlapping (aggregation, amalgamation, merging). This has led other authors (e.g. Ruas and Lagrange 1995, Plazanet 1996) to extend this classification by adding operators and by refining definitions of existing ones. The composition of a comprehensive set of generalization operators is still the subject of an on-going debate; it is hoped that having it would assist the development of adequate generalization algorithms as well as their integration into comprehensive workflows.

No matter what set of operators is defined, however, the relationship between generalization *operators* and generalization *algorithms* is hierarchical. An operator defines the transformation that is to be achieved; a generalization algorithm is then used to implement the particular transformation. Commonly, several algorithms are possible for each operator. In particular, a wide range of different algorithms exists for line simplification in vector mode (cf. Subsection 7.2).

5 Generalization – The Role of Algorithms

It should by now have become clear from the above discussion that generalization is a complex process. What makes generalization so particularly hard to treat is not only the complexity of geometric problems involved but also the fact that the objectives are often ill-defined, owing to subjective, intuitive elements of cartographic design. Note that this is not the case in model generalization (which is non-graphical in nature), but it certainly holds for cartographic generalization which forms the focus of this survey and also the major thrust of generalization research.

So, what is the role of algorithms if we are trying to solve problems whose objectives are so weakly defined? One consequence is that in terms of meeting the functional objectives we may not expect to develop optimal algorithms, but only plausible ones. Another effect is that algorithms are probably not the only approach that should be used to tackle the problem comprehensively.

Knowledge-based methods are often mentioned as an alternative to algorithms. Yet, a look at the history of research in cartographic generalization reveals that neither algorithmic methods (Lichtner 1979, Leberl 1986) nor knowledge-based techniques such as expert systems (Fisher and Mackaness 1987, Nickerson 1988) have been capable of solving the problem comprehensively. While the former suffered from a lack of flexibility (since they are usually designed to meet a certain task) and from weak definition of objectives, the development of the latter was impeded by the scarcity of formalized cartographic knowledge and the problems encountered in acquiring it (Weibel et al. 1995). More recent research has therefore concentrated on approaches that more closely follow the decision support system (DSS) paradigm, a strategy often used to solve ill-defined problems. A particular approach along this vein builds on the integration of algorithmic and knowledge-based techniques and has been termed *amplified intelligence* (Weibel 1991).

As visualization and generalization are essentially regarded as creative design processes, the human is kept in the loop: key decisions default explicitly to the user, who initiates and controls a range of algorithms that automatically carry out generalization tasks (Fig. 6). Algorithms are embedded in an interactive environment and complemented by various tools for structure and shape recognition giving cartometric information on object properties and clustering, spatial conflicts and overlaps, and providing decision support to the user as well as to knowledge-based components. Ideally, interactive control by the user reduces to zero for tasks which have been adequately formalized and for which automated solutions could be developed.

In such a setup, ***algorithms*** serve the purpose of implementing tasks for which sufficiently accurate objectives can be defined:

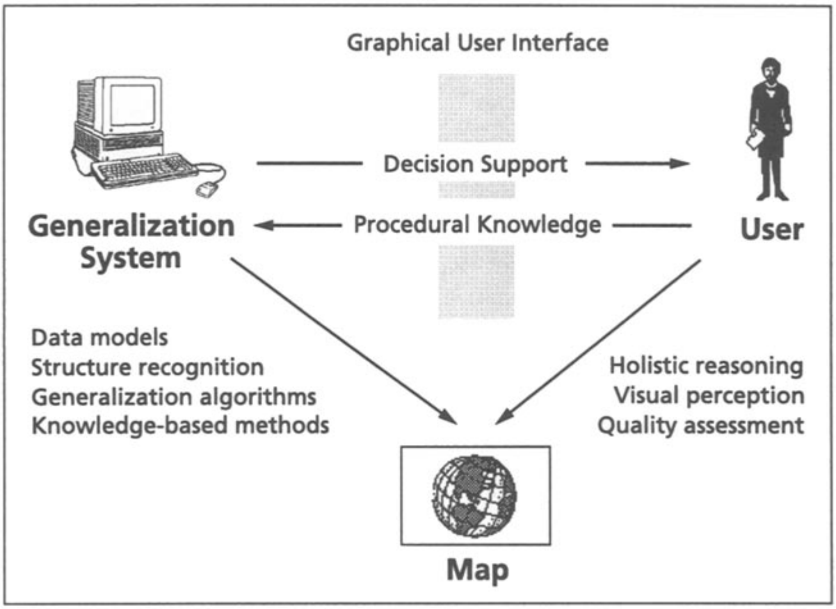

Fig. 5. The concept of amplified intelligence for map design and generalization.

- *generalization operators:* selection, simplification, aggregation, displacement, ...
- *structure recognition* (cf. Subsection 4.1): shape measures, density measures, detection of spatial conflicts, ...
- *model generalization* (cf. Section 9)

Knowledge-based methods can be used to extend the range of applicability of algorithms and code expert knowledge into the system:

- *knowledge acquisition:* machine learning may help to establish a set of parameter values that control the selection and operation of particular algorithms in a given generalization situation (Weibel et al. 1995)
- *procedural knowledge, control strategies:* once the expert knowledge is formalized, it can be used to select an appropriate set and sequence of operators and algorithms and establish a strategy to solve a particular generalization problem.

In summary, an ideal system builds on a hierarchy of control levels. The human expert takes high-level design decisions and evaluates system output. Knowledge-based methods operate at an intermediate level and are responsible for selecting appropriate operators and algorithms and for conflict resolution strategies. Finally, algorithms are the work horses of a generalization and form the foundation of everything else.

6 The Choice of Spatial Data Models – Raster or Vector?

Besides underlying theoretical principles and algorithms, spatial data models and data structures form a third component of the foundation that allows to build generalization methods. The choice of spatial data model has a great impact on the way and completeness in which properties of real world objects can be digitally represented and thus also directly governs the quality achievable by algorithms that are developed on top of them (cf. Section 13).

It is common to distinguish two major classes of spatial data models available in GIS: tessellations, of which the raster model is the most widely used, and vector models, in particular the topological vector model. The two classes of data models represent quite different concepts of representing space. The raster model, as a space-primary model, has advantages in representing continuously varying phenomena (e.g., scalar fields) or regularly sampled categorical data (e.g., landuse data derived from remote sensing imagery), and it also eases the computation of distance transformations. On the other hand, object representation is lost in raster models and severe discretization problems may be caused by the tessellation structure. The vector model, as an object-primary model, basically has reversed properties. It excels in its capabilities for object representation and accurate geometric coding, but it puts an additional burden on the computation of proximity relations and makes it almost impossible to represent continuous phenomena.

The debate over the advantages and disadvantages of raster vs. vector models has been one of the most persistent disputes in GIS research for many years. It is therefore not surprising that the debate also affected generalization research. Some authors have proposed to develop specific generalization operators for raster models different from those for vector models (McMaster and Monmonier 1989). While the differences between the two forms of representation may be considerable, it is not advisable, however, to depart from the conceptual hierarchy of tasks, operators, and algorithms. It is possible to develop solutions for all operators for both vector and raster models, although obviously some operators will be easier to implement for some data structures than others. The focus should therefore be on the generalization tasks (what are the objectives that the generalization has to meet?) and on the requirements of object representation (what data model adequately represents the structure and properties of the given real world objects?). Given these requirements, a suitable set of operators and algorithms then needs to be developed and applied, using the optimal data model. In some situations this may be a raster model, in others a vector model may be better suited. Most probably, complex problems will require a combination of different data models including auxiliary data structures, with functions to convert between them. Section 12.3 presents an example of terrain generalization which uses a combination of raster models to represent the terrain surface and 3-D vector models to represent topographic structure lines. The two models are converted into each other by object extraction and interpolation processes, respectively.

For lack of space we will focus on methods based on vector data models in this survey. Raster models are predominantly used in landuse generalization (since most landuse databases are in raster format or derived from remote sensing imagery) and as auxiliary representation to ease spatial search and distance transformations (e.g., in object displacement). A review of raster-based methods can be found in Schylberg (1993) and Jäger (1991).

7 Basic Algorithms – Context-Independent Generalization

In this section, we will describe a few basic algorithms for three simple operators: *selection/elimination, simplification and smoothing*. These operators all have in common that they are applied to individual objects independently of their spatial context. For instance, objects that are close to a line that is simplified may be affected (e.g., the new line may overlap with them), but the simplification process really only relates to the line object. This kind of generalization can be termed ***context-independent generalization.*** In contrast to that, ***context-dependent generalization*** involves operators such as aggregation or displacement which can only be triggered and controlled following an analysis of the spatial context (spatial relations of objects, object density, etc.). Context-dependent operators will only be described in Sections 11 to 13.

7.1 Object Selection/Elimination

Object selection (or defined by the antonym, object elimination) may be a simple operator, but is also an effective one as it makes space available on the map by omitting objects that are deemed irrelevant for the target map. Three questions must be addressed:

- How many objects are selected?
- Which objects are selected?
- What constraints govern the selection process?

Commonly, however, objects are not only omitted but the remaining objects are also repositioned in order to maintain the visual impression of the original arrangement of objects on the source map. Object selection thus most often only represents a first step of a series of operations.

Number of objects. The first of the above questions has been addressed in the 1960s by Töpfer and his co-workers (Töpfer 1974). Empirical rules were established in extensive studies involving the comparison of published map series. The basic empirical principle derived from these studies was termed the 'Principle of Selection' or 'Radical Law' (in German: *'Wurzelgesetz'*):

$$n_T = n_S \sqrt{\frac{s_S}{s_T}} \qquad (1)$$

Given the number of objects at the source scale n_S and the denominators of the source scale (s_S) and target scale (s_T), this simple formula allows to compute the number of objects n_T that should be maintained on the target map.

Selecting specific objects. Töpfer's principle of selection has been extended in various ways by adding further terms to take into account special cases and specific feature classes, but no matter how detailed the equation is, it still does not give any indication which objects should be selected. The selection of an actual set of objects can only be carried out using object semantics. Assuming that each of the map features is characterized by a set of attributes, objects may be selected by a query on the attributes. If the number of attributes is large and/or attributes are at different scales of measurement (e.g., interval/ratio vs. ordinal), a ranking approach is more useful. The values of each attribute are ranked over all objects, and a total score is computed for each object, allowing to establish a rank order. Following that, the top n_T objects are selected.

Note that attributes are not limited to purely non-geometric properties of an object. Geometric properties may also contribute to object semantics. For instance, when selecting towns for a small scale map, it certainly makes sense to select the places with the largest population, but places at important highway junctions or remote settlements may also be retained regardless of their lack of population. In a desert, an oasis of just ten inhabitants (but with fresh water) is something to look forward to. Measures of proximity or remoteness can be derived from an analysis of the spatial distribution of objects, for instance, by analyzing the size of Voronoi regions of the map objects (Roos 1996).

Constraints to selection. The example of the preceding paragraph shows that objects semantics obviously exert an influence on object selection. Apart from that, selection may be governed by topological constraints. A typical example is the selection of edges in a graph. In a road network the logic of circulation must be maintained. Detached roads don't make sense; similarly, major road axes (e.g., an interstate highway) should be maintained all the way through.

River networks, which usually exhibit a tree-like structure, can only be pruned from the leaves (i.e., sources) towards the root (i.e., outlet). Quantitative geomorphology has developed a number of so-called stream ordering schemes (Horton 1945, Strahler 1957, Shreve 1966). These ordering schemes reflect the topological order of edges in the river tree from the sources to the outlet (Fig. 6) and can thus be usefully exploited for the generalization of river networks. Edges in the network can be selected at successively higher levels, ensuring the topological consistency of the resulting pruned tree. Of all the ordering schemes in use today, the Horton scheme has proven to be the most useful one for generalization (Rusak Masur and Castner 1990, Weibel 1992), because it combines topological order with metric properties (the longest branches in the tree are assigned the highest order).

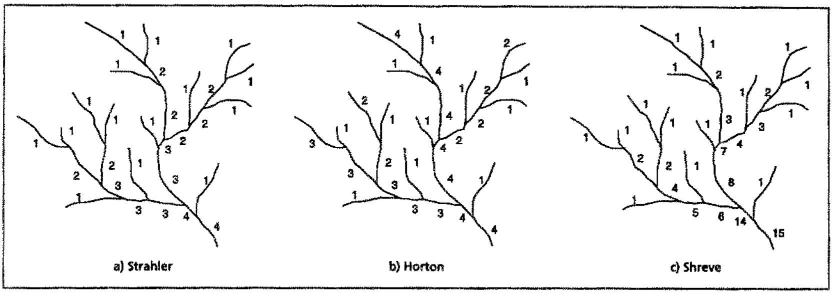

Fig. 6. Stream ordering schemes. a) Strahler order. b) Horton order. c) Shreve order.

7.2 Line Simplification

Line simplification is regarded by many as the most important generalization operator. The majority of map features are either directly represented as lines (e.g., road centerlines, streams), or form polygons which are bounded by lines (e.g., administrative regions, soil polygons, forest stands). Simplification reduces the amount of line detail and thus visibly contributes significantly to the generalization effect. If line simplification is implemented as a vertex elimination algorithm (which is the usual case), it automatically reduces data volume. Simplification algorithms are also highly useful for eliminating high frequency detail on lines digitized by continuous point sampling (stream mode digitizing) or scan-digitizing.

A seemingly countless number of line simplification algorithms has been developed over the past three decades. Commonly, simplification algorithms start with a polyline C made up of two endnodes and an arbitrary set of vertices V. C is then turned into a simplified polyline C' by reducing the number of vertices V to V', while keeping the endnodes fixed. $V/$ is thus a subset of V, and no further vertex locations are introduced nor vertices displaced (Fig. 7). The classical criteria which guide vertex elimination are the following: 1) minimize line distortion (e.g., no vertex of C' should be further away from C than a maximum error ε); 2) minimize V'; and 3) minimize computational complexity.

McMaster (1987a) and McMaster and Shea (1992) give overviews of some of the classical simplification algorithms used in cartography and GIS. Hershberger and Snoeyink (1992) and de Berg et al. (1995) list some algorithms from the computational geometry and the image processing literature. Lecordix et al. (1997) describe a comparative implementation of a wide range of algorithms in a research system. Based on the geometric extent of computation, simplification algorithms have been assigned to five categories by McMaster (McMaster 1987a, McMaster and Shea 1992):

1. Independent point algorithms
2. Local processing algorithms
3. Constrained extended local processing algorithms

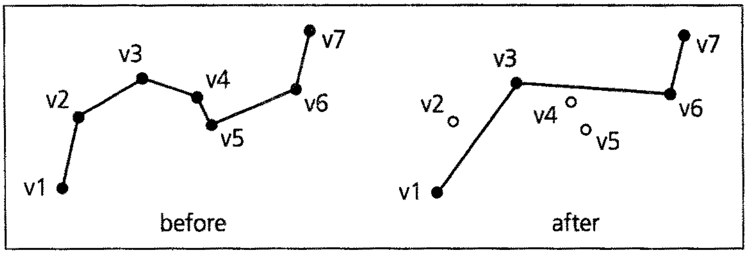

Fig. 7. Line simplification as a vertex elimination process ('weeding'). Note that although this definition is prevailing in the literature on digital generalization, it does not reflect the manual operation. In manual line drawing, simplification of the shape of a line also includes displacements along the line.

4. Unconstrained extended local processing algorithms
5. Global algorithms

Simplification algorithms may alternatively be distinguished with respect to the geometric criterion used to drive the selection of so-called critical points. Figure 8 illustrates some of these criteria, including retained length, angular change, perpendicular distance, and areal displacement.

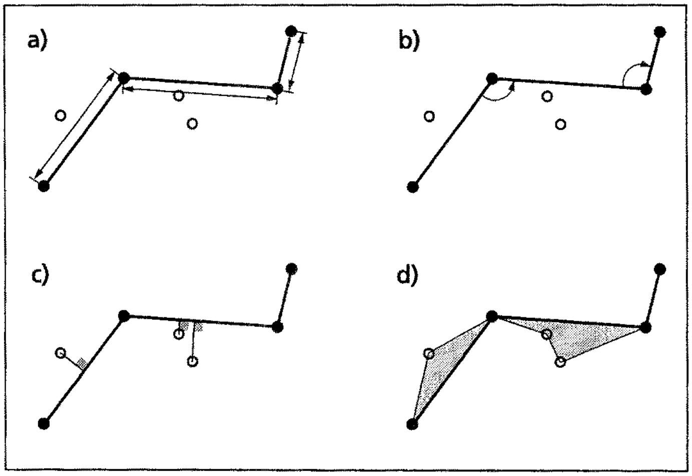

Fig. 8. Alternative geometric criteria which can be used for the selection of critical points in line simplification algorithms.

In the remainder of this subsection, we will describe a few (classical) algorithms from the area of GIS/cartography. See McMaster (1987b) and McMaster and Shea (1992) for details on related algorithms. The following discussion to some extent also reflects the historical evolution of line simplification techniques. Methods are illustrated using the same sample line shown in Figure 9.

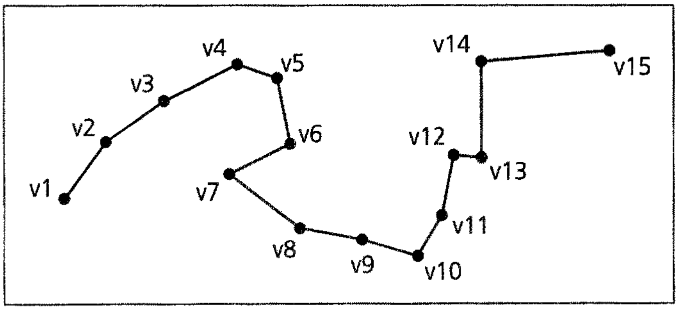

Fig. 9. The sample line used to illustrate line simplification algorithms.

Independent point algorithms. Algorithms of this class do not account for the geometric relationships with the neighboring vertices and operate independently of line topology. Examples are the n^{th} point algorithm (every n^{th} vertex of a polyline is selected, the others eliminated) as well as random selection of vertices. Obviously, these algorithms will only very rarely pick the salient vertices along a polyline (by chance) and may thus result in major line distortions. They are therefore no longer used today.

Local processing algorithms. As the name indicates, the characteristics of immediate neighboring vertices are used in determining selection/elimination of a vertex. Examples of such local criteria are the Euclidean distance between two consecutive vertices, the perpendicular distance to a base line connecting the neighbors of a vertex, as well as angular change in a vertex (McMaster 1987a). The complexity of these algorithms is linear in the number of vertices. As an empirical study has shown (McMaster 1983, 1987b), these algorithms generate less distortion than independent point algorithms, but are inferior to algorithms described below. Nevertheless, due to their localized nature they can be usefully applied for light on-the-fly weeding in line drawing.

Constrained extended local processing algorithms. Algorithms of this class search beyond immediate vertex neighbors and evaluate sections of the polyline. The extent of the search depends on a distance, angular, or number of vertices criterion. A prominent representative of this category is the Lang algorithm, which is also one of the earliest published simplification algorithms (Lang 1969). The extent of the local search is controlled by the so-called 'look-ahead' parameter in this algorithm; the amount of filtering is governed by a perpendicular distance tolerance ε. Thus, the Lang algorithm is frequently assigned to a class of algorithms termed ***tolerance band algorithms*** or ***bandwidth algorithms***, along with other algorithms utilizing a perpendicular distance tolerance to a base line (e.g., Douglas and Peucker 1973, Reumann and Witkam 1974, or Opheim 1982).

Figure 10 schematically depicts the working principle of the Lang algorithm for a look-ahead of 5 points. Perpendicular distances of intermediate vertices to a base line between the beginning point (1) and a floating endpoint (6 = starting point + look-ahead) are computed to evaluate if any vertices exceed the distance tolerance ε (Fig. 10 a) If so, a new floating endpoint is selected (Fig. 10 b) until all vertices fall within tolerance (Fig. 10 c), in which case they are eliminated. Subsequently, the last floating endpoint (4) becomes the new beginning point, and the algorithm continues.

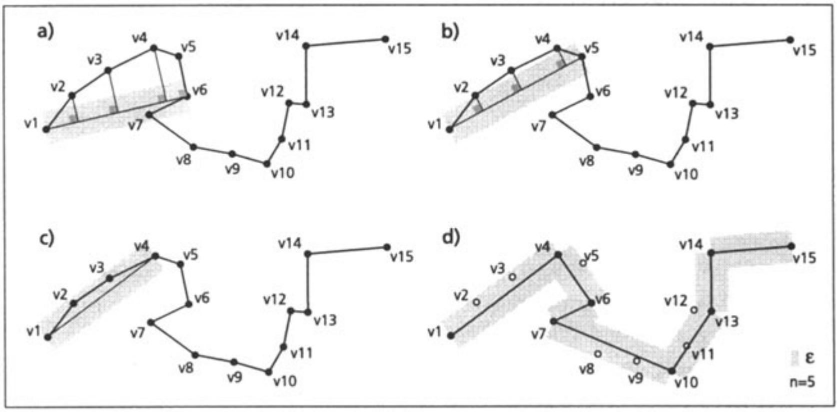

Fig. 10. The Lang algorithm. Figures **a-c** show the first iteration for a complete look-ahead. Figure **d** depicts the resulting line segments, with eliminated vertices shown in white. Note that the result of Figure **d** happens to be the same as for the Douglas-Peucker algorithm for this particular line and ε (cf. Fig. 12 d).

Unconstrained extended local processing algorithms. As with the previous class, local search extends beyond immediate neighbors of a vertex that is being tested, and sections of the line are evaluated. The extent of the search, however, is constrained by the shape complexity of the line, rather than by an arbitrary criterion. The example shown in Figure 11 depicts the algorithm by Reumann and Witkam (1974).

A search corridor made up of two parallel lines at distance ε to either side of the digitized line is extended forward until one edge intersects the digitized

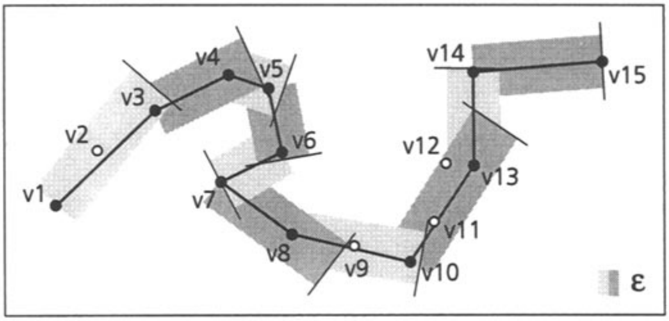

Fig. 11. The Reumann-Witkam algorithm. Vertices which were eliminated are shown in white.

line. All vertices falling within the corridor except the first and last one are eliminated. A new corridor is then extended starting with the last vertex that fell within the subsequent corridor, and the line is sequentially processed until all vertices have been tested.

Global algorithms. Global simplification algorithms consider the entire line and iteratively select critical points, while weeding out vertices within tolerance. The algorithm by Douglas and Peucker (1973) – probably one of the best known simplification algorithms – falls within this class. Just like the Lang and Reumann-Witkam algorithms, it is also a prominent representative of tolerance band algorithms. The algorithm by Visvalingam and Whyatt (1993), on the other hand, is based on an area tolerance controlling areal displacement.

Douglas-Peucker algorithm. While the Douglas-Peucker algorithm may be the line simplification method that is referenced most frequently in the literature it should be noted that nearly identical algorithms were developed independently by Ramer (1972) and Duda and Hart (1973) around nearly the same time. The method was originally developed as a weeding algorithm for removing excessive detail on digitized lines falling within the width of a source cartographic line. The algorithm is illustrated in Figure 12. It starts by connecting the two endpoints 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 e from the base line, these vertices may be eliminated and the original line can be represented by the base line. If any of the intermediate vertices falls outside e, however, the line is split into two parts at the furthest vertex and the process repeated recursively on the two parts.

Several reasons may be responsible for the popularity of the Douglas-Peucker algorithm. The global tolerance band concept makes it intuitively appealing (although the related theory was only developed *post factum*; see Peucker 1975). A very practical reason for the wide-spread use of this algorithm, however, may

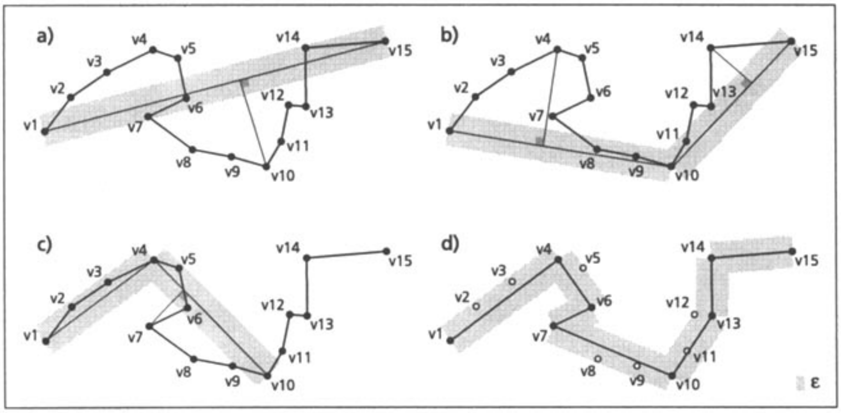

Fig. 12. The Douglas-Peucker algorithm. **a**) Initial base line with furthest vertex ($v10$). **b**) First split into two parts, again with furthest vertices ($v4$, $v14$) shown. **c**) Second split of left part. Vertices $v2$ and $v3$ are now within ε, while the second part must be split further at vertex $v7$. **d**) Corridors that were eventually generated along the original line. Vertices which were eliminated are shown in white. Note that the final result happens to be the same as for the Lang algorithm for the given line and ε.

also be found in the fact that the first author made a Fortran implementation available at an early stage, leading to the algorithm's adoption in virtually all the GIS packages on the market.

The fact that the Douglas-Peucker algorithm recursively subdivides the original line in a hierarchical fashion has been usefully exploited by other authors. Buttenfield (1985) has used the segmentation generated by this algorithm to build a strip tree that represents a compact geometric description of a line. A strip of a line segment is formed by the minimum bounding rectangle along its base (anchor) line. For each strip, geometric measures are calculated and stored. Van Oosterom and van den Bos (1989) and Cromley (1991) have independently proposed to build a tree data structure for on-the-fly generalization. Using the Douglas-Peucker algorithm, the elimination sequence and the perpendicular distance to the base line are pre-computed for all vertices except the endpoints and stored in a binary tree called the Binary Line Generalization (BLG) tree (van Oosterom and van den Bos 1989) or simplification tree (Cromley 1991), respectively. Once this data structure has been built, the retrieval of vertices, to the desired tolerance, becomes a simple tree search. Only those vertices with a perpendicular distance greater than the specified tolerance are retrieved on-the-fly from the tree for line drawing.

Visvalingam-Whyatt algorithm. McMaster (1983, 1987b) developed two classes of measures to assess the quality of line simplification algorithms similar to the geometric criteria shown in Figure 8. The first class relates to *attributes* of the cartographic line such as length, total angularity and curvilinearity; the second class includes measures which characterize the amount of *displacement* induced by simplification, expressed by the length of displacement vectors and the displacement area between the original and the simplified line. In an empirical study, tolerance band algorithms – in particular the Douglas-Peucker algorithm – showed superior performance relative to these measures (McMaster (1987b). Similar results were reported by perceptual studies involving subject testing (Marino 1979, White 1985), based on the concept of critical points as a psychological measure of curve similarity. However, it should be noted that the algorithms compared in these studies are either extremely simple techniques (e.g., nth point) or themselves representatives of the tolerance band approach. Results can therefore be expected to be biased.

Visvalingam and Whyatt (1993), point out a few deficiencies of the tolerance band approach. In particular, they argue that the selection of the furthest vertex outside the tolerance band as a critical point to be retained is unreliable because this point may be located on spikes (errors) and on minor features. In an attempt to preserve salient shapes and entire features rather than selecting specific points they present an algorithm which eliminates vertices on a line based on their effective area. The effective area E of a vertex is defined as the area of the triangle formed by the vertex and its immediate neighbors (Fig. 13). It represents the area by which the line would be displaced if the vertex was discarded.

The algorithm is simple. It makes multiple passes over the line. On each pass, the vertex with the smallest effective area is considered as least significant and removed. When a vertex is eliminated the effective areas of adjacent vertices need to be recalculated before the next pass. The algorithm repeats until all vertices except the endpoints are tagged with their effective area and their elimination sequence is recorded. The tagged vertices may then be filtered at runtime by interactive selection of the tolerance value for E. This approach of first pre-computing the elimination sequence is similar to the approach used for the Douglas-Peucker algorithm by van Oosterom and van den Bos (1989) and Cromley (1991). As the empirical study presented in Visvalingam and Williamson (1995) suggests, the Visvalingam-Whyatt algorithm indeed seems to perform better on the elimination of entire shapes (caricatural generalization), while the Douglas-Peucker method appears to be better at minor weeding (minimal simplification).

Simplification as an optimization problem. All of the above algorithms have in common that they exploit some kind of heuristic to determine which vertices along a line should be retained. These heuristics may produce adequate results in many cases, but it is difficult to say whether the result is better than that of another algorithm, let alone to determine whether it is optimal with respect to a particular geometric criterion. Only *a posteriori* empirical analysis (McMaster 1983, 1987b) can assess the geometric performance of such heuristic methods.

In reaction to this weakness of heuristic techniques, Cromley and Campbell (1991, 1992) re-formulated the line simplification problem as an optimization problem. Initially, they presented an algorithm that produces an optimal

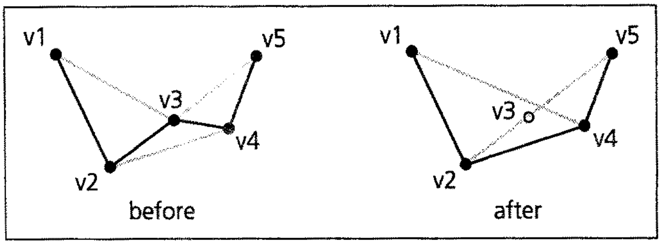

Fig. 13. The Visvalingam-Whyatt algorithm. Effective areas are computed for each vertex except the endpoints using the area of the triangle formed by each vertex and its immediate neighbors. Vertex $v3$ then is the first one to be eliminated, and the area of neighboring vertices $v2$ and $v4$ needs to be recomputed (after Visvalingam and Whyatt 1993).

simplification with respect to the tolerance band criterion using mathematical programming techniques (Cromley and Campbell 1991). This method was subsequently extended by integrating qualitative criteria such as those shown in Figure 8. Using these types of criteria, line simplification is stated as the problem of minimizing (or maximizing) a particular geometric property of a line (e.g., maximize line length, minimize areal displacement), subject to a constraint on the number of individual vertices retained in the simplified line. The maximum number of retained vertices is obtained from Töpfer's selection formula (cf. Subsection 7.11). To solve this multi-criteria problem, the digitized line is considered as a directed acyclic graph of all possible $\frac{n(n-1)}{2}$ segments which connect the n vertices (Fig. 14). Each segment is attributed a cost value c_{ij} which represents the cost of traversing the segment. Each c_{ij} corresponds to a geometric performance measure such as the line length or areal displacement associated with a line segment. Optimal line simplification is then approached as a form of shortest path problem, in which a path through the graph is to be found that minimizes the total cost.

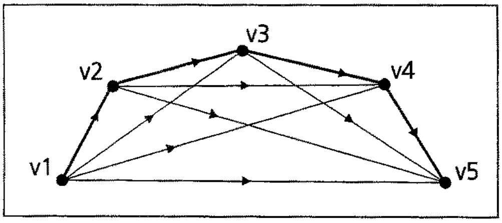

Fig. 14. Alternative paths for a simplified line connecting endpoints 1 and 5 (after Cromley and Campbell 1992).

7.3 Line Smoothing

Line smoothing in many ways forms a complement to line simplification. According to the definition commonly used, line smoothing techniques "shift the position of points in order to improve the appearance [of a line]. Smoothing algorithms relocate points in an attempt to plane away small perturbations and capture only the most significant trends of the line. Thus smoothing is used primarily for cosmetic modification" (McMaster and Shea 1992: 84-85). Figure 15 illustrates this process. Note, however, that this definition of line smoothing is currently under re-examination by several authors (Ruas and Lagrange 1995, Plazanet 1996, Weibel 1996).

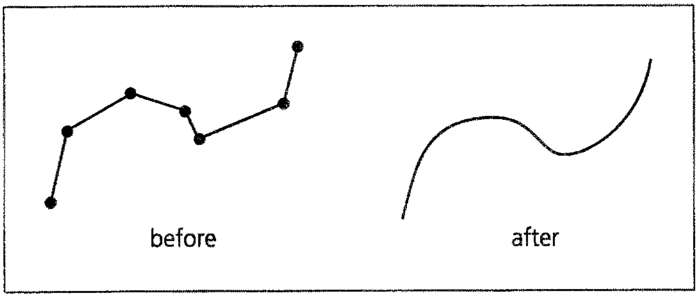

Fig. 15. Principle of line smoothing.

McMaster (1989) distinguishes three groups of smoothing algorithms:

1. Weighted averaging techniques
2. Epsilon filtering techniques
3. Mathematical approximation

Weighted averaging techniques are based on averaging of vertex coordinates. Mathematical approximation methods encompass techniques such as Gaussian filtering as well as curve fitting (Rogers and Adams 1990). We restrict the discussion to a representative of the second group. Epsilon filtering methods are based on the paradigm that generalization is a process of reducing the spatial resolution of a map such that no detail is displayed that is smaller than the smallest perceptible size ε.

Li-Openshaw algorithm. Li and Openshaw (1992) proposed a 'natural principle' of line generalization based on a concept similar to so-called Epsilon filtering (Perkal 1966), and presented three different algorithms which implement this principle: a vector-mode algorithm, a raster-mode algorithm, and a mixed raster-vector algorithm. We describe the raster-vector algorithm only, as it produces the best generalization results according to the authors.

In a first step, the size F_c of the 'smallest visible object' (SVO) at the target scale is estimated according to the formula

$$F_c = S_t D \left(1 - \frac{S_s}{S_t}\right) \tag{2}$$

where S_t is the scale factor of the target map; D is the diameter of the SVO at the target map scale (in map units), within which all information can be neglected; and S_s is the scale factor of the source map.

A local square grid with a spacing equal to F_c is then overlaid on each cartographic line, with the origin centered at the beginning node (Fig. 16). Next, intersections with the grid are calculated along the original line. In addition to the endpoints of the line, resulting points on the output line consist of the midpoints of each pair of consecutive intersections.

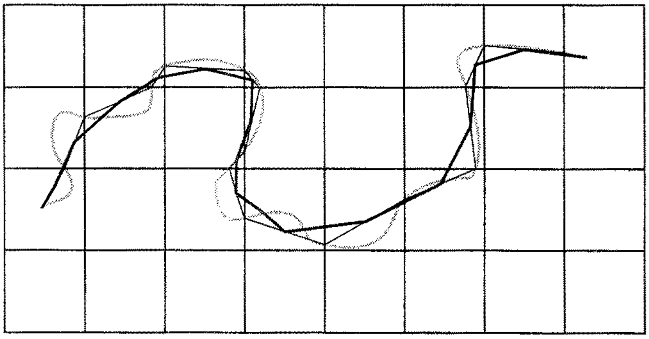

Fig. 16. The Li-Openshaw method: raster-vector algorithm. The side length of a grid cell reflects the size F_c of the smallest visible object (SVO). The heavy gray curve shows the original line. The fine line represents the segments connecting intersections of the original line with the grid. Finally, the heavy solid line represents the resulting line connecting the midpoints of the fine line segments.

8 Operator Sequencing

Subdividing the overall generalization process into individual operators alleviates the development and implementation of specific tools for generalization, allowing to address particular generalization tasks in a flexible way through a combination of different operators, algorithms, and parameter sets. On the other hand, the functional break-down into generalization operators also requires that the appropriate combination and sequence of operators (and associated algorithms) must be determined for a given generalization problem.

Unfortunately, there is no sequence of operators that is valid for all scale ranges, map purposes, and combinations of feature classes. It is possible to some extent to develop generic sequences for a particular *class* of generalization problems and product specifications (e.g., for generalization of landuse maps at medium to small scales). For each specific generalization problem, however, the operator/algorithm combination and sequence has to be fine-tuned and calibrated specifically. It is also a common fact that if two algorithms – with the same parameter values – are applied in reverse order, the result will not be the same (Monmonier and McMaster 1991, Plazanet 1996).

Research in operator and algorithm sequencing is still relatively sparse to date. Examples include Lichtner (1979) who proposed a generic sequence for large scale topographic map generalization, McMaster (1989) and Monmonier and McMaster (1991) with studies on sequential effects of line simplification and smoothing algorithms, and Lecordix et al. (1997) with empirical comparisons of line caricature algorithms. McMaster (1989) proposes a detailed procedure for generalizing linear data in which both smoothing and simplification are applied in two phases. In the first phase, smoothing precedes simplification (both using conservative parameter values) in order to remove spurious effects of digitizing. During the second phase, simplification is used for an initial generalization to target scale, with subsequent smoothing to improve the aesthetic quality of line drawing.

In today's interactive generalization systems the issue of operator and algorithm sequencing is even more important due to the large number of algorithms offered in systems of the type described by Lee (1995). Interactive systems, however, also allow to establish algorithm sequences under interactive control, with the option to fine-tune and 'train' parameter sets on representative sample data in order to subsequently apply them globally to the entire data set. Finally, some pragmatic general guidelines for operator sequences can be derived from cartographic practice:

- *Selection/elimination:* Is applied first as it eliminates insignificant details and features and increases available space.
- *Aggregation/amalgamation/merging:* These operators combine selected features and thus save space. Additionally, they induce a transition of topological type (e.g., point to area, double lines to single line, area to line) and thus must precede line processing operators (simplification, smoothing, etc.).
- *Simplification:* Reduces detail and contributes to line caricature. Should therefore be applied at an early stage.
- *Smoothing:* Contributes to aesthetical refinement. Follows simplification.
- *Displacement:* Used to resolve spatial conflicts created by previous operators.

9 Model Generalization – The Example of TIN Filtering

As was explained above model generalization functions are crucial to the development and derivation of *databases* at multiple levels of resolution. Frequently, model generalization relies on the exploitation of hierarchies which are inherent to spatial data. For instance, in the attribute (i.e., thematic) domain, the classical example of inherent hierarchies are categorical data such as land use or soil classifications. Such intrinsic relations can be formalized for storage and retrieval (Molenaar 1996b, Richardson 1994). In this section, however, we concentrate on a single example of a *geometric* model generalization process. Further examples of model generalization methods – involving aspects of thematic and temporal model generalization – can be found in the reader edited by Molenaar (1996a).

Filtering operations for data reduction are an essential component of terrain modeling systems. As more and larger datasets are being processed and new methods for high-density data collection are being put to use, the necessity for an adaptation of secondary models to the desired resolution and accuracy is becoming more urgent. For instance, it is not necessary to carry all the minute details contained in a particular model through the generation of an animated sequence if the result does not show them. TIN filtering can be desirable to eliminate redundant data points within the sampling tolerance, to detect blunders, to save storage space and processing time, to homogenize a TIN, or to convert a gridded terrain model into a TIN. It may also be used as a component of terrain generalization (Weibel 1992; cf. Subsection 12.3).

The ***objective of TIN filtering*** is to find an approximate representation of a field of elevations, that is, a bivariate function which nowhere deviates from the original surface by more than a specified tolerance Δz. TIN filtering thus essentially forms the 3-D equivalent of the simplification of plane curves (Subsection 7.2).

Van Kreveld (1997) gives further references to algorithms for TIN filtering. Our discussion focuses on the role of TIN filtering in model generalization and briefly presents a particular incremental algorithm developed by Heller (1990), termed 'adaptive triangular mesh (ATM) filtering'. It is based on a coherent approach of successive construction of Delaunay triangulations. However, the method can be used to reduce the data volume of both grids and TINs. A grid is just considered as a special case of a TIN, with nodes arranged in a rectangular grid. The general flow of the algorithm is as follows:

1. Start with an initial set of points: selected points on the convex hull and the *significant extremes.*
2. Triangulate these points to build an initial triangulation.
3. Determine the priority of the remaining points forming the initial priority queue. The priority of a point is calculated as the vertical distance to the current triangular mesh weighted by the inverse of the tolerance Δz.
4. Select the point with the largest priority and insert it into the triangulation, swapping edges of affected triangles to maintain the Delaunay criterion.
5. Readjust the priorities of the affected points.
6. Repeat steps 4 and 5 until no point remains whose vertical distance exceeds the user-specified tolerance Δz.

A few auxiliary data structures are used to achieve an efficient algorithm. The priority queue of points waiting to be inserted into the triangulation is organized in a heap. The points pertaining to each triangle are linked into a list. The insertion of a point requires a local retriangulation which consists of swapping all necessary triangles to maintain the Delaunay criterion, and readjusting the priorities of all affected points. It is obvious that the time required for retriangulation is proportional to the number of readjusted points and the logarithm of the number of queued points. Therefore, a heuristic is used to start the process with as many significant points as possible.

The set of initial points is formed by selected points on the convex hull and the *significant extremes*. The points which are selected on the hull include all consecutive hull points which are not collinear (i.e., not in line with respect to their planimetric location). Collinear points are handled specially, which is particularly important when the input points originate from a regular grid, since all points on the edge of the grid are collinear. A variant of the Douglas-Peucker algorithm is applied to the profile of collinear points using Δz as a distance tolerance. Local extremes form further candidates for the initial point set. The following definitions are used to select the *significant extremes*:

- A local minimum is considered as significant if it is the global minimum in a basin of depth greater or equal Δz.
- A local maximum is considered as significant if it is the global maximum on a hill of height greater or equal Δz.

These definitions lead to a straightforward approach for the determination of local minima and maxima. The local minima are sorted by their altitude by inserting them into a priority queue. Then, the following step is repeated until all minima in the queue are tested. The lowest remaining minimum z_i is selected, and the points in its neighborhood traversed radially until the lowest point along the 'wavefront' of this traversal is higher than $z_i + \Delta z$. If a local minimum is found in this process, it can be removed from the priority queue. As soon as a point is found which is lower than z_i, the traversal is aborted and the current minimum discarded. The same method is also used in an analogous way to determine significant maxima.

An example of ATM filtering is shown in Figure 17: starting from a gridded digital terrain model (68,731 points), a TIN with a tolerance of 5 m (with 11,450 points or 16.7remaining), and a TIN with a tolerance of 10 m (5,732 points or 8.3) were obtained.

The fact that for the determination of the significance of a point the vertical distance is weighted offers the potential for useful extensions. In the normal case, the weight is set to the inverse of Δz, and therefore constant. However, the weight can also be modified individually according to the specific properties of each point. For instance, points on structure lines can be assigned higher weights than others, thus enabling the preservation of linear structural features. In a similar way, the level of detail of perspective views can be adjusted according to viewing depth. Height values of points can be weighted according to some function that is proportional to the inverse of the distance of a point to the viewpoint (Hess 1995, Misund 1996). Points near the viewer are thus assigned higher weights than distant ones, causing more points to be removed from the TIN in distant regions which are less likely to be discernible.

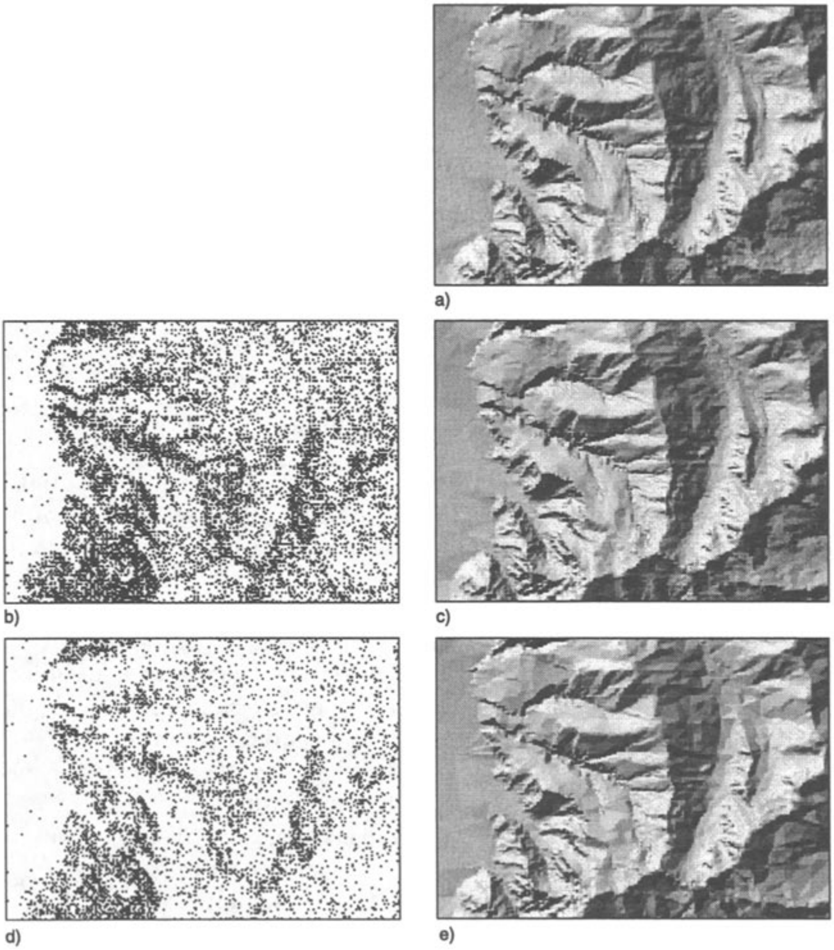

Fig. 17. Sample runs of ATM filtering for data reduction and grid-to-TIN conversion. **a)** Original grid digital terrain model (311 x 221 = 68,731 points; 25 m spacing). **b)** Remaining points (11,450 or 16.7) after ATM filtering with $\Delta z = 5$ m. **c)** Hillshading of corresponding TIN. **d)** Remaining points (5,732 or 8.3) after ATM filtering with $\Delta z = 10$ m. **e)** Hillshading of corresponding TIN. (DTM data courtesy of Swiss Federal Office of Topography, DHM25 ©1997, (1263a))

10 An Assessment of Basic Algorithms

Basic generalization operators and algorithms as discussed in Sections 7 to 9 largely represent the state of the art of available generalization tools in current commercial GIS (Schlegel and Weibel 1995) and also form the core of special-purpose generalization systems such as Intergraph's MGE Map Generalizer (Lee 1995, Weibel and Ehrliholzer 1995). A number of deficiencies have been observed and documented in the literature (Muller 1990, Beard 1991, Plazanet et al. 1995, de Berg et al. 1995, Weibel and Ehrliholzer 1995, Lecordix et al. 1997). This section presents a brief assessment of basic generalization methods, attempting to identify weaknesses as well as key areas for future research. Note that the discussion primarily focuses on *functional* deficiencies and improvements, rather than on aspects of computational efficiency. In an evaluation of the quality of today's generalization methods computational efficiency is only secondary to a functional assessment. Many methods just don't do what they are expected to do, so producing garbage fast is not really an objective. However, we certainly appreciate the importance of computationally efficient methods in the context of interactive generalization and databases of increasing size.

10.1 What's Wrong with Basic Algorithms?

Based on the study of the above literature as well as empirical investigations (Schlegel and Weibel 1995, Weibel and Ehrliholzer 1995) we have identified a number of weaknesses of basic generalization methods with respect to algorithms and data structures, which can be summarized as follows:

- Independent processing of individual features neglects spatial context.
- Structure and shape recognition for the characterization of map objects is restricted to simple heuristics (such as the tolerance band). It is not explicitly represented in terms of shape measures and spatial relations.
- Algorithms are unspecific; they are not tailored to the properties of specific feature classes (e.g., simplification of building outlines).
- Algorithms to implement context-dependent operators (displacement, amalgamation, aggregation, caricature, etc.) are largely missing.
- Feature representations and data structures commonly used offer little support for structure recognition and context-dependent operators.

10.2 What Should Be Improved?

The necessary improvements of basic algorithmic methods and the development of more advanced algorithms basically fall into three (strongly interrelated) areas:

- *Constraint-based methods:* Algorithms must observe the spatial and semantic constraints imposed by map context.
- *Methods for structure and shape recognition:* Structure recognition must be made explicit. Analysis of shape and structure of map features must precede the execution of generalization algorithms. It is necessary to select an appropriate set of operators, algorithms and parameter values.
- *Alternative data representations and data models:* Generalization requires a rich data model encompassing a combination of different data representations and auxiliary data structures.

In the remainder of this survey, we briefly discuss selected representatives of topical work for each of these areas, rather than presenting a comprehensive review of current research. Note that beyond the problems of algorithmic nature, further deficiencies can be observed with non-algorithmic issues which prompt an equally strong need for future research:

- *Knowledge-based methods:* There is a lack of procedural knowledge in generalization, and knowledge acquisition (KA) has proven to be a major bottleneck. New methods for KA must be developed, including techniques of computational intelligence (Weibel et al. 1995). Integration of knowledge-based and algorithmic techniques is also a major issue.
- *Quality assessment:* Criteria and methods (quantitative and qualitative) for the assessment of the quality of generalization methods are largely missing. Development of criteria and measures and evaluation methods to implement them are required (Weibel 1995b).
- *Human-computer interaction:* Current user interfaces are not designed specifically for generalization. Optimized user interfaces, strategies of sharing the responsibility between system and user must be developed.
- *Practical issues:* In commercial GIS, there is still a problem with the adoption of results from advanced research (the Douglas-Peucker algorithm is frequently the only method offered). Also, current systems often offer little decision support to the user, low quality graphics function (e.g., cartographic drawing), cryptic GUIs, etc.

11 Constraint-Based Methods

Context-independent generalization algorithms as outlined in Sections 7 to 9 exhibit a fundamental problem: they process each map object individually, neglecting the context which the object is embedded in. Most basic algorithms concentrate purely on metric criteria and even the simplest topological or semantic constraints are ignored. As a result, lines may intersect with themselves, with other lines nearby, or points may fall outside polygons, to name but a few of the most frequent problems (Muller 1990, Beard 1991, de Berg et al. 1995, Fritsch and Lagrange 1995).

In terms of the development of methods that can satisfy additional non-metric constraints, the simplification of polygonal subdivisions has recently attracted research interest. Polygonal subdivisions are a frequent data type in GIS applications (political boundaries, vegetation units, geological units, etc.) and present particular problems to basic line simplification algorithms (Fig. 18). Weibel (1996) has attempted to identify the constraints that govern polygonal subdivision simplification, proposing a typology of metric, topological, semantic and Gestalt constraints and reviewing relevant previous research. Two basic alternatives exist to resolve problems such as the ones illustrated in Figure 18: 1) the problems are cleaned up in a post-processing operation or 2) the simplification algorithm incorporates the corresponding constraints and thus avoids the problems in the first place. Muller (1990) has presented a post-processing method to remove self-intersections created by spurious line simplification algorithms. Intersections are detected and affected vertices displaced to eliminate the problem. Alternatively, de Berg et al. (1995) proposed an algorithm for the simplification of chains of polygonal subdivisions that extends the basic simplification techniques and satisfies four different constraints. If C is a polygonal chain and P a set of points that model special positions inside the regions of the map (e.g., cities in countries), then it is required from its simplification C':

1. No point of the chain C has a distance to its simplification C' exceeding a prespecified error tolerance.
2. C' is a chain with no self-intersections.
3. C' may not intersect other chains of the subdivision.
4. All points of P lie to the same side of C' as of C.

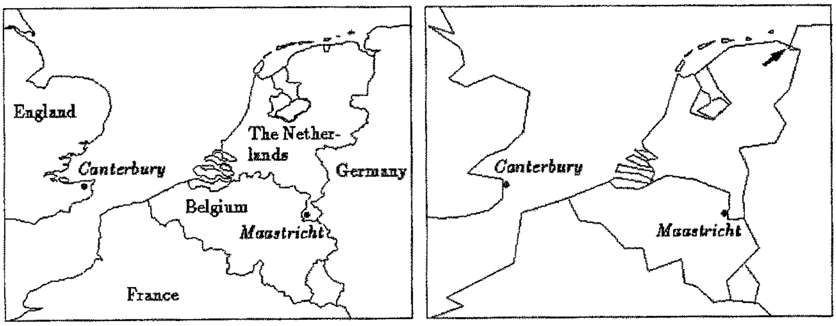

Fig. 18. An example of an inconsistent simplification of a subdivision (source: de Berg et al. 1995).

Instead of trying to satisfy all conditions at once, each condition is dealt with individually and the final result extracted from the combination of partial solutions. A polygonal chain is understood as a directed acyclic graph G, with vertices v_i forming the nodes of the graph. Each line segment, called a shortcut, that is valid relative to a particular condition is added to G. For each of the four conditions, a separate graph $G_1, \ldots, G_4$ is created. The final graph G is built from shortcuts that are allowable in all graphs G_i representing the partial solutions. In G, the resulting minimum vertex simplification of the polygonal chain C is found as the shortest path between the endpoints.

In order to build the graph G_1 that satisfies the first condition, the algorithm by Imai and Iri (1988) is used. In the version of the paper published in de Berg et al. (1995) the input chains are required to be x-monotone. The second condition is thus met automatically as no self-intersections can occur in x-monotone chains, and G_2 is equivalent to G_1. The solutions for the third and fourth condition can be combined: Vertices of the chains of polygons adjacent to C are added to point set P. G_4 thus need not be established. Furthermore, only the points falling inside the convex hull of the chain C being simplified could possibly end up to the wrong side of C'. The actual number of candidate points in P can thus be reduced further. The algorithm for determining consistent shortcuts (with respect to the locations of points in P) leading to G_3 is described in detail in de Berg et al. (1995).

De Berg et al. (1995) have shown that their algorithm for simplifying a polygonal subdivision with N vertices and M extra points runs in $O(N(N+M)\log N)$ time in the worst case. Empirical studies with real data will need to establish whether this close to quadratic time behavior actually shows up.

12 Methods for Structure and Shape Recognition

12.1 Motivation and Objectives

As Section 4 discussed, structure and shape recognition (i.e., cartometric evaluation) is logically prior to the application of generalization operators (Brassel and Weibel 1988, McMaster and Shea 1992). Structure recognition allows to determine when and where generalization needs to be applied and furnishes the basis for the selection, sequencing, and parametrization of an appropriate set of generalization operators for a given problem. Cartographic data often are relatively unstructured. Entity definition in most spatial databases stops at the level of individual map features; parts of features (e.g., a hairpin bend on a road or an annex of a building) are rarely coded explicitly. Most spatial databases also contain little semantic information in terms of the relative importance of individual map objects – which is crucial in generalization since the purpose is to distinguish between important and insignificant features. Finally, little information is normally stored on shape properties of map features. The *objectives of structure recognition* are therefore the following:

- to structure the source data according to the requirements of the intended generalization
- to 'enrich' the source data
- to derive secondary metric, topologic and semantic properties:
  - metric: shape characteristics, density, distribution, object partitioning, proximity relations
  - topologic: topologic relations not represented in the source data
  - semantic: relative importance (priority) of map objects, logical relations between objects

12.2 Characterization and Segmentation of Cartographic Lines

Our first example of structure recognition is concerned with the analysis of linear data. Since such a great share of cartographic data are of linear type, this task is of considerable importance. As Figure 19 shows, analysis and segmentation of cartographic lines into meaningful components is essential in order to decide how individual shapes need to be treated during generalization. Depending on shape properties such as sinuosity, but also depending on context information, different operators and algorithms may be applied.

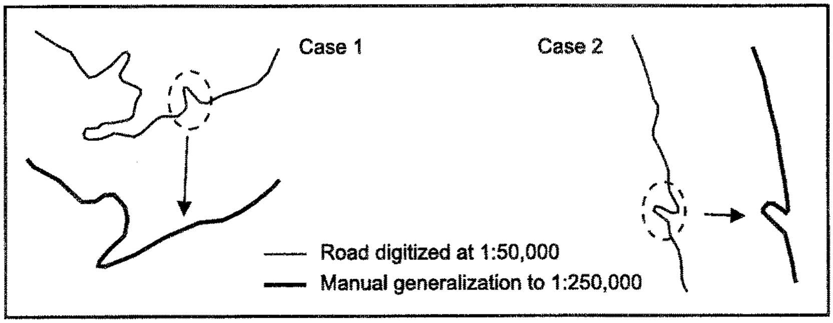
Case 1
Case 2
Road digitized at 1:50,000
Manual generalization to 1:250,000

Fig. 19. Example of different generalization alternatives for the same shape (after Plazanet 1995).

First attempts at cartographic line characterization and segmentation have been made by Buttenfield (1985) who based her segmentation procedure on the partitioning scheme resulting from the Douglas-Peucker simplification algorithm (cf. Subsection 7.2). Recently, an approach which is not biased towards a particular generalization algorithm has been presented by Plazanet (Plazanet 1995, Plazanet 1996, Plazanet et al. 1995). The procedure generates a hierarchical segmentation of the line according to a *homogeneity* criterion. The resulting tree structure is called the *descriptive tree.* Figure 20 illustrates such a tree. The homogeneity of the individual sections is intuitively apparent. The homogeneity definition used to split up the line is based on the variation of the distances between consecutive inflection points. Inflection points where the sign of the difference between the mean of distances and the distance between the current and consecutive inflection point changes are considered as ‘critical points’ for segmentation (Fig. 21).

The objective of segmentation is mainly to obtain sections of the line that are geometrically sufficiently homogeneous to be tractable by the same generalization algorithm and parameter values. Further information may be added to the descriptive tree which characterizes the *sinuosity* (and thus the prevailing geometric character) of each line section. A variety of measures can be obtained from the deviation of the cartographic line from a trend line formed by the *base line* connecting consecutive inflection points. These measures are calculated for each individual bend (Fig. 22) and then averaged to yield the values for the corresponding line section. Using the proposed segmentation procedure and shape measures, lines can be segmented and the resulting line sections classified

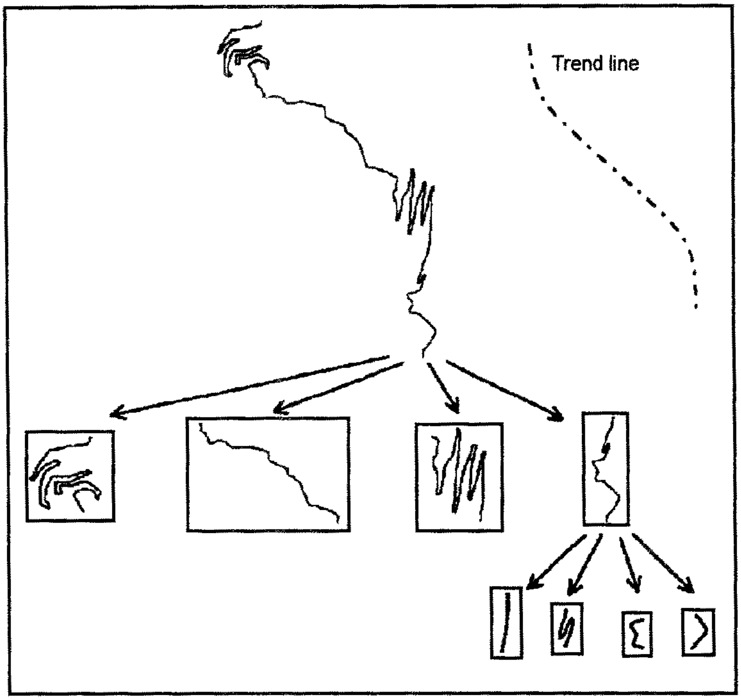

Fig. 20. An example of line segmentation (courtesy of C. Plazanet, IGN France).

according to their degree of sinuosity (e.g., using cluster analysis). Results of classification can be found in Plazanet (1995), Plazanet et al. (1996) or Plazanet (1996).

12.3 Terrain generalization

Our second example of structure recognition is intended to show the use of structural and shape information in terrain generalization. Several techniques for terrain generalization have been reported in the literature (see Weibel 1992 for a brief review). Weibel (1992) proposes three classes of generalization methods which are integrated into a common strategy whose purpose it is to select the appropriate method based on an analysis of the character of the terrain represented by the digital terrain model (DTM). This initial analysis step is termed *global structure recognition.* It is similar in nature to the approach for 2-D line characterization by Plazanet (1995, 1996) in that it segments the continuous terrain surface into patches of homogeneous terrain character based on the computation of a set of geomorphometric measures for each DTM point.

The three classes of terrain generalization methods are termed global filtering, selective filtering and heuristic generalization. *Global filtering* consists of a variety of smoothing filters (in the spatial and frequency domain), combined

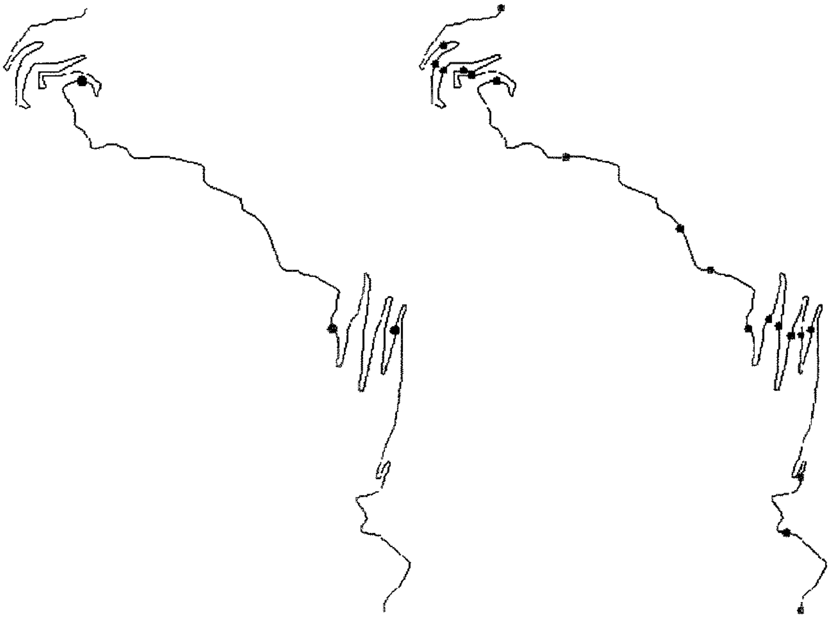

Fig. 21. Right: Detected inflection points. Left: Critical points retained automatically (courtesy of C. Plazanet, IGN France).

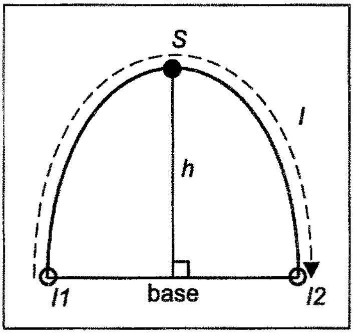

Fig. 22. Sinuosity measures (after Plazanet 1995).

with filters for enhancement. It is intended for use in smooth, rolling terrain and minor scale changes. *Selective filtering* is equivalent to ATM filtering described in Section 9. It is both aimed at cartographic generalization (minor scale change in rugged terrain) and model generalization (for the purposes outlined in Section 9). *Heuristic filtering* is potentially the most flexible method as it is the only approach which is based on explicit structure recognition. It is intended for use in complex fluvially eroded terrain.

Heuristic terrain generalization is based on an emulation of principles used in manual cartography. Manual cartographers use sketches of structure lines (drainage channels, ridges, and other breaklines) as a skeletal representation of the continuous terrain surface in order to guide the generalization process. Heuristic terrain generalization is thus preceded by a step called *local structure recognition*. This process yields a model called the *structure line model (SLM)*. The networks of drainage channels and ridges are extracted from the DTM, as well as the associated area features (i.e., drainage basins and hills) and additional significant points (for a review of relevant algorithms, see van Kreveld 1997). All features are tagged with descriptive geomorphometric attributes. Note again the similarity of the SLM approach to the 2-D descriptive tree of Plazanet (1995, 1996).

This rich information of the SLM is then used as a 3-D skeletal structure of the DTM and forms the basis of heuristic generalization operations. The 3-D skeleton of the SLM can be generalized by eliminating and simplifying network links, resulting in a derived version of the SLM. The reduced network still represents a 3-D, though simplified, structure. Thus, the generalized surface can be reconstructed by interpolation from the generalized structure lines. Figure 23 shows a sample terrain surface (Fig. 23 a) which was generalized using this procedure (Fig. 23 b). Compared to the original model, the derived model has been modified in many ways. Smaller landforms have been dropped or combined as a result of the elimination process. Detail has generally been reduced, the major structures have been retained, though.

Figure 23 c) shows a further processing step. By means of an interactive DTM editor, the essential landforms have been retouched and enhanced. As a consequence, the map image has become clearer and more expressive. Thus, the important structures remain clearly discernible even at small scales, as the reduced versions of Figure 23 c) demonstrate. The interactive editor used for final retouching was developed in a project that focused on dynamic modification of DTMs (Bär 1995). This editor offers a range of tools which allow to modify the surface of a grid DTM in real-time, in close analogy to tools of daily life such as spatula and iron. The individual tools are implemented as local filters in the spatial domain that can both be controlled interactively and be guided automatically along predefined lines. For any tool, its size, basic shape (footprint) and the degree of cutting and filling can be specified.

13 Alternative Data Representations and Data Models

The evaluation of geometric data structures given in Subsection 10.1 can be summarized as follows: Generalization algorithms cannot be expected to advance much beyond their current state unless certain limitations of presently used geometric data structures are overcome. That is, alternative schemes of data modeling and representation must be exploited. The problem must be addressed at two levels:

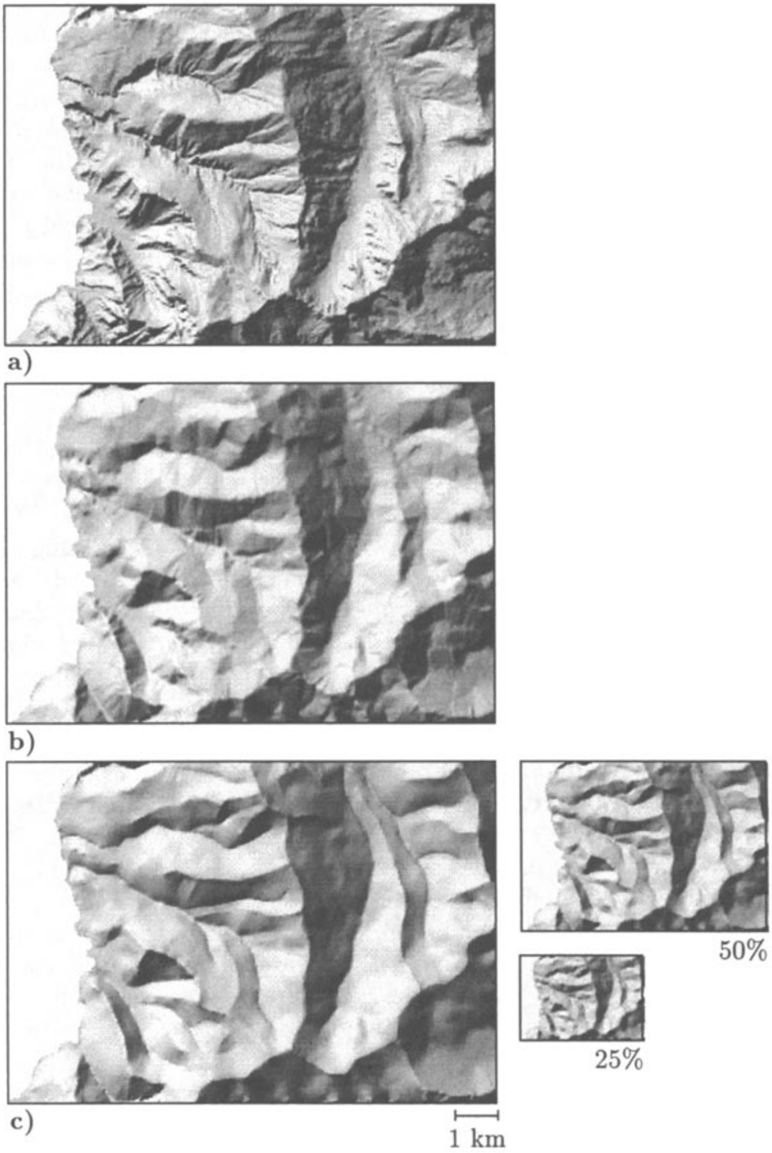

Fig. 23. Terrain generalization. **a)** Original surface. **b)** Generalized surface, resulting from the extraction and generalization of the network of topographic structure lines (channels and ridges). **c)** The automatically generalized surface has been modified further by interactive enhancement. (DTM data courtesy of Swiss Federal Office of Topography, DHM25 ©1997, (1263a))

- *Representations for geometric primitives:* Basic schemes available for the representation of geometric primitives (points, lines, etc.). Examples of commonly used representations are the polygonal chain (or polyline), raster or mathematical curves.
- *Complex data models and data structures:* Complex data models allow to integrate primitives into a common model and record their spatial and semantic relations. Examples are the topological vector data model, but auxiliary data structures (uniform grid, quadtree, Delaunay triangulation, etc.) are also of use in this context.

13.1 Representations for Geometric Primitives

In vector mode generalization, polygonal chains (polylines) are by far the most commonly used scheme for representing geometric primitives. They are easy to implement and handle, intuitive to understand and they can approximate any desired shape accurately (provided the vertices are sampled sufficiently densely). On the other hand, the polyline representation also imposes severe impediments on the development of generalization algorithms (Werschlein 1996, Fritsch and Lagrange 1995). Allowable generalization operators are essentially restricted to removing points (i.e., line simplification by vertex elimination) or displacing points (i.e., line smoothing). The fact that a polyline is equivalent to a chain (i.e., sequence) of points implies that it is difficult to model entire shapes in a compact term.

The polyline representation certainly still has its merits in many generalization applications, but it should be extended by complementary representations. The work by Affholder on geometric modeling of road data (reported in Plazanet et al. 1995) is an example of fitting the representation scheme more closely to the object that needs to be represented. Road data are commonly represented as polygonal chains, neglecting the fact that these man-made features are constructed using mathematical curves rather than free-form chains of points. Affholder models roads by a series of *cubic arcs*, leading to a more compact and also more natural representation which offers potential for the development of novel algorithms. For each bend of a road between two inflection points, a pair of cubic arcs is used to approximate the left and right half of the bend, respectively.

Other representations that bear potential for complementing polylines in a useful way are parametric curve representations and wavelets. *Curvature-based curve parametrizations* can be usefully exploited for shape analysis since critical points (such as inflection points) show up as extremes (Werschlein 1996). In addition to that, the magnitude of these extremes also exhibits the size of the shape associated with a critical point and thus allows to prioritize. *Wavelets* have potential for both shape analysis (Plazanet et al. 1995, Werschlein 1996) and as a basis for novel generalization algorithms (Fritsch and Lagrange 1995, Werschlein 1996). Wavelet coefficients can be analyzed to locate critical points and shapes, and they can also be filtered yielding generalized versions of the original feature. Since wavelets are localized, it is possible to eliminate entire shapes by setting the coefficients of the wavelets supporting the shape to zero (Werschlein 1996).

13.2 Complex Data Models

While the search for alternative representations for geometric primitives is mainly guided by the requirements of shape representation and shape analysis, research for improved complex data models is driven by the need to develop adequate algorithms for the operators of *context-dependent generalization*. That is, data models used for generalization must be extended to allow improved representation of spatial and semantic relations between individual features and feature classes. The requirements for improved data models can be summarized as follows:

- Representation of relevant metric, topological and semantic relations must be possible between objects of the same feature class and across feature classes. In particular, representation of proximity relations (metric) must be improved.
- Object modeling:
  - Multiple primitives per object (e.g., a coastline is partitioned into different sections – sandy beach, estuary, rocky shore)
  - Grouping (groups of objects of the same feature class), complex objects
  - Shared primitives between objects of different feature classes
- Integration of auxiliary data structures for computing and representing proximity relations (triangulations, regular tessellations)

As a consequence of these requirements the main data model should be an object-oriented extension of the basic topological vector model (as opposed to layer-based). Data models of this kind are now beginning to appear in some commercial GIS. Integrated auxiliary data structures for proximity relations are not yet available in commercial systems, but research is under way in that direction.

Most approaches to represent proximity relations between map objects have concentrated on the use of Delaunay triangulations or Voronoi diagrams (Ruas 1995, Ruas and Plazanet 1996, Ware et al. 1995, Jones et al. 1995, Ware and Jones 1996). An example of the use of a regular triangular tessellation for line generalization has been presented by Dutton (1996a). Dutton's quaternary triangular mesh (QTM) scheme is interesting for a variety of reasons other than generalization (outlined in Dutton 1996b). It offers a method for planetary geocoding of both local and global geospatial data as an alternative to the traditional latitude/longitude coordinate notation. Starting with an octahedron inscribed to the globe, the eight faces of the octahedron are successively subdivided into four equilateral triangles (and vertices projected to the surface of the globe), yielding eight quadtree-like structures. Triangles are coded into 64-bit words. A QTM location code (QTM ID) consists of an octant number (from 1 to 8) followed by up to 30 quaternary digits (from 0 to 3) which name a leaf node in a triangular quadtree rooted in the given octant. For example, the 18-level QTM ID for the building housing the Geography Department at the University of Zurich is 1133013130312301002. This encodes the geographic location 47^o 23' 48" N, 8^o 33' 4" E within about 60 meters, close to the precision obtained from measuring on a 1:100,000 scale map. Adding more digits increases the locational precision. At level 30, locations are encoded to a precision of about 2 cm. This code just fills one 64-bit word, equivalent to two single precision floats used for latitude/longitude encoding. However, 32 bit latitude/longitude coordinates do not allow for the same level of precision as QTM encoding.

Not only does QTM encoding offer a very space-efficient location scheme, it also allows to adapt the level of resolution used for encoding (and therefore the locational precision that can be achieved) to the accuracy of the data. In other words, it offers a convenient way to handle locational uncertainty contained in geographical data. Since the QTM location scheme is hierarchical, it is also inherently related to scale-changing. Dutton (1996a) has proposed an algorithm for line generalization that makes use of this property. Lines whose vertices had been encoded to a certain QTM level (e.g., level 20, which relates to 1:25,000) are weeded to the locational precision of a coarser QTM level (e.g., level 18, roughly equivalent to 1:100,000). Any vertices that fall within the same 'QTM attractor' (the hexagonal region formed by the six triangles surrounding a QTM node) are replaced by the median point of the corresponding section of the line. Beyond this direct use of the hierarchical structure of QTMs, there is also potential for using it in conflict detection and spatial search.

The data models used by Ruas (1995; see also Ruas and Plazanet 1996) and by Ware et al. (1995; see also Jones et al. 1995, Ware and Jones 1996) are both based on Delaunay triangulations (cf. van Kreveld 1997). Both approaches concentrate on the support of methods for the detection and resolution of spatial conflicts between features (e.g., features that overlap, features that are too close, etc.). Additonally, both use the space subdivision scheme as a means to compute proximity relations, compute displacement vectors (in the case of conflict), and keep track of displacements. Beyond these similarities, however, the two data models are built on a different approach.

Ruas (1995) attempts to embed the use of her proposed data model in a comprehensive strategy of generalization (see also Ruas and Plazanet 1996). Generalization – in Ruas' case the generalization of built-up areas – is seen as a process of conflict detection and resolution. A conflict is defined as an infringement on a cartographic principle (such as minimum visual separability of map features, avoidance of overlaps, etc.). Conflict detection proceeds in a hierarchical fashion; it is first carried out at the global level (i.e., the entire map), then the map space is subdivided according to the hierarchy of the road network (Fig. 24). Within each of the resulting partitions, conflict detection and resolution again takes place, starting at level 1 and proceeding to finer levels.

The Delaunay triangulation is then built within each partition that is currently worked on. Note that this is an unconstrained Delaunay triangulation and that it is kept local (i.e, to the elements of the current partition only) and

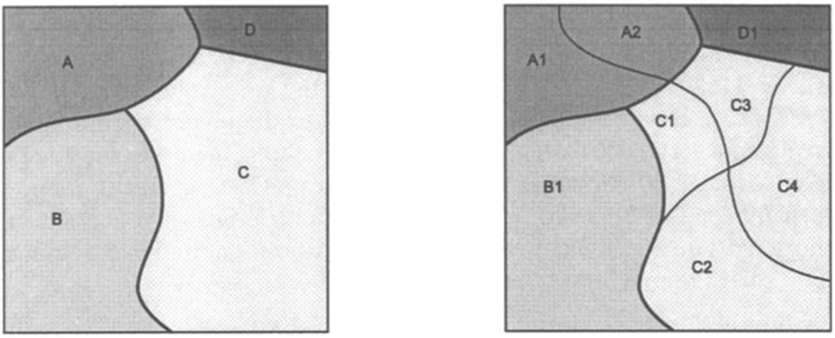

Fig. 24. Hierarchical subdivision scheme based on hierarchy of road categories (after Ruas 1995).

temporary (i.e., it is not saved). The triangulation connects the centroids of the small area objects and point objects falling within the tile, as well as projection points on the surrounding roads forming the tile boundary (Fig. 25). The edges of the triangulation are classified according to the objects they connect (Fig. 25). If the shape of a bounding road is changed or houses are enlarged or moved, the triangulation is used determine any conflicts that might have arisen. Displacement vectors are then computed from the proximity relations between objects and displacement propagation activated using decay functions (Fig. 26).

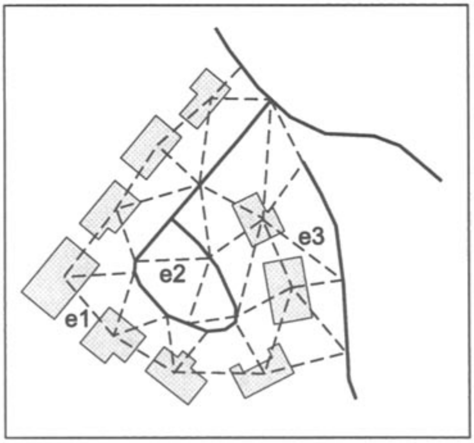

Fig. 25. Local Delaunay triangulation between buildings and adjacent roads (after Ruas 1995). Edge e_1 denotes an edge connecting two buildings; e_2 connects two vertices on a road; and e_3 connects a building and a road.

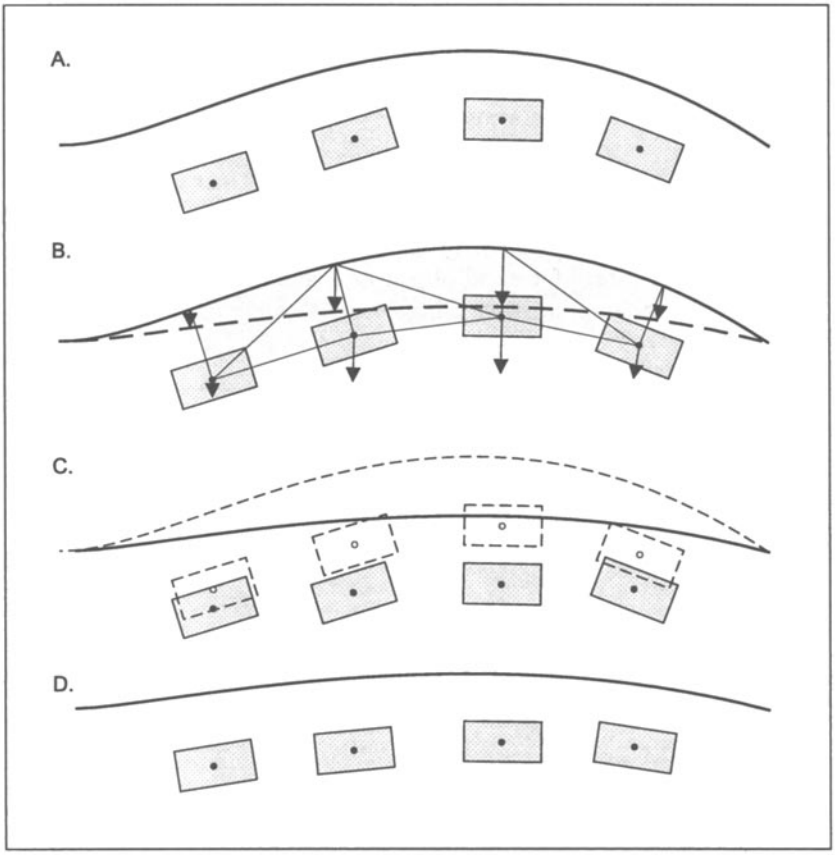

Fig. 26. Displacement of buildings after simplification of a road (after Ruas and Plazanet 1996).

There are several points in which the triangulated data model developed by researchers at the University of Glamorgan (Ware et al. 1995, Jones et al. 1995, Ware and Jones 1996) differs from the approach chosen by Ruas (1995). The triangulation forms the core of the data model. As a consequence, it is not only computed temporarily, but maintained continuously. Not only the centroids of map objects are connected, but a constrained Delaunay triangulation of all the vertices of all map objects is built (Fig. 27). The resulting simplicial data structure (SDS) is represented through a set of relations between objects, triangles, edges and vertices. These relations are stored by pointers corresponding to the entity relationships illustrated in Figure 28.

Jones et al. (1995) claim several useful properties for the SDS model: the explicit representation of all space on a 2-D map; precise representation of object boundaries from vector-structured data; ease of measurement; maintenance of topological relationships between points, lines and polygons; ease of determination of proximal polygons between objects; malleability; and a dynamic data structure. Since the SDS comprises all the geometric information of the original objects, all generalization operations are carried out directly on the SDS. The target application for the MAGE system built around the SDS is the generalization of large-scale topographic map data from the Ordnance Survey of GB. To that end, a palette of generalization operators has been developed including object exaggeration (enlargement); object collapse (constructing the centerline of road casings); operators for areal object amalgamation (direct merge, adopt merge); and building simplification using corner flipping of triangles.

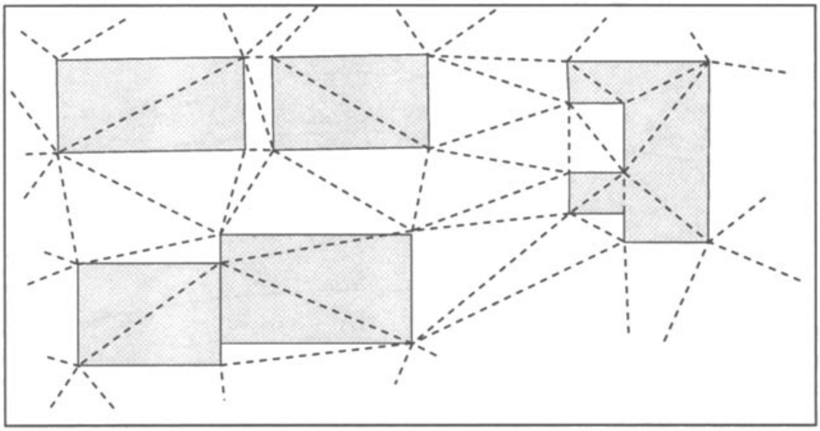

Fig. 27. Sample section of the constrained Delaunay triangulation forming the simplicial data structure (after Ware et al 1995).

The above examples all argue for an enrichment of data models used in generalization. A number of more fundamental questions, however, remain to be resolved by future research: How far does the representation of spatial and semantic relations in data models need to be extended? In what ways will this increase the cost of building spatial databases? Which relations can be determined computationally, and which ones need to be coded 'manually'? Which relations should be stored in the database, and which ones can be computed on-the-fly?

14 Conclusions

The importance of the generalization of spatial data as a function of GIS is accentuated by the current growth of the number and volume of spatial databases and by the need to produce data to specific requirements and share them among

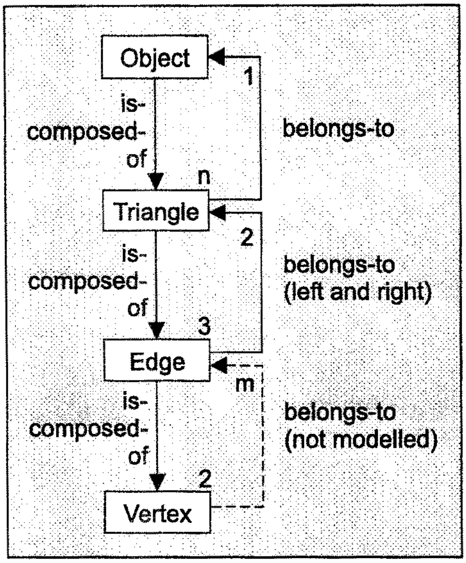

Fig. 28. Entity relationships in the simplicial data structure (after Jones et al. 1995).

different user groups. After a time of relative stagnation during the late 1970s and the 1980s, generalization has again attracted significant research interest in the GIS community and beyond. The topic is well represented at key GIS conferences and several working groups of international organizations – e.g., of the International Cartographic Association (ICA), the European Organization for Experimental Photogrammetric Research (OEEPE), and the International Society of Photogrammetry and Remote Sensing (ISPRS) – are trying to coordinate research efforts.

Although non-algorithmic methods certainly will play an increasingly significant role in automating generalization, algorithms still are of crucial importance because they form the foundation on which the other techniques must build. Computational geometry could contribute substantially to improving the functional and computational performance of current methods to address the geometrical aspects of generalization. Key areas awaiting better solutions are generalization algorithms that observe multiple constraints, robust methods for structure recognition, and the exploitation of alternative data representations and data structures to build more complex algorithms, particularly for context-dependent generalization.

Acknowledgments

I would like to thank Frank Brazile for helping with the preparation of illustrations and Geoff Dutton for reading parts of the draft manuscript. A number of people have generously provided illustrations or helped with the compilation of figures, including Corinne Plazanet and Anne Ruas of IGN France, and Chris Jones of the University of Glamorgan. Support from the Swiss NSF through project 2100-043502.95/1 is also gratefully acknowledged.

References

Bär, H.R. (1995): Interaktive Bearbeitung von Geländeoberflächen – Konzepte, Methoden, Versuche. (Ph.D. Dissertation) *Geoprocessing Series,* Department of Geography, University of Zurich, vol. **25**, 140 pgs.

Beard, K. (1991): Theory of the Cartographic Line Revisited: Implications for Automated Generalization. *Cartographica,* **28**(4): 32-58.

Brassel, K.E. and Weibel, R. (1988). A Review and Framework of Automated Map Generalization. *International Journal of Geographical Information Systems,* **2**(3): 229-44.

Buttenfield, B.P. (1985): Treatment of the Cartographic Line. *Cartographica,* **22**(2): 1-26.

Buttenfield, B.P and McMaster, R.B. (1991, eds.): *Map Generalization: Making Rules for Knowledge Representation.* London: Longman.

Clinton, W.J. (1994): Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure. *Federal Register,* 13 April 1994, Executive Order 12906, 59(71): 17671-17674.

Cromley, R.G. (1991): Hierarchical Methods of Line Simplification. *Cartography and Geographic Information Systems,* **18**(2): 125-131.

Cromley, R.G., and Campbell, G.M. (1991): Noninferior Bandwidth Line Simplification: Algorithm and Structural Analysis. *Geographical Analysis,* **23**(1), 25-38.

Cromley, R.G., and Campbell, G.M. (1992): Integrating Quantitative and Qualitative Aspects of Digital Line Simplification. *The Cartographic Journal,* **29**(1), 25-30.

De Berg, M., van Kreveld, M. and Schirra, S. (1995): A New Approach to Subdivision Simplification. *ACSM/ASPRS Annual Convention and Exposition,* Vol. 4 (Proc. Auto-Carto 12): 79-88.

Devogele, T., Trevisan, J. and Raynal, L. (1996): Building a Multi-Scale Database with Scale-Transition Relationships. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS research II* (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 6.19-6.33.

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

Duda, R. and Hart, P. (1973): *Pattern Classification and Scene Analysis.* New York: John Wiley.

Dutton, G. (1996a): Encoding and Handling Geospatial Data with Hierarchical Triangular Meshes. In: Kraak, M.J. and Molenaar, M. (eds.):

Advances in GIS research II (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 8B.15-8B.28.

Dutton, G. (1996b): Improving Locational Specificity of Map Data – A Multi-Resolution, Metadata-Driven Approach and Notation. *International Journal of Geographical Information Systems,* 10(3): 253-268.

Fisher, P.F. and Mackaness, W.A. (1987): Are Cartographic Expert Systems Possible? *Proc. Auto-Carto 8* (8th Int. Symposium on Computer-Assisted Cartography), Baltimore, MD: 530-534.

EUROGI (1996): GI2000 – Towards a European Policy Framework for Geographic Information. *Discussion Paper,* European Umbrella Organisation for Geographic Information (EUROGI).

Fritsch, E. and Lagrange, J.-P. (1995): Spectral Representations of Linear Features for Generalisation. In: Frank, A.U. and Kuhn, W. (eds.): Spatial Information Theory – A Theoretical Basis for GIS (Proceedings COSIT '95). *Lecture Notes in Computer Science* **988**, Berlin, Springer-Verlag: 157-71

Grünreich, D. (1985): Computer-Assisted Generalization. *Papers CERCO Cartography Course.* Institut für Angewandte Geodäsie, Frankfurt a. M.

Grünreich, D. (1992): ATKIS – A Topographic Information System as a Basis for GIS and Digital Cartography in Germany. In: Vinken, R. (ed.): From Digital Map Series in Geosciences to Geo-Information Systems. *Geologisches Jahrbuch* Series A, Vol. **122**. Hannover: Federal Institute of Geosciences and Resources: 207-216.

Hake, G. (1975): Zum Begriffssystem der Generalisierung. *Nachrichten aus dem Karten- und Vermessungswesen,* Sonderheft zum 65. Geburtstag von Prof Knorr: 53-62.

Heller, M. (1990): Triangulation Algorithms for Adaptive Terrain Modeling. *Proceedings Fourth International Symposium on Spatial Data Handling,* Zurich, July 1990, **1**: 163-174.

Hershberger, J. and Snoeyink, J. (1992): Speeding Up the Douglas-Peucker Line-Simplification Algorithm. *Proceedings 5th International Symposium on Spatial Data Handling,* Charleston, SC, 134-143.

Hess, M. (1995): *Erweiterung von Methoden zur automatischen Erzeugung panoramischer Ansichten.* MSc Thesis, Department of Geography, University of Zurich.

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.

ICA (International Cartographic Association) (1973): *Multilingual Dictionary of Technical Terms in Cartography.* Wiesbaden: Franz Steiner Verlag.

Imai, H. and Iri, M. (1988): Polygonal Approximations of a Curve – Formulations and Algorithms. In: Toussaint, G.T. (ed.): *Computational Morphology.* Elsevier Science Publishers, 71-86.

Jäger, E. (1991): Investigations on Automated Feature Displacement for Small Scale Maps in Raster Format. *Proceedings 15th International Cartographic Conference,* Bournemouth (UK): 245-256.

Jones, C.B., Bundy, G.Ll. and Ware, J.M. (1995): Map Generalization with a Triangulated Data Structure. *Cartography and Geographic Information Systems,* **22**(4): 317-331.

Kidner, D.B. and Jones, C.B. (1994): A Deductive Object-Oriented GIS for Handling Multiple Representations. In: Waugh T.C. and Healey, R.G. (eds.): *Advances in GIS research* (Proceedings Sixth International Symposium on Spatial Data Handling), London: Taylor & Francis: 882-900

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

Leberl, F.W. (1986): ASTRA – A System for Automated Scale Transition. *Photogrammetric Engineering and Remote Sensing,* **52**(2): 251-258.

Lecordix, F., Plazanet, C. and Lagrange, J.-P. (1996): PlaGe: A Platform for Research in Generalization. Application to Caricature. *GeoInformatica,* **1**(1).

Lee, D. (1995): Experiment on Formalizing the Generalization Process. In: Müller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): *GIS and Generalization: Methodological and Practical Issues,* London: Taylor & Francis, 219-234.

Li, Z., and Openshaw, S. (1992): Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization. *International Journal of Geographical Information Systems,* **6**(5): 373-389.

Lichtner, W. (1979): Computer-Assisted Processes of Cartographic Generalization in Topographic Maps. *Geo-Processing,* **1**: 183-199.

MSC Mapping Science Committee (1993): *Toward a Coordinated Spatial Data Infrastructure for the Nation.* Washington, DC: National Research Council, National Academy Press.

Marino, J.S. (1979): Identification of Characteristics along Naturally Occurring Lines: An Empirical Study. *The Canadian Cartographer,* **16**(1): 70-80.

McMaster, R.B. (1983): *A Quantitative Analysis of Mathematical Measures in Linear Simplification.* Ph.D. Thesis, Dept. of Geography and Meteorology, University of Kansas, Lawrence, Kansas.

McMaster, R.B. (1987a): Automated Line Generalization, *Cartographica,* **24**(2): 74-111.

McMaster, R.B. (1987b): The Geometric Properties of Numerical Generalization. *Geographical Analysis,* **19**(4): 330-346.

McMaster, R.B. (1989): The Integration of Simplification and Smoothing Algorithms in Line Generalization. *Cartographica,* **26**(1): 101-121.

McMaster, R.B. and Monmonier, M. (1989): A Conceptual Framework for Quantitative and Qualitative Raster-Mode Generalization. *Proceedings GIS/LIS '89*, Orlando, FL: 390-403.

McMaster, R.B., and Shea, K.S. (1992): *Generalization in Digital Cartography.* (Resource Publications in Geography). Washington, D.C.: Association of American Geographers.

Misund, G. (1996): Varioscale TIN Based Surfaces. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS research II* (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 6.36-6.45.

Molenaar, M. (1996a, ed.): Methods for the Generalization of Geo-Databases. *Publications on Geodesy,* New Series, Delft: Netherlands Geodetic Commission, **43**.

Molenaar, M. (1996b): The role of topologic and hierarchical spatial object models in database generalization. In: Molenaar, M. (ed.): Methods for the Generalization of Geo-Databases. *Publications on Geodesy,* New Series, Delft, Netherlands Geodetic Commission **43**: 13-36.

Monmonier, M.S. and McMaster, R.B. (1991): The Sequential Effects of Geometric Operators in Cartographic Line Generalization. *International Yearbook of Cartography.*

Muller, J.-C. (1990): The Removal of Spatial Conflicts in Line Generalization. *Cartography and Geographic Information Systems,* **17**(2): 141-149.

Muller, J.-C. (1991): Generalization of Spatial Databases. In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.): *Geographical Information Systems: Principles and Applications.* London: Longman, **1**: 457-475.

Müller, J.-C., Lagrange, J.-P., and Weibel, R. (1995a, eds.): *GIS and Generalization: Methodological and Practical Issues.* London: Taylor & Francis.

Müller, J.-C., Weibel, R., Lagrange, J.-P., and Salgé, F. (1995b): Generalization: State of the Art and Issues. In: Müller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): *GIS and Generalization: Methodological and Practical Issues,* London: Taylor & Francis, 3-17.

Nickerson, B.G. (1988): Automated Cartographic Generalization for Linear Features. *Cartographica,* **25**(3), 15-66.

Opheim, H. (1982): Fast Reduction of a Digitized Curve. *Geo-Processing,* **2**: 33-40.

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

Peucker, T.K. (1975): A Theory of the Cartographic Line. *International Yearbook of Cartography,* **16**: 134-143.

Plazanet, C. (1995): Measurements, Characterization, and Classification for Automated Line Feature Generalization. *ACSM/ASPRS Annual Convention and Exposition,* Vol. **4** (Proc. Auto-Carto 12): 59-68.

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

Plazanet, C. (1996): *Analyse de la géométrie des objets linéaires pour l'enrichissement des bases de données. Intégration dans le processus de généralisation cartographique des routes.* PhD Thesis, Université Marne la Vallée.

Ramer, U. (1972): An Iterative Procedure for the Polygonal Approximation of Plane Curves. *Computer Graphics and Image Processing,* **1**: 244-256.

Reumann, K. and Witkam, A.P.M. (1974): Optimizing Curve Segmentation in Computer Graphics. *International Computing Symposium.* Amsterdam: North Holland; 467-472.

Richardson, D.E. (1994): Generalization of Spatial and Thematic Data Using Inheritance and Classification and Aggregation Hierarchies. In: Waugh T.C. and Healey, R.G. (eds.): *Advances in GIS research* (Proceedings Sixth International Symposium on Spatial Data Handling), London: Taylor & Francis: 901-20

Rieger, M. and Coulson, M. (1993): Consensus or Confusion: Cartographers' Knowledge of Generalization. *Cartographica,* **30**(1): 69-80.

Rogers, D.F. and Adams, J.A. (1990): *Mathematical Elements for Computer Graphics.* Second Edition. New York et al.: McGraw-Hill.

Roos, T. (1996): Voronoi Methods in GIS. *This volume.*

Ruas, A. (1995a): Multiple Representations and Generalization. *Lecture Notes for 1995 Nordic Cartography Seminar.* St.-Mandé: Institut Géographique National. Available as PostScript document from anonymous ftp <sturm.ign.fr>.

Ruas, A. (1995b): Multiple Paradigms for Automating Map Generalization: Geometry, Topology, Hierarchical Partitioning and Local Triangulation. *ACSM/ ASPRS Annual Convention and Exposition,* Vol. 4 (Proc. Auto-Carto 12): 69-78.

Ruas, A. and Lagrange, J.-P. (1995): Data and Knowledge Modelling for Generalization. In: Müller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): *GIS and Generalization: Methodological and Practical Issues,* London: Taylor & Francis, 73-90.

Ruas, A. and Plazanet, C. (1996): Strategies for Automated Generalization. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS research II* (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 6.1-6.18.

Rusak Mazur, E., and Castner, H.W. (1990): Horton's Ordering Scheme and the Generalisation of River Networks. *The Cartographic Journal,* **27**: 104-112.

Schlegel, A., and Weibel (1995): Extending a General-Purpose GIS for Computer-Assisted Generalization. *17th International Cartographic Congress of the ICA,* Barcelona (E), 2211-2220.

Schylberg, L. (1993): *Computational Methods for Generalization of Cartographic Data in a Raster Environment.* PhD Thesis, Department of Photogrammetry, Royal Institute of Technology, Stockholm.

Shreve, R.L. (1966): Statistical Law of Stream Number. *Journal of Geology,* **74**: 17-37.

Spiess, E. (1995): The Need for Generalization in a GIS Environment. In: Müller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): *GIS and Generalization: Methodological and Practical Issues,* London: Taylor & Francis, 31-46.

Strahler, A.N. (1957): Quantitative Analysis of Watershed Geomorphology. *Transactions of the American Geophysical Union,* **8**(6): 913-920.

Swiss Society of Cartography (1977): Cartographic Generalization – Topographic Maps. *Cartographic Publication Series,* vol. **2**. Zurich: Swiss Society of Cartography.

Töpfer, F. (1974): *Kartographische Generalisierung.* Gotha, Leipzig: VEB Hermann Haack.

van Kreveld, M. (1997): Digital Elevation Models: Overview and Selected TIN Algorithms. *This volume.*

van Oosterom, P. and van den Bos, J. (1989): An Object-Oriented Approach to the Design of Geographic Information Systems. *Computers & Graphics,* **13**: 409-418.

van Oosterom, P. and Schenkelaars, V. (1995): The Development of an Interactive Multiscale GIS. *International Journal of Geographical Information Systems,* **9**(5): 489-507.

Visvalingam, M., and Whyatt, J.D. (1993): Line Generalisation by Repeated Elimination of Points. *Cartographic Journal,* **30**(1): 46-51.

Visvalingam, M. and Williamson, P.J. (1995): Simplification and Generalization of Large Scale Data for Roads: A Comparison of Two Filtering Algorithms. *Cartography and Geographic Information Systems,* **22**(4): 264-275.

Ware, J.M., Jones, C.B. and Bundy, G.Ll. (1995): A Triangulated Spatial Model for Cartographic Generalisation of Areal Objects. In: Frank, A.U. and Kuhn, W. (eds.): Spatial Information Theory – A Theoretical Basis for GIS (Proceedings COSIT '95). *Lecture Notes in Computer Science* **988**, Berlin, Springer-Verlag: 173-192.

Ware, J.M. and Jones, C.B. (1996): A Spatial Model for Detecting (and Resolving) Conflict Caused by Scale Reduction. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS research II* (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 9A.15-9A.26.

Weibel, R. (1991): Amplified Intelligence and Rule-Based Systems. In: Buttenfield, B.P., and McMaster, R.B. (eds.): *Map Generalization – Making Rules for Knowledge Representation.* London: Longman, 172-186.

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

Weibel, R. (1995a): Map Generalization. Special Issue of *Cartography and Geographic Information Systems,* **22**(4).

Weibel, R. (1995b): Three Essential Building Blocks for Automated Generalisation. In: Müller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): *GIS and Generalization: Methodological and Practical Issues,* London: Taylor & Francis, 56-69.

Weibel, R. and Ehrliholzer, R. (1995): An Evaluation of MGE Map Generalizer. *Internal Report,* Department of Geography, University of Zurich, 36 + 18 pgs.

Weibel, R., Keller, St., and Reichenbacher, T. (1995): Overcoming the Knowledge Acquisition Bottleneck in Map Generalization: The Role of Interactive Systems and Computational Intelligence. In: Frank, A.U. and Kuhn, W. (eds.): Spatial Information Theory – A Theoretical Basis for GIS (Proceedings COSIT '95). *Lecture Notes in Computer Science* **988**, Berlin, Springer-Verlag: 139-156.

Weibel, R. (1996): A Typology of Constraints to Line Simplification. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS research II* (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 9A.1-9A.14.

Werschlein, T. (1996): *Frequenzbasierte Linienrepräsentationen für die kartographische Generalisierung.* MSc Thesis, Department of Geography, University of Zurich.

White, E.R. (1985): Assessment of Line Generalization Algorithms Using Characteristic Points. *The American Cartographer,* **12**(1): 17-28.