Using Vector and Raster-Based Techniques in Categorical Map Generalization

Beat Peter and Robert Weibel
Department of Geography
University of Zurich
Winterthurerstrasse 190
8057 Zurich (Switzerland)
Phone: +41-1-635 52 55
Fax: +41-1-635 68 48
e-mail: {beatp | weibel}@geo.unizh.ch

Abstract

Categorical data are a frequent data type in GIS and thematic cartography. Therefore, comprehensive methodologies for the generalization of categorical data in both the vector and the raster model are urgently needed. After the presentation of a general framework and recommended workflow, generic cartographic constraints governing the generalization of categorical data are specified. In the next step, these constraints will be translated into tools for assessing the need and the quality of generalization solutions as well as tools for achieving the necessary generalization transformations. These tools are usually associated with a particular data model. Typical and frequent generalization problems are used to study how the constraints can be best translated and parameterized in a particular data model, and how particular solutions perform in comparison to others. The last component of this paper relates to the conversion of data between the vector and the raster model and vice versa and explores how such operations can be usefully integrated into a generalization strategy.

1. Introduction

Categorical maps are a frequent data type in GIS applications and in thematic cartography. Examples include maps (or databases) of soil, geology, vegetation, or classified remote sensing images. Networks of political or administrative boundaries can be considered a special case of this maps type. Categorical maps are commonly modeled as either vector data (i.e., as polygonal maps or polygonal subdivisions) or as raster data. Raster categorical data mainly originate from grid samples, remote sensing imagery, or interpolated and classified point samples. Vector data are usually digitized from the corresponding categorical maps. Although there are tools available in current commercial GIS and cartography systems that allow processing raster and vector categorical data for purposes of analysis and display, specific methods for automated generalization of such data are less well developed. They represent mere adaptations of methods developed elsewhere to the problem of categorical map generalization. For vector categorical maps line generalization algorithms are used instead of polygon-oriented methods. Techniques to “generalize” raster data sets are essentially equivalent to simple pixel-based image processing operations, not respecting the object nature of “raster polygons” (regions, connected components). More sophisticated methods of preliminary nature for both raster and vector categorical maps have been proposed in the research literature, such as Schylberg (1993), Su et al. (1997) and Jaakkola (1998) for raster data, or Muller and Wang (1992), de Berg et al. (1998) for vector models respectively. However, they need further improvement and integration into a coherent framework and workflow if the generalization of categorical maps is to be solved more comprehensively.

The research reported here builds on previous work (Weibel 1996, Bader and Weibel 1997, Peter 1997) and has two main objectives: to improve current methods for categorical map generalization and to evaluate if and how vector and raster-based techniques can be usefully integrated into a comprehensive generalization methodology. In order to meet these objectives, three elements are looked at. These elements will be discussed in sections 3 to 5, following the presentation of a general framework in section 2. The first element concerns the specification of so-called generalization constraints, that is, conditions of geometric, topological, semantic, and Gestalt nature, which govern the process of categorical map generalization. The second element of an integrated methodology has to do with the translation of the constraints into tools for assessing the need and the effect or quality of generalization (assessment tools, measures) and tools for achieving the necessary generalization transformations (transformation tools, generalization algorithms). Invariably, these tools will be associated with a particular data model. Using typical and frequent generalization tasks, it is studied how the constraints can be best translated and parameterized in a particular data model, and how particular solutions perform in comparison to others. Finally, the third component of this research relates to the conversion between raster and vector data in both directions. Conversion may be necessary because categorical data in raster or vector format need to be combined with or aligned to other feature classes, which exist in another data model. It is also potentially of help to substitute for missing generalization functionality in one domain by exploiting generalization algorithms in the other, as has been suggested by Su et al. (1998), for instance. Section 5 discusses some of the problems that may occur during the conversion process, and proposes ways to cope with these difficulties as well as possible applications. The final section of the paper presents conclusions and directions for future research.

2. Framework for Categorical Map Generalization

While most frameworks for the generalization of cartographic data, like for instance those by Brassel and Weibel (1988) or Ruas and Plazanet (1996), provide generic procedural information, the one presented in this paper is designed more specifically for categorical data and provides more detail (see figure 1). Large parts of this framework may be considered generic as well (e.g., constraints definition and constraints translation). However, all parts are defined and instantiated specifically for categorical generalization, and apply almost exclusively to categorical data (e.g. thematic generalization).

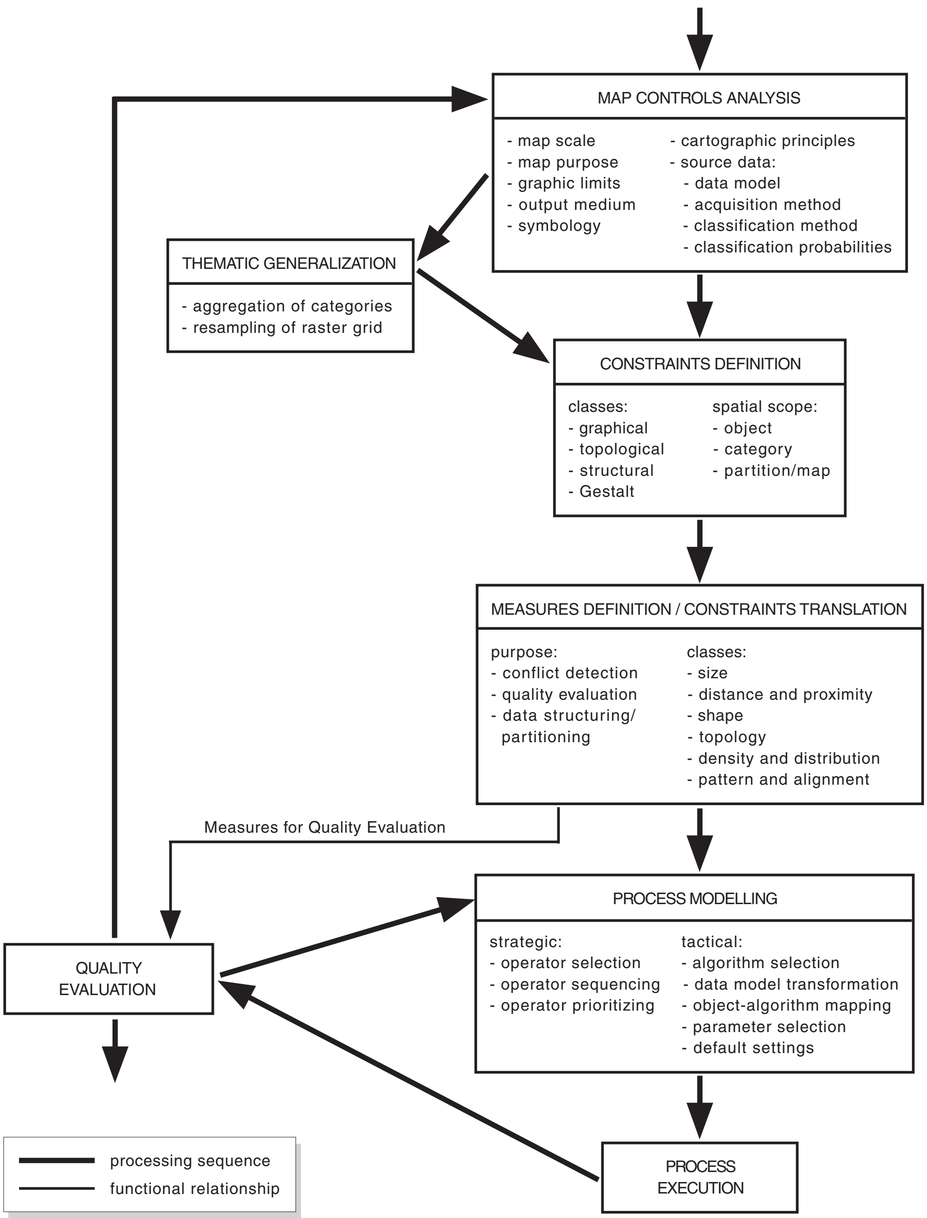

Figure 1: Framework for categorical data generalization

When analyzing *map controls* relevant for a generalization problem, specific issues for categorical data arise mainly from the way data has been obtained. Categorical data is frequently acquired from remote sensing sensors, point sampling methods or fieldwork. Therefore, aspects like the classification method used play a major role when developing a global generalization strategy. These issues influence the quality of the results that can be usefully achieved as well as the methods that are applicable to a wide degree. For instance, when a maximum likelihood classifier was used, the resulting classification probabilities can be exploited in generalization operations where a distinction can be made between ambiguously and unambiguously classified regions.

The next step in the process, *thematic generalization*, is unique to categorical data. While in small-scale topographic maps features like minor roads can just be omitted, this is not possible in categorical maps since a continuous network of polygons is required by definition. Thematic generalization is in general possible for both vector and raster data and does not require any geometric transformations. Resampling of data to a coarser resolution using appropriate interpolation methods is, of course, a method that can only be applied to raster data, while aggregation of categories (e.g., deciduous and coniferous forest to forest) is not bound to a specific data model.

The next two steps in process flow direction, *constraints definition* and *measures definition/constraints translation*, are generic and apply to constraint-based generalization in general. These two items together roughly represent what is termed structure recognition in the Brassel and Weibel (1988) conceptual framework. Both constraints definition and constraints translation will be discussed in more detail and with respect to categorical data in the following sections.

Process modeling is the next step in the framework. Here, in contrast to thematic generalization, geometric transformations of the data are applied. This is valid for both vector and raster data since although, in principle, raster data is generalized by reclassifying cells, the concepts on how to do this rely on (discrete) geometrical transformations. Process modeling can be decomposed into a strategic and a tactical part. The first part is to develop a generalization strategy within the limits of the defined and formalized constraints as well as the predefined map controls. For categorical data, global structural accuracy is in most cases more important than local positional accuracy, especially when data is used as thematic background (e.g. for linear or network data). After appropriate operators have been selected, their sequence and their tolerances regarding predefined values or bandwidths has to be decided upon (prioritizing). After a strategy has been formally declared, tactical considerations follow. The possibility of data model transformation has to be included in tactical considerations since, as we will see, some algorithms can are considerably easier to implement, require less computational effort or generate visually more convincing solutions in either the vector or the raster model. Data model transformation is possible on a global or local basis and can be temporary or permanent depending on the desired data structure of the resulting map.

Evaluation and *validation* of the results follow the execution of the selected strategy. Measures for this purpose are provided by the constraints translation step, as the corresponding arrow in figure 1 indicates. The main goal of the evaluation step is to verify that all identified conflicts have been resolved, that no new conflicts have been introduced during generalization (e.g., self-intersection of outlines due to line filtering algorithms) and to decide if values that implement structural and Gestalt measures lie within the specified bandwidths. Validation of results includes, of course, also visual inspection since, as we will see, complex concepts like *visual balance* cannot be formalized comprehensively. Therefore, a result can be rejected, although all specified parameters may lie within acceptable limits. For this reason, the last part of the framework forms a feedback loop. Several passes with changes of parameters, tactics or even the entire strategy might be necessary until a formally and visually acceptable result is achieved. Should this not be possible with the available means, the loop to the starting point has to be chosen, requiring a complete re-evaluation of the map controls.

3. Constraints to Categorical Data

The goal of this section is to identify constraints that apply to the generalization of categorical data. No restrictions regarding the underlying data model (vector or raster) are made at this point. After discussing basic issues of constraints with respect to the generalization of categorical data, some basic properties for this data type are introduced and a classification scheme is presented. A detailed list of generic constraints to the generalization of categorical data follows, which allows developing a strategy for the representation of categorical data at various scales and for a variety of purposes.

3.1 Key Aspects of Constraints

A *constraint* in the context of generalization can be defined as a design specification to which the solutions to a generalization problem should adhere (Weibel and Dutton 1998). A constraint is meant to limit the number of possible solutions without binding it to a particular action. This concept reflects the idea that more than one acceptable solution may exist to a given generalization problem. It is therefore well suited for developing flexible systems. Constraints originate from specific *map controls* applicable to a generalization problem. They are usually specified as something to *maintain* or to *avoid*. The expression “to respect” is used for something to be adhered to as far as possible while “to preserve” is used for topology related constraints in the same sense as to maintain. Some constraints can be termed *absolute* (e.g., minimum size) while others designate issues to be *optimized* (e.g., respect size distribution). Constraints like minimum size are termed *intrinsic* since they consider only one state of an object in a database while *extrinsic* constraints require two states (before and after generalization) for quality evaluation.

Most constraints do not work independently, they are contextually related and affect one another. A sub-system for prioritizing and managing constraints is therefore required indicating the *priority* and maximum tolerable *severity* for each constraint. Priority designates the importance of a constraint in relation to others, while severity indicates the degree to which a constraint can be violated under certain conditions. Finally, it is important to mention that not every constraint is needed in every situation. Constraints and appropriate mechanisms for their management provide the means for the development of global strategies for holistic solutions to generalization problems, including both spatial and semantic aspects.

3.2 Constraints vs. Properties and Rules

For categorical data, a number of *properties* are defined. Although they could also be formulated as constraints they are listed here separately since they reflect either intrinsic, low level aspects of this kind of data or just definitions made for data dealt with in this paper. Since only few properties are defined, the applicability of the presented concepts to most types of categorical data should not be compromised. Properties of categorical data enforced in this work are:

- Data covers the entire plane either as *polygonal subdivision* or as a *grid of raster cells*. No unclassified polygons or raster cells are permitted to occur as a result of generalization.
- In the case of vector data, a data model using *shared primitives* (i.e., shared arcs and nodes) within every feature class must be used. If technically possible, shared primitives can also be used between feature classes to constrain common boundaries to each other.
- No object has common boundaries with other objects of the *same* category. If such a case occurs, the separating boundary is dropped.

Rules are, compared to constraints, more fixed and not dynamically modifiable since they usually clearly indicate what particular action to take under a certain condition. They follow a notation of the type IF <condition> THEN <action>. Since for the treatment of generalization problems many variations in spatial and attribute characteristics have to be considered, a very large and hardly manageable number of rules would result. Working with constraints is therefore better suited for developing flexible generalization systems (Beard 1991).

3.3 Constraints and Data Model

Datasets with categorical data consist either of *polygons* for vector data or an *array of cells* for raster data. Using the technique of *connected component labeling*, so-called *regions* can be formed for raster data from connected cells of the same category. Depending on the resolution and spatial structure of the data, *4* or *8 cell connectivity* can be used to form a region. In many cases 4 cell connectivity is more appropriate since very large and complex regions can result if 8 cell connectivity is chosen. GIS systems in use provide functionality for connected component labeling but no topologic information is usually computed and associated with the formed regions. Instead of the term object, the expression *patch* is used from now on if both polygons and raster regions are meant.

It is important to mention that constraints, in principle, express cartographic design specifications which should be completely independent from specific data models. Cartographic constraints are first of all defined in terms of continuous geometry. Vector models offer the most direct translation of continuous to discrete geometry, while raster models entail a significant discretization by virtue of discrete and systematic spatial sampling. Hence, some constraints – particularly size and distance constraints – cannot be easily and usefully accommodated in raster models. The sampling interval (i.e., spatial resolution) has a premier influence on the potential of translating constraints into raster models. Depending on whether the resolution is below or above the minimum visual separability distance, some constraints are simply not applicable and cannot be considered.

3.4 Classification of Constraints

For the classification of constraints the scheme of Weibel and Dutton (1998) is used. Constraints are classified according to their *function*, which seems appropriate for generalization problems in a digital environment. Further subdivision relates to the spatial *application scope* of constraints, which can either be a single patch, all patches of a category or a group of patches, a partition of the map or the whole map respectively. It has to be pointed out that, although constraints for single patches can be identified, at least two patches are always involved in the actual generalization process since patches in categorical maps share boundaries as per definition. Therefore, constraints for single patches are mainly related to the *selection process* rather than to the actual transformation. Four types of constraints are distinguished:

Graphic constraints mainly deal with aspects of perceptibility such as size, width and separability. Categorical data consist entirely of area features where each category is assigned a different color fill. Depending on the color, different parameters for minimum perceptibility may apply. Furthermore, symbolization effects have to be considered since the line weight of the outlines of polygons or regions is affected by scale change. Figure 2 presents an overview of distance related constraints within and between patches for both vector and raster data.

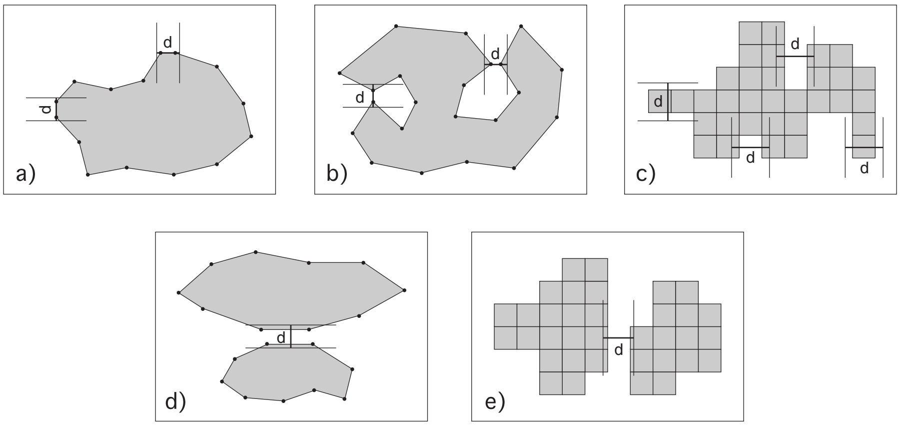

Figure 2: Distance related constraints within or between patches

Topological constraints deal with basic topological relationships like connectivity, adjacency and containment, which should be maintained when generalizing data. Self-intersection and overlapping patches are also issues related to topology. Since a data model using shared primitives is used by definition, overlapping patches do not exist and cannot be introduced with generalization. However, self-intersecting boundaries or intersections between different boundaries can occur as a result of erroneous line generalization algorithms.

Structural constraints define criteria that describe *spatial* and *semantic properties* of the data. Spatial structural constraints deal mainly with the preservation of typical shapes (on the patch level) or with the preservation of *patterns* and *alignments* if multiple patches are involved. Semantic structural constraints deal with the preservation of the *logical context* of patches. For these constraints *auxiliary data* such as road and river networks or terrain models are necessary as well as heuristics and domain knowledge about the nature of the data being generalized.

Gestalt constraints relate to aesthetic aspects. These include the preservation of the patch characteristics as well as the retention of the overall visual balance when multiple patches or the whole dataset is considered. Gestalt constraints are complex and difficult to formalize for use in digital systems but nevertheless important since they represent aspects of cartographic knowledge, which is not necessarily formalized. Gestalt constraints are enforced by the global strategy rather than by tactical decisions.

Constraints Related to Patches

1. *Minimum size* (graphical): Patches, which are too small, can be either deleted or enlarged

2a. *Minimum distance* (graphical, vector): The distance between consecutive vertices of a polygon outline should not be less than the minimum visual separability distance (see figure 2a)

2b. *Minimum distance* (graphical, raster): The distance between any parallel edges of the outlines (horizontal or vertical) of a region should not be less than the minimum visual separability distance (see figure 2b)

2c. *Self-coalescence* (graphical, vector): The distance between any vertices of a polygon outline should not be less than the minimum visual separability distance (see figure 2c)

3. *Separability* (graphical): The distance between two patches should not be less than the minimum visual separability distance (see figure 2d and e)
4. *Separation* (topological): Avoid separation of patches when deleting parts of it
5. *Islands* (topological): Patches, which can be identified as islands may be deleted or enlarged but should not be amalgamated with other patches of the same category
6. *Self-intersection* (topological): Avoid introduction of self-intersection of patch outlines
7. *Amalgamation* (structural): Disjoint patches of the same category may be amalgamated
8. *Collabsability* (structural): The area of eliminated patches should be distributed among the neighboring patches
9. *Shape/Angularity* (structural): Respect the global shape and angularity of patches

Constraints Related to Categories

10. *Size ratio* (structural): Respect the size ratio for each category relative to the total area
11. *Shape/Angularity* (structural): Respect typical shapes and angularity of patches of each category
12. *Size distribution* (structural): Respect the given size distribution of patches for each category
13. *Alignement/Pattern* (Gestalt): Preserve typical alignments and patterns of patches of a category

Constraints Related to Partitions or Groups of Patches

14. *Neighborhood relations* (topological): Preserve given neighborhood relations
15. *Spatial context* (structural): Avoid introduction of illogical neighborhood relations (e.g., house in a lake)
16. *Aggregability* (structural): Allow aggregation of categories if required and suitable super-categories exist
17. *Auxiliary data* (structural): Observe constraints imposed by auxiliary data (e.g., roads, rivers, point features)
18. *Alignment/Pattern* (Gestalt): Preserve typical alignments and patterns of patches within the map or within a group of patches
19. *Visual balance* (Gestalt): Avoid gross changes in shape and distribution of patches, unless required by extreme scale change
20. *Equal treatment* (Gestalt): Ensure equal treatment within a partition of the map and avoid highly unequal treatment across all partitions

The analysis of the above list of constraints shows that, compared to topographic maps, only relatively few topological issues have to be observed for the generalization of categorical data. According to Weibel (1996), it can be hopped that, if all other constraint classes are satisfied, aesthetic principles (Gestalt constraints) are met to a large degree as well. The above list is dominated by graphical and structural constraints with the graphical constraints exclusively on the patch level. This indicates that a generalization strategy should tackle the problem from two sides simultaneously. On the patch level, methods take care of conflict identification between or within patches (intrinsic graphic constraints) while simultaneously decisions on which alternative to chose for conflict resolution are made on a higher spatial level (category, whole map), considering and monitoring the structural changes for an entire category or the whole dataset.

4. Translating Constraints to Measures and Generalization Algorithms

The first part of this section discusses key aspects and general requirements for geometric and semantic measures. Due to the overwhelming number of measures that can be found in the literature their review within this paper seems neither possible nor useful. Instead, the second part of this section demonstrates the process from instantiation of constraints over the selection of appropriate measures to the enforcement of constraints with the help of two frequent and typical generalization problems. This procedure should allow to better illustrating problems and alternatives as well as advantages and restrictions that arise from the data model.

4.1 Key Aspects of Measures

A *measure* is defined as a procedure for computing *measurements* (numerical values). Measures are the basis for formal descriptions of relevant characteristics of geographical entities at the patch, category and map level. They allow assessing the *need* for and the *success* of generalization. A measure can be a simple formula (e.g., area calculation) or a complex algorithm, which may even require the computation of auxiliary data structures like a *Delaunay triangulation*. Measures can be either *absolute* (intrinsic), meaning that they can be interpreted and applied according to the analysis of one state of the database or be *relative* (extrinsic), which means that measurements of two states of the database have to be compared and evaluated to decide if a solution is acceptable or has to be rejected. This includes also subsequent testing for side effects (e.g., self-intersection) that can be introduced by certain generalization algorithms (e.g., line simplification). Most measures exist for vector and raster data but employ different methods for their computation. The key concept for using measures in generalization systems is *database enrichment*. The measures computed are added to the database as attributes. This includes numerical values as well as topological information (if not computed automatically) or just *flags* that identify a patch e.g. as island or “undeletable”. Computation of *statistical measures* (e.g., histograms) is also considered very useful for the analysis of the distribution and variability of patches and the evaluation of changes.

4.2 Classification of Measures

Measures can be classified according to the main characteristic they represent. This schema is influenced to a large degree by the constraints defined in section 3. However, some measures may express more than one property, for example *core area* (FRAGSTATS 1994) which is used to characterize size in the first place but contains also information about the shape of a patch. The following classes of measures are distinguished:

- Size measures
- Distance and proximity measures
- Shape measures
- Topological measures
- Density and distribution measures
- Pattern and alignment measures

4.3 Requirements for Measures

A useful measure should satisfy the following criteria: Ideally, it should

- describe the intended property as precise as possible and should not be influenced by other effects (orthogonality),
- be insensitive to outliers (robustness),
- be invariant to geometric transformations (geometric invariance),
- produce different results for different configurations of characteristics and similar results for similar configurations (differentiation),
- be easy to calculate (ease of calculation),
- be easy to use (with only a limited number of parameters) (ease of use), and
- be easy to interpret (ease of interpretation). Ideally for a certain value (a measurement) only one possible configuration of the measured property should exist.

While simple measures (e.g., area, perimeter) can fully meet the above criteria, this may not be the case for complex measures, especially those formalizing *structural* or *Gestalt* constraints (e.g., description of pattern and alignment of patches). Abstract measures such as the *fractal dimension* (representing shape aspects) are very difficult to interpret and should therefore be used with care. In many cases, a set of measures is needed to cover the main characteristics of a spatial or thematic entity sufficiently. Many statistical measures such as *patch size standard deviation* assume *normal distribution* of data. Variability measures (e.g., patch size coefficient of variation) should only be interpreted together with the total number of patches to avoid misinterpretation. For raster data measures are computed for regions that were formed using the connected component labeling method. Measurements can vary dramatically depending on the connectivity rule employed for region building (4 or 8 cell). Of course, measures for quality evaluation of generalized raster data can only be usefully applied if the spatial resolution of the raster grid is not changed during the generalization process.

4.4 Translating Constraints to Measures

The translation process of constraints to formal measures is executed by the system designer at research and design time. Optimally, at run time, system users need only to specify the *priority* and the *maximum tolerable severity* of the various constraints for a given mapping task.

Translating constraints to measures is a complex and crucial process within a constraint based generalization system. The underlying concepts of constraints and measures are quite different. As stated in section 3, the goal of a constraint is to limit the number of acceptable solutions to a problem without binding it to a particular action. A measure on the other hand is a formal mathematical concept, which makes use of clearly defined formulae or algorithms. Only few constraints, e.g. minimal size, can be translated to a measure on a 1:1 basis. Most concepts, such as shape or visual balance are rather fuzzy and ill-defined terms. Hence, it is almost impossible to formally describe all properties that characterize such a spatial concept comprehensively. Translating constraints to measures is therefore also a *selection process*. The goal of the process is to make the main properties of a spatial entity available to formal mathematical descriptions. The degree to which this goal can be achieved has a major influence on the results of the generalization process. Generalization algorithms cannot deal with properties of patches or spatial entities that have not been formalized nor can changes be evaluated for possible rejection of solutions. In general, graphical constraints can be formalized more easily and precisely than structural or Gestalt constraints.

4.5 Examples

Using typical and frequent generalization problems, the following two examples demonstrate how constraints are instantiated and how they can be translated to measures. While the constraints themselves are unspecific to any particular data model, algorithms for data enrichment and conflict detection (e.g., measures) and data generalization must be specialized for vector or raster data. Problems and advantages of each data model resulting from this fact are discussed. Only basic properties of categorical data have been incorporated in the description of generalization algorithms. Countless others can be specified to modify the generalization process, respecting specific aspects of the data used. Furthermore it should be pointed out that the problems and solutions presented in both examples are normally part of an integrated generalization strategy of interrelated operators and algorithms and should not be looked at in isolation.

Example 1: Detecting and Resolving Conflicts Imposed by the Minimum Size Constraint

Minimum size constraints are straightforward and easy to translate to a measure. Conflict identification is simple and methods for conflict resolution are not very challenging. However, we will show that for the selection of the appropriate generalization operator additional information needs to be considered.

The measure for size is *area*. For polygons, area calculations are standard GIS functions. For raster data, the area of a region is represented by the number of consecutive cells of the same category. The lower limit of the value for minimum area is the *minimum perceptibility size* but should be selected higher due to other relevant map controls (e.g. map purpose or output media). In principle, two operators exist for conflict resolution. A patch violating the minimum size constraint can be either *deleted* or *enlarged* until the size is above the specified minimum. The possibility to amalgamate patches of the same category will be discussed in example 2 and is not considered here. Every patch, even if it is very small, represents not only itself but also its category in the map. To be able to decide which operator to use for a particular patch, further constraints, especially structural and Gestalt constraints have to be instantiated and translated to measures allowing to look at a patch in its spatial context. Such measures are:

- Total number of patches of a category
- Total area of a category relative to total map area
- Ratio of the area of a category relative to areas of other categories
- Size distribution of the patches of a category
- Spatial distribution or concentration of the patches of a category
- Topological information about neighborhood relations of patches
- Semantic information about neighborhood relations of patches

Observing these *statistical measures*, or their modification by generalization respectively, allows assessing potential structural changes that influence the visual appearance of a map. The strategic goal when resolving size related conflicts is to preserve the given distribution of the patches of a category as far as possible. If, for example, most patches of a frequent category are very small, the overall structure would not be maintained if the deletion operator was selected for all patches. Structural and Gestalt constraints (e.g., maintain visual balance) would be violated. With the help of the above mentioned measures, different scenarios can be computed to assess if changes in total area and distribution can be tolerated before the actual generalization is carried out. For a first scenario, patches with an area just below the defined minimum size could be enlarged while patches where conflict violation is severe could be deleted. As a second criterion, distances to nearest patches of the same category could be observed as well. The further away a patch is from others of the same category, the more structurally important it is for its category. Further criteria (e.g., the importance of the categories involved) are possible depending on specific properties of the data.

Several algorithms that implement these operators can be found in the literature. For patches with only a single neighboring patch (islands), elimination is easy. The polygons' coordinates are simply deleted from the database (vector) or all cells of a region are assigned the value of the surrounding region. Cases where a polygon has more than one neighbor require more effort. Bader and Weibel (1997) have evaluated methods for this operator. They propose a solution based on the computation of a *skeleton* for the polygon to be eliminated (see figure 3). The area of the polygon is distributed equally among its neighbors and introduction of topological error is avoided.

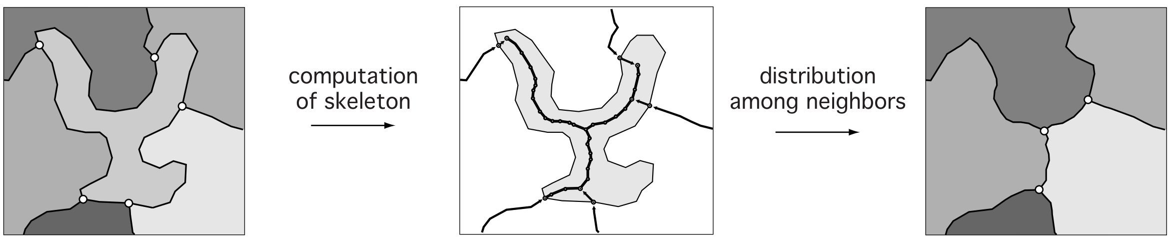

Figure 3: Elimination of a polygon using a skeleton algorithm (Bader and Weibel 1997)

A simpler method works with the polygons' nodes, which are displaced in the direction of the center of gravity. With this method, however, introduction of topological error is possible if complex polygons are eliminated (see figure 4).

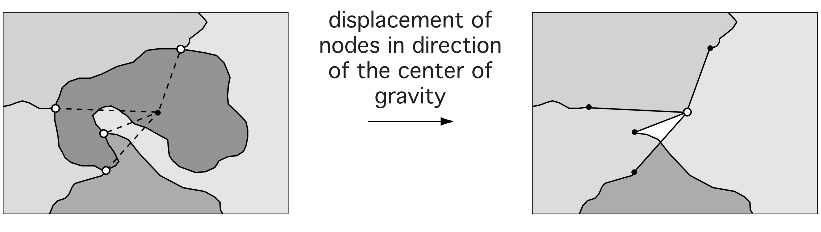

Figure 4: Topological error after application of the node displacement method (Bader 1997)

Methods for raster data require less algorithmic effort due to the data model. Normally, regions are eroded from the outside using for example a *majority filter*. More sophisticated rules reflecting structural and semantic knowledge can be defined to control the erosion process (Peter 1997). Polygons can be enlarged radially by scaling the vector between the center of gravity and the polygon nodes. This is basically the same method as the one described for elimination; therefore the same restrictions apply. Uneven expansion is possible (e.g., weighted with respect to "strong" and "weak" neighbor regions). Raster regions are enlarged by reclassification of cells of adjacent regions. As for vector data, rules for controlling enlargement can be implemented easily.

Detecting and resolving minimum size conflicts while respecting structural and Gestalt constraints can be conducted equally well for both vector and raster data. No model offers significantly better methods that would justify data transformation from one model to the other. In general, methods for vector data are computationally more complex but working with continuous data allows better control of results which might not always be possible with raster data. The advantage of raster data is that implementation is straightforward and rules respecting specific properties can be easily integrated.

Example 2: Amalgamation of Disjoint Patches of the Same Category

This example discusses measures and generalization algorithms available for *amalgamation* of patches of the same category. Amalgamation may be required for resolving minimum distance conflicts and, more generally, to reduce the number of patches and spatial variability in a map to meet specific map controls (e.g., map purpose). In addition to the measures presented here, the statistical measures mentioned in example 1 have to be observed as well to prevent violation of structural and Gestalt constraints.

Searching for candidate patches for amalgamation requires *distance measures* to be computed. One possibility is the computation of *buffers* for each individual polygon to both of its sides. Buffer width would be set to half the distance up to which patches should be amalgamated (i.e., half the minimum visual separability distance). Intersecting buffers will identify the desired situations. For polygonal data, a *cell* can be calculated as a measure for the degree of overlap (Bader 1997, Bader and Weibel 1997). As illustrated in figure 5 this cell can serve as a basis for the actual amalgamation process as well as for other operators such as *displacement* (Bader and Weibel 1997).

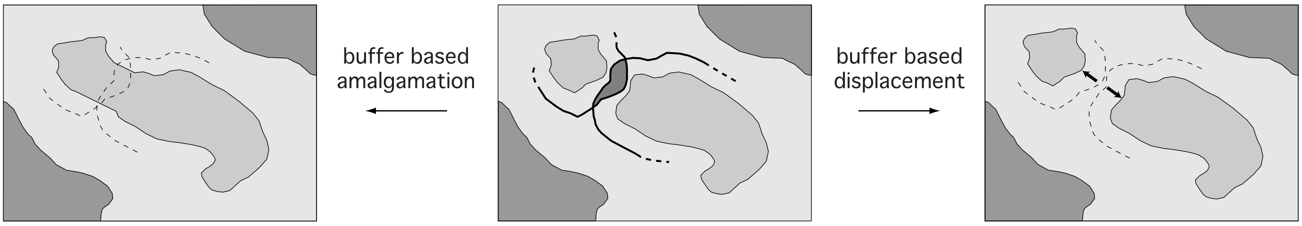

Figure 5: Buffer operations for amalgamation and displacement operators (Bader and Weibel 1997)

A second possibility is the computation of a *conforming Delaunay triangulation*. In this case, a global triangulation representing distance is calculated before decisions are made where amalgamation would be possible and useful (Bader 1997). The main advantage of this method is that, once computed, several alternatives can easily be tested. Triangulations may also be of good use for the actual generalization process. In general, triangles connecting the polygons are reclassified to the category of the polygons to be amalgamated. This may produce visually not very convincing solutions., More sophisticated solutions using curves to connect polygons are possible but require complex and computationally expensive algorithms. Methods based on Delaunay triangulations have for instance been implemented by Bader (1997) and Jones et al. (1995) who use constrained Delaunay triangulations not only for the amalgamation operator but also for polygon exaggeration and collapse (see figure 6).

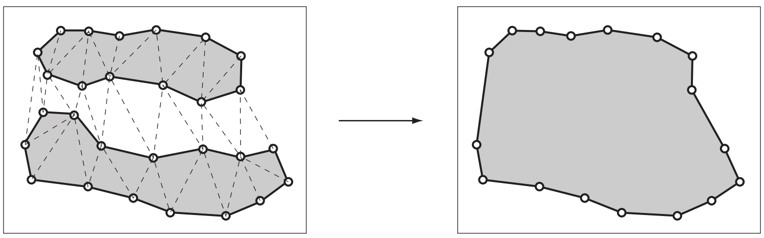

Figure 6: Amalgamation of polygons based on a Delaunay triangulation after Jones et al. (1995)

The use of *cost-distances* instead of Euclidean distances is a major advantage of raster data. This concept allows easy integration of semantic information and knowledge in the amalgamation operator. Important regions or their category respectively can be given very high costs to prevent parts of them from being eliminated due to amalgamation. On the other hand, assigning low costs to the respective cells can facilitate amalgamation over objects of unimportant categories. With this method it is, for instance, possible to prevent amalgamation of two forest regions over a narrow lake.

Algorithms for the amalgamation operator have been implemented by Schylberg (1993), Jaakkola (1998) and Peter (1997) for landuse/landcover raster data. Schylberg (1993) uses a simple *grow-and-shrink* algorithm. Objects overlapping or touching in grown state remain connected after re-shrinking by the same amount of cells. Although very simple to implement, this method might not be adequate in situations where spatial variability is high (see figure 7).

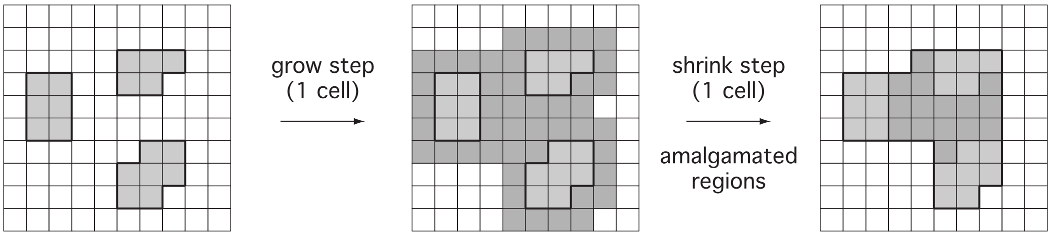

Figure 7: Grow-and-shrink algorithm for the amalgamation operator

Peter (1997) has adapted and modified a method proposed by Brown et al. (1996) using cost-distances. Cells are weighted according to the cost-distance to the *least cost path* between candidate regions. As an additional criterion, Euclidean distance to the nearest candidate region is considered as well. This results in a classification of the cells between and around candidate regions with the lowest values right between them provided the respective categories were given low costs. With this method naturally looking results can be achieved. Figure 8 illustrates this method schematically. Amalgamation of regions of category A is promoted over cells of category C (low costs) while cells of category B act as a barrier (high costs).

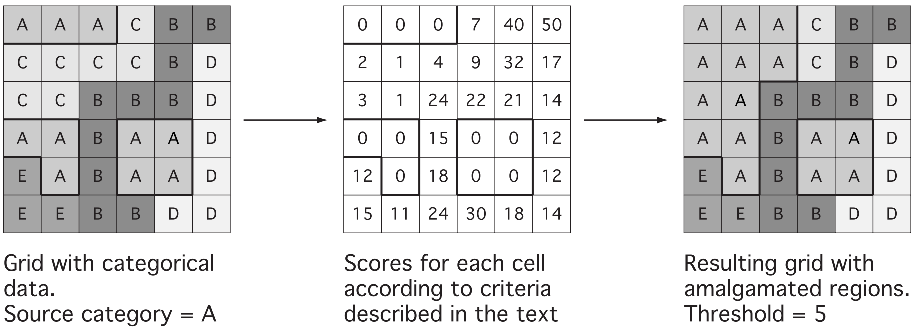

Figure 8: Amalgamation of regions based on cost-distance methods

In general, raster based methods for the amalgamation operator offer more flexibility and are easier to implement than methods for vector data. Appropriate measures are more easily computed and the use of cost distances is a major advantage. Furthermore, computation of triangulations is not trivial since numerous special cases have to be respected (Bader 1997). On the other hand, and this is the major problem with raster data, quality of the visual appearance of solutions depend to a large degree on the spatial resolution of data.

5. Integration of Vector and Raster-Based Methods

The purpose of this section is to show that although technically speaking the conversion of categorical data is a straightforward process, it is by no means trivial to handle when integrated with the generalization process. After discussing general issues related to generalization procedures where data model transformations are involved, examples are presented to demonstrate when and how conversion from raster to vector and vector to raster can be usefully integrated in a generalization strategy. The remainder of the section discusses specific possibilities of local data transformation.

5.1 Data Model Transformation: Applications and Limitations

In the previous sections it has been shown that integrated generalization systems can be developed for categorical data in a vector as well as in a raster environment. Means for translating constraints and generalization algorithms are commonly available for both data models. Since some methods can be implemented more easily or more precisely in one model, a generalization strategy incorporating algorithms from both data models, using their respective advantages has to be considered.

Several authors, for instance Bo et al. (1998), have proposed methods where vector data is converted to the raster structure, then generalized and finally transformed back to the vector model (vector – raster – vector). In general, this strategy makes more sense than the reverse (raster – vector – raster) since most raster operators, especially those involving neighborhood and contextual operations, are simpler and easier to implement than their counterparts in the vector domain. This may compensate for the relative loss of precision and semantic expressiveness that occurs when data is transformed. When data is reconverted after generalization, the source data (vector) might help with the interpretation of possible ambiguities that arise in the raster to vector phase. No convincing argument, however, can be thought of that would justify a generalization strategy where raster data is converted to the vector model for generalization with subsequent re-transformation back to raster structure. When a generalization strategy involving *bi-directional* conversion is implemented, various effects and problems have to be analyzed. Unavoidably, transformation of data between continuous and discrete reference systems and vice versa results in a loss of information and/or precision. Piwowar et al. (1990) have implemented several conversion algorithms and have evaluated them based on qualitative, quantitative and efficiency criteria. None of them could satisfy all requirements at the same time, meaning for example that an algorithm which minimizes changes in region area (for a given cell resolution) may displace and distort the same region heavily. Since bi-directional conversion usually *doubles* the effects mentioned and these effects cannot be easily controlled, we do not recommend it for general use despite the fact that some generalization algorithms might be easier to implement in a raster environment. In addition, parameters for vector to raster conversion (e.g., sampling interval) are usually defined globally for the whole dataset. This may result in partial or complete loss of important local geometric information that cannot be taken care of during the generalization process. Using *local* conversion, which will be discussed later, might provide a solution for this kind of problem.

We propose to distinguish between the *source representation* and the *target representation* of a map or a dataset. Commonly, one should tend to *maintain* the representation of the source data. That is, conversion is to be avoided unless it serves a specific purpose, certain operators are only available in a particular representation, or the intended target representation is different from the source representation. A specific purpose is, for instance, if data from different sources need to be integrated. Given a raster landuse dataset and several vector datasets, the landuse data would then be generalized in raster mode, transformed to vector and finally integrated (i.e. matched) with the vector data. This data or the constraints imposed by them should already be considered and respected during data generalization as far as possible to avoid integration problems as well as topological and semantic error (e.g., isolated rivers could exist after a small lake has been deleted). *Smoothing* of the outlines of patches is an operator that is executed more precisely and flexibly with continuous than with discrete data and may therefore require data structure transformation from raster to vector. Converting data to the raster model applies a discrete sampling distance to a vector dataset. Theoretically, if the sampling interval (i.e., the spatial resolution) was chosen to be equal to the machine precision (e.g., float), then a regular raster could represent the geometry as precise as vector data. However, that seems impractical. Even if storage costs could be neglected, the excessively high resolution would do away with the advantages of raster in neighborhood operations because the "instantaneous field of view" (e.g., a 5x5 kernel) would only cover minute portions of the dataset. The *sampling theorem* is more practical to define and optimizes the resolution of a raster dataset. It can make sure that geometric accuracy is not lost unintentionally. In that case the resolution of a dataset has to be selected twice as high as the dimensions of the smallest patch that should be resolved. However, oversampling as mentioned above can be used deliberately in order to obtain a smoothing effect.

5.2 Raster to Vector Conversion

Smoothing of the outlines of complex patches is a typical example, which may require conversion from raster to the vector structure. The visual quality that can be achieved by smoothing algorithms in the raster domain, for instance *mode filtering*, or *erode smoothing* (Monmonier 1983) is limited by the resolution of the dataset. Other raster based methods work with *resampling* of the raster grid to a higher resolution (*oversampling*). Depending on the oversampling factor chosen (e.g., 4) each raster cell would then consist of a number of *sub-cells* (e.g., 16 for factor 4). The smoothing effect is achieved by removing or adding sub-cells as illustrated in figure 9. This method can result in a considerable increase of the amount of data and stepped lines may still be visible unless the size of the sub-cells is below the minimum visual separability distance.

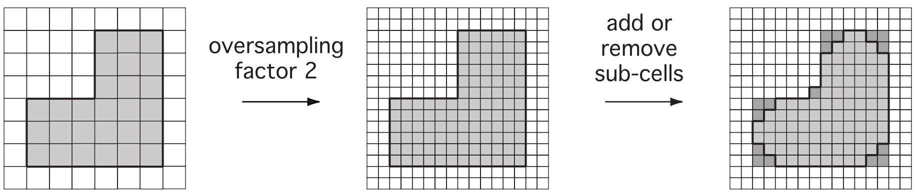

Figure 9: Raster mode generalization based on oversampling

Continuous geometry provides better means for smoothing the outlines of patches. Conversion to the vector model produces polygons that exactly match the outlines of the regions they represent, meaning that only right angles occur. The main task of the smoothing operation is to remove *stepped lines*. As Peter (1997) has shown, the commonly used line simplification algorithms (e.g., Douglas Peucker) will not yield the desired results and may even destroy the effects of the previous generalization operations. A simple yet effective algorithm has been developed by Herzog et al. (1983).

Designed for the simplification of boundaries extracted from raster data, the algorithm identifies regularly stepped portions of polygon outlines and replaces them by straight lines (see figure 10). This method can easily be modified to meet specific requirements. After its application, various line simplification and smoothing algorithms can be employed for further refinement.

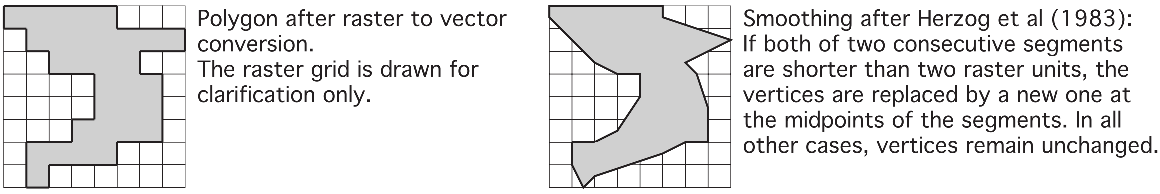

Figure 10: Smoothing of transformed raster regions in vector mode after Herzog et al. (1983)

5.3 Vector to Raster Conversion

Cases where vector data is converted to the raster structure for generalization without subsequent re-transformation (bi-directional conversion) are rather rare. A possible application might be that the *target representation* is raster and that vector data (e.g., line or point features) need to be integrated with an existing raster dataset. In such cases integration should take place *before* any generalization process is executed. Since positional precision and semantic information are partly lost during the conversion process, it would not be of great use to generalize vector data prior to vector-raster transformation. Furthermore, the structure of the existing regions will alter when data is integrated, as will the pre-conditions for the generalization process. Basically, all cells that intersect with a line are assigned the value of the newly integrated category. A *thinning* algorithm can be applied to reduce the width of rasterized linear features to one cell for datasets with a course resolution. This operation improves the visual quality of rasterized lines but may cause considerable displacement. If regions are built, 8 cell connectivity should be used to allow diagonal connectivity of cells. The size of a point feature in the raster model is always one cell. Symbolization will be lost during conversion as will, at least to a certain degree, positional accuracy. If the resolution of the dataset is below the minimum visual separability size at target scale, rasterized point features have to be either enlarged or cannot be represented at all. Rasterized point features, or their containment information respectively, can be used to avoid topological error from being introduced by generalization. Rasterized lines can be given high costs to control amalgamation when cost-distance algorithms are used as mentioned in section 4. Figure 11 illustrates the concept of converting and integrating vector data with raster data and subsequent raster based generalization.

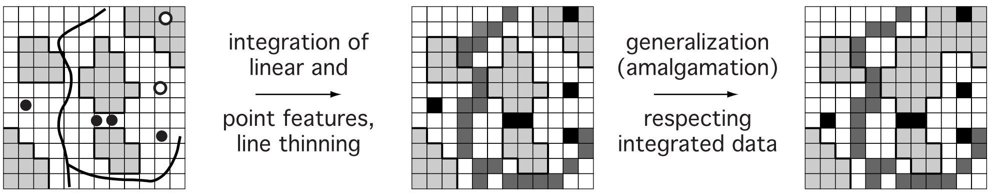

Figure 11: Integration of vector data through vector to raster conversion

5.4 Local Conversion

Some of the above mentioned problems with generalization strategies involving bi-directional data conversion can be avoided if the transformation is kept *local*. The principle is illustrated in figure 12. Such transformations normally only occur from vector to raster, where generalization operations are applied, followed by subsequent re-transformation to the vector model (vector – raster – vector). The main advantage of local conversions is that the sampling resolution for the raster part can be coordinated with the specific properties of the patches involved and the planned generalization algorithms. Algorithms in question are mostly those which are more easily implemented for raster than vector data, like methods that use cost-distances or involve neighborhood operations. For the implementation of a local transformation, the minimum bounding rectangle of the desired polygons or area of interest is calculated. A margin is added to avoid edge effects. After applying the generalization algorithms, data is re-converted to the vector model. A major drawback of this method is that the computational effort might be considerable for large datasets and/or smaller portions where methods involving local conversion are applied. Furthermore it should be pointed out that, in principle, the same problems and restrictions as noted for global transformation apply to local transformations as well. But since the amount of data will be rather small in most cases and all relevant parameters can be selected for a particular local situation, loss of information and positional precision can be better controlled.

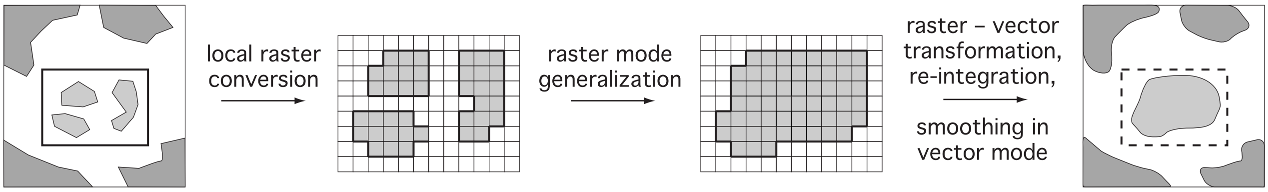

Figure 12: Generalization based on bi-directional local data conversion

6. Conclusions

Appropriate means to flexibly decide which operator to use in a particular situation at run time are not currently available in generalization systems. These systems neglect important constraints imposed by cartographic principles which may therefore lead to detrimental results of generalization solutions. The approach presented in this paper provides the fundamentals for the development of comprehensive strategies for the generalization of categorical data. Based on a set of generic constraints representing cartographic principles governing the generalization process and their subsequent formalization, tools are provided to control and steer the generalization process on all levels of observation as well as to evaluate the results. However, until a fully operational constraint-based generalization system is available further research is necessary. Besides the development of improved algorithms for the treatment of categorical data in vector and raster format, problems of system coordination at run time, such as operator and algorithm selection and adequate prioritizing of constraints, still remain partly unsolved.

The use of generalization systems involving bi-directional conversion cannot be recommended for general use. In most cases the advantages of raster based methods (ease of implementation, neighborhood operations) will be more than neutralized by the problems caused by the transformations operations. In principle, we recommend to avoid data conversion if possible unless it is required because the target representation is different from the source representation. The only situation where conversion might be useful is smoothing of patches outlines which is more precisely executed in vector mode where no restrictions due to coarse resolution exist. Methods that involve local conversion look promising since many of the problems reported can be solved if conversion is kept local but issues related to performance, optimal sampling resolution and data partitioning require further investigation and empirical testing.

Acknowledgements

This research is part of project "Integrating Methods for Quality Evaluation in Cartographic Generalization" supported by the Swiss National Science Foundation under contract 20-52759.97.

References

Bader, M. (1997): Methoden zur Erkennung und Lösung von Metrischen Konflikten in der Generalisierung von Polygonmosaiken. *M.Sc. Thesis*. Geographisches Institut der Universität Zürich.

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

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

Brassel, K.E. and Weibel, R. (1988): A Review and Conceptual Framework of Automated Map Generalization. *International Journal of Geographic Information Systems*, **2**(3):229-244.

Brown, N.J., Wright, S.M. and Fuller, R.M. (1996): A Technique for the Removal of Outliers During a Computerised Map Generalization Process. *Cartographic Journal*, **33**(1):11-16.

deBerg, M., van Kreveld, M. and Schirra, S. (1998): Topologically Correct Subdivision Simplification Using the Bandwidth Criterion. *Cartography and Geographic Information Systems*, **25**(4):243-257.

FRAGSTATS (1994): Spatial Pattern Analysis Program for Quantifying Landscape Structure. *Documentation of Version 2.0*. Oregon State University.

Herzog, A., Brassel, K., Bopp, M. and Dorigo, G. (1983): Ein System zur Generalisierung von Grenzliniendateien Administrativer Einheiten der Schweiz. *Geoprocessing Series*, Vol. 3. Geographisches Institut der Universität Zürich.

Jaakkola, O. (1998): Multi-Scale Categorical Databases with Automatic Generalization Transformations Based on Map Algebra. *Cartography and Geographic Information Systems*, **25**(4):195-207.

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.

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

Monmonier, M.S. (1983): Raster Area Generalization for Land Use and Land Cover Maps. *Cartographica*, **20**(4):65-91.

Muller, J.C. and Wang, Z. (1992): Area-Patch Generalization: A Competitive Approach. *The Cartographic Journal*, **29**(2):137-144.

Peter, B. (1997): Ableitung von Generalisierten Bodennutzungskarten aus der Arealstatistik der Schweiz 1979/85. *M.Sc. Thesis*. Geographisches Institut der Universität Zürich.

Piwowar, J.M., Ledrew, E.F. and Dudycha, D.J. (1990): Integration of Spatial Data in Vector and Raster Formats in a Geographic Information System Environment. *International Journal of Geographic Information Systems*, **4**(4):429-444.

Ruas, A. and Plazanet, C. (1996): Strategies for Automated Generalization. *Proceedings of the 7th International Symposium on Spatial Data Handling*, Delft, The Netherlands, Vol.2:6.1-6.17.

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

Su, B., Li, Z., Lodwick, G. and Müller, J.C. (1997): Algebraic Models for the Aggregation of Area Features Based upon Morphological Operators. *International Journal of Geographic Information Science*, **11**(3):233-246.

Su, B., Li, Z. and Lodwick, G. (1998): Morphological Models for the Collapse of Area Features in Digital Map Generalization. *GeoInformatica*, **2**(4):359-382.

Weibel, R. (1996): A Typology of Constraints to Line Simplification. *Proceedings of the 7th International Symposium on Spatial Data Handling*, Delft, The Netherlands, Vol.1:9A.1-9A.14.

Weibel, R. and Dutton, G.H. (1998): Constraint-Based Automated Map Generalization. *Proceedings of the 8th International Symposium on Spatial Data Handling*, Vancouver, BC, pp. 214-244.