Section 08

Generalization
Généralisation

Session / Séance 43-D

Automated Cartographic Generalization: Approach and Methods

Jean-François Hangouët, Sylvie Lamy

Laboratoire Cogit, Institut Géographique National
2-4 av. Pasteur, 94165 Saint-Mandé Cedex France
{Jean-Francois.Hangouet , Sylvie.Lamy} @ign.fr

Abstract

Cartographic generalization is the traditional process of drawing a new map from a reference map. Although geographical databases, as digital counterparts of paper maps, have been equipped with tools for interactive derivation, the full automation of the generalization process, a pointless or Utopian accomplishment for some indeed, remains an open and much-investigated issue. Wording out the intrinsic complexity of generalization not as a useless truth (eg. "generalization is an art") nor as a collection of fortuitous problems but as a remote and fragile imitation *of geography, the first part of our paper ends with the conclusion that geographical reality must be the focus of and the inspiration for all of the researcher's activities. The following section describes a rigorous, epistemological approach to bring this reality ever-present in mind, that is called* phenomenology *(with some differences however from philosophers' phenomenology). Phenomenology is about how to search through data for the geographical phenomena they stand for, and how to design specific measure or transformation algorithms consequently. In order to identify meaningful topographical organizations, spatial and contextual relationships have to be recovered on the data that come separately within so-called "geographical" databases. Phenomenology here advocates that they be reconstructed with methods that are respectful of the data in their wholeness. Such methods exist: the third section describes the* Voronoï diagram on segments*, a mathematical structure on the objects' full geometry that is the exact dual of the data themselves (which usual Voronoï tessellations or Delaunay triangulations on points are not). With these methodological and practical foundations, automated generalization proves both feasible and tractable. Incidentally, it appears that algorithms and practices already in use are neither negated nor swept out by our new approach and methods but find instead a framework where they can be objectified or evaluated. The fourth and last part of the paper describes the application of our principles to the programming of the* typification *operator, applied to buildings in a peripheral cityscape of scattered buildings and streets.*

On the Difficulty of Cartographic Generalization

The difficulty of automating generalization is often expressed in computer terms: « *automated generalization is (...) an 'NP-complete' problem (i.e. a computational solution cannot be devised)* » [Müller *et al*. 1995], or « *the complexity of the rules and knowledge involved in cartographic compilation (and generalization) is truly mind-boggling...*» [Morehouse 1995], or « *it is very hard to imagine that information theory will provide a model for all aspects of generalization* » [Bjørke 1993]. Another argument lies in contrasting the systematicity or stubbornness of the computer with the art of the cartographer: « *With generalization, art enters into the making of maps* » (an early quotation of Eckert given in [McMaster 1989]), and art cannot be modelled. The aim of this section is to reformulate why automating generalization is difficult, with an effort to free the description from the bias of both purpose ("automation") and means ("computer") and to refocus instead on the object at stake, viz. generalization *per se*.

Cartographic Generalization

Cartographic generalization may be defined by the range of activities involved: *simplification, classification, induction, symbolization* [Robinson and Sale 1969], or *selecting, sketching, harmonizing* [Cuenin 1972]. It may also be defined by a range of commanders: das "*Fallen unter dem Maßstab*" ("falling under the -map-scale", objects must be drawn with what tools are used by the cartographer), *legibility* (objects must be seen on the map), *expressiveness* (objects must be conveyed on the map) [Eckert 1921]. It may still be defined by a range of motivations: *economic*-, *data robustness*-, *multipurpose*- and *display/communication requirements* [Müller 1991]. A scrupulous definition, striving for explicit exhaustive coverage, would either be fastidious or turn into a textbook.

Suffice in fact to say that cartographic generalization is the drawing of a map from a map or from field data.

This straightforward enough definition obviously encompasses what the above activity names suggest, with the exception of "induction" — of which it must be said however that it turns out to be an imprecisely evocative denomination: broadly referring to the « *application of logical inference processes to cartography* » [Brassel 1990 : p.7] (trans. from the German), even to keen commentators it either seems a « *more abstract element* » [McMaster and Shea 1992 : p.19] or tends to elude description: « *induction cannot be entirely consciously considered* » [Morrison 1974 : p.119]. *Induction*, as the creative process of establishing laws from the evidence of facts, fits with what is intuitively conveyed by "drawing a map from a map" only convolutedly. In fact, efforts to bring it into cartographic disposition amount to construct on a confusion started by the simple play-on-word that rashly prompted "induction": that between "generalization" in its cartographic acceptation and "generalization" in its logical acceptation. Those two activities actually differ as much from each other as copying differs from creating.

If the intrinsic picturesqueness of the expression "induction" fails to provide us with any explanation, its very awkwardness to enlighten "map generalization" points however to the fact that generalization is no easy performance. The difficulty of generalization, as will now be seen, is explained naturally and confidently when it is considered as an *imitation*.

Cartographic Generalization and Imitation

To draw a map from a map is to make an *imitation*. Such a statement, describing "drawing a drawing from a drawing" as "imitating", may appear at first poorly informative because self-redundant; yet by ranking generalization as a copying process it insists on its "deterministic" quality (output cannot depart much from input). Second thoughts may object that output can be much different indeed from input — but imitation is a matter of copying, not a matter of copying out. Before clarifying this point by detailing what exactly is imitated, a few remarks are necessary to dissipate whatever uneasiness may be felt at seeing a word charged with deprecation applied to so prestigious an activity as map-making.

"Imitation" is not a "positive" word indeed. In the wake of Plato who said of the imitator that he « *is a long way off the truth* » [Plato X : p.365], that he is a maker of a lesser kind [Plato X : p.373], the imitator has been considered with tacit and even open distrust in everyday life, arts, politics, science… Probing into the intrinsic characteristics of "imitation", instead of lazily taking the expression for granted, will help us reveal the conditions of successful imitation and thus reassess the imitator's task. The list below summarises a discussion given in [Hangouët 1999]. For each characteristic are added a short explanation and an illustrative map case.

- *Resemblance is not symmetric*

- What imitates strives to approach what is imitated, while what is imitated stands passively aloof.

- Measurements of the Hausdorff distance between generalized and reference features have shown that the generalized map is closer to the reference map than the reference map is close to the generalized map [Abbas-

Hottier 1992 : pp.32sqq.].

- *Both what is imitated and what imitates have sense*
- What is imitated has sense by itself, and what imitates has sense by itself and by referring to what is imitated.
- Both reference and generalized map are meant to be used.

- *The generality of what is imitated explains what imitates*
- What is imitated can be mistaken as what imitates without prejudice.
- Large-scale maps may be used to solve problems that are solvable with small-scale maps.

- *What imitates practically limits the generality of what is imitated*
- What imitates cannot be mistaken as what is imitated without prejudice.
- Small-scale maps cannot be used for all applications of large-scale maps.

- *Actual comparison makes things clear*
- When in doubt, actual comparison only makes it possible to identify what is imitated and what imitates.
- Comparing a reference map with its derivative helps to see into either's meaning.

It becomes obvious that the imitator's task is not merely to make a product that looks like the original, but, while so doing, to be constantly aware of the dissymmetric distance between what he copies and what he makes, either to deceive or to serve. The counterfeiter will endeavour to blur any possible clues for authentication, while the open imitator will stimulate the users to discern the original sense through the evidence he proposes. Our imitator does not imitate by chance and, contrary to Plato's imitator (« *the imitator has no knowledge worth mentioning of what he imitates* » [Plato X : p.371]), he does have a keen understanding of what he imitates, and a deep respect for it: the imitator must be admired, besides his evident skills, for his ethical rectitude.

As to the cartographer, he also is aware of the representative status of his production, as the solemn use of the word "fix" in the following quotation by Max Eckert conveys: « *When it comes to cartographically fix any form, there must be carried out a serious reflection beforehand to recognise the characteristics of the general organisation and to render it clearly* » [Eckert 1921 : p.331] (trans. from the German). His accomplishment also is decided on ethical involvement, as is recalled or testified by Mark Monmonnier's provocative book title « *How to lie with maps* » or the unexpected word "honestly" in the following quotation by Jean Barbier: « *the work of the cartographer* (...) *is to completely comprehend the subject, to dominate all its aspects, to show all its mechanisms and to reconstruct for the reader a graphical image illustrating honestly all he has been able to understand of the reality, as far as he can go into detail having regard to the scale and to the level of the map* » [Barbier 1965: p.168].

Cartographic Generalization as Imitation of Geographical Reality

The latter quotation, incidentally, provides us with what exactly is imitated with generalization: not the reference map, but what the reference map itself imitates, viz. *reality*. Not any kind of reality however, but *geographical reality*, i.e. *what, in the real world* ("reality"), *graves the Earth* ("geo-graphic". Note however that, from a linguistic point of view, the analogy between "grave" and "graphic" may be purely accidental: according to [OED 71 : entry "grave"], between Old English "grafan" and Greek "graphein" « *the connexion* (…) *is no longer accepted by philologists* »).

The Difficulty of Cartographic Generalization

The difficulty of cartographic generalization can thus be said to lie in the ethical submission to geographical reality. The cartographer's own complacency and creativity must be dedicated to the tracing of reality through the given map or field data and to the conveyance of it in the produced map.

The following section proposes a description of the cartographer's apprehension of geographical reality through the given map or field data, and how it can benefit or inspire an approach to the research on the automation of generalization.

Geographical Phenomena

In this section, we describe what to the cartographer's eyes inflates the symbols he sees with geographical meaning as *geographical phenomena*. The notion is extended to digital geographical data, and it is argued that the obvious lack of ethical involvement on the computer's part that makes it blind to the data it will have to handle and process can be compensated by a rigorous attitude on the researcher's part, that of thinking in terms of *phenomena* instead of *data*. This attitude or method is described under the name of *phenomenology*.

The Geographical Phenomenon, from the Traditional Generalization Standpoint

What the cartographer sees, in the reference map or in the field data which he has to rework into a new map, is not a collection of coloured lines and shapes or of labelled items. It is not a collection of signs he sees either, but rather what the signs stand for. We will call what with the sign attracts the perception and use of it to its intended geographical meaning the "geographical phenomenon" of the sign. In fact, the sign, being a drawing on a sheet of paper, is contingent, while the phenomenon, which is grounded in the real world and of which the sign is but an echo, is unconditional: if the notion of "geographical phenomenon" can be introduced from the familiar notion of sign, it deserves however to be defined for itself, more independently.

The "geographical phenomenon" is the pair constituted by:

1. the real world feature homologous to the sign
2. the immersion of this homologue in a geo-graphic explicability

"Geo-graphic" with an hyphen: to insist that the adjective is to be taken in its full sense of "that graves the surface of the Earth".

Take this map. The curved bamboo stick is a sign. The real world feature it stands for is the local dominant swell. The swell is at its purest at some distance from the islands, from which it seems to radiate (or does it ?).

The cartographer in his activity musters not only his draughtsmanship, his obedience to the intended map's specifications, the given in the reference map — but also his "knowledge" and "experience" of the real world.

The Geographical Phenomenon, from the Automated Generalization Standpoint

Computers have no knowledge, « *neither aptness nor experience* » [Rimbert 1990 : p.76] (trans. from the French), nor for the matter ethical involvement which enables one to seize the world in its fullest. Computers are designed to munch data, while the meaning of the processes and the meaning of the data stay with the designers. If this happens to sound obvious to some, it can be reworded more incisively: Designers may be contaminated by the computer's blankness, e.g. they may be led to mistake a compression algorithm (the famous Douglas & Peucker) for a generalization operator. While we researchers tend naturally to think in terms of phenomena (the "road", the "house", the "ridge"…), we may also overlook the scope of the compromise imposed by the simple-mindedness (if any) of the system to which geography is entrusted and thus be led to content with useless or even wrong approximations. The situation of geography in the computer can be compared with that of exact arithmetic, but for the fact that, while much effort is being devoted to the automation of exact mathematics, the computer is still widely considered as the vector of a new rather than a limited cartography. This is why fixing what digital data stand for, i.e. insisting again on the geographical phenomena that deepen them with meaning, is no idle effort.

The “geographical phenomenon” is the pair constituted by:

1. the real world feature homologous to the data stored in the database,
2. the immersion of this homologue in a geo-graphic explicability.

This definition regroups the importance of data, real world, and carto-graphy at once. None of them must be sacrificed on account of would-be efficiency. To the researcher that hastily makes do with purely computational algorithms, the definition opposes the loyalty to the real world and the faith in one’s knowledge and experience. To the researcher that despairs of the feasibility of automation because of the profusion of cases in the real world, it recalls the inescapability of data and the grasp they provide on the real world.

Our definition of the expression “geographical phenomenon”, which researchers widely use in their articles to justify allusively their works, but never care to dissect nor justify, objectifies the motivation for their researches. Nevertheless, as such, it leaves one puzzled. So what ? Shall I post this as a motto over the door ? Shall I have it blinking or flapping through my screen ? Or, to the less cynical, How can I remain true in my allegiance to geography ?

Phenomenology

We have decided to call *phenomenology* the approach of the automation of generalization that is based on the acceptation of the definition of the “geographical phenomenon”. Phenomenology can be considered either as an explicit method (when it is sensed to be prescriptive) or, rather, as an attitude (when it is recognised as descriptive). Starting as soon as the researcher is about to research, or even beforehand, it includes the following five movements:

1. The awakening to the continuous manifestation of the phenomena in the real world. Any fleeting or powerful signs from the world which evoke the generic geo-graphical phenomenon under a real feature need not be pondered upon yet, but they must be accepted as such and rejected on no account.
2. The wilful, obstinate questioning of the phenomenon to see it and circumscribe its explicability better. All means can be used, the more the better: the ‘naïve’ or regular or renewed practice of the landscape where one lives, photographs, novels, maps, poems, manuals, monographs, discussions with experts and non-experts, paintings, other researches and methods on similar data etc.
3. The frank condensation of the essence of the phenomenon, that is, with words, a formulation that is chosen or constructed for its expressiveness, that is at once descriptive (i.e. it sums up the accumulated images and explicability of the phenomenon), indisputable (no real occurrence is incompatible without ingrained explanation), and unconditional (no reference is made yet to the data).
4. The conforming of digital data with the phenomenon. This, from a computer standpoint, corresponds to a (spatial) analysis step to regroup or fracture data according to the phenomenon’s explicability principles. Most of the time, this consists in restoring the full surfaceness of features that are stored by their contours or axes, the natural segmentation of features that are stored globally, or the relative dispositions of features that are stored as lists instead of 2-D or 3-D pictures. The restoration must be consistent with what is allowed by the explicability of the phenomenon. When the measuring tools designed or used fail to totally trace what the phenomenon demands to be traced, this should be documented so that *a/* users be not deceived by what could turn out to be a pretext phenomenon, *b/* the tools be reworkable or prolongeable by subsequent research.

5.The conception of automated transformation tools, articulated around the lines of force peculiar to the geographical explicability of the phenomenon, and taking both groups of data and cartographic constraints as mere input data and parameters.

The first two movements aim at picking up the world through intensive consideration, mustering an attention that is required to reach wider than what scientific cartography usually suggests; the last two movements aim at making data tractable more intuitively. The central movement, the *reduction* of the phenomena to its simplest yet most potent expression, as well as the *reason* of the automated processes, is the necessary, crucial articulation.

The five movements do not pass off in turn but cumulate instead, each movement in the later movements adding its ever-deployment to its traceable influence. The ultimate stages thus tauten and drone under the tension of multiple new questionings - this however in the phenomenological context is no longer the frailty of hasty implementation or algorithmic distraction, but the only natural frailty of spatial analysis: the risk of missing something of what the ever-surging world has to tell us.

The appellation “phenomenology”, to be read in our context as “how geo-graphic phenomena can be discoursed upon generally and specifically”, is the natural elaboration on the expression “geographical phenomenon”. In our decision to choose “phenomenology”, we had to face the wide-reaching pregnancy of the word in philosophy. Our phenomenology actually partakes to Husserl’s phenomenology, inasmuch as what in our method could be discarded off as irrelevant subjectivity by cartesian scientists (i.e. the insistence on the researcher’s involvement with the phenomena instead of presuming of his abstracted observation of objects — which, to put it aggressively, would rank him next to computer), is seen by Husserl and modern phenomenologists as the very condition for reliable understanding and knowledge: « (Husserl reproaches Descartes) *with having abstracted the truths of geometry from the methodical doubt* ». [Desanti 1963 : p.25] (trans. from the French).

Voronoï Diagrams on Segments

Our general, conceptual approach to the automation of generalization being exposed, it remains to show or illustrate how it can be practically conducted. This section describes a general, mathematically sound structure that makes it possible to reconstruct and measure proximity relationships between objects, thus restoring the computerized lists of hollow data into map-like pictures. Section 4 will illustrate how this structure can serve the phenomenological designing of the generalization operator called *typification*.

Basic Notions

The Voronoï diagram of separate objects located in a metric space and called sites is the geometric locus of points that are minimally equidistant to the sites. The diagram tessellates the space into as many cells as there are sites; there is a cell around each site; any point within a given site’s cell lies closest to the site than to any other.

In our applications, the sites are the segments and points that compose the geometry of the features as they are stored in a vector database. The Voronoï diagram is thus made up of segments of lines (equidistance between two points, equidistance between two segments) and segment of parabolas (equidistance between a point and a segment).

Voronoï Diagrams in Geomatics

While Voronoï diagrams are well-known on punctual sites, and widely used (either as such or in the guise of the dual Delaunay structure) to identify proximity relationships between features (see e.g. [Peng *et al.* 1995], [Ware and Jones 1998], [Ruas 1998]), few papers describe Voronoï diagrams on segment sites. Most of them are to be found in computational geometry literature and focus on the computation of the diagram (an O(n . log n) problem made delicate by computer arithmetic, see e.g. [Boissonnat and Yvinec 1995] or the wonderful http://www.scs.carleton.ca/~csgs/resources/cg.html). Yet some of the properties of Voronoï diagrams on segments are reviewed in [Okabe *et al*. 1992] ; and their powerful applications for topological reconstructions in GIS have been investigated and demonstrated in a series of works by Christopher Gold, downloadable from the author's home page (http://www.scs.carleton.ca/~csgs/resources/cg.html). A variety of mathematical applications useful for GIS and generalization operations, including convex hull, minimal spanning tree, computation of the Hausdorff distance, skeletonization, dilation and erosion in vector mode, have been exposed in [Hangouët and Djadri 1997]. The reader is invited to consult these references for a thorough training over Voronoï diagrams on segments. In this paper, we will address what makes this structure on geographical data so magical from a conceptual point of view only.

On the Effectiveness of the Voronoï Diagram on Segments

The Voronoï diagram on segments misses nothing of the features stored in geographical vector databases, being computed on the segments and points that are the atomic components of their geometry. As argued in [Hangouët and Djadri 1997], Delaunay and (point) Voronoï structures in use are no more than unconsciously performed approximations of the Voronoï on segments. In these applications, a few points are strategically or empirically chosen on the data, then triangulated. Different geographical configurations can end up with the same triangulation, when the points to be processed happen to be chosen at the same locations. In addition to some kind of generalization (based on what geographical principles ? Why for a one choose the buildings' centroids and the streets' vertices, and not the buildings' corners and their projections on streets ?), this leads to a highly procedure-dependent and complex interpretation, which shows for example from the descriptions of [Ware and Jones 1998].

The Voronoï diagram on segments, conversely, is equivalent to the geographical data involved: there is only one diagram for a given figure, and one figure for a given valid diagram. In fact, the computation of the diagram being not of linear complexity, but higher, the Voronoï diagram adds something to the available geometry: the very cementation of the features into a whole and the possibility to measure exactly the empty spaces in between. When the Voronoï edges are "semanticized" as edges inside a same contour, edges radiating from a same feature, edges separating a same pair of features (see Figure 1), shape computations can be performed, distance computations can be performed, proximity computations can be performed: geographical data, originally stored minimally as separated axes or contours, are brought to their map-like status at last.

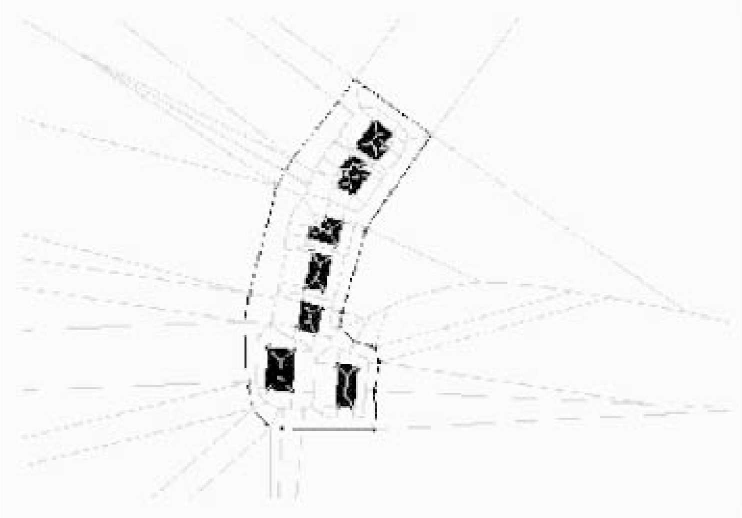


Figure 1. The Voronoï diagram is made up of line and parabola segments that limit the respective influence areas (loci of the closest points in the plane) of the points and segments that make up the figure. Edges within a contour constitute its "skeleton"; edges that separate two different features show them to be neighbours and make it possible to measure their proximity. Any geographical object participates fully to the construction of the diagram (here, buildings and streets).

The Voronoï diagram on segments, by taking into account the full geometry and semantics of the data stored in the geographical database to restore them in their map-like disposition, is the adequate tool for our phenomenological approach, one principles of which forbids to sacrifice data… How to use the tool efficiently for generalization purposes, without sacrificing geography either, is illustrated in the next section with the description of the *typification* operator.

An Application: Typification

Typification is one of the elementary generalization operators traditionally identified in the literature on automated generalization. It consists in transforming a group of features without defacing the group's appearance. The present section describes how the operator has been designed for groups of buildings, with an insistence on its phenomenological foundations and rectitude.

The Cartographic Compromise

With scale reduction, small objects may be legible no longer if not enlarged. Enlargement has no side effect when objects are isolated. When the density is very high, individual enlargements lead to a blackening of the area, which is visually sound and, computationally speaking, easily reprocessed into an aggregated object for subsequent handling in the computer. In intermediate densities however, individual enlargements of features lead to over-occupation: in fact, free spaces should also be enlarged to be legible — but the enlargement of both objects and spaces is physically impossible. The traditional solution consists in making room by getting rid of some of the objects and in reorganizing the remaining objects over the original footing: this is what typification is about. Lakes harmoniously spaced out, bends obstinately succeeding one another crawling up the steep mountain slope, streets that criss-cross the town, houses in the residential area: all are liable to be typified.

Typification, a Definition

Concreting a variety of descriptions from the literature on generalization, typifying can be defined as bringing into a cartographic representation, where the identified distribution is preserved, features of a same nature that happen to be grouped locally by some identifiable geographical process (see [Hangouët 1998 : p.227] for a list of references).

The Typification of Buildings

Our typification method is designed for rows of buildings.

Phenomenological grouping of buildings

Buildings come separately in the databases, their groups have to be indicated interactively, or computed. While the groups are sometimes computed on Gestalt principles, most notably by [Regnauld 1998] ("notably", because of the impressive results obtained) , i.e. on methods meant to emulate the perception of geometrical groupings, we have chosen the phenomenological approach. The phenomenon that explains that buildings are grouped in the real world is that of accessibility by streets: buildings are constructed along streets. This is why the area covered by the database is segmented into as many cells as there are minimally looping roads. Within each cell, standing for city blocks, each building is associated to the closest lying street's side (a straightforward operation with the Voronoï computed on buildings and streets). Then, buildings are grouped on their adhesion to a same street's side.

The group just obtained is of no use however for automatic processing. The succession of the buildings along the street has to be elicited. This again is straightforward with the Voronoï diagram on buildings: chain the buildings on their "sharing a Voronoï separation edge" characteristics. Our typification operator is designed to process rows only, yet not all natural groups are linear: this again reads clearly and quickly from the Voronoï diagram, as soon as a building of the group indeed happens to have more than 3 neighbours in the group, the group is labelled as "not typifiable by this method".

The row just obtained is of no use however for automatic processing. How could it be decided which buildings are to be selected for representation at the final scale? A big building is an important landmark, or an administrative building, or a strangely-shaped building… 7 characteristics may mark a building as different: nature, size, shape, absolute orientation, orientation on the street, distance to the street, distances with its neighbours [Hangouët 1998 : pp.202sqq.]. Except for the "nature" quality, all can be computed with precision and refinement by means of the Voronoï diagram [*id.*]. The characteristics of each building are compared to the mean characteristics on the group to decide on the difference or regularity of the building.

The row, informed with the contribution of each building, is ready to be typified.

Phenomenological transformation of grouped buildings

The final group will be represented in the original footing of the group, for both positional accuracy and topological consistency reasons — i.e. for obvious geographical reasons. The *n* buildings of the group are inflated with a factor *f* to be legible (*f* is a parameter of the algorithm. It is equal to the scale ratio when buildings are represented by a same symbol on the source and generalized maps). The number of buildings to be represented, *m*, is computed so as to best approximate the original occupation ratio within the original span of the row. Each *i*th final position is interpolated between the *j*th and (*j*+1)th initial positions, with $i/m = j/n$ (this trick ensures that much of the dynamics of the original configuration is conveyed). The algorithm is parametrized with cartographic preferences: buildings showing the indicated characteristics will be favoured over plain buildings: The *j*th and (*j*+1)th initial buildings are made to compete for representation at the *i*th final position: the building which has the more desirable characteristics is chosen. If the two rank equally in the preferences, the building closer to the final position is chosen. If the two lie equally close, there is no means nor reason to distinguish one from the other and either will do. Each selected building is represented either in its original if somewhat over-scaled shape or as an assembly of rectangles (buildings being built with right-angled walls), the choice being computed in function of the output scale, of the building's size and complexity, and of a specific "rendering preferred" parameter indicated to the algorithm.

This results in unsurprising, natural renderings (see Figure 2).

Figure 2. Buildings within a minimally-looping set of streets in a peri-urban area, and the typified version (scale ratio = 0.7)

Conclusion

Considering map-making as the imitation of what is engraved on the surface of the Earth has provided us with a natural approach to the automation of cartographic generalization: the researcher must think in terms of geographical phenomena instead of data. The imperious principle of phenomenology, "*every geographical phenomenon with its own automated methods*", seemingly announcing titanic and endless tasks, first inspires awe. It turns out however that those specific methods and algorithms rely on the expression of spatial relationships which Voronoï diagrams render systematic and straightforward. Our approach and methods are not limited to generalization purposes. Phenomenology applies to research in any domain where geographical data are expected, either together or in isolation, to be usable as maps of the world.

References

Abbas, I. and Hottier, P. (1992).Contrôle du tracé planimétrique d'une carte - Contrôle ponctuel et contrôle linéaire. Bilan Recherche '92, Bulletin d'Information de l'I.G.N. (1993) pp.30-36.

Barbier, J., (1965). Thematic Cartography. Problems Particular to Illustration. Annuaire International de Cartographie, vol. V, 167-171.

Bjørke, J.T. (1993). Information theory as a tool to formalize cartographic knowledge. In NCGIA Initiative 8 Specialist Meeting on Formalizing Cartographic Knowledge, October 1993. 45-52

Boissonnat, J-D. and Yvinec, M., (1995). Géométrie algorithmique. Édiscience international. Paris.

Brassel, K. (1990). Der Generalisierungsbegriff in der Kartographie und anderen Disziplinen, Schweizerische Gesellschaft für Kartographie : Kartographische Publikationsreihe Nummer 10. pp.5-16

Cuenin, R. (1972). Cartographie générale (tome 1 : Notions générales et principes d'élaboration). Eyrolles, collection scientifique de l'Institut géographique national.

Desanti, J.T. (1963). Phénoménologie et praxis. Editions Sociales, Paris.

Eckert, M. (1921). Die Kartenwissenschaft, Erster Band (vol.1). de Gruyter. Berlin und Leipzig.

Hangouët, J-F., and Djadri, R. (1997). Voronoï Diagrams on Line Segments: Measurements for Contextual Generalization Purposes. In S.C. Hirtle amd A.U. Frank (Eds.). Spatial Information Theory. A Theoretical Basis for GIS (Cosit'97). Springer Verlag, 207-222.

Hangouët, J-F. (1998). Approche et méthodes pour l'automatisation de la généralisation cartographique ; application en bord de ville. Thèse de doctorat, Université de Marne-la-Vallée, 352p.

Hangouët, J-F. (1999). On the Nature of Spatial Analysis. Submitted to Spatial Information Theory (Cosit'99), in press.

McMaster, R. (1989). Introduction to 'numerical generalization in cartography'. Cartographica 26(1), 1-6.

McMaster, R. and Shea, K.S. (1992). Generalization in Digital Cartography. Association of American Cartographers.

Monmonnier, M. (1991). How to lie with maps. University of Chicago Press.

Morehouse, S. (1995). GIS-map compilation and generalization. In J-C. Müller, J-P. Lagrange, and R. Weibel (Eds). GIS and Generalization, GISDATA n.1 (series Eds. I. Masser and F. Salgé), Taylor & Francis, pp.21-30.

Morrison, J. (1974). A Theoretical Framework for Cartography, Annuaire International de Cartographie, XIV, 115-127.

Müler, J-C., (1991). Generalization of spatial databases. In Maguire, Goodchild, Rhind (Eds). Geographical Information Systems vol.1 Principles. Longman Scientific Technical. 457-475

Müller, J-C., Weibel, R., Lagrange, J-P. and Salgé, F. (1995). Generalization: State of the Art and Issues. In J-C. Müller, J-P. Lagrange, and R. Weibel (Eds). GIS and Generalization, GISDATA n.1 (series Eds. I. Masser and F. Salgé), Taylor & Francis, pp.3-17

Okabe, A., Boots, B. and Sugihara, K., (1992). Spatial Tessellations, Concepts and Applications of Voronoi Diagrams. Wiley & Sons.

Oxford English Dictionary (1971). Oxford University Press.

Peng, W., Sijmons, K. and Brown, A. (1995). Voronoï diagram and Delaunay triangulation supporting automated generalization. Proceedings of the 17th International Cartographic Conference - 10th General Assembly of ICA. Barcelona, Sept. 1995. pp.301-310

Plato. The Republic. Vintage Classics 1991 (Eng. trans. by Jowett).

Regnauld, N., (1998). Généralisation du bâti : Structure spatiale de type graphe et représentation cartographique. Thèse de doctorat, université de Marseille.

Rimbert, S., (1990). Carto-graphies. Hermès.

Robinson, A. and Sale, R. (1969). 'Elements of Cartography'. 3d edition. John Wiley and Sons.

Ruas, A., (1998). OO-Constraint modelling to automate urban generalisation process. Proceedings of the 8th International Symposium on Spatial Data Handling. pp.225-235

Ware, J.M. and Jones, C.B. (1998). Conflict Reduction on Map Generalization Using Iterative Improvement. GeoInformatica, 2(4), 383-409.

Session / Séance C7-D

The Future of Automated Map Generalization

Brian E. Smith, Henry L. Jackson, *and* **Reid L. Maier**

Intergraph Corporation
Huntsville, Alabama, USA
besmith@ingr.com hljackso@ingr.com rlmaier@ingr.com

Abstract

One of the most prevalent trends in the map production industry today is the multiple-use of existing data stores. Geospatial data is expensive to capture and maintain, and the era of single-product data flow lines is rapidly being replaced by the need to produce, and sell, numerous representations and custom datasets from the same data store. In a digital GIS world, where data is essentially scaleless, there is the need for smart data selection or filtering, coupled with a variety of presentation capabilities and advanced processing techniques. In the cartographic environment there is an additional requirement for efficient, consistent and quality data generalization.

This paper will address the use of topological data structures and generalization tools in the map production process. The focus of discussion is on issues, requirements and solutions for generalizing digital vector data. Also discussed will be computer-assisted generalization techniques incorporating complex feature and spatial interrelationships, which go beyond the simple geometric rules of the past. The approach taken involves visual interaction between the user and a dynamic topological data structure in the selection and definition of generalization algorithms and parameters, followed by automated processing to achieve high-quality, reproducible, and cartographically pleasing results – every time.

Introduction

Map generalization is a process that alters the presentation of high-resolution map data so that it can be displayed appropriately to a user at a smaller scale. This process, it is well understood, is very subjective and requires the cartographer to make varying decisions about the presentation of map data based on feature content, density, and detail. Any attempts to automate the map generalization process requires capturing the knowledge a cartographer has as to where, why, and how generalization is performed. By capturing this type of information the generalization process can be reproduced in a consistent manner. This is a very challenging task for those trying to develop an automated map generalization system. Although the types of decisions a cartographer makes during generalization are numerous, they all require a similar analysis of the detail and density of the data with respect to the needs the user has for the map. A system to totally automate the map generalization process seems impossible at this time, but the tools do exist to make it considerably more efficient and reproducible in the digital world.

Regardless of the level of automation, any system designed to perform map generalization should provide a fundamental set of tools that can be used to analyze the map data and alter its display as necessary. Given a quality set of tools the cartographer has a good start in performing map generalization. These tools should allow the cartographer to view and analyze the map data to determine where generalization is required and also what type of generalization function should be applied. The cartographer should be able to view the results of different generalization functions on an area and make choices as to what best provides the desired results.

Once a choice is made the cartographer should be able to save the generalized map data as well as the information about what generalization functions occurred and all the associated parameters related to this generalization function. In this way a knowledge base can be created and maintained that will, over time, facilitate a higher level of automation.

The Architecture of a Generalization System

The architecture of a map generalization system should not only provide the tools required by a cartographer to perform generalization in a digital environment, but should also have certain other important characteristics. First and foremost it should be flexible. Every generalization process is unique and is dependent on many factors including the scale reduction taking place and the type of product being generated. The cartographer should have the freedom to generalize different features or regions of the map in the order he feels are appropriate. The ability to choose how a feature is generalized and to easily control the extent they are generalized is also very important. The generalization system should also be extendible. Every map production shop and every cartographer has a unique perspective of how generalization should be performed. The system should be able to incorporate these ideas. Also, as new innovations are developed, it should be possible to easily introduce them into the system. Figure 1 depicts a high level view of an architecture of a generalization system and the tools or components that comprise it.

Generalization Operators

The basis for any automated map generalization system is the generalization operators. A generalization operator is any discrete generalization function a cartographer may want to use to alter the detail or content of all or part of the map data. Some examples of these are the smoothing of a line to attenuate its detail or the aggregation of two or more areas or polygons into a single area. In every map production shop and for every type of map produced there are typical sets of operators used. A system to perform map generalization should provide this typical set of operators but also allow for the addition of more operators over time as the cartographer learns about the generalization process in the digital world. There are several generalization operators that have been well documented over time that provide a good base for a generalization system.

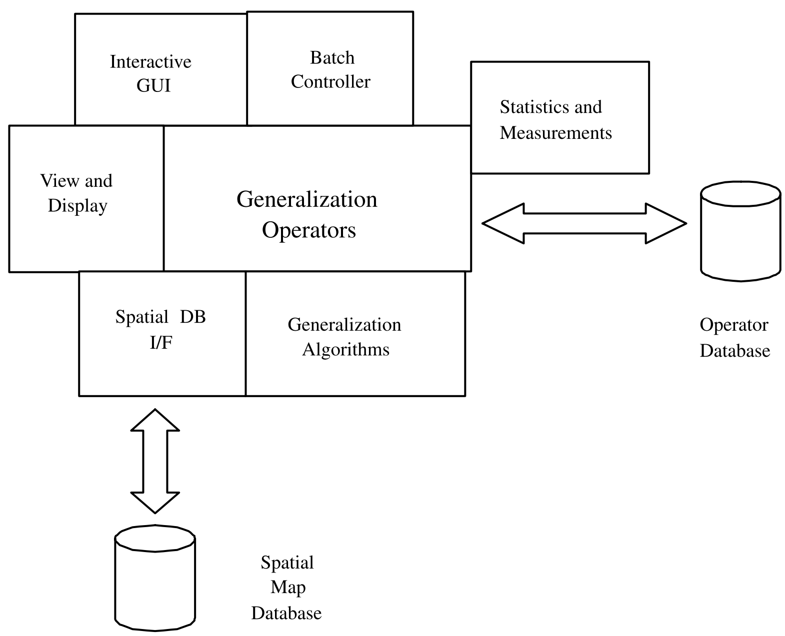


Figure 1.

Line Simplification

The line simplification operator removes unwanted detail from a linear geometry by removing those vertices on the line that are determined to be non-critical to its portrayal.

Line Smoothing

The line-smoothing operator attenuates the detail of a linear geometry by inserting and/or moving vertices on the line. This is done so that the line has a smoother appearance to the user.

Area Aggregation

The area aggregation operator merges two or more areas or polygons into a single area or polygon.

Point Aggregation

The point aggregation operator combines two or more points into a line or area.

Feature Collapse

The feature collapse operator replaces the geometry of an existing feature with a simpler geometry of lesser dimension. Some examples of feature collapse are the replacing of an area geometry with a line geometry along the centerline of the area or the replacing of an area with a point geometry located at its centroid.

Typification

The typification operator removes features in a specified region that is determined to be too dense. The features are removed in such a way so that those that remain leave a representative pattern of the features in this region.

Feature Elimination

The feature elimination operator removes features solely on the basis of whether the feature serves a purpose to the user of the map. Features may be eliminated by feature type or attribute value.

Conflict Detection and Displacement

The displacement operator moves one or more features that are in close proximity to other features. This is done so that the features are more easily distinguishable.

Generalization Algorithms

Behind each of the generalization operators are the algorithms used to derive the generalization results. Each operator can supply a number of algorithms designed to produce different results based on the input provided. The best example of this is line simplification or line thinning. A number of different algorithms have been developed over the years, each of which produces a simplified version of an input line based on its operating parameters. Some examples of line simplification algorithms are Douglas, Lang, Point Relaxation and Nth Point. Each of these algorithms is unique and might produce a different version of a simplified line given a specific set of parameter values. A map generalization system should provide a number of different algorithms for each of the generalization operators so the cartographer can experiment with these algorithms to determine an algorithm and parameter set that produces the best results.

View and Display

It is well understood that generalization by its very nature requires the cartographer to view the map data at product scale. In a digital world the ability to view the data at different scales can enhance the cartographer's decision-making capability.

Map Display

Because the display of a map on a computer screen is much different than how it would appear on paper in its final form, the need arises to be able to give the cartographer a more realistic display of the map data. Being able to display the data at the map or target scale on the computer screen is a necessity. Given this capability the user can make more appropriate decisions about whether the detail or density of the map data is appropriate.

Dynamic Display

Given the capability to view the data in its real context, the cartographer can determine in what regions generalization is required and apply the appropriate operators. However, each of the generalization operators will provide different results based on the algorithm chosen and the parameters used. It is important, especially for those cartographers just learning how to generalize in the digital world, to be able to experiment with different operators, algorithms, and parameters sets to see which ones provide the desired results for each situation encountered. A dynamic display where the cartographer can quickly change the algorithm or parameter values and see the effects of these changes is important. This gives the cartographer instant visual feedback, and once the desired results are obtained the map data can be saved to the permanent data store.

Operator Database

As the cartographer works in the interactive environment and applies various generalization operators, algorithms, and parameters, it is important to capture what generalization is being performed in different situations and why. By capturing this information the cartographer can apply similar algorithms and parameters in like situations. This not only saves time by not making the cartographer duplicate the experimental process for situations already encountered, but also ensures a more consistent application across the entire region being generalized. It is apparent that factors such as scale reduction, product type, and input data characteristics influence what generalization operators, algorithms, and parameters are to be applied. Therefore it is a good idea to partition this database by these factors.

Measurements and Statistics

It is important for any generalization process to record what and how features are being generalized. This allows the cartographer to get a good understanding of the types and extent of the changes that are occurring to the map data. Capturing the number and types of features generalized by a specific operator is relatively simple to do and can provide useful information to the cartographer. Also, being able to measure certain size and shape characteristics of a generalized feature to compare with the original feature could provide information as to whether a feature is being over-generalized or not. A simple example might be the change in coverage of an area feature or length of a line feature. A line feature whose length changed by more than a specified amount during the generalization process might be an indication that it has been over-generalized.

Spatial Map Database

A very important component of the generalization system is the database where the map data is stored. An intelligent database allows for more automation to be achieved with higher quality. Important to the quality of the generalization process is the maintenance of the spatial relationships between features. All generalization operators should be able to check for and report changes in spatial relationships and maintain these spatial relationships if desired. Any modification of the spatial relationships between features by the generalization operators gives a misrepresentation of the data. This type of quality checking is important for interactive generalization processes, but more so for batch processes where the cartographer does not have the visual capability to view the changes to the data as they occur. Generalization operators should maintain spatial relationships if possible but at a minimum report these changes to the cartographer for review.

Given the types of checks that need to occur in a generalization process in order to maintain the spatial relationships between features, an intelligent database where these relationships are stored or can be easily derived from is essential. A topological database where geometry is shared between features and contains spatial information makes this type of checking relatively easy. An example of a generalization operator applied in a topological database will show how quality can be maintained.

Line Simplification/Smoothing

When performing line simplification or line smoothing it is important that existing spatial relationships between features are maintained. For example features that are coincident should remain coincident, features that intersect should remain intersecting at the geographic location they originally intersected at, and features to the left or right of a feature being simplified should remain to the left or right.

In a topological database, linear geometry can be shared between coincident features. Common geometry is stored in the database once and is pointed to by the features that use it to convey its location and shape. When this common geometry is simplified or smoothed the detail of the features that use this geometry is simplified or smoothed. Both features are modified in the same way. Coincidence is one type of spatial relationship that is stored in this type of database. Another is an intersection. When two or more linear geometries cross, the linear geometries are broken with the intersection point serving as an endpoint to these new geometries. Because intersection points are stored in the database, they can be easily retrieved and used as critical points to a line simplification process. Once critical points are identified, it can be ensured that they are not deleted or moved. A final example of a relationship that is stored in a topological database is orientation. The regions to the left and right of all linear geometries are stored making it easy to determine what features are to their left and right. Using this information the simplification and smoothing operators can ensure that what is to the left or right of a linear geometry remains to the left or right after line simplification or smoothing occurs. An example would best illustrate how a topological data model could help in the simplification process (Fig. 2a). A river is being simplified that is composed of three linear geometries L1, L2, and L3. The linear geometry L2 also belongs to a tree area. L2 helps define the geometry of both the river and the tree features. The river can

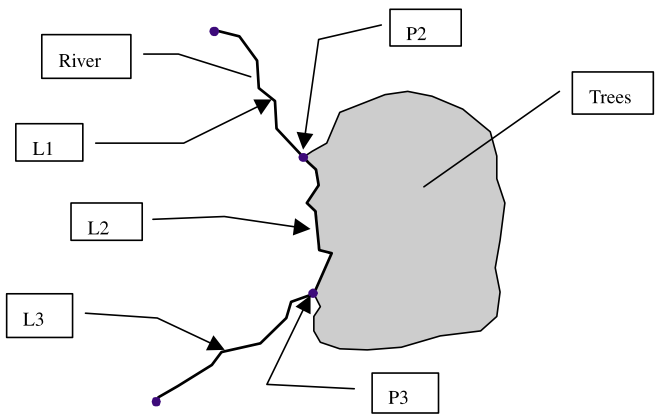


Figure 2a.

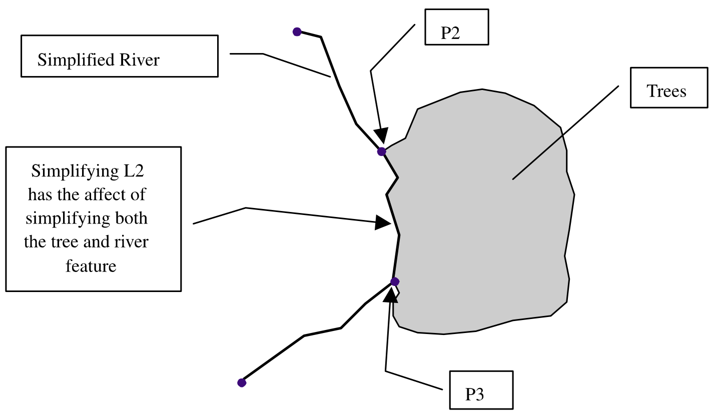


Figure 2b.

be simplified by simplifying the three linear geometries that compose it. When L2 is simplified this in effect simplifies that part of the tree area coincident with the river. Coincidence is maintained between the river and the tree area (see figure 2b). Note that simplifying each geometry separately also ensures that the points of intersection between the river and trees (P2 and P3) are maintained. This is just one example of how an intelligent data model that contains spatial relationships can help ensure that the generalization process does not compromise the integrity of the map data.

Generalization Workflow

When deriving a workflow to perform generalization in the digital world, it is important to understand the factors that could affect generalization results and thus influence the workflow. One of these factors is the scale reduction that is taking place between the source and target scales. For example if the scale reduction is large - say 1:10000 to 1:500000 - it is clear that much of the work required by the generalization process will be eliminating features. Another factor is the product being produced. Certain features may not typically be shown on certain products, therefore they need to be eliminated. Also with a digital product some of the detail that would otherwise be removed for a hardcopy product could remain. The order that generalization operators are applied also affects the results of the generalization process. Therefore, careful consideration must be given to the workflow of steps used to generalize features. Data that defines feature relationships, is well attributed, and contains spatial relationships can potentially have more generalization operators applied in an automatic process. These and other factors influence the cartographic results of generalization therefore it is important to provide a system that has a wide variety of tools that the cartographer can use and allow for the development of a flexible workflow that best fits the given situation.

Interactive Generalization

As much as a totally automated system to perform map generalization is desired, at this time nothing can replace the decision making process of the cartographer. Given the appropriate tool set the cartographer can view the data on the display, determine locations where generalization are required, and apply generalization operators as needed. Since map generalization is such a subjective process there is a need to give the cartographer visual feedback of the results of a generalization process so it can be viewed, modified and eventually accepted as part of the generalized map.

Interactive generalization can be both a pre and post process to an automatic generalization process. As a preprocess interactive generalization can be used to analyze the map data to determine the generalization operators to be applied, set up and save the generalization operators and parameters to be executed in batch, identify features and regions not to be generalized, and apply generalization operators that require the cartographer to give an approval. As a post process interactive generalization can be used to view the results of a batch generalization process, resolve any queue entries generated, apply generalization operators to regions that were not generalized appropriately, and set up for additional batch processes.

Interactive generalization must provide several approaches or methods for performing generalization. Cartographers that are very familiar with the different generalization operators and algorithms may want to perform generalization tasks by operator. For example the cartographer may choose to perform line simplification to all the features prior to performing area aggregation. Another approach might be to generalize by feature class. The cartographer may perform all the generalization operators to rivers or drainage features before moving on to roads and buildings. Once a generalization database has been developed and saved, a cartographer may want to perform generalization tasks by invoking an ordered sequence of generalization functions as defined in the database. Finally a good generalization application must allow the cartographer to initiate generalization operators against groups of features as well as individual features. These groups could be defined via queries of the data, local areas of interest defined by the cartographer using fences, or class of feature defined by theme, name, or attribution set.

Automatic Generalization

Although to this point most of what has been discussed is the application of generalization operators interactively, considerable time can be saved in a generalization process if some or all of the desired operators can be applied globally to an entire region being generalized. Although some generalization operators require more control by the cartographer due to the number of unique results that can be obtained through the adjustment of parameters, other more simple operators and algorithms have a limited number of parameters and thus produce a limited number of results. These operators can be applied in a batch environment because they typically produce the desired results regardless of the input features being generalized. An example of this type of operator is the collapse of an area building to a point or a line. If the product being generated is to have the buildings displayed as points and the input data has the building collected as an area or polygon, the feature collapse operator with an algorithm that places a point at the centroid of the area could be applied. This type of algorithm is very consistent and requires little if any input from the operator to view and approve the results. This algorithm could in most instances be applied to an entire region in a batch execution of the operator. Another example could be aggregation of two or more areas that share a common boundary. These areas were captured and stored as distinct areas in the source data. Because the product being generated has no requirements for them to be displayed as separate entities, the areas may be combined into a single area with the boundaries of the input areas serving as the geometry of the generalized feature. These are only a few examples of some operators that could be executed in a batch environment. Others operators can be executed automatically as the cartographer becomes more and more comfortable with their execution. Also, advances made in the algorithms could make it possible over time for more algorithms to be executed without the intervention of the cartographer.

Any automatic generalization process should be able to produce queues that contain both possible problems encountered during generalization and features or regions that are potential generalization candidates. The queue indicating potential problems should contain the results of the generalization operators the cartographer needs to review. The types of problems could include spatial relationships that were not maintained, algorithm limitations that were exceeded, or measurements that indicate over-generalization. The potential generalization queue should contain features that are good candidates for generalization but need to be generalized interactively to produce the desired results. An example may be a cluster of areas that are within a specified distance of one another and may need to be aggregated.

Generic Workflow

It is easy to visualize a workflow to perform generalization being a combination of interactive and automatic processes as defined above. By performing incremental and iterative generalization as shown in Figure 3 it is believed good quality can be achieved.

Initial Analysis and Data Preparation

This is an initial phase of any generalization process. The cartographer should review the data to determine its content. Here an initial plan is developed as to what generalization operators are to be applied and in what order. Regions where generalization should not take place should be marked so that the batch and interactive processes that follow will exclude them from processing.

Interactive Generalization and Batch Initialization

During this process interactive generalization is performed using the generalization operators, viewing capabilities and stored parameters. Generalization operators that require more visual feedback are applied here. Initially few of these operators may be executed because it may be more desirable to have certain operators such as feature elimination be executed first in a batch process. The set of operators to be applied in the next automatic process are identified.

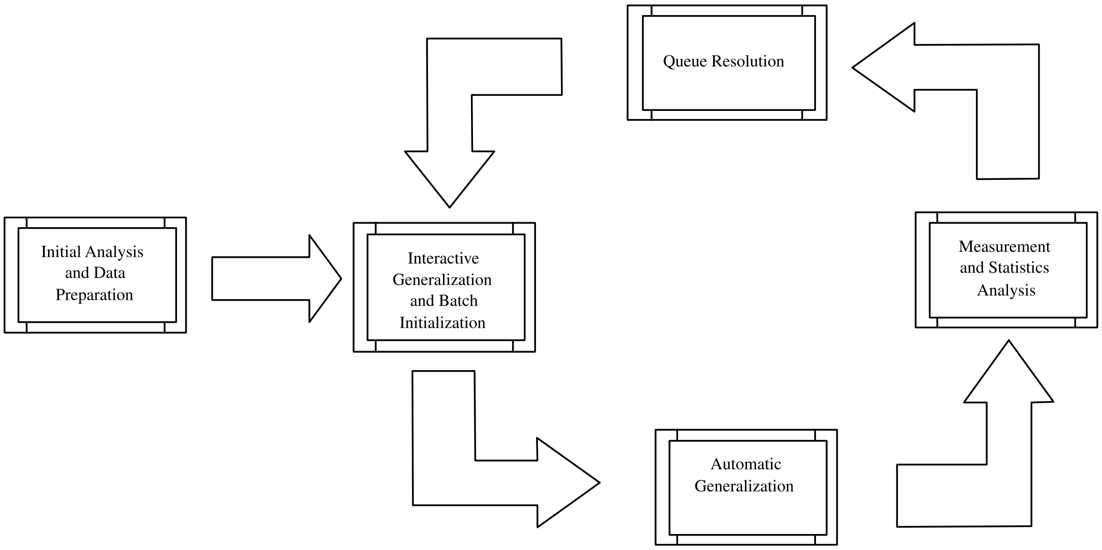


Figure 3.

Automatic Generalization

The generalization operators identified in the prior interactive process are executed sequentially as specified. Queues are generated as situations are encountered during the process that will require the intervention of the cartographer. Statistics are kept and measurements made of the generalization operators that are executed and the features generalized.

Statistics and Measurement Analysis

The number and types of features generalized by the individual generalization operators is reviewed. The cartographer makes decisions as to whether the automatic process performed as expected and if the data was generalized to the appropriate level.

Queue Resolution

Any queue entries made by the batch generalization process are resolved as needed. In order to resolve the queues the cartographer may choose to perform additional generalization operators on the queued features or may retrieve the original features and perform any of the generalization operators with varying parameters. All queue entries will be resolved before starting the next iteration.

Conclusions

Map generalization is a very challenging process to automate. Although it is desirable to have a totally automated process where the cartographer need only specify a few parameters and have the generalized output generated, it is more realistic at this time to have a system that provides good tools and gives the cartographer the ability to make the decisions of where and how to generalize. It is believed that such a system will not only provide a means of performing generalization in the digital map-making world of today, but also provide a basis for a development of more advanced generalization systems in the future.

Session / Séance 43-B1

Issues and Solutions to Displacement in Map Generalisation

William A. Mackaness *and* **Ross Purves**

Department of Geography
University of Edinburgh
Drummond St, Edinburgh EH8 9XP, Scotland
wam@geo.ed.ac.uk

Abstract

Displacement is one of a number of methods used to improve the legibility of symbolised features in a map. In this paper a number of alternate paradigms for displacement are considered and compared. It is apparent that effective displacement can be achieved by sharing small movements among a group of objects in order to minimise loss of positional accuracy and maintain the gestalt and topology of the map objects. The proposed algorithm works by considering, for each object in turn, a number of alternate positions close to its current location and the location which minimises overlap among the neighbouring objects is chosen. In contrast to other approaches to the modeling of displacement, this idea is relatively simple but produces topologically consistent solutions to the displacement of large numbers of objects in very short order with very low processing overhead. This paper describes the algorithm and illustrates its application and evaluation. Suggestions for further work form part of the conclusion.

Introduction

The objective of map generalisation has always been the meaningful ordering, grouping, and symbolisation of phenomenon commensurate with the scale of display and screen resolution. As we view data at different scales, so the patterns of the relationships among the phenomenon being represented change. Accordingly, in automated cartography, a range of generalisation methods has been developed to support this process of abstraction, in grouping and ordering the phenomenon being represented. Increasingly research has focused on further development of these generalisation methods so that they can work within a set of constraints (Ruas 1999), and without the need for intensive user interaction. Some of these methods produce radical changes in the form and composition of the map (such as selection, and symbolisation), whilst others produce more subtle changes,(such as smoothing or displacement). Much of the research in automated design is premised on the idea of identifying ‚conflict', the idea of competition for space at reduced scales. It is important to appreciate that the suitability of any given method to a design problem depends on

1) the number of design conflicts,
2) the distribution of those conflicts across the map, and
3) the composition of those conflicts.

Radical changes in map composition are required where large number of conflicts occur, that are comprised of mixed phenomenon. At the other extreme where few conflicts arise, and are of a similar type, then subtle application of generalisation methods (such as displacement) is all that is required. For any displacement method, its application should ensure that

1) features should be more distinguishable after application that prior to it,
2) that movement should not create new conflicts,
3) that the pattern of distribution should be conserved,
4) that the general defining characteristics of an object should be conserved (qualities such as orientation, size, angularity/smoothness, shape),
5) that objects should be moved a minimum distance from their true location.

For methods capable of autonomous operation (minimal user intervention), we add the following operational criteria; 1) that any developed method is capable of detecting overlap, 2) capable of knowing when to stop, 3) can provide meta data on the quality of the solution (as input to the application of alternate methods as part of a design strategy). It should be pointed out that the evaluation of the result, (beyond that of meta data relating to variables such as total displacement, number of conflicts solved, etc.), has proved hard to define (McMaster 1996). And finally, from an operational sense, we consider it important for the algorithm to be robust, and to work with the minimum of information necessary to achieve a workable solution.

Defining the constraints on displacement

Harrie (1999) proposed a method by which objects are considered both individually and in context. He proposes that the three issues governing the limits of displacement are: the geometry (whether point, line or area geometry), the limit of displacement for any one object, and whether the internal geometry of the object can be modified. Natural and relatively large objects (such as lake boundaries) can 'absorb' quite high levels of distortion, whilst small anthropogenic features have to be treated as being rigid but 'floating' (Figure 1).

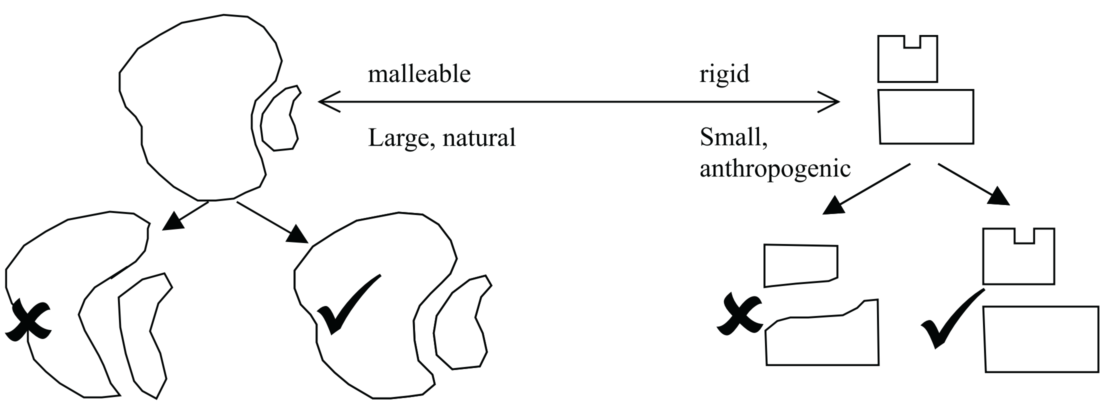


Figure 1: The continuum of object types governing 'acceptable' levels of distortion and displacement.

The inherent interdependencies that exist between geographic phenomena complicate the process of displacement. Irrespective of the degree of plasticity or levels of displacement allowable for any one object, it is often surrounding features that limit the degree of movement. The most obvious example is the way network features limit the movement of point objects. A road network in a city will limit the movement of buildings within a block (Regnauld 1996), and the displacement of river sections (for example using the accordion algorithm developed by the IGN (Lecordix et al 1997)) will require other objects to move if we are to conserve the topology of those objects. A further example is given by Nyerges (1991) who observed that lines of communication often run parallel to one another, and their coincidence and cross over points constrain the degree of movement allowable along different sections of the communication lines. Such an idea is captured in Figure 2, adapted from Nyerges (1991), which shows how each of these four object types (railways, roads, rivers, and bridges) constrain the displacement among them.

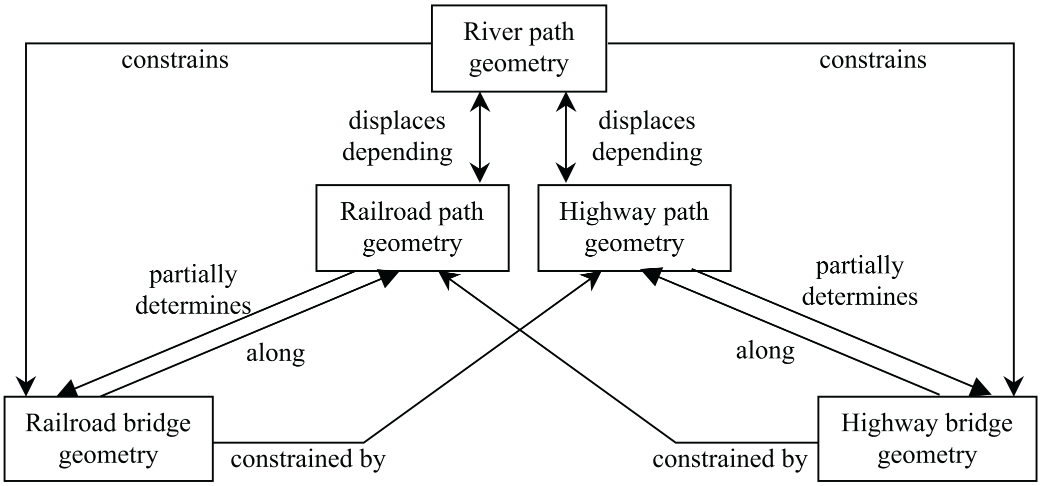


Figure 2: The spatial relationships between map features with regards to displacement (after Nyerges 1991, 72).

Proposed Solutions

Methods proposed for detection include cluster analysis (Mackaness 1994) and via the symbolisation process (Nickerson 1988; Ruas 1999) however identifying groups or clusters of objects does not necessarily mean that the overlap is confusing and requires displacement. But once features are found to overlap or to be in proximity, the issue is which strategy is best able to resolve the conflict. Nickerson (1994) and Christ (1979) addressed the issue by considering pairs of objects, but the application of this technique can lead to the 'ricochet' effect whereby movement of one pair of objects generates a conflict with surrounding objects, which after subsequent displacement recreates the original conflict with the first pair (Figure 3).

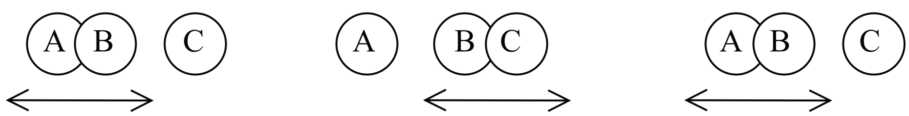


Figure 3: Ricochet effect recreating the original conflict.

Ormsby and Mackaness (1999) argue that for all these reasons it is important to consider not just the geometry of the object, but the semantics of the object and its effect on the neighbourhood of objects surrounding it. This is to ensure that relative relationship, homogeneity and topological qualities are conserved during displacement. For these reasons it is important to develop overarching methodologies that enable us to characterise and measure the criteria governing the triggering, synthesis and evaluation of displacement activities. Figure 4, adapted from Ruas (1998) reflects a maturing of ideas and a realisation of the cyclic dependency within methods such as displacement and between this method and other methods.

Common to a number of these displacement methods, is the high processing overhead; first in the detection of overlap and proximity, second in the calculation of vectors of displacement, and finally in the evaluation of the solution. Typically a space exhaustive tessellation of the space (Delaunay or its dual the Voronoi) is used to define the neighbourhood of a set of discrete objects and provides the basis for calculating the vectors of displacement (Jones, et al. 1995) . Once displacement has occurred, the tessellation is recomputed to validate and evaluate the outcome. Certain approaches require that each point along the boundary of a building be compared with each point on a neighbouring building to calculate the closest distance between the two (Ruas 1997) . These have proved to create good solutions but for small numbers of objects.

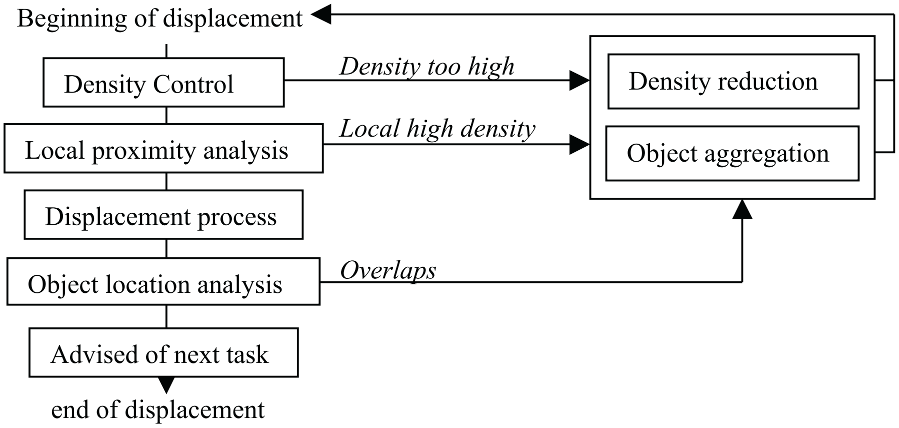


Figure 4: A flow diagram of the displacement process as proposed by Ruas 1998, 793.

An Alternative Approach

As systems increasingly begin to support the idea of 'intelligent zoom' (changing levels of detail with changing map scale), so it is less critical that any one display be a cartographic masterpiece. In the context of the dynamic process of display, emphasis is shifting towards solutions that can work in real time - i.e. methods that can handle large number of objects, and provide robust solutions in every instance (as Dutton (1999) argues, we should move away from slavishly mimicking an object's shape, representation and location at larger scales). This paper proposes a new model for displacement that addresses the above issue. It begins by discussing the ideas of Dewdney (1987), on which the algorithm is based, and illustrates the incremental development and implementation of the algorithm and associated results. Though the idea is relatively simple (with low processing overhead) it produces topologically consistent solutions to the displacement of large numbers of objects in very short order with very low processing overhead. It is applicable where the composition of the conflict is of mixed type (preventing the merging of objects), where the objects are sufficiently salient or contextual to be retained (cannot be omitted), and where the overlap is sufficiently small as to avoid changes in topology after displacement. Effective displacement is achieved by sharing small movements among a group of objects in order to minimise loss of positional accuracy and maintain the gestalt and topology of the map objects. This research was driven by a desire to implement a simple but effective way of handling large numbers of map objects that may require displacement. The idea came from a paper by Dewdney (1987) in which he models the social relationships between a group of people at a party. Among such a set, there might be some people who are attracted to specific individuals, but those same individuals would run a mile to escape their approaches! Thus each guest has a preferred separation distance from each of the other guests - in effect a matrix of distances that define optimal 'happiness' among the guests. It is unlikely that each guest has the same preferred profile of distances and it will therefore be necessary for each guest to always move in a direction that minimize unhappiness - gravitating towards those they like, and away from those they wish to avoid. By iteratively considering alternate movements among the guests, the proposed algorithm provides a means of modeling the social dynamics we so enjoy observing at parties! The party space is divided into a regular grid, and Dewdney's algorithm works by considering, for each guest, the eight surrounding neighbourhood squares (a 3*3 kernel). For each square in the kernel, the program calculates the total unhappiness of the guest and moves to the square in which unhappiness is at a minimum. The program runs through a series of iterations in which people effectively migrate around the room in search of happiness. The program reaches a stable state in which some guests may remain partially unhappy but for which spatial constraints prevent them from reaching happier states.

This idea can be applied laterally to the problem of reducing overlap between discrete objects in the map space. It works by considering, for each object in turn, a number of alternate positions close to its current location. For each location, the program calculates the total ‘unhappiness’ of the object, and moves to the position in which unhappiness is at a minimum. Minimum unhappiness (or happiness) is defined as a position among a set of alternate positions which produces the greatest reduction in overlap between itself and its local neighbours. The program runs through a series of iterations in which objects effectively migrate small distances within the map space in search of happiness. The program reaches a stable state in which some objects may remain partially unhappy (still overlapping) but for which spatial constraints prevent them from reaching happier states. Just as the party guests are constrained by the room, so the map objects are constrained by the areal extent of the map.

First Implementation

In the first implementation, circles of common radius were used. Each circle in turn, considers one of eight locations, a radial distance of r/2 from the centre of the circle, where r is the radius of the circle (Figure 5b). For each location, the circle calculate the degree of overlap with any neighbouring circles, and chooses the location that minimises overlap (Figure 5c).

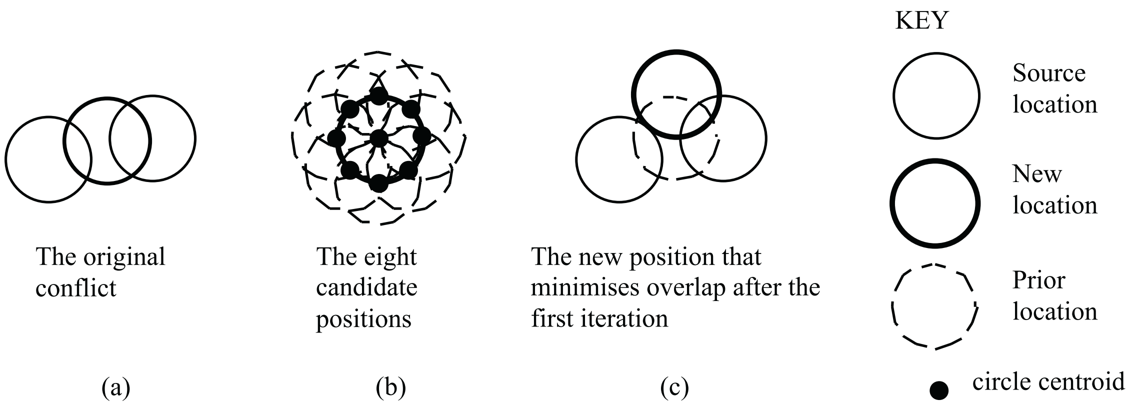


Figure 5. Choosing a location that minimises overlap.

When applied to a random set of circles, it was observed that the overall pattern of distribution remained the same and that the algorithm reached a near complete solution after just four iterations. Some conflicts remained however, for which alternate generalisation methods would be required. The next stage was to extend the implementation so as to consider typical building shapes instead of circles. Comparison of overlap or coalescence between concave and convex shapes is complex and has high processing overhead. Since circles enable very simple comparison, it was decided to adopt the methodology by fitting a circle to each building and adapt the above implementation to work for circles of differing radius. Figure 6 shows the various stages in the displacement cycle. The algorithm takes as input, a set of building footprints defined as a sequence of co-ordinate pairs. The buildings are enlarged in anticipation of scale reduction (Figure 6b). For each building in turn, the bounding circle that encompasses the building is fitted with at least two points on the building’s footprint touching the bounding circle (Figure 6c). The result is a set of circles, each with known centre and radius, which are then iteratively displaced (Figure 6d). The circle acts as a ‘required buffer of separation’ between buildings. The new location of the circle becomes the new location for the building (Figure 6f).

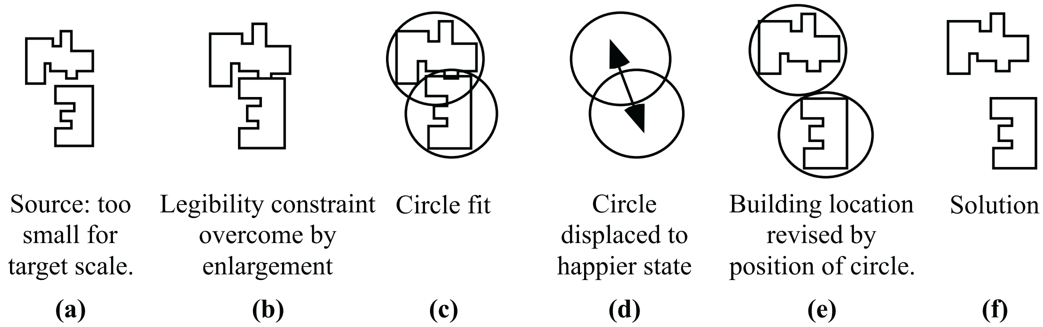


Figure 6: the stages of displacement for building objects.

Figure 7 shows the results of applying the algorithm illustrated in Figure 6, to a set of buildings. In order to compare the source with the target, they are displayed at the same scale in Figure 7a and 7b but it should be remembered that the process of enlargement and displacement is in anticipation of scale reduction. The test data are a set of buildings from the IGN topographic sheet (BD Carto), which contains about 200 buildings and is in anticipation of a scale reduction from 1: 50000 to 1: 100 000.

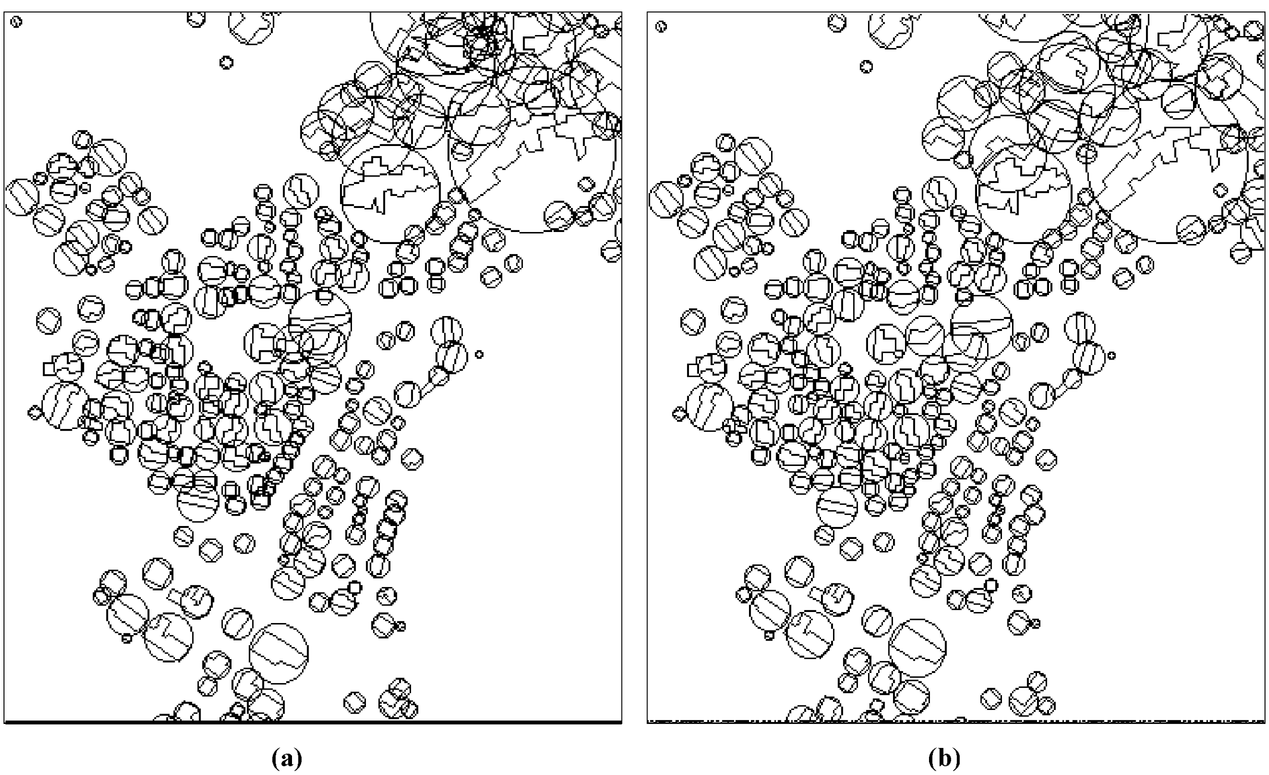
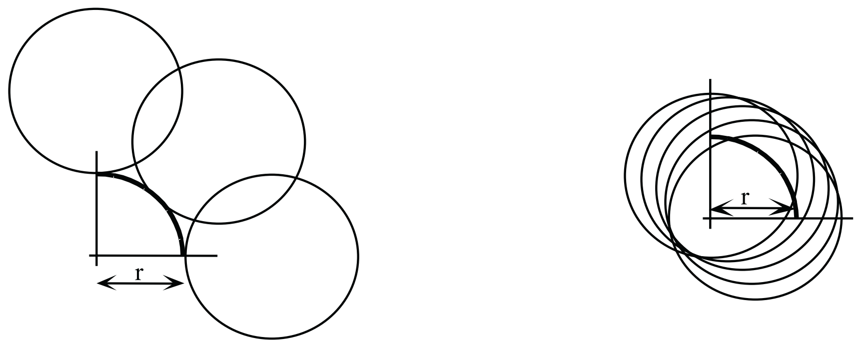


Figure 7. Results of application of the method after six iterations.

Refinement of the technique

From empirical observations, it was noted that three issues critically affected the rate at which a solution was reached. The first was the number of radials used in generating candidate solutions, the second was the maximum amount of movement allowed in any one iteration, and the third was the number of candidate solutions considered along the length of each radial. It was observed that if the movement was large (say r, where r is the radius of the circle) and the number of radials considered was small (Figure8a), then most conflicts were solved in very few iterations, but the solution was cartographically unacceptable. If only small movements were considered (say r/4), and for many radials (Figure 8b), the converse was true, (resulting in higher processing times but more acceptable solutions).



a) displacement r, radial increment π/4 **b)displacement r/4, radial increment of π/8**

Figure 8: Changes in the number of candidate solutions and the amount of displacement.

Furthermore the quality of the solution was affected by the number of candidate solutions considered along each radial. It was found that considering one value of r along the radials sometimes resulted in movements that were unnecessarily large. To overcome this, the algorithm was refined so that for each radial a number of candidate solutions were generated along the length of the radial. Figure 9a shows the eight centroid locations for r/2 and angle step of p/4 radians. Such an implementation generates the solution shown in Figure 9a. Figure 9b shows the centroids of the candidate solutions, for increments of r/8 from r/2 to r, and for radial angle step of p/4. This generates a finer result. The question then becomes, what is the optimum number of candidate solutions to consider in any one iteration, and what should be the increment in angle step size, and increment step along each radial?

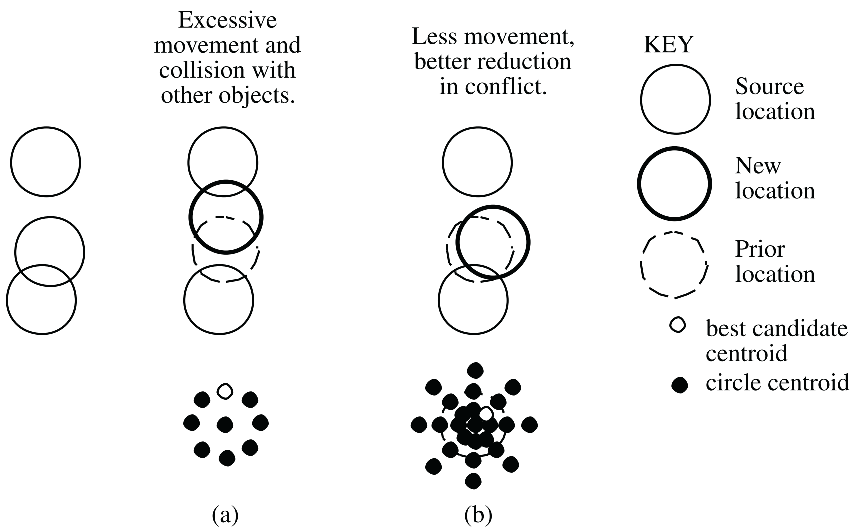


Figure 9: Better solution achieved by considering larger number of candidate solutions in one iteration.

Optimisation of Parameters

A set of experiments were conducted to examine the best compromise between speed and the quality of solution, by creating a set of solutions for which varying combinations of maximum radius, radius step and radial increment were used. Figure 10 shows the number of conflicts against the number of iterations. The lines are exponential in form. For the bottom three lines (maximum displacement of 2r, r and r/2), the large value of r results in a rapid stabilisation and solutions being found to most of the conflicts in few iterations, with few conflicts remaining. But the results were found to be cartographically poor. For the top two lines (using small increments of movement, and smaller steps), it takes longer to reach a solution and slightly more conflicts remain, however the solution from a cartographic perspective is much more acceptable because the distribution of buildings more closely reflects the source set (Figure 11b is the result for line 4, after nine iterations, maximum displacement of 0.25r, steps of r/16 and angle step size of p/4).

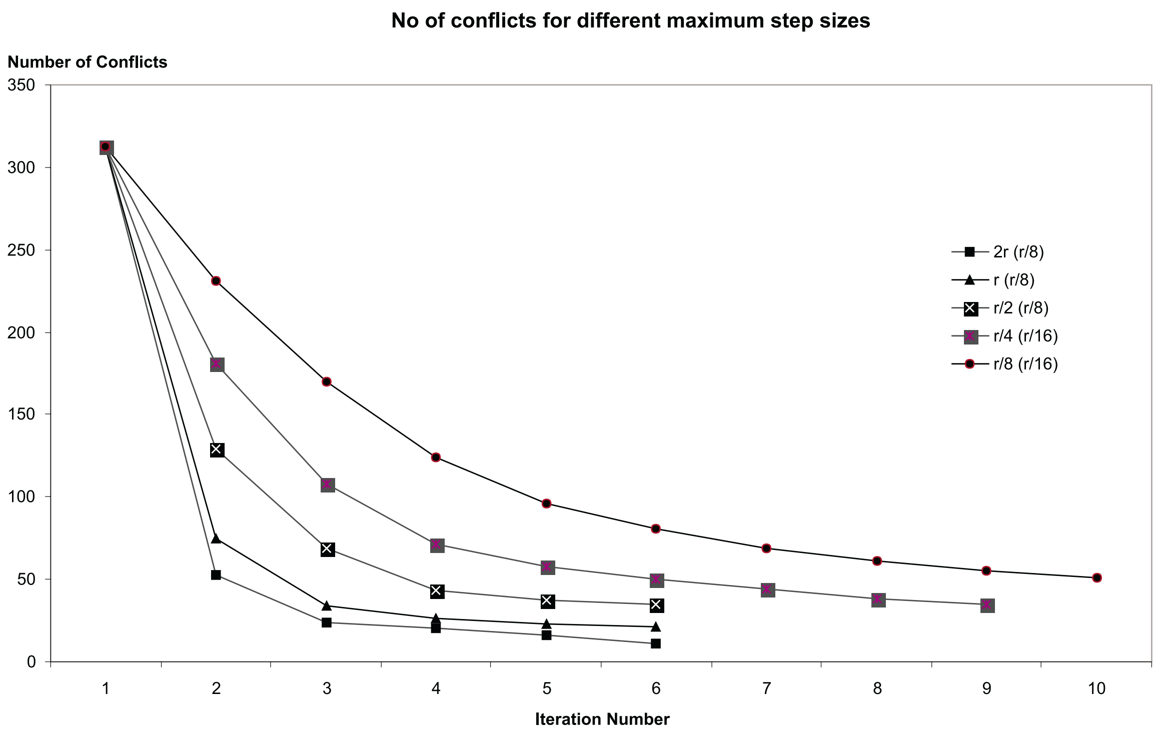


Figure 10: A graph of conflicts against iteration for varying levels of r and radial incrementation.

Though the values of line 5 produced a marginally better result, the cost in terms of processing and the larger number of conflicts remaining did not justify its use.

Limitations of the approach

The closer the building approximates a regular square, the closer the circle is to its generalised form. For buildings with low compactness ratio (length to area ratio) the less appropriate the circle is as a surrogate for its form. The consequence of this is the displacement of buildings that, in reality, do not require displacement. This can be overcome by fitting an ellipsoid to the building for those buildings with low compactness. The algorithm could be modified so that for each building, its compactness ratio is first calculated and if below a certain threshold, an ellipse rather than a circle is fitted. Displacement and rectangle fitting would then follow the same procedure as described above. The programming and associated increase in computational process in determining overlap was such that it could not be considered in the context of this initial implementation. In the implementation by Dewdney (1987), fixed objects (such as the drink's table at the party) were modelled as a raster matrix underlying the circles. For each iteration, the individual considers a set of locations but if the location is occupied by fixed objects, then the location is not considered as a candidate solution. This implementation has considered only objects of type building. Roads (across which buildings may not pass) could be formed as a matrix of fixed objects into which objects could not be displaced. Again, this is considered part of future implementation work. However, with the absolute limit imposed on the movement of any one building and the emphasis on small displacements, it is not clear how significant an improvement this modification will achieve.

Figure 11. a) original set of buildings, b) the solution using the parameters for line 4 of Figure 10.

Conclusion

It is important to realise that displacement by itself does not lead to a decrease in map content. It stands to reason that the displacement of objects works in a relatively narrow band of circumstance and is based on the premise that there is some free space into which objects can move. This paper has presented a brief overview of past attempts at displacement and discussed the constraints surrounding the use of displacement as a generalisation technique. In presenting a new algorithm (based on Dewdney's 'unhappiness' algorithm) the paper argues that there is a need for algorithms that can meet these constraints but avoid the processing overheads typically associated with modeling displacements among a large number of objects. The algorithm is considered to be an effective way of both identifying and displacing objects, and providing cartometric information for use by other generalisation algorithms.

There is no reason why this approach could not be used in concert with other generalisation operators. A subset of buildings could first be removed before the happiness algorithm was applied. Indeed the happiness algorithm could be used to identify unhappy buildings and this criteria could be used to select candidate buildings for removal.

Acknowledgment

The authors are very grateful for funding for this research from the European Union via the Esprit funded AGENT project, 24 939 LTR.

References

Christ, F. 1978. A Program for the Fully Automated Displacement of Point and Line Features in Cartographic Generalization. Information Relative to Cartography and Geodesy Translations (Frankfurt), vol. 35, no. pp. 5-30.

Dewdney, A. K. 1987. Computer Recreations Diverse personalities search for social equilibrium at a co. Scientific American Sept 87, vol. no. pp. 104 - 107.

Dutton, G. 1999 Scale, Sinuosity and Point Selection in Digital Line Generalisation Cartography and Geographic Information Systems (special issue).

Jones, C. B., G. L. Bundy and J. M. Ware. 1995. Map generalization with a triangulated data structure. Cartography and Geographic Information Systems, vol. 22, no. 4, pp. 317-331.

Harrie, L. 1999 Modelling Displacement Behaviours in Automated Generalisation Cartography and Geographic Information Systems (special issue).

Lecordix,F, Plazanet,C and Lagrange,JP “A platform for research in generalization: Application to caricature” Geoinformatica , 1(2):161-182, August 1997

Mackaness, W. A. 1994. An Algorithm for Conflict Identification and Feature Displacement in Automated Map Generalization. Cartography and Geographic Information Systems, vol. 21, no. 4, pp. 219-232.

McMaster, R. B., and Comenetz, J., 1996, Procedural and Quality Assessment Measrues for Cartographic Generalisation: GIS/LIS 96, p. 775-785.

Monmonier, M. 1991. Role of interpolation in feature displacement. In Map generalization: Making rules for knowledge representation, B. P. B. a. R. B. McMaster, (ed) Essex: Longman Scientific & Technical, pp. 189-203.

Nyerges, T.L 1991. Representing Geographic Meaning in Buttenfield B.P and McMaster, R.B (eds) Map Generalisation: Making Rules for Knowledge Representation 59-85. London: Longman.

Regnauld, N., 1996, Recognition of Building Cluster for Generalization: Proceedings of the 7th International Symposium on Spatial Data Handling, p. 185-198.

Ruas, A. 1997. An algorithm for reactive displacement in automated map generalization. International Journal of Geographical Information Systems, vol. no. pp.

Ruas, A., 1998, A Method for Building Displacement in Automated Map Generalisation: International Journal of Geographic Information Science, v. 12, p. 789-803.

Ruas, A., 1999, Modèle de géneralisation de données géographiques à base de constraintes et d’autonomie, Thèse de doctorat de L’université de Marne La Vallée, Paris, Marne La Vallée. Unpublished PhD Thesis.

Schittenhelm, R. 1976. The problem of displacement in cartographic generalization Nachrichten Aus Dem Karten Und Vermessungswesen II/33, vol. no. pp. 65 - 74.

Session / Séance 39-D

Holistic Cartographic Generalization by Least Squares Adjustment for Large Data Sets

Tapani Sarjakoski *and* **Tiina Kilpeläinen**

Finnish Geodetic Institute
Department of Cartography and Geoinformatics
P.O. Box 15, FIN-02431 Masala, Finland
Tapani.Sarjakoski@fgi.fi Tiina.Kilpelainen@fgi.fi

Abstract

In holistic generalization all the geographic objects in the data set are considered simultaneously. This paper presents for holistic generalization a method in which the least squares adjustment (LSA) is applied. The central motivation for the approach is that if we can formulate a cartographic generalization task as an adjustment problem, we are able to utilize powerful methods developed for its numerical treatment. Such methods have a long tradition in geodetic and photogrammetric applications, based on simultaneous adjustment of different kinds of observations. They have been shown in many cases to be superior over methods based on a sequential solution of the problem at hand.

In cartographic generalization, a central requirement is that the spatial and topological relations between the objects are to be preserved throughout the generalization process. However, the displacement of objects can result in violation of this requirement. It has been shown in earlier studies that the problem can be solved by formulating it as equations for the least squares adjustment. The holistic approach results in a simultaneous adjustment applied to large data sets. The paper demonstrates how sparse matrix techniques and an iterative conjugate-gradient method (CGM) can be used for the efficient numerical solution of the related linear equation system. The principle of holistic generalization is also studied, regarding the integration of various generalization operators. A process model including the least squares adjustment is also presented in the paper.

1. Introduction

There is a long tradition in geodesy and photogrammetry for using the least squares adjustment (LSA). There the primary task is to compute coordinates for points from observed quantities, such as geodetic angles and distances and photogrammetric image coordinates. The main motivation for its usage is to get optimal solution when there are more observations than strictly necessary to determine the unknowns. The theory behind the least squares adjustment is sound and well-developed.

Recently least squares adjustment has gained attention also related to cartographic generalization. Burghard and Meier (1997) describe a method for using snakes for displacing linear features. Højholt (1998) has used finite-element method for the solution of spatial conflicts related to displacement. Both of these two methods are inherently based on least squares techniques. Finally, Harrie (1998, 1999) has presented a method for solving spatial conflicts. His method also uses the least squares adjustment in the solution. Harrie's work also presents a rather complete set of geometric constraints that can be applied in different situations. He also demonstrates the usage of the method with a practical example including a small number of geographic objects.

Our paper elaborates further the approaches presented in Højholt (1998) and Harrie (1999). We put emphasis on linking the methods with the general theory and practice of the least squares adjustment, as applied especially in photogrammetry. Our goal is also to show that the least squares adjustment offers a promising alternative for sequentially combining different generalization operators.

It was pointed out in Harrie's work (1999) that his constraint method is computationally demanding. To overcome the problem, we show how sparse-matrix techniques and the conjugate-gradient method can also be used in cartographic generalization for solving large linear equation systems. These techniques have been used frequently e.g. in photogrammetric applications.

2. Least squares adjustment

The least squares adjustment, as usually applied in photogrammetry and geodesy, is based on a functional dependence

$$\mathrm{F}(\mathbf{X},\mathbf{L}) = \mathbf{0}$$

between a vector of observations $\mathbf{L}$ and a vector of unknowns $\mathbf{X}$. The purpose of the adjustment is to find estimates $\hat{\mathbf{X}}$ and $\hat{\mathbf{L}}$ satisfying the least squares condition

$$\mathrm{F}(\hat{\mathbf{X}},\hat{\mathbf{L}}) = \mathbf{0}, \quad \hat{\mathbf{v}} = \mathbf{L} - \hat{\mathbf{L}}, \quad \hat{\mathbf{v}}^t \mathbf{W} \hat{\mathbf{v}} = \min!,$$

where $\mathbf{W}$ is a weight matrix. Although the functional dependence between observations and unknowns is usually non-linear, in majority of cases we end up, after linearization, solving an overdetermined equation system of form

$$\mathbf{v} = \mathbf{l} - \mathbf{A}\mathbf{x}$$

$$\mathbf{v}^t \mathbf{W} \mathbf{v} = \min!$$

with following notations:

$\mathbf{l}$	vector of observations
$\mathbf{v}$	vector of residuals
$\mathbf{x}$	vector of unknown parameters
$\mathbf{A}$	design matrix (Jacobean matrix)
$\mathbf{W}$	weight matrix, usually diagonal.

The system is often solved using normal equations

$$\mathbf{N}\mathbf{x} = \mathbf{u}$$

$$\mathbf{N} = \mathbf{A}^t \mathbf{W} \mathbf{A}$$

$$\mathbf{u} = \mathbf{A}^t \mathbf{W} \mathbf{l}$$

with

$\mathbf{N}$	coefficient matrix of normal equations
$\mathbf{u}$	constant vector of normal equations.

The number of unknown parameters in these linear equation systems can be very large in photogrammetric applications, typically several thousands but sometimes exceeding even some millions. As a consequence, special consideration has to be given for the methods to solve an equations system, even with current computers.

Some alternatives are available but it is characteristic for all of them that the sparsity of the coefficient matrices is used. The Cholesky-factorization is perhaps the most commonly used, combined with techniques to utilize the sparsity of the normal equation coefficient matrix. This method is applied especially to banded or banded-bordered matrices. It has the advantage of being suitable for also producing a partial inverse, which is needed for statistical evaluation of the parameters. The use of iterative methods that also utilize the sparsity is another alternative. The so-called conjugate-gradient method is one of them and will be treated in detail in section 4.

3. Least squares adjustment for map generalization

The least squares adjustment is based on analytical formulation of a problem. The formulation must yield an overdetermined equation system that can be solved following the least squares principle. Regarding cartographic generalization, the approach presented by Højholt (1988) is targeted for solving the problem displacement and spatial conflicts. It uses the regularization of the widths of the roads as an example. The method is based on constrained Delaney triangulation and the triangles are the basic elements for a finite element method. The stiffness of the triangles is assigned such that buildings will retain their original shape. For the triangles forming a road area, constraints are assigned forcing them to be deformed such that the road gets a even predefined width. The finite element method is essentially based on least squares solution. Højholt pointed out that the method can be extended in many ways, for example so that the orientation of objects would be retained. The triangulation-based approach has also the property of rigorously preserving the topology. This is important to avoid spatial conflicts or inconsistencies in the resulting data set.

Harrie (1999) has applied the least squares adjustment to solve the problem of displacement and spatial conflicts, calling it a constraint method. The approach was object-oriented and the objects were categorized as movable/immovable and also as stiff/non-stiff objects. The degree of movability and stiffness of objects was indicated by assigning suitable weights for the related equations. The analytically formulated constraints included equations for object movement, object stiffness, line curvature and angle of crossings. As concluded by Harrie, this set is not complete and could be further extended e.g. for the preservation of the spatial distribution of objects.

The earlier work by Burghardt and Meier (1997) has used the snake concept for solving the problem of displacement and spatial conflicts, in the case of lines. The snake concept is closely related to splines. The treatment of snakes can also be interpreted as a least squares adjustment problem. Snakes have also been applied for feature extraction from digital images, see e.g. Trinder and Li (1995). In this case the general goal is to find an optimal position of features, like roads or edges, with respect to certain curvature conditions (called internal energy) and intensity values of the digital image (called external energy). The degree of generalization of the curve is increased when the internal energy is increased. Related to feature extraction, the least squares adjustment has also been applied as a method to make the outline of buildings regular, e.g. so that all the corners are rectangular (Förstner, 1995; Grün and Wang, 1998). The constraints that have been applied have a close resemblance to those described by Harrie (1999).

Referring to the work of Højholt (1988), a finite element method has also been applied to create digital elevation models. One of the early works has been presented by Ebner et al. (1978; 1980). Their approach is based on regular grid, i.e. square elements. The height values in the corners of the grid are the unknown quantities in the equation system. Observed heights are entered in the system as equations based on bilinear interpolation within the corresponding grid element. The continuity constraints based on first order derivatives of the surface are used for controlling the stiffness of the surface. Taking an extreme case as an example, use of continuity constraints with a very high weight will force the elevation surface to become a (tilted) plane, adjusted to the set of observed heights. The least squares adjustment is used in this method for solving the overdetermined equation system.

As apparent from the review above, the least squares adjustment is a generic method that can be applied in numerous ways in tasks related to generalization of geospatial data. Two observations are essential. Firstly, the problem must be formulated analytically as functional dependencies. Secondly, the strength of the method is in the simultaneous solution of all the displacements and deformations related to the data set under process. This is a significant advantage over sequential methods, which may suffer from the phenomenon that the solution of one conflict will produce another problem.

An overall flowchart for an automatic generalization using the least squares adjustment is presented in Figure 1. *Design of map series* and *Semantic mapping of objects* refer to the intellectual processes which are supposed to be carried out only once for each map series. *Structural generalization*, *Least squares adjustment*, and *Post analysis* refer to the computerized processes for creating the generalized data set. *Structural generalization* is a process for creating a topologically correct data set with approximate geometry. *Least squares adjustment* will fine tune the geometry to satisfy the spatial constraints. The distinction between the structural generalization and the least squares adjustment coincides to some extent, but not exactly, with model generalization and cartographic generalization as defined by Kilpeläinen (1997). Finally, the purpose of *Post analysis* is to check the result for remaining conflicts and possibly refine the topology and constraints for a repetition of the least squares adjustment. In principle conflicts should occur only in cases where a spatial problem cannot be solved by fine tuning the geometry.

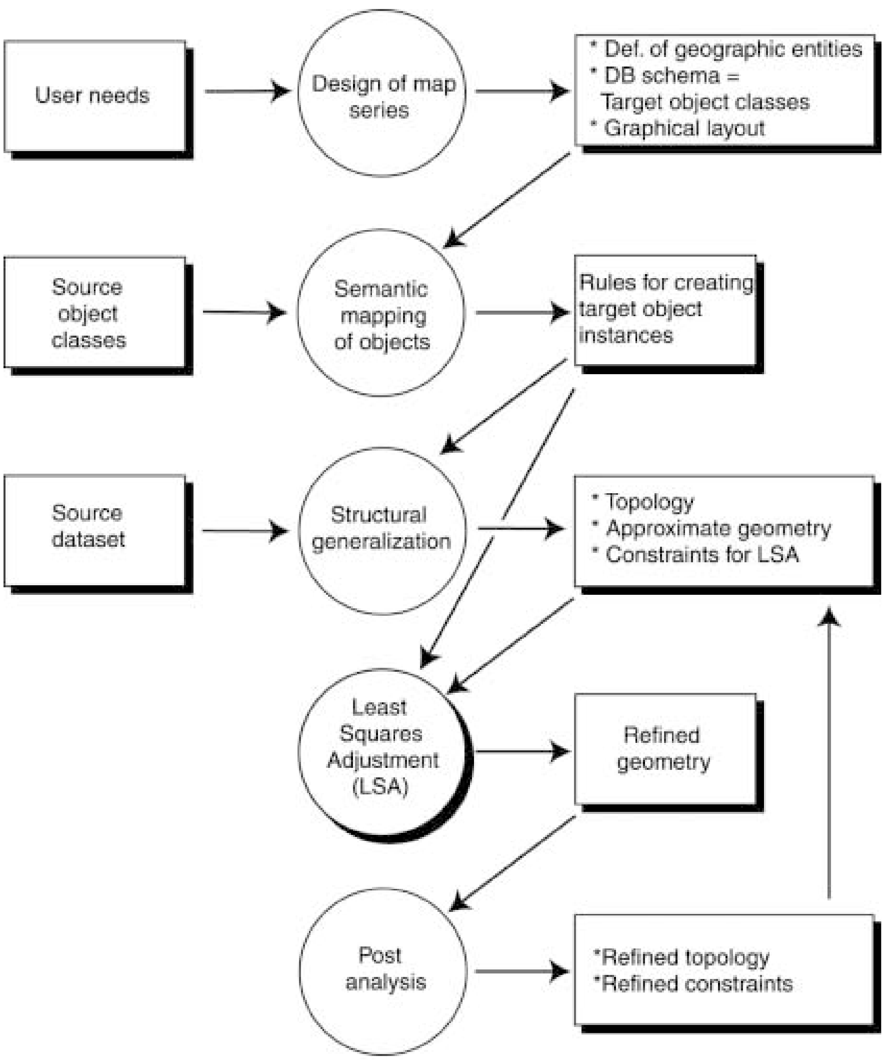


Figure 1. An overall flowchart for automatic generalization using the least squares adjustment

Regarding the functional dependencies or constraints, some categorization is useful. The purpose of *preserving constraints* is to force the original state to be preserved while *modifying constraints* are targeted to force a change. We can also make distinction between *internal*, *relative*, and *absolute* constraints. Internal are those being for each object fully independent of the global coordinate system and the position of the object. Area and length are the most obvious examples but more advanced metrics for describing a shape are certainly also needed. Relative constraints are used between two or more objects for describing the mutual dependencies. Finally, absolute constraints are used to relate objects to the global coordinate system. Position and orientation are good examples of functional dependencies that could be absolute but also relative. A resembling categorization has been presented by Ruas (1998).

4. Conjugate-gradient method for solving large, sparse equation systems

The conjugate-gradient method (CGM) is one of the iterative methods that have been developed for solving systems of linear equations. Its special characteristics are that it is suited for positive-definite matrices and also that it actually produces an algebraically exact solution, although in practice the iteration is terminated much earlier, when a desired accuracy has been reached.

Without going into the details of the theory of the CGM, its implementation is usually based on the following formulation, although also other variants exist.

A central entity in the CGM is a discrepancy or residual vector

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

that functions as a measure of the numerical accuracy of the solution. The iteration will be started with $\mathbf{x}_o = \mathbf{0}$, thus $\mathbf{r}_o = \mathbf{u}$. The iteration is usually terminated when the Euclidean norm of the residual vector, $\sqrt{\mathbf{r}_i^t \mathbf{r}_i}$, is below a predefined threshold value. After selecting $\mathbf{p}_o = \mathbf{r}_o$, the following entities will be computed during each iteration cycle:

$$\alpha = \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{p}_i^t \mathbf{N}\mathbf{p}_i}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\mathbf{p}_i$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha\mathbf{N}\mathbf{p}_i$$

$$\beta = \frac{\mathbf{r}_{i+1}^t \mathbf{r}_{i+1}}{\mathbf{r}_i^t \mathbf{r}_i}$$

$$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta\mathbf{p}_i$$

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

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

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

In many practical situations the convergence of the CGM can be improved considerably by scaling the normal equation matrix $\mathbf{N}$ into one having unit diagonal elements (Reid, 1971; Haljala 1974). This can be realized by defining a diagonal scaling matrix

$$\mathbf{Q} = diag(\mathbf{N})$$

and by inserting it into the original linear equation system

$$\mathbf{v} = \mathbf{l} - \mathbf{A}\mathbf{Q}^{-1/2}\mathbf{Q}^{1/2}\mathbf{x}$$

with

$$\mathbf{A}' = \mathbf{Q}^{-1/2}\mathbf{A}$$

$$\mathbf{x}' = \mathbf{Q}^{1/2}\mathbf{x}$$

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

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

with

$$\mathbf{N}' = \mathbf{A}'^{t}\mathbf{W}\mathbf{A}' = \mathbf{Q}^{-1/2}\mathbf{A}^{t}\mathbf{W}\mathbf{A}\mathbf{Q}^{-1/2} = \mathbf{Q}^{-1/2}\mathbf{N}\mathbf{Q}^{-1/2}$$

$$\mathbf{u}' = \mathbf{A}'^{t}\mathbf{W}\mathbf{l} = \mathbf{Q}^{-1/2}\mathbf{A}^{t}\mathbf{W}\mathbf{l} = \mathbf{Q}^{-1/2}\mathbf{u}$$

$$\mathbf{x}' = \mathbf{Q}^{1/2}\mathbf{x}$$

For implementation purposes it is often desirable to avoid the prescaling of the equation systems and instead to carry out the scaling "on-the-fly" during the iteration. This can be realized with the formulation:

$$\mathbf{p}'_o = \mathbf{r}'_o = \mathbf{Q}^{-1/2}\mathbf{u}$$

$$\mathbf{p}''_o = \mathbf{Q}^{-1/2}\mathbf{p}'_o$$

$$\alpha' = \frac{\mathbf{r}'^{t}_i\mathbf{r}'_i}{\mathbf{p}''^{t}_i\mathbf{N}\mathbf{p}''_i}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha'\mathbf{p}''_i$$

$$\mathbf{r}'_{i+1} = \mathbf{r}'_i - \alpha'\mathbf{Q}^{-1/2}\mathbf{N}\mathbf{p}''_i$$

$$\beta' = \frac{\mathbf{r}'^{t}_{i+1}\mathbf{r}'_{i+1}}{\mathbf{r}'^{t}_i\mathbf{r}'_i}$$

$$\mathbf{p}''_{i+1} = \mathbf{Q}^{-1/2}\mathbf{r}'_{i+1} + \beta'\mathbf{p}''_i$$

where $\mathbf{p}''$ is a double scaled conjugate vector. Algebraically this formulation is identical with the approach of applying the original formulation of the CGM to a prescaled equation system.

5. CGM in the least squares adjustment for cartographic generalization

The CGM has several characteristics that make it a strong candidate to be used as a tool for solving the linear equation systems related to the least squares adjustment for cartographic generalization. From the implementation point of view, the CGM is rather simple to program and also well suited for sparse equation systems. The convergence of the iteration process is also very rapid in the case of well-conditioned equation system, which usually occurs when the block comprises a geometrically rigid configuration. This statement is based on our own experience related to photogrammetric block adjustment (Sarjakoski, 1982), and similar experiences has been observed in other applications (Schlüter, 1998). We also have practical, however not documented, experience on using the CGM for approximating digital elevation models from randomly distributing points using the finite element method by Ebner et al. (1980). Using a rather straightforward implementation, it was possible to compute grids of size 1000*1000 (one million unknowns) with central memory requirements less than 100 MB. The number of iterations required for convergence was typically less than 50 and the processing time some minutes on a Pentium 200 MHz computer. The very low number of iterations required can be explained by the fact that the unknowns are significantly correlated with each other only within a rather limited spatial neighbourhood. The CGM also has the property of yielding a solution even if the rank of the equation system is less than the number of unknowns. This makes the CGM suitable for handling cases with linear dependencies between the unknowns.

6. Discussion and conclusions

We have here reviewed the foundation of the least squares adjustment, the research of its applicability for cartographic generalization and related fields like feature extraction and digital elevation models. Based on the demonstrations presented by Højholt (1998) and Harrie (1999) we can conclude that in many cases the least squares adjustment is a well working solution for cartographic displacement. The usage of splines for line smoothing is an established technique in cartography. Regularization of the shapes of building outlines in photogrammetry demonstrates the applicability of the least squares adjustment for a slightly different kind of generalization. Approximation of digital elevation models using finite element method serves as an example how the least squares adjustment, supported with conjugate gradient method, can handle large data sets.

From the point of view of holistic cartographic generalization it is noteworthy that all the techniques discussed above could be integrated in a simultaneous adjustment. It can also be visioned that the list of incorporated generalization operators could be extended and that even an integrated 3D generalization could be carried out. Generalization based on the least squares adjustment is limited, to a great extent, to the fine tuning of the geometry. Therefore the automatic process for generalization process as a whole must include other methods for structural generalization.

The least squares adjustment for generalization is a declarative approach in the sense that we define the constraints whereafter the adjustment engine takes care of the actual computations. In this simultaneous solution the formulation of the equation system becomes critical. In this context it is not so relevant to use the notion generalization operators, but rather to express analytically the constraints. The analytical approach stresses the need to develop more properties defining metrics of the generalized data set. The least squares adjustment seems to bring the methodology of map generalization closer to the mainstream of the approximation theory, as applied in other subareas of geospatial information science.

When writing this paper, our prototype implementation for the least squares adjustment based generalization is still under development. We predict, however, that the least squares adjustment as such will not be very problematic. On the other hand, the process for structural generalization seems to remain as the challenging part of the implementation.

References

Bader, M., R. Weibel, 1997. Detecting and Resolving Size and Proximity Conflicts in the Generalization of Polygonal Maps. In: Ottoson, L. (ed.), *Proceedings of the 18th ICA/ACI International Cartographic Conference*, Stockholm, Vol. 3, pp. 1525-1532.

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

Ebner, H., P. Reiss, 1978. Height Interpolation by the Method of Finite Elements, *Nachrichten aus dem Karten- und Vermessungswesen*, Reihe II, Heft 36, pp. 79-94.

Ebner, H., B. Hofmann-Wellenhof, P. Reiss, F. Steidler, 1980. HIFI - A Minicomputer Program Package for Height Interpolation by Finite Elements. *Proceedings of the 14th ISP Congress, International Archives of Photogrammetry*, Vol. 13, Part B4, Hamburg, pp. 202-215.

Förstner, W., 1995. Mid-Level Vision Processes for Automatic Building Extraction. In: Grün, A., O. Kübler, P. Agouris (eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Birkhäuser Verlag, Basel, pp. 179-188.

Grün, A., X. Wang, 1998. CC-Modeller: A Topology Generator for 3-D City Models. In: Fritsch, D., Englich, M., M. *Sester, GIS - Between Visions and Applications, Proceedings of ISPRS Commission IV Symposium, International Archives of Photogrammetry and Remote Sensing*, Vol. 32, Part 4, Stuttgart, Germany. pp. 188-196.

Haljala, S. 1974. Method of Conjugate Gradients for the Solution of Normal Equations in the Analytical Block Adjustment. *The Photogrammetric Journal of Finland*. Vol. 6, No. 2, pp. 160-165.

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

Harrie, L., 1998. *Generalization Methods for Propagating Updates between Cartographic Data Sets*. Licentiate Thesis. Lund University, Lund Institute of Technology, Department of Surveying, Lund, Sweden, 127 p.

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

Kilpeläinen, T. 1997. Multiple Representation and Generalization of Geo-Databases for Topographic Maps. Doctorate thesis. *Publications of the Finnish Geodetic Institute*, n:o 124, 229 p.

Ruas, A., 1998. A Method for Building Displacement in Automated Map Generalization. *International Journal of Geographical Information Science*, Vol. 12, No. 8, pp. 789-803.

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

Schlüter, M., 1998. Multi-Image Matching in Object Space on the Basis of a General 3-D Surface Model instead of Common 2.5-D Surface Models and its Application for Urban Scenes. In: Fritsch, D., Englich, M., M. Sester, *GIS - Between Visions and Applications, Proceedings of ISPRS Commission IV Symposium, International Archives of Photogrammetry and Remote Sensing*, Vol. 32., Part 4, Stuttgart, Germany, pp. 545-552.

Trinder, J., H. Li, 1995. Semi-Automatic Feature Extraction by Snakes. In: Grün, A., O. Kübler, P. Agouris (eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Birkhäuser Verlag, Basel, pp. 179-188.

Session / Séance 43-C

A “Genuine” Approach to Line Generalization

Albert H. J. Christensen

*7836 Muirfield CourtPotomac, MD 20854301 983 9004*E-mail: AHCHRIS@erols.com

This presentation introduces a fully automated line generalization procedure based on the two operations known as waterlining and Medial-Axis Transformation (MAT). The approach is deemed genuine because its results and those achieved by a trained cartographer, — working with the time honored manual method —, would be exactly the same. Julian Perkal's proposal (1959) is referred to as having the same genuine objectives, which this presentation proves can be reached by a combination of waterlining operations. Computer waterlining is briefly described as the counterpart of the device skilled copper engravers in the 1800's and early 1900's resorted to for marking off water areas in maps. The MAT employed is said to be a vector-mode operation that yields a well-organized centerline or skeleton of an arbitrary shape. In this approach to line generalization the medial-axis is proposed as the best choice for spanning Perkal's boundary zones. Remarks are made on the need for spanning those zones, something Perkal could not have foreseen in 1959. A series of generalizations of a group of islands is illustrated showing that the method preserves very well the general shape for lower scale factors and that tends to the logical end, a rounded shape, for larger factors. The exercises discussed do not include any line thinning, applications of which are understood to precede and follow the generalization proper. Each scale change is proven to strongly depend on the breadth of the waterlining operation, which breath can be equated to the quantity known as e *in the generalization literature. This dependence, absent in the thinning algorithms usually miscalled generalizations, is presented as extremely useful to the interpretation and handling of difficult configurations.*

Waterlining and the Medial-Axis Transformation

Waterlining is defined in [Neumann, 1997] as *lines representing water, drawn parallel with the edge of a water feature, which decrease in proximity and strength away from that edge.* Any reader who had examined old maps would recognise waterlining as the device used to mark out wide rivers and open water areas. The lines, although not defined in glossaries, have been called in the literature water lines, water-lines and waterlines. The last variant is used in this paper. Although waterlining was a very elegant artifice when executed by experienced engravers, it was also very expensive. Moreover, it could not be applied to narrow double-sided streams and water areas carrying bathymetric data. It is not surprising then that waterlining was outmoded by the popularisation of screen-printing that started at the turn of the last century.

Although banned from topographic maps, waterlining still has some technical value, as it has been demonstrated in the various papers by this author listed in the references. The approach to Cartographic Line Generalization presented in this paper is one of those applications. Waterlining was also the primary inspiration for the application: as the lines travel away from the shoreline, they simplified themselves by eliminating or by smoothing the concavities seen from the waterside. If one imagines the waterlines also traced on the land side the simplification would operate on the concavities seen from that side. To complete the scenario it would be necessary to imagine that the last waterlines to be drawn at both sides of the shoreline are in turn used as input to waterline in the opposite directions (towards the shoreline). The last lines to be traced would coincide partly with the original shoreline. Where they not, the shoreline needs generalization. The repeated waterlinings lead to results that are identical with those obtained with the approach proposed by Julian Perkal [Perkal, 1958]. However, to become a fully automated application, no matter what method is used to replicate it, Perkal's proposal needs an extension. The extension is provided by the Pattern Recognition operation known as Medial-Axis Transformation (MAT) [Blum, 1973]. In the generalization presented in this paper the MAT is also waterlining-derived. The geometry of the medial-axis is known by several names. Centerline; skeleton; axis of symmetry, and midline are some of them. Loosely, all these concepts mean a set of lines that are equidistant from the perimeter of a shape. If the perimeter can be reconstructed from the set of lines, then the set of lines plus the data needed for the reconstruction is called a medial-axis.

A description of the method developed to derive a medial-axis from waterlines can be found in [Christensen, 1999b]. In the same paper as well as in the others listed in the references, there are explanations on how waterlines are created from shapes. Figure 1 shows the relationship that exists between the waterlines and the medial-axis (in heavier lines) of a shape.

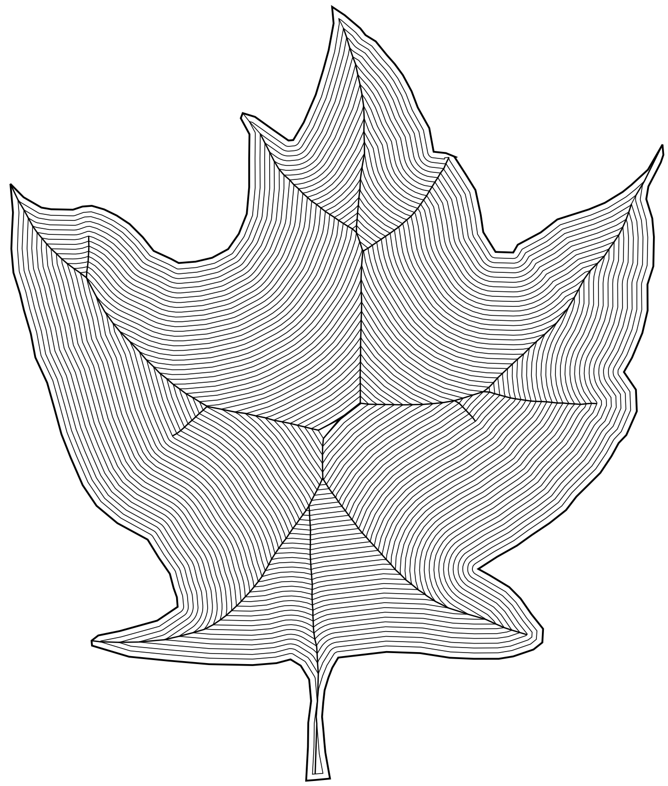


Figure 1. Waterlines at constant intervals and the medial-axis of a shape

The goodness of this method can only be assessed by inspecting the generalized shapes. As the reduction in the number of vertices is not a measure of that goodness, this paper does not offer any statistics in that respect. As a matter of fact, the generalized lines are described in general by fewer vertices than the originals, but that reduction is not the purpose of the generalization. It is just a boon. Therefore, it should not be used to grade quality. The reduction in the number of vertices, the province of the popular thinning algorithms, should be performed before the generalization, if there is any surplus of vertices, or afterwards if the tolerances for the generalization were set too high.

Perkal's Proposal in the Computer

Perkal's proposal is centered on the assignment of a scale-dependent lineweight to the line to be generalized. A segment of the original line is e-convex if a band of width e centered on the original line doesn't overlap itself, e being the lineweight to be assigned to the lines at the final scale. If the final natural scale is S and the lineweight is w, e = S.w. If the segment of the original line is not e-convex, then the overlapping areas would become ink blobs at scale S. Perkal determined the overlapping areas, which he called *generalized edges* (also *boundary zones* in [Nystuen, 1966]), by rolling without slipping a

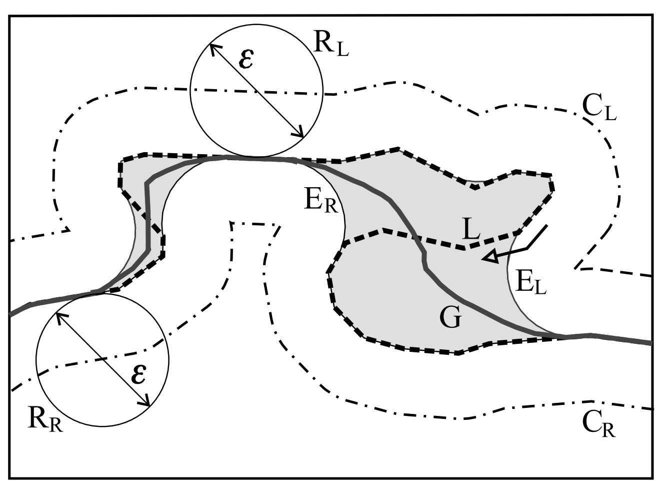


Figure 2. The geometry of the generalization

roulette of diameter e in contact with the line. The operation was performed from both sides of the line. Where the roulette could not touch the line because the mouth of a line concavity was smaller than e, the arc P-N of roulette edge between the previous and the next instantaneous rotation centers of the roulette was taken as a part of a boundary zone. Figure 2 is a small area (173m by 240m) of the hydrography of Cat Island in the Gulf of Mexico (from USGS Digital Line Graph (DLG) for 1:100,000 quad 30089-A1-TM-100). The shoreline is represented in a dashed pattern with an arrow denoting the listing sequence. The elements in the figure and their designations therein are:

L: the shoreline to be generalized

$R_L(R_R)$: the edge of the rolling roulette with diameter e=50m to the left (right) of L

$C_L(C_R)$: the loci of successive positions of the geometric center of $R_L(R_R)$

$E_L(E_R)$: the loci of the instantaneous rotation centers of $R_L(R_R)$ and arcs P-N

G: the generalized line

The area bound by C_L and C_R represent the path along L of the tip of the pen of diameter e at scale S. The shaded areas, bound by E_L and E_R, are the boundary zones. Where E_L and E_R coincide the line is said to be e-convex and invariant through the generalization. The computer implementation selects C_L and C_R as the last of the waterlines that cover an area of breath d = e / 2 at both sides of L. In reality, C_L and C_R could also be obtained by the GIS operation known as *buffering*. However, buffering doesn't in general resolve interferences between shapes nor some of the concavities that are small with respect to d. Waterlining avoids those pitfalls by gradually incrementing the distances to the input line and by resorting to GIS techniques to solve interferences.

C_R is generated from L listed in the original sequence. To generate C_L, L is reversed. E_L and E_R are generated in a similar manner, with C_L and C_R as input. In summary, four waterlining applications are required to turn Perkal's proposal into a fully automated procedure. If the procedure were used with infinite precision, E_L and E_R would coincide exactly along the e–convex edges. However they don't. The shifts between the two sections are due to multiple causes, the main one being the finite nature of the intervals and the circle stroking used in waterlining. To reduce E_L and E_R to a single line in the e–convex sections a procedure is invoked that is similar to the sliver-removal function with which vector GIS are provided. The reduction not only yields a single line but also supplies the sections of E_L and E_R that shape the boundary zones. This shaping is executed with a polygon formation procedure also similar to the fundamental GIS function. With e–convex edges as single lines and polygons as boundary zones, the procedure then classifies the endpoints of single lines with respect to the polygons as either entry (P_{EN}) or exit points (P_{EX}). The classification is based on the listing sequence of the lines. In the next step the perimeter of each polygon is walked in clockwise fashion starting from any P_{EN}. The next encountered P_{EX} is paired to the precedent P_{EN}. If there are more than two connected e–convex edges, the other (P_{EN}; P_{EX}) pairs are determined in the same manner.

Perkal's proposal ends at this point, with the definition of the boundary zones. In his time and in terms of manual generalization, that definition was all that was needed to meet his objective, a method for letting the cartographers know which sections of a line should be generalized. However, a computer generalization needs more: a way to connect the e-convex lines across the boundary zones. In [Christensen, 1999a] there is a detailed description of an extension to Perkal's proposal that performs the needed function. The next section contains a summary outline of that extension.

Joining e–convex Edges with Medial-axes

There is no theoretical justification for this method. However, the choice of a medial-axis as the path of the generalized line across a boundary zone seems a natural if not a neutral one: if a line has to be traced across a shape between two perimetral points A and B, what could be more neutral than selecting the middle course? The other two alternatives that can be readily incorporated into a systematic solution, the two halves of the

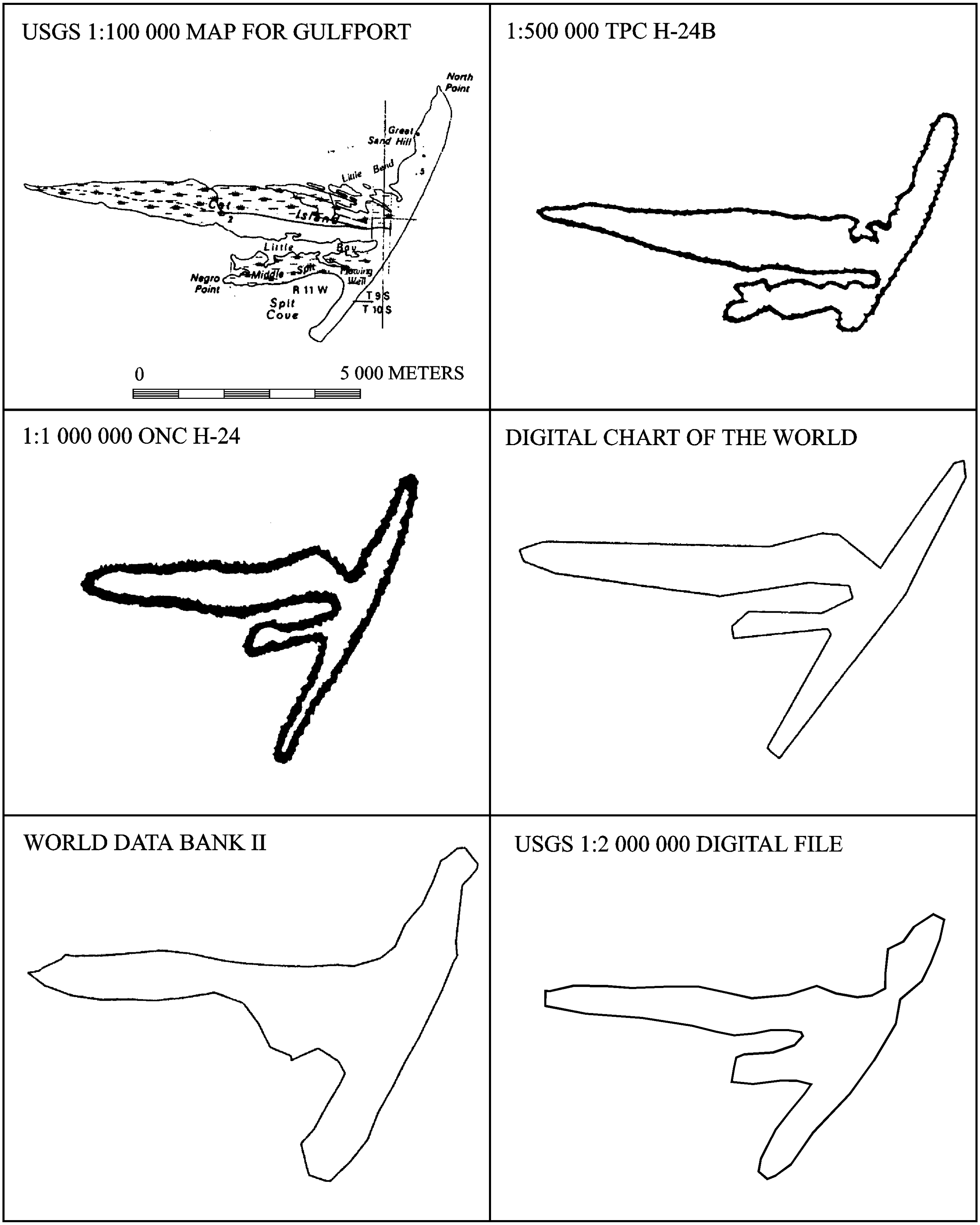


Figure 3. Six representations from Cat Island and their sources

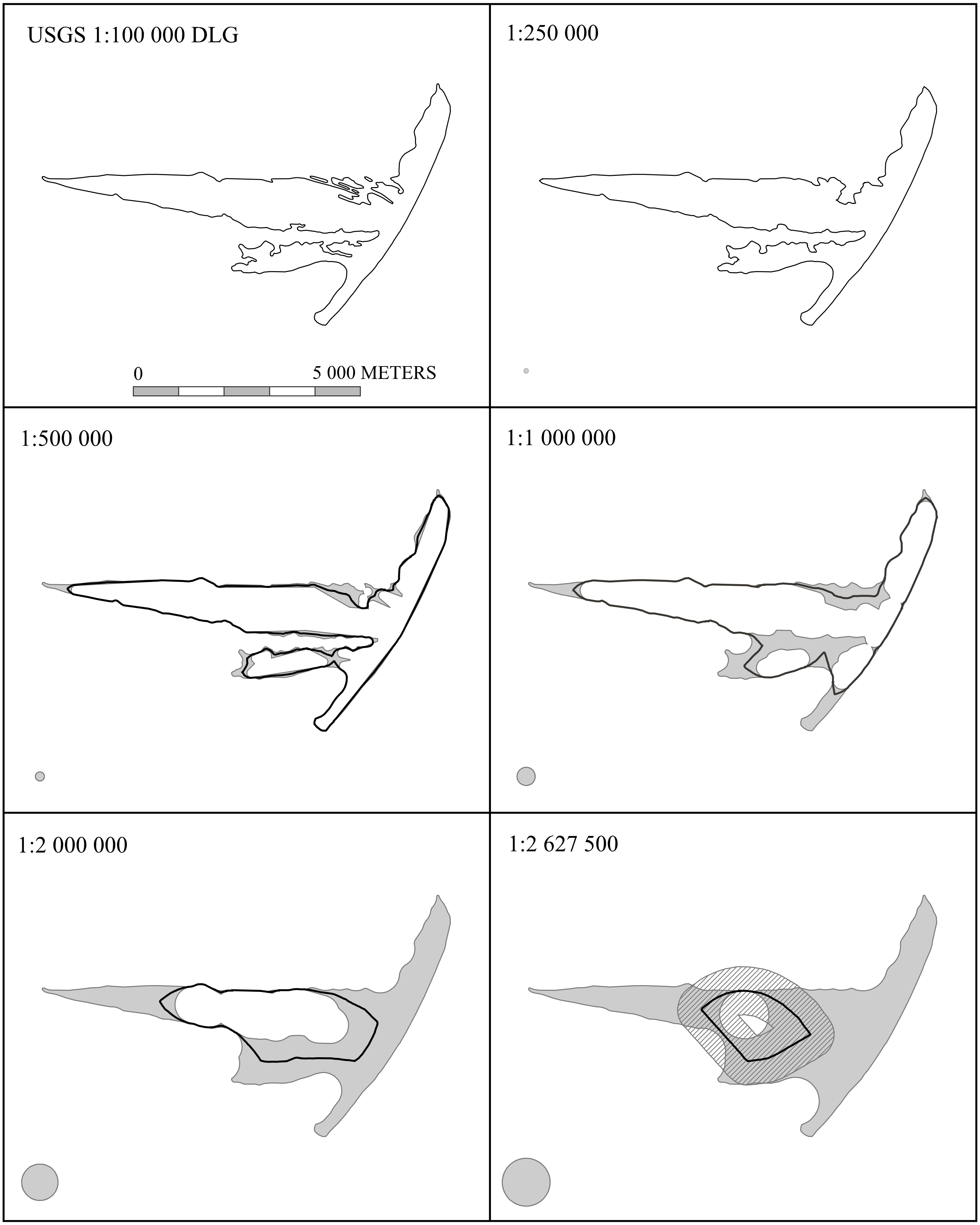


Figure 4. Five generalizations of Cat Island

perimeter (from A to B in clockwise and counterclockwise sequences, which are respectively parts of E_L and E_R), yield lines unacceptable to the eye of any cartographer [Christensen, 1999a, p. 21-24]. Furthermore, the MAT procedure works perfectly well in the most common of boundary zone configurations, those in which a boundary zone is linked to only two e–convex edges. Although the procedure may need enhancements in the other cases (four or more connections with e–convex edges), those enhancements can be introduced in a systematic way by resorting again to waterlining and MAT [Christensen, op cit].

The MAT invoked in this generalization yields an axis structured as a network and described in two files: the first contains the geometry and attributes of the arcs as well as pointers to nodes. The second is a set of node-to-arcs pointers. As it can be observed in Figure 1, a medial-axis consists of stem and branch arcs. Given a pair (P_{EN}; P_{EX}) the procedure tests the network for the two dangling ends D_1 and D_2 of branches that are respectively closest to P_{EN} and P_{EX}. Next, the segments P_{EN} -D_1 and D_2- P_{EX} are appended to the network. The line that joins P_{EN} to P_{EX} is the shortest medial-axis path between them. A navigation algorithm constructs the path.

The final generalized line is obtained as an orderly sequence of pairs (e–convex edge; medial-axis path).

Examples of Generalizations based on Waterlining and MAT

The subject for this example is Cat Island and four rocks in the Gulf of Mexico. The group is shown in six reproductions (see Figure 3) from different sources. Two of them (3b and 3c) were enlarged from paper maps in a copy machine, to match as best as possible the scale of the digital reproductions. In that way the generalizations that led to the originals could be compared with each other, as well as with the automated results illustrated in Figure 4. The sources are listed below.

3a (top left):	USGS 1:100,000 quad for Gulfport (Mississippi-Louisiana)
3b (top right):	1:500,000 TPC H-24B
3c (middle left):	1:1,000,000 ONC H-24
3d (middle right):	Digital Chart of the World
3e (bottom left):	World Data Bank II
3f (bottom right):	USGS File digitized at 1:2,000,000

The input data for the example of automated generalization was extracted from the DLG for the 1:100,000 Gulfport map(Figure 4a). The other figures are generalizations (in bold lines) of that input to these scales:

4b (top right):	1:250,000
4c (middle left):	1:500,000
4d (middle right):	1:1,000,000
4e (bottom left):	1:2,000,000
4f (bottom right):	1:2,627,500

The e-values for each generalization appear as diameters of the circles in each of the five figures. In addition, Figures 4c to 4f show the boundary zones (shaded polygons) obtained during the processes. The selection of an unusual scale for Figure 4f is explained later.

The first three generalizations, to 1:250,000, 1:500,000 and 1:1,000,000, preserve the general shape rather well. Beyond that, the island is too small with respect to the e-values being used and the shape tends to a circle, a logical conclusion if one considers that an object seen at a distance much larger than itself appears as a point feature right before it vanishes. More on this in the next section.

If Figures 3 and 4 are compared and Perkal's proposal is taken as standard, one should conclude that the generalization in Figure 3e is better than the one in 3d, which is somewhat surprising because the World Data

Bank II, source of Figure 3e, is known to have been digitized from the ONCs (source of 3d). Perhaps expediency on the part of the digitizer deserves the credit.

If compared with 4c, the generalization in TPC (Figure 3b) looks almost correct. Only missing is the southern projection, which again is somewhat puzzling. The exaggeration of Little Bay (see Figure 3a) in the paper map at 1:1,000,000 (Figure 3c) and its digital version (Figure 3d) is too large, again according to Perkal's proposal, as Figure 4d indicates. The 1:2,000,000 file (Figure 3f) is even more so. If Little Bay had to be shown at those scales there should have been reduced to a single line (See Pannekoek, 1961, p. 59).

The Feature Dropping Case

For features such Cat Island and if the e-value is gradually increased, all the boundary zones may end by combining themselves into a polygon with one outer perimeter and one or more inner perimeters. (If the original shoreline is listed clockwise, the inner and outer perimeters are respectively the loci designated as E_R and E_L in Figure 2 and vice versa). If the e–value if further increased, the inner perimeters may vanish altogether (right before the last one vanishes, its shape is very approximately a circle of radius e). Perkal suggested (op cit, p. 9) that this circumstance should be taken as an indication that the feature under that particular generalization be replaced by a symbol or be dropped from the representation. It is interesting to see a computer procedure confirming that thesis and its practical value, because with the limit e-value, the inner perimeters disappear and the computer procedure is unable to create the boundary zones and even less able to generalize the shoreline. However, if the lack of solution is disregarded, the logical connection between the shape of boundary zone and need to dispose of the feature in the manner described seems rather tenuous. More logical or at least more intuitive is the same conclusion based instead on the inability of the generalized feature to "hold color". This inability could be tested by rolling a roulette of diameter e centered on the generalized line. The polygon bound by the envelopes of the roulette positions would represent the area inked by a pen with a e-wide tip centered on the generalized line. If there is only an outer envelope, the generalized feature could not hold color and should then be disposed of. So two criteria, based each on a different type of inner polygon, could be proposed. Unfortunately, at such limit e-value the generalized line does not exist, so the second criterion is inapplicable. Perkal's insight was remarkable indeed. His criterion is not only applicable but also eminently practical. Moreover, in a way it converges with the second criterion. To witness, Perkal's proposal, if complemented by a MAT, yields boundary zones and inked areas that lack inner perimeters at very approximately the same limit e-value. Yet, Perkal never mentioned a medial-axis nor even suggested a way to reduce the boundary zones to single lines.

An approximation to the e-value that triggers the vanishing of both inner perimeters can be determined by approaching the unknown e-value from both sides. The limit e -value for Cat Island, found in that manner, falls somewhere between 525.5 and 526 meters. Figure 6f was created with the first value and includes both the boundary zone (light shading) and the inked area (dark shading). The combination of parameters e=525.5m and d=0.2mm determined the unusual 1:2,627.500.

Perkal's criterion is one of the several contributions that intermediate results can make to the automated interpretation and handling of generalization results. Others are: the connection of single-line streams to a shoreline; the connection of segregated polygons with each other and with the main feature (such as strings of water polygons resulting from the process of a strait or fjord) and the correction of undesirable features.

The need for corrections to the procedure's results, also mentioned in a previous section, is demonstrated by the nick in the Spit Cove (Figure 3a) generalized to 1:500,000 (Figure 4c). This nick and its opposite on the generalized outline of Little Bay (Figure 3a) are unwanted features of the solution to a boundary zone linked to four e-convex edges. The options for smoothing and correcting those features are detailed in [Christensen, 1999a].

Final Remarks

Among the positive aspects of the solution presented in this paper, it was noted as of particular importance that the solution yields results that directly depend on the e-value selected for the operation, i.e., that there is a firm and clear relationship between final scale and results. Thus it was demonstrated that an issue often raised against the thinning algorithms commonly and improperly called generalizations cannot be raised against this solution. With respect to Perkal's proposal, this paper established that MAT furnishes a solution to the joining of e-convex edges across boundary zones, a problem left out of Perkal's framework. Furthermore, Figure 4 shows that the solution also performs the generalization operation known as feature aggregation (islets aggregated to Cat Island). In addition, proposing Perkal's criterion, of immediate and simple application within the generalization procedure, made the case for feature dropping. On the negative it can be said that the application of waterlining and MAT to Cartographic Generalization requires a rather complex development and results in long processing times. The seriousness of the first objection, software complexity, can be greatly lessened by honoring throughout the development a modern and strict programming discipline. The second objection, long processing times, loses importance every time a new advance in computer technology is announced.

Although the basic procedures have been tested with many types of data sets, there is still room for improvement and for the treatment of yet uncovered singularities and procedural errors. Because waterlining and even more so, MAT, are useful in other cartographic problems as well as in different disciplines, the author trusts that future developments would help in perfecting and honing both tools.

Acknowledgements

The help and enthusiasm of Bruce Wright, USGS, Reston, VA, USA is gratefully acknowledged. To his sister, Sara C de Luchilo, the author is thankful for yet another skilful edit of a manuscript.

References

Neumann, Joachim (1997). *Enzyklopädisches Wörterbuch Kartographie in 25 Sprachen*. KG Saur, Munich

Blum, H. (1973). Biological shape and visual science. *Journal of Theoretical Biology*, 38, 205.

Christensen, A. H. J. (1986). Parallel pairs in automated cartography. *Cartographica*, 23, 4, 62-78.

Christensen, A. H. J. (1987). Fitting a triangulation to contour lines. *Proceedings of AUTOCARTO 8 Conference*, Baltimore, Maryland, USA, 57-67.

Christensen, A. H. J. (1998a). A global method for the delimitation of maritime boundaries. *Hydrographic Journal*, 89, 3-6.

Christensen, A. H. J. (1998b). GIS-operations on hydrographic overlays. *GIM, Geomatics Info Magazine*, 12, 8, 47-49.

Christensen, A. H. J. (1999a). Cartographic line generalization with waterlines and medial-axes. *Cartography and Geographic Information Science*, 26, 1, 19-32.

Christensen, A. H. J. (1999b). The revival of a Victorian art. Waterlining with a computer. *Cartographic Journal*, 36(1).

Nystuen, J. D. (1966). Effects of boundary shape and the concept of local convexity. *Discussion paper number 10*. Department of Geography, University of Michigan, Ann Arbor, Michigan.

Pannekoek, A. J. (1961). Generalization of coastlines and contours. *International Yearbook of Cartography*, 2, 55-74.

Perkal,J. D. (1958). An attempt at objective generalization. Translation by W Jackowski from Proba obiektywnej generalizacji, Geodezia I Kartografia. Tom VII, Zeszyt 2, 130-142. *Discussion Paper No. 10*, Department of Geography, University of Michigan, Ann Arbor, Michigan, USA.

Session / Séance 10-B

Customisable Line Generalisation using Delaunay Triangulation

Peter van der Poorten

University of Glamorgan
pmvander@uglam.ac.uk

Christopher B. Jones

University of Glamorgan
cbjones@uglam.ac.uk

Abstract:

A method of generalising linear cartographic features using an approach based on the determination of the structure of such features is proposed. This structure is determined by examining the space surrounding the feature. This is achieved using Delaunay triangulation. A variety of statistical measures are used to determine the nature of parts of this structure corresponding to segments of the line, allowing the line to be intelligently generalised and giving the user control over the style of generalisation to be produced.

Generalisation is the process of creating a legible map at a given scale from a more detailed geographical dataset. The art of map making lies in selecting both what features to include and how to represent them [Shea and McMaster 1989]. Features may require simplification if they are to be legible at a reduced scale, a process which has traditionally been performed manually by cartographers. It is highly desirable to be able to automate this process (and many methods have been proposed to do this [McMaster, 1987; Buttenfield, 1985]). One outstanding problem in this regard is the automatic generalisation of linear features.

A failing common to most linear generalisation methods so far devised is that they treat the cartographic line as an abstract entity. In so doing they fail to take into account the line's geographical nature - the fact that it represents a real physical feature. The line may represent a road, a coastline, or a river, say — and its various curves and indentations may represent significant sub-features — such as a dangerous bend, a peninsula, or a delta. Most existing algorithms generally do not 'see' such sub-features, and may remove them or distort them inappropriately – turning a peninsula into an island or grossly distorting a road in the vicinity of a sharp bend, for example. Consequently, while they perform well as point reduction techniques, they are not capable of achieving true generalisation.

Some recent work has attempted to identify sub-features, essentially by looking for so-called 'critical points' at different scales (points of maximum curvature and points of inflection) [Plazanet et al, 1995; Thapa, 1988;Wang and Muller, 1988]. This is potentially a far superior approach, possibly capable of achieving true generalisation. Using critical points is, however, a somewhat indirect means of finding sub-features. It requires additional processing to move from the points to the features [Wang and Muller, 1988]. Additionally, some form of smoothing is required in order to obtain a hierarchy of features at different scales.

This paper explores an alternative approach to the problem of identifying sub-features.

An Area Based Approach

Critical point methods approach the problem of segmenting the line into distinct 'features' by examining the line itself. An alternative approach is to identify such features by examination of the space surrounding the line. The hope is that such an approach would allow sub-features to be identified in a more direct fashion than in the former method. Additionally it should be possible to calculate a variety of descriptive statistics about the sub-features so identified.

It was decided to use the method of Delaunay triangulation for the purposes of investigating the space surrounding the line. Such an approach (strictly speaking *constrained* Delauney triangulation) has proved fruitful in exploring other aspects of cartographic generalisation. For example, the use of triangulated networks is helpful for handling the various operations (e.g. aggregation, enlargement and elimination) necessary for generalising areal objects [Jones et al, 1995].

The benefits of triangulation derive particularly from the rich neighbourhood relationships that are encoded in the triangulation. This leads for example to very efficient search procedures, as well as the identification of local proximal relations that can be exploited in triangle transformations such as collapse and re-attribution.

The essential procedure is to enclose the line(s) to be generalised in a containing box and apply a constrained Delaunay triangulation on the resulting area. The constraint being that the line segments making up the given lines and the bounding box must be retained as edges within the triangulation.

The triangulation so obtained is then examined in an attempt to gain an idea of its structure. At present this is achieved by simply counting the number of 'internal' neighbours for each triangle, where an internal neighbour is defined as sharing a common edge with the triangle under consideration where that edge is not part of the original line.

Triangles with only a single such neighbour can be recognised as a 'leaf' – that is, they lie at the terminus of a feature. Those with two neighbours form the body of a feature, and those with three can be considered as branching triangles, where two branches connect. Figure 1 shows a sample line feature and corresponding triangulation. In this case the leaf triangles are dark grey, the 'trunk' triangles are very white and the branching triangles are light grey.

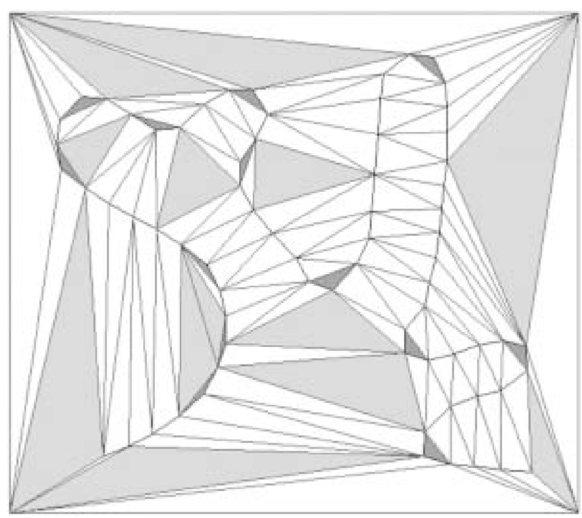


Figure 1 – A Triangulation

Once the triangles have been so categorised, it is possible to examine the hierarchy of features implicit in the triangulation. This is achieved by taking each branching node in turn and then following each of the three branches that stem from it. A number of statistics about these branches are calculated and then stored as part of a record associated with the branch node.

The statistics (or 'metrics') include:

(i) The area of the branch

(ii) The length of the boundary of the branch

(iii) The length of the branch

(iv) The average width of the branch

(v) The standard deviation of the branch width

(vi) The base angle of the branch

The first two of these are self-explanatory (i.e. exactly what they say they are). The length of the branch is calculated by taking each triangle that makes up the branch and calculating the distance between the midpoints of its two internal edges, and then adding up the total of these values. For the leaf triangle the relevant distance is that from the midpoint of its (single) internal edge to its opposing vertex.

The average width of a branch is considered to be the average width of the triangles that constitute the branch, where the width of a triangle is defined as the distance from the midpoint of its external edge to the vertex not on this edge. (This is not exactly what one would intuitively consider as the width - a better definition would perhaps be the distance between opposite edges of a branch perpendicular to its skeleton – but it is the closest that can be calculated from the triangulation.)

The 'base angle' metric is a measure of the increase in the angularity of the line that would result from cutting the relevant branch off at the base. This is a less than satisfactory choice of metric, for reasons discussed below, but is presently the only way of resolving certain difficulties with the procedure.

The use of these different metrics allows decisions to be made about processing the line so as to achieve different styles of generalisation. At present this simply allows a choice of different criteria for 'branch pruning', the selective removal of sections of the line.

Each statistic allows a different style of pruning. For example, the use of the boundary length is equivalent to using the area multiplied by a 'compactness factor' – meaning that more circularly shaped features will be more likely to be removed than less compact ones of the same area. The two most useful are the branch length and average width. Thus the generalisation can be performed in such a fashion as to remove all the narrow branches or all the short ones, or some combination of the two. This might be particularly useful when generalising a complex coastline, say.

In future work it is hoped to be able to use these, and possibly other measures, in order to be able identify the nature of branches so as to manipulate them more intelligently. This might include identification of specific types of geomorphological feature, and caricature operations that include exaggeration of retained branches. However, so far the only form of generalisation processing implemented is selective pruning of branches. This is accomplished using the following procedure.

All the branching triangles are checked and the smallest branch, according to a selected metric, is found. The segment of the line that defines this branch is then deleted, and the area is retriangulated and the branch sizes recalculated. The process is then repeated until the smallest remaining branch (measured by the chosen metric) is above the given threshold value. In general one would specify thresholds for each possible metric and remove all branches that fall below all the relevant thresholds. For example, one could specify a length threshold and a width threshold and remove all branches whose length falls below the former *and* whose width falls below the latter. However, it is still necessary to specify one of the selected metrics as the *primary* one. This is due to the two-sided nature of the line. When a branch is removed, the branches adjoining it (on the other side of the lines that constitute its boundary) are affected and will almost certainly change size. These affected branches may consequently become newly eligible for pruning or cease to be so eligible, requiring metrics to be recalculated before pruning continues. An apparently simpler strategy would be to simply remove all the branches with metrics below the chosen thresholds in one pass. However if this strategy to be employed it is quite possible that some of the new branches thus created would have smaller measurements than those deleted. Thus, pruning must be done sequentially, with metrics being recalculated each time a branch is pruned, with the process ceasing when no more eligible branches are found. This requires one particular metric to be chosen to determine the *order* of pruning, though all the chosen threshold values are used to determine *which* features are pruned. In practice which metric is chosen as primary does not appear to greatly affect the outcome.

For example, if a width and a length threshold is specified, but the former is chosen as primary the procedure will occur as follows:

All the branching triangles are checked and all the branches which are both shorter than the length threshold, and narrower than the width threshold are found. Of these, the narrowest is selected for deletion. The segment of the line that defines this branch is then deleted, and the area is retriangulated and the branch sizes recalculated. The process is then repeated until all the remaining branches are either too long or too wide to be deleted.

The choice to combine thresholds by means of an 'AND' operator (that is, in the example above branches are eligible for removal if they are both too short AND too narrow) is a somewhat arbitrary one. One could just as well decide to use 'OR' (remove all branches that are either too short OR too narrow). The choice of operator might in future be added as an additional user specifiable option.

An additional option that could be added, in addition to the metrics discussed above, is the option to declare lines, or segments thereof, to be single sided. What this would mean is that only branches identified on one (specified) side of the line would be considered. This would greatly simplify the processing, as the deletion of a branch would not require retriangulation and re-evaluation of the area on the opposite side of the line. More to the point, however, it would provide another option when determining the style of map to be produced. This might be particularly relevant when considering features such as coastlines in which promontories such as peninsulas only exist as areal features on the landward side.

A relatively simple further development would be to allow for widening (and hence exaggerating) branches that fell below a certain width threshold. This would require, however, a means of ensuring such an adjustment did not reduce the width of the neighbouring branches below the allowable threshold.

Benefits of an Area Based Approach

The primary benefit of this approach is that it offers a method of obtaining a relatively direct access to the structure of the line, allowing for the possibility of more intelligent means of generalisation such as caricature. A further important benefit is the fact that topological consistency is maintained implicitly. A major drawback of most existing methods, simple point filtering algorithms in particular, is their tendency to create fictional intersections between lines or even within the same line. These problems often have to be cleared up with post-generalisation processing [Muller, 1990]. With an area-based method such problems generally do not arise, provided all the linear features on a particular map are generalised together, producing a single triangulation.

Another positive feature of the approach is that it allows a significant degree of control over the style of generalisation produced. While many existing algorithms allow the specification of parameters, generally these parameters merely control the degree of point reduction obtained. In fact, for the existing point reduction type methods having multiple parameters to tune is often seen as a drawback, as it is not clear what each parameter in fact *means*. Instead the user is confronted with multiple means of achieving the same end, a reduction in the number of points used in the line, with no clear indication of what the difference is between tweaking parameter A or parameter B in terms of the type of generalisation obtained. In the method discussed here, it is possible for the user to specify a particular style of generalisation, even with just the simple pruning routines so far implemented.

Sample Results

Some example results are shown in Figures 2-5. Figure 2 shows a line subjected to 'pruning' where the constraining metric is the branch length. Figure 3 shows the same line pruned by branch width, while Figures 4 and 5 the pruning is performed by area, with high and low threshold values respectively. . In each case the original line is shown in black, the 'pruned' form in grey. Note that the average width metric is not entirely accurate for branches that are only one triangle in size. This will be corrected in future; however such branches are generally appropriate for pruning in any case.

Note that in each of these examples only one metric is used at a time in order to best demonstrate the effects of each metric. It would of course be possible to specify length, width and area thresholds at the same time.

As one would expect, when pruning is performed based on branch width the long narrow features are removed while the short, wide features are retained. With a length metric the reverse is the case.

Figure 2 - Pruning by path length

Figure 3 - Pruning by path width.

Issues to be addressed:

(1) *The difficulty of distinguishing between ‘corners’ and genuine features.* The problem arises because the resultant branches of the triangulation are very hard, if not impossible to distinguish - the difference lies in what happens at the edges of the branches.

An example of this situation is shown in Figure 6. The V-shaped indentation in the line is clearly a meaningful feature, but the triangulation appears to detect very similar shaped branches where the line bends sharply. These branches do not correspond to features in the same sense, as at least one of the base points lies on a nearly perfectly straight line rather than at a bend.

At present this problem is partially solved by use of the ‘base angle’ metric. This measures, for a given branch, the increase in angularity of the line that would result from amputating the branch at its baseline. By only removing branches in which this number is positive or zero (i.e. for which pruning would not introduce a sharp angle in the line where non previously existed) one can, usually, avoid chopping the corners off of features, while still pruning identically sized genuine features. However, this approach is unsatisfactory because it depends entirely on the local properties of the line at the base of the relevant segment. The angle concerned is measured relative solely to the points either side of the points determined to represent the base of the feature, and they are not necessarily typical of the shape of the line as a whole. To obtain a more genuine representation of the angle between a detected ‘feature’ and the rest of the line it would be necessary to look at a smoothed representation of the line or its curvature - and this would bring us back to the previously discussed ‘inflection point’ type methods.

There are also other specific cases where there is a difficulty in getting from the shape of the triangle network to the structure of the line. Partly this is a question of finding more sophisticated metrics to describe features and of identifying what these metrics are telling us about the line. At present none of the measures used give much information about the relations between branches, for example at what angle a sub-branch branches off from its parent.

However it may be that certain kinds of information are not easily extracted from the triangulation. It is expected that switching to the use of skeletons will help here.

(2) *The fact that it is essentially a sub-setting method.* This reduces the freedom of manipulation. The triangulation method isn't perfect at identifying exact branches of the original line. This is also a problem with inflection point detection methods [Wang and Muller, 1988]. It means that simply pruning branches tends to leave ugly 'stumps'. This can be fixed by an ad-hoc solution to correct the branch base, but this isn't really addressing the underlying problem. The problem is concealed to some extent in the existing approach by the use of the 'base angle' metric, but this is itself unsatisfactory, for reasons discussed above, and for the additional problem that it prevents the creation of 'stumps' by simply preventing that branch being pruned at all.

The real problem is that sub-setting methods are appropriate for point filtering algorithms (which is what most existing methods really are) but what we really want is to extract the underlying conceptual shape of the line, rather than making a fetish of the individual points. It is the branch we wish to prune, not the individual points making up the branch. Again, moving to skeletons would help get away from the existing points towards a more abstract representation of the structure of the line.

(3) *The fact that a line has two sides.* This is the fundamental complicating factor of the approach. However, as discussed earlier, the ability to process a line from the vantage point of one particular side may, in some circumstances, be an advantage. The major effect of this issue is the necessity for processing to be done sequentially, with adjustments to the line being done in a clearly defined order. For example, with simple branch pruning, a branch, the smallest judged by some appropriate metric, is deleted, the area retriangulated (to allow

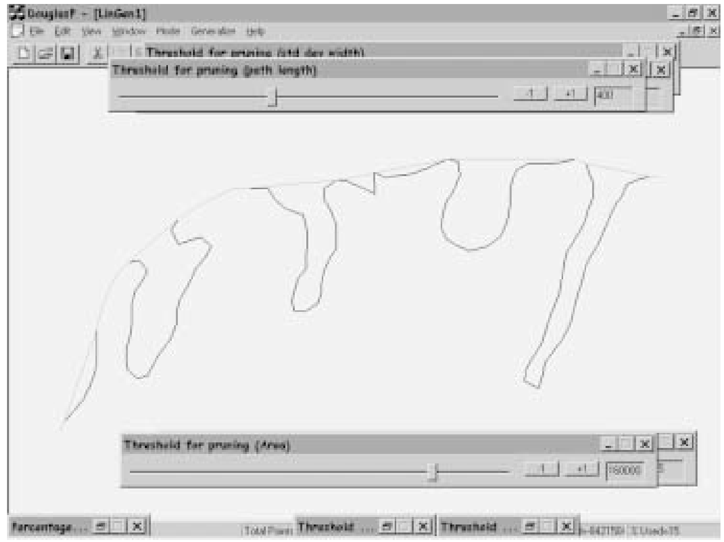


Figure 4 - Pruning on the basis of area (large threshold value)

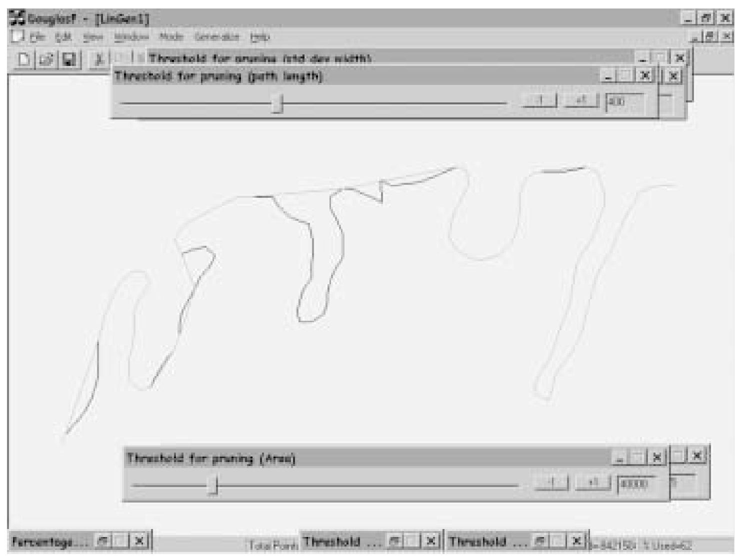


Figure 5 – Pruned by Area (smaller threshold)

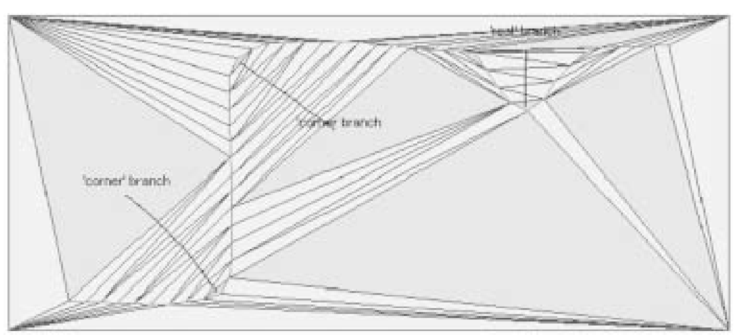


Figure 6 – Note the artificial branches at the 'corners'

for the fact that removal of this branch has changed the form of the triangulation on the *other side* of the line) and then the process is repeated. Unlike the two previous problems this problem is liable to be exacerbated by a move to a skeleton representation. With triangulation methods it is relatively simple to use local retriangulation in the area affected by the removal of a branch. Recreating the skeleton is likely to be a more computationally expensive procedure.

Future Work

As mentioned, an obvious way forward for this work would be to switch from triangulations to the use of explicit skeletons. The fundamental advantage of this would be that it would represent the line structure at a more abstract level. This abstract representation could then be analysed and simplified. This would require the devising of a method for reconstructing the original line from the modified skeleton. Some work has already been done on this question in relation to the simplification of shapes [Ferley et al, 1997], and this avenue looks quite promising. The major question is how to regenerate the line from the skeleton without creating undesirable ‘artefacts’, such as ‘wrinkles’.

Such a skeleton could be obtained from the (line) Voronoi diagram, or, more simply, an approximate skeleton could be derived from the triangulation itself, by connecting either the centroids of the triangles or the bisections of their edges.

Once such a skeleton had been obtained it could be point-filtered, pruned, smoothed, and structurally analysed, allowing such procedures as caricature and exaggeration. An issue to be considered is choosing an appropriate data structure to be used for representing the skeleton.

Conclusion

It is clear that existing point-filtering algorithms are essentially incapable of true generalisation. Such true generalisation requires a method capable of identifying the structure of the line that is to be generalised, allowing intelligent cartographic operations to be applied to it. Existing work aimed at analysing line structure relies on the detection of such features as points of inflection and maxima of curvature. This paper describes an alternative approach, which offers a number of advantages over inflection-point type methods, including the ability to maintain topological consistency implicitly, and the ability to offer control over the style of generalisation desired. It is expected that further progress will come from a switch from the use of triangulation to explicit skeletons, which would represent a further step away from point filtering towards a more abstract representation of line structure.

References:

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

Shea, S. and McMaster, R.B. (1989), Cartographic Generalisation in a digital environment: how and when to generalise, *Auto-Carto 9*, Baltimore, ACSM/ASPRS, 56-67.

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

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

Thapa, K. (1988), Automatic Line Generalization Using Zero Crossings, *Photogrammetric Engineering and Remote Sensing*, 54(4) (April ‘88), 511-517.

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

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

Ferley, E.,Cani-Gascuel, M.P., Attali, D. (1997), Skeletal Reconstruction of Branching Shapes, *Computer Graphics Forum*, 16(5), 283-293.

Wang, Z. and Muller, J.C. (1988), Line Generalisation based on analysis of shape characteristics, *Cartography and Geographic Information systems*, 25(1), 3-15

Session / Séance 43-A

Généralisation Automatique du Linéaire : Quelques outils pour mesurer la forme des routes

Xavier Barillot

Laboratoire COGIT IGN 2 avenue pasteur 94160 Saint-Mandé
C. élec : Xavier.Barillot@ign.fr

Résumé

Le contexte est la généralisation automatique et indépendante du linéaire et plus particulièrement celle des routes : nous disposons d'une polyligne (suite ordonnée de points représentant une route) trop détaillée et qu'il faut transformer pour la représenter à une échelle inférieure. Qu'il s'agisse de savoir quelle opération effectuer sur cette polyligne (lissage, filtrage, caricature, ...) ou bien de valider le résultat obtenu (qualité intrinsèque et comparaison avec la polyligne origine), des outils sont nécessaires : il s'agit entre autres de mesures de position, de topologie et surtout de forme.

La caractérisation de la forme d'une polyligne pose divers problèmes dont le premier est dû à sa nature discrète : cette caractérisation devra être indépendante du niveau de détail de la saisie. Il faudra soit créer des outils indépendants de cette nature discrète, soit prévoir un prétraitement des polylignes destiné à les rendre continues; c'est cette dernière optique qui a été retenue: on travaille sur des modélisations de polylignes; plusieurs modélisations polynomiales ont été étudiées, dont celle par paraboles/cubiques.

Ensuite, la caractérisation de forme d'une polyligne peut se concevoir à des niveaux très différents, allant du global (sinuosité, ...) jusqu'au très local (forme d'un virage, ...). Sachant qu'on a toujours la possibilité de construire le global à partir du local, l'accent est mis actuellement sur le deuxième niveau, sans toutefois oublier le premier.

Le problème central est donc de trouver et d'analyser des outils de détection et de caractérisation des virages. Après avoir constaté les difficultés pour délimiter perceptuellement les virages, nous montrerons les faiblesses de la définition classique en généralisation qui est la partie d'une courbe située entre deux points d'inflexion. Ensuite, nous étudions dans quelle mesure le calcul de la courbure peut permettre d'isoler les virages. Puis, il restera à fixer les paramètres nécessaires et suffisants pour caractériser un virage, sans perdre de vue notre objectif de généralisation. Pour conclure, nous verrons les problèmes posés par le passage du niveau local au niveau global.

Introduction

Pour généraliser automatiquement une route, nous disposons d'une palette d'algorithmes très variée (filtrage, lissage, caricature, …) qui nous permettent de façon interactive de répondre à la plupart des cas réels. Le problème est de comprendre pourquoi, pour telle route ou tel tronçon de route, un algorithme avec tel paramétrage est choisi, pourquoi après tel algorithme et au vu du résultat, tel autre algorithme est lancé, pourquoi encore tel résultat a été jugé si insatisfaisant que l'opération est annulée et le processus réinitialisé.

Pour se donner les moyens de réaliser ces choix de manière automatique, nous proposons de qualifier les objets à transformer à deux niveaux: d'une part pour choisir l'algorithme adapté à la ligne et d'autre part pour évaluer la qualité du résultat (Cf. Figure 1). Concernant le premier niveau, on suppose que l'application, manuellement ou interactivement, d'un même algorithme ou d'une même opération à deux objets provient du fait qu'ils ont au moins une caractéristique en commun; de plus, on suppose que la qualification de cette caractéristique pourra permettre de fixer l'intensité avec laquelle cet algorithme va être appliqué, c'est-à-dire la valeur des paramètres. Deuxièmement, après application d'un algorithme, on estimera la qualité intrinsèque (acceptabilité) et extrinsèque (comparaison avec l'original) du résultat.

Ces outils qui vont nous permettre de qualifier et comparer les objets (routes initiale et généralisée), nous les appellerons des mesures [McMaster, 1983; Buttenfield, 1984]. Il pourra s'agir aussi bien de mesures évidentes comme la longueur de la route, le nombre de points d'intersection entre la route et sa correspondante lissée, que de plus complexes comme le nombre de virages ou la sinuosité. Comme nous l'avons mentionné ci-dessus, ces mesures vont être utiles principalement à deux niveaux dans le processus (phases 1 et 2 dans la Figure 1 et pour plus d'informations concernant le processus, nous renvoyons à [Ruas et al., 1996; Ruas, 1999]):

- Premièrement, elles vont nous permettre de décider quelle opération choisir ou quel algorithme utiliser pour traiter une ligne donnée. Ceci constitue bien entendu un gros problème qui est l'établissement des règles de correspondance entre les types de lignes caractérisées par ces mesures et les séquence d'algorithmes (des travaux sont en cours actuellement à l'IGN qui vont dans ce sens: ils sont entre autres basés sur des techniques d'apprentissage automatique en temps différé ou réel. Ce problème de mise en correspondance ne sera pas abordé ici : Cf. [Ruas, 1999]).
- Deuxièmement, ces mesures serviront à estimer la qualité du résultat donné par l'algorithme choisi. En effet, même si on suppose que la base de règles de correspondance est parfaite, il conviendra d'évaluer la qualité du résultat, éventuellement de proposer une autre valeur ou un intervalle de valeurs pour l'algorithme utilisé ou bien de proposer un autre algorithme à appliquer si le résultat intermédiaire est acceptable ou encore de revenir à l'état initial; lorsque le résultat obtenu est acceptable, il nous permettra d'enrichir et d'améliorer la base de règles (pour la mise en correspondance). Dans le cas contraire, la caractérisation sera mise en cause : il faudra créer d'autres outils de caractérisation, des mesures plus fines.

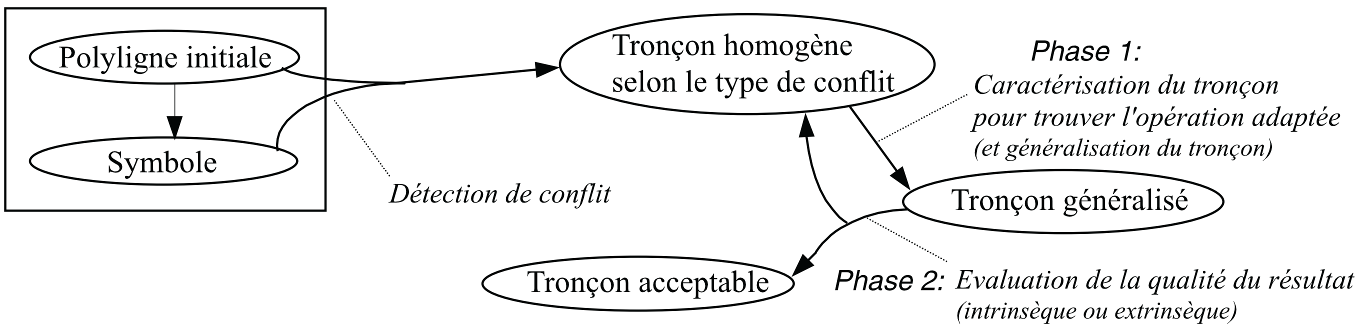


Figure 1 : Processus simplifié

Nous allons nous intéresser ici aux problèmes rencontrés et aux solutions envisagées pour les résoudre lorsqu'on cherche à qualifier la forme d'une ligne (qu'il s'agisse de caractérisation de la ligne de départ ou bien d'évaluation du résultat).

Notons que ce problème de reconnaissance de forme est récurrent dans d'autres domaines mais que la spécificité du point de vue de la généralisation nous contraint à reconsidérer le problème: en effet, dans notre domaine, les objets sont donnés et il ne reste qu'à qualifier leur forme, contrairement à l'analyse d'image par exemple, où le principal travail est de construire les objets à partir d'une représentation en mode raster. Néanmoins, les recherches déjà menées dans ces domaines peuvent nous être utiles: en particulier, il est reconnu qu'il est illusoire de chercher à décrire la forme d'un objet avec un seul indicateur, qui permettrait de distinguer les objets de forme différente et de regrouper les semblables.

1. Modélisation

Les objets sur lesquels nous travaillons sont représentés par des *polylignes* consistant en une suite ordonnée de points, c'est-à-dire une ligne brisée représentant la route et ne pouvant pas s'auto-intersecter. Il s'agit d'une approximation qui va avoir un effet sur le résultat des mesures destinées à qualifier cette route. Pour une mesure de longueur, on peut supposer que le biais engendré est négligeable, mais pour calculer la pente, la courbure ou d'une façon générale qualifier la forme de la ligne, on conçoit facilement qu'il faille prévoir un prétraitement.

Une première méthode consiste à considérer la courbure de la ligne brisée comme une distribution qui est ensuite lissée (avec un lissage gaussien) plus ou moins fortement suivant le degré de sélection des détails désiré [Barrault, 1998; Fritsch, 1999]. La méthode choisie ici est différente: au lieu de partir de la courbure, nous partons de la polyligne elle-même que nous transformons pour la rendre dérivable en tous points. Ainsi, on pourrait qualifier la première méthode d'*ascendante* (on part de la courbure pour éventuellement remonter à la ligne) et la deuxième de *descendante* (la pente, puis la courbure sont déduites d'une modélisation de la ligne).

Nous proposons une interpolation polynomiale en cubiques et paraboles. Ce choix a premièrement l'avantage d'être simple au niveau du calcul, de faciliter l'analyse, et deuxièmement d'assurer le nombre de points d'inflexion minimal (Cf. ci-dessous le critère de choix du degré du polynôme). Entre chaque doublet de points consécutifs de la polyligne, nous interpolons un polynôme de dimension deux ou trois; la courbe qui est issue de ce processus est donc une suite de portions de cubiques et de paraboles.

D'abord, nous assurons la dérivabilité de cette courbe en chacun des points de la polyligne en imposant que les pentes à gauche et à droite, c'est-à-dire les pentes des polynômes choisis de part et d'autre de chaque point, soient égales. De plus, afin d'éviter une dérive (oscillations grandissantes de la courbe) pendant la construction de la courbe, nous imposons une valeur pour les pentes aux points de la polyligne, avant le calcul des polynômes: en un point donné, la valeur de la pente est fixée à celle de la droite joignant les deux points voisins (Cf. Figure 2 : la pente en P_2 est celle de la droite P_1P_3); notons qu'étant donné que les points sont ordonnés, on tiendra compte du sens de parcours: les angles des pentes seront donc compris dans un intervalle [0,2ð[; par exemple, une pente d'angle ð/2 ne sera pas equivalente à une pente d'angle 3p/2: dans le premier cas, la courbe arrive au point par en-dessous et dans l'autre, par au-dessus. Nous résumons ci-dessous les paramètres de cette interpolation pour un segment $[P_i,P_{i+1}]$ de la polyligne:

(1) a. Le polynôme doit passer par les points P_i et P_{i+1}.

b. Les angles de pente en P_i et P_{i+1} sont imposées dans [0,2p[: resp. angles du vecteur $P_{i-1}P_{i+1}$ et du vecteur P_iP_{i+2}.

Ensuite, il faut se donner le critère de choix pour le degré du polynôme; on ne choisira naturellement une cubique que lorsqu'on a un changement de progression de la pente des segments et donc un point d'inflexion; dans tout les autres cas, l'interpolation sera parabolique. Le processus de modélisation proposé comporte donc trois étapes rappelées ci-dessous:

(2) a. Calcul des pentes en chacun des points de la polyligne initiale.

b. Détermination du type de polynôme (parabole ou cubique) pour l'interpolation.

c. Interpolation entre deux points successifs de la polyligne avec contraintes de pente.

Notons que la propriété de non auto-intersection des polylignes impose pour les pentes aux extrémités d'un segment des valeurs telles qu'il existe toujours *une* parabole (unique) et une *infinité* de cubiques obéissant aux contraintes fixées en (1). Le choix de la cubique est guidé par le souci de proximité entre modélisation et polyligne (ie cubique et segment); elle est calculée dans le repère dont l'axe des abscisses est la bissectrice des pentes aux extrémités du segment considéré.

A la place d'une ligne brisée, nous disposons à présent d'une courbe dérivable; cette modélisation va nous être utile pour caractériser la forme de la route, le calcul de la courbure sur la polyligne étant impossible puisqu'elle n'est pas dérivable.

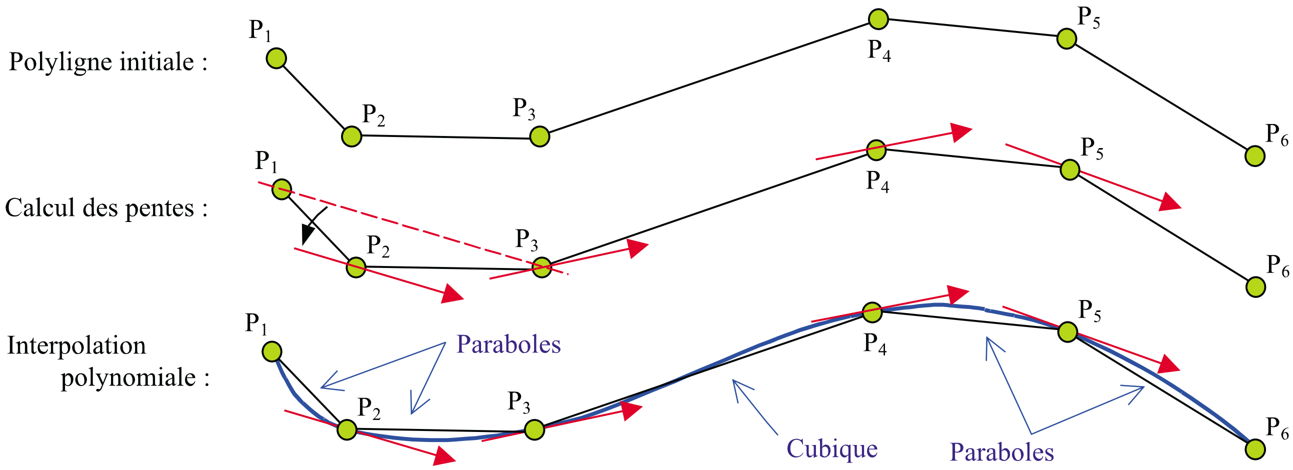


Figure 2 : Modélisation

2. Evaluation de la forme d'une polyligne

2.1 Qu'est-ce que la forme d'une polyligne ?

Commençons par délimiter ce que nous entendons par *forme*, sans perdre de vue que l'objet que nous cherchons à caractériser, une polyligne, est en relation avec un objet réel sur le terrain (une route) par l'intermédiaire d'un objet cartographique avec une symbolisation donnée: pour chacun de ces trois niveaux, le concept de forme peut se définir par contraste avec les autres paramètres de description (position, orientation, taille, densité, …). En figure 3, dans le but de fixer les choses, on a des exemples d'objets ne variant que par l'un de ces concepts:

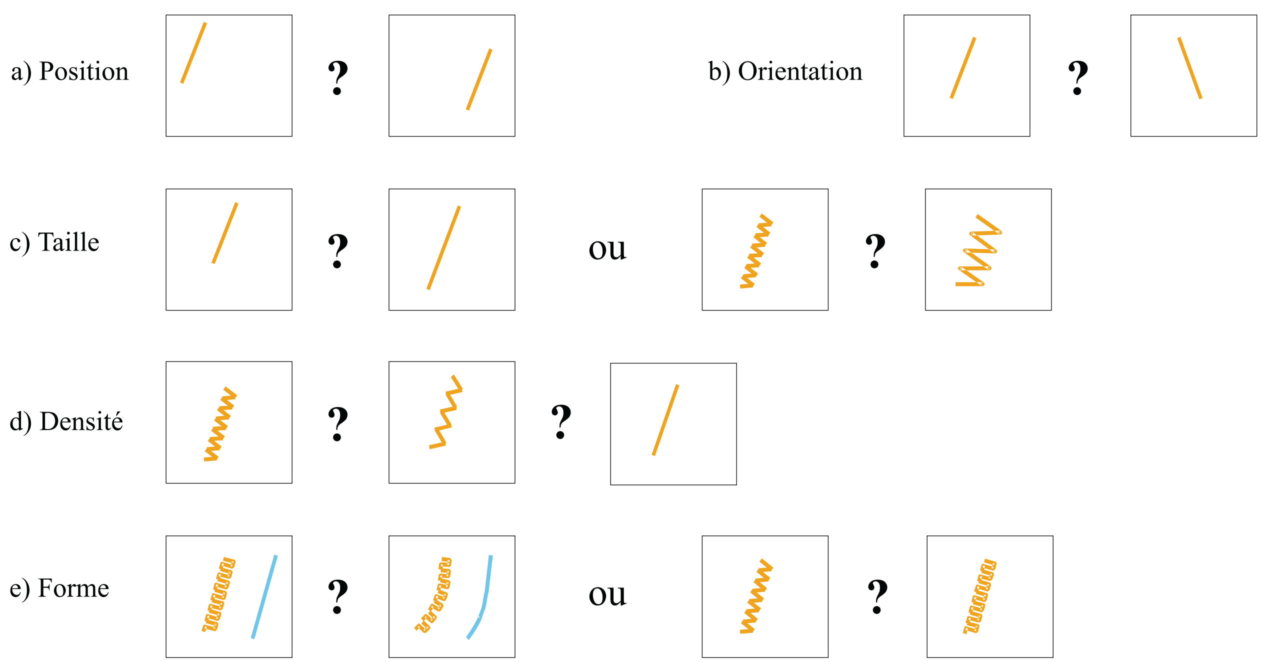


Figure 3 : Paramètres de description d'une ligne

Lorsqu'on compare deux objets, on peut effectuer les tests suivants:

a) Evaluer leur *position*: une translation permet de les superposer (rem: la position d'un objet est celle de son barycentre).

b) Evaluer leur *orientation*: une rotation permet de les confondre (rem: certains objets n'ont pas d'orientation (ex: disques)).

c) Evaluer leur *taille*: on se ramène à une taille identique, soit par une homothétie de centre le centre de gravité de l'objet (à gauche en Figure 3c), soit par une affinité orthogonale dont l'axe est l'orientation de l'objet (à droite en 3c).

d) Evaluer leur *densité*: une affinité orthogonale d'axe perpendiculaire à l'orientation et passant par le centre de gravité de l'objet permet de les rendre équivalentes (rem: la taille de l'objet ne sera pas conservée, car le calcul de densité repose sur une caractéristique locale: il ne se conçoit que sur des lignes présentant un motif répété de manière sensiblement identique; il faut noter que les concepts de densité, de taille et de forme sont très liés).

e) Comparer leur *forme*: une fois réalisés les quatre tests ci-dessus, si les deux objets sont superposables, c'est qu'ils ont la même forme (rem: on verra plus bas que la forme d'un objet peut se concevoir à plusieurs niveaux allant du global au local).

Ainsi, une différence de forme entre deux objets est tout ce qui ne pourra pas être résorbé par une tranformation simple globale (translation, rotation, homothétie et affinité). Il s'agira de ce qui est invariant par translation, rotation, … (ce qui est le cas en particulier de *la dérivée seconde de la polyligne,* c'est-à-dire *la courbure, les inflexions, ...)* :

A présent que nous avons une vision plus claire de ce que recouvre ici le concept de *forme*, voyons comment s'y prendre pour l'appréhender. Commençons par remarquer, comme on le constate en e) dans la figure 3 ci-dessus, que la forme d'une ligne peut se concevoir à plusieurs niveaux allant du *global* (à gauche) jusqu'au très *local* (à droite: c'est-à-dire au niveau d'un virage ou éventuellement d'une série de virages identiques).

Deux approches sont envisageables pour qualifier la forme d'une ligne, soit nous considérons la ligne dans sa globalité, soit nous partons d'un niveau plus bas, à savoir le virage: dans cette deuxième optique, la polyligne est considérée comme une succession de virages. C'est cette dernière approche qui sera privilégiée dans cette communication et à laquelle nous allons à présent nous consacrer. Indépendemment des impératifs géométriques ou mathématiques, ceci se justifie par le fait que le virage est une entité primordiale aussi bien pour le conducteur d'un véhicule que pour le lecteur de la carte, que ce soit dans un souci de sécurité ou de reconnaissance du terrain. Il faut d'abord décomposer la ligne (dans le processus proposé par l'IGN (Cf. Figure 1 en introduction), la ligne est d'abord segmentée en portions selon le type de conflit de symbolisation [Mustière, 1995]; l'objet de départ est donc plus exactement une portion de ligne) en virages et ensuite caractériser chaque virage, puis éventuellement les regrouper pour construire des tronçons homogènes, en particulier du point de vue de la forme, ce qui donne trois phases :

(3) a. Décomposition en virages (niveau *micro*).

b. Caractérisation de chaque virage (forme, taille, orientation, …).

c. Regroupement en tronçons homogènes / Caractérisation de la forme de la ligne (niveau *macro*).

Il ne sera question ici que des deux premières étapes, mais avant même de considérer la première, il convient d'étudier ce qu'on peut entendre par *virage* et comment accéder à cette entité.

2.2 Le concept de virage

2.2.1 Qu'est-ce qu'un virage ?

La réponse n'est pas facile, on s'en aperçoit dès qu'on essaie de décomposer des courbes un peu complexes. Bien sûr, il est fréquent d'avoir des hésitations sur la localisation du début et de la fin d'un virage, mais plus grave, il arrive aussi qu'on obtienne un nombre de virages différent (Cf. figure 4 où le nombre de virages et leur découpage va dépendre du point de vue qui peut être celui du dessinateur de la carte, du conducteur d'une automobile, d'un mathématicien, …).

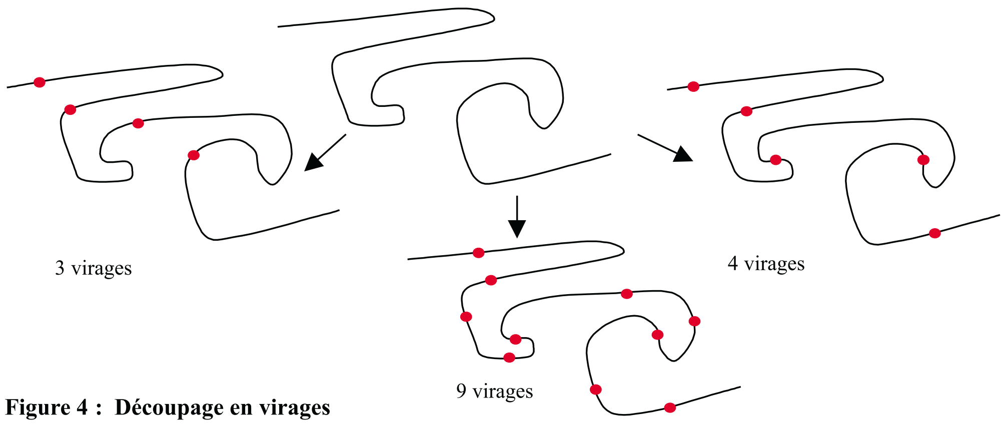


Figure 4 : Découpage en virages

On constate donc que la notion de virage est loin d'être évidente, même si, dans la plupart des cas, on constate un relatif consensus sur le découpage (en particulier sur le nombre de virages). Il reste à présent à relier ces intuitions à des critères mathématiques.

La définition la plus courante est «tronçon maximal de même concavité», c'est-à-dire la *portion de courbe comprise entre deux points d'inflexion consécutifs*. Mais, on pourrait aussi donner d'autres définitions comme par exemple *portion de courbe comprise entre deux Parties Droites*. Selon qu'on choisit l'une ou l'autre de ces deux définitions, le découpage est dans certains cas sensiblement différent, comme on le constate dans la figure 5 :

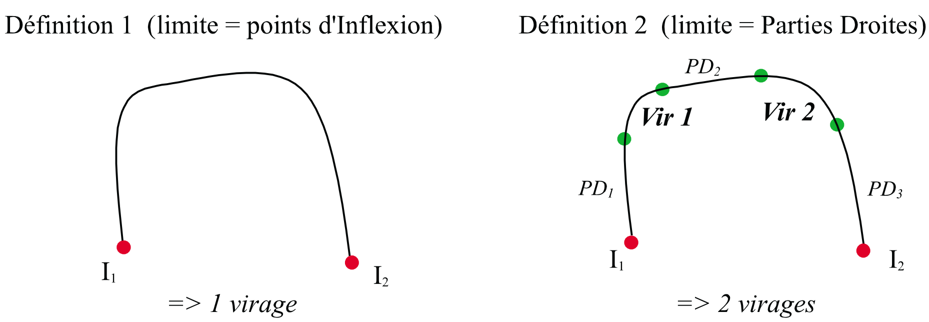


Figure 5 : Limites de virage

Selon les cas, nous choisirons l'une, l'autre ou les deux définitions. De toute façon, il faudra prévoir le calcul des parties droites *et* des points d'inflexion (et plus largement de la courbure) pour chaque polyligne dont nous voulons caractériser la forme. Comme nous allons le voir dans les sections suivantes, ces deux résultats seront indispensables, que ce soit au niveau de la segmentation en virages ou de leur caractérisation.

2.2.2 Les outils de détection des virages et leurs inconvénients

Le calcul des *points d'inflexion* est basé sur la modélisation présentée en partie 1; il y en a un pour chaque segment interpolé par une cubique. On obtient donc une série de points comme dans la figure ci-dessous:

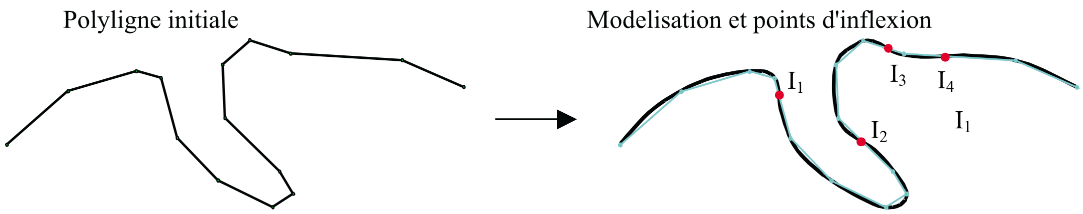


Figure 6 : Points d'inflexion

Dans la figure 6, on constate l'émergence de plusieurs problèmes. D'abord, on aimerait négliger certains virages par rapport à d'autres (par exemple, celui délimité par I_3 et I_4). Ensuite, les points d'inflexion ne sont pas toujours placés face à face comme c'est le cas du virage délimité par I_1 et I_2, ce qui complique la caractérisation des virages, en particulier le calcul de l'amplitude ou de la base. La délimitation des virages à partir des points d'inflexion est donc une première étape, mais insuffisante. Nous verrons en 2.2.3 des moyens de résoudre ces problèmes.

La détection des *parties droites* d'une courbe (i.e. modélisation polynômiale d'une polyligne) pose le problème du choix du seuil; en effet, vu que la courbe en question est une succession de paraboles et de cubiques, elle ne comprend jamais de segments de droite. Un tronçon de la courbe sera donc dit *partie droite* si la courbure en chacun de ses points est inférieure à un seuil donné. Toute la difficulté réside dans le choix de ce seuil.

2.2.3 Propositions pour un meilleur découpage en virage

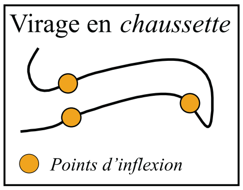
Lorsqu'on découpe une ligne en virages d'après la position des points d'inflexion, trois principaux problèmes surgissent:

a) Mauvaise délimitation des virages (Cf. virage I_1I_2 dans la figure 6):

Une solution peut être de calculer les parties droites de la ligne modelisée; sauf exception, chaque point d'inflexion est inclus dans une partie droite. La méthode consiste alors à choisir les limites des virages dans ces parties droites.

b) Virages en *chaussette* (Cf. figure ci-contre) :

Nous avons un cas de virage particulièrement typique dans le linéaire routier en zone montagneuse, appelé *virage en chaussette* [Plazanet, 1996], et que le découpage par les points d'inflexion ne permet pas de constituer. Il faut donc prévoir autre chose comme par exemple:
- Considérer les virages dont la différence des angles de pente aux extrémités est supérieure à π (rem: on rappelle que les pentes sont orientées et les angles compris dans $[0,2\pi[$).
- Calculer la pente et la position du point suivant et les comparer avec celles du 1er point.



c) Pas de hiérarchisation des virages (Cf. virage I_3I_4 par rapport au virage I_1I_2 dans la figure 6):

Pour détecter les virages secondaires (ou du moins, la présence de différents niveaux), deux méthodes sont utilisées:

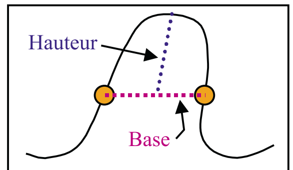
- Test si une portion comprise entre deux points d'inflexion consécutifs appartient à une partie droite.
- Calcul des différences de pente entre deux points d'inflexion consécutifs et classification des virages en fonction.

Nous disposons à présent de moyens pour découper chaque polyligne en virages; il convient alors d'établir des outils pour qualifier la forme de chacun d'eux.

2.3 Caractéristiques de forme d'un virage

Une fois connue la position, l'orientation et la taille d'un virage, il reste à qualifier sa forme. Les mesures les plus couramment citées pour cela sont:

- Rapport de la hauteur sur la base.
- Symétrie du virage par rapport à l'axe.
- Nombre de points d'inflexion.
- Nombre d'extrema de la fonction de courbure.
- …



Hauteur & base

2.3.1 Calcul de la hauteur/base (Cf. Figure ci-contre)

Cette mesure est un indicateur de l'aspect resserré ou au contraire épaté d'un virage. Dans le cadre de la classification donnée dans la figure 3, elle permet de caractériser la densité d'une série de virages; cependant, lorsque ceux-ci sont en nombre insuffisant ou lorsqu'ils sont très hétérogènes et que cela interdit de parler de *série*, cette mesure devient simplement un indicateur de forme de chaque virage.

La hauteur/base est basée sur leur délimitation, problème déjà évoqué plus haut dans la section 2.2. Ainsi, pour des virages très resserrés, on n'aura pas de problèmes pour calculer la base, contrairement à la hauteur, tandis que c'est l'opposé pour les virages épatés : les conséquences d'une erreur de localisation des limites de virage sont en effet moins importantes sur le cacul de la base que sur celui de la hauteur dans le cas de virages resserrés. Si on ne dispose pas d'une bonne délimitation, on peut proposer le processus suivant:

a) On détermine si un virage est resserré ou épaté : pour cela, on peut calculer la hauteur/base *brute*, c'est-à-dire avec la délimitation basée sur les points d'inflexion, ainsi que le *vecteur orientation* du virage et le degré de resserrement sera donné par le quotient de la norme du vecteur par la distance séparant les points d'inflexion.

Méthode de calcul du vecteur d'orientation :

1. se placer dans un repère où l'abscisse des points d'inflexion I_1 et I_2 est nulle.
2. Déterminer le point M_0 où la dérivée s'annule.
3. Calculer le symétrique P_0 de I_2 par rapport à M_0.
4. *Le résultat est la somme des vecteurs I_1M_0 et M_0P_0.*

b) Si le virage est épaté, on base le calcul sur la délimitation existante.

c) Si le virage est resserré, on en calcule l'orientation et on en place les limites sur la droite perpendiculaire à l'orientation et passant par le milieu des points d'inflexion.

d) Dans ces deux cas, une vérification pourra être faite sur la délimitation choisie en considérant les parties droites et les distances aux sommets de part et d'autre.

2.3.2 Etude de la fonction de courbure

Comme nous l'avons vu au chapitre 1, nous pouvons calculer la courbure en chaque point de polyligne à partir de la modélisation. La fonction de courbure associe abscisse curviligne et courbure d'un point de la modélisation. Pour chaque virage, cette fonction est en forme de cloche plus ou moins régulière (convexe ou concave) puisque la courbure est nulle aux extrémités (notons que la courbure n'est nulle aux extrémités du virage que si les points d'inflexion ont été conservés comme limites; dans le cas contraire, on n'a pas exactement une valeur nulle.). Ensuite, l'étude des extrema de cette cloche permet d'établir une typologie de forme des virages:

- Cloche avec *un seul* extremum: dans ce cas, la forme du virage est comparable à celle de la courbure, cette dernière étant généralement plus resserrée (Cf. figure 7). De tels virages seront appelés *pointus*.
- Cloche avec *trois* extrema: si les deux maxima ont la même intensité et que le minimum les séparant est conséquent, on aura des virages dits *émoussés* (Cf. figure 8). Dans le cas contraire, on est ramené à des virages *pointus*.

- Choche avec *au moins cinq* extrema: le plus souvent, on peut se ramener à l'un des deux cas précédents en éliminant les extrema secondaires (Cf. figure 9).

Lorsque le nombre d'extrema est supérieur à 3, nous avons besoin d'outils de caractérisation plus fins de la fonction de courbure d'un virage que le nombre d'extrema pour déterminer son type (pointu, émoussé, …).

Le point commun de tous les virages de la figure 7 ci-dessous, corrolairement à leur forme *pointue*, est que la détermination du sommet ne pose pas de problème. Les critères manquants pour caractériser leur forme sont principalement l'aspect plus ou moins épaté (donné par la *hauteur/base* qu'on a vu ci-dessus) et la symétrie.

Dans la figure 8 , nous avons deux virages émoussés (le premier et le dernier); les deux pics de courbure sont comparables. Dans l'exemple du milieu, bien que la fonction de courbure comporte trois extrema, nous n'avons pas affaire à un virage émoussé; le premier pic de courbure est en effet plus important que le deuxième (nous verrons plus loin d'autres critères pour distinguer ces cas des virages vraiment émoussés).

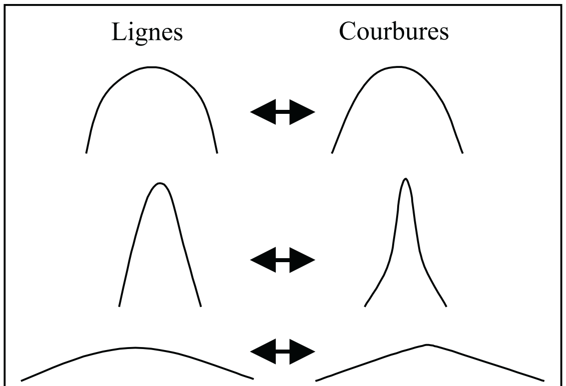


Figure 7 : 1 extremum de courbure

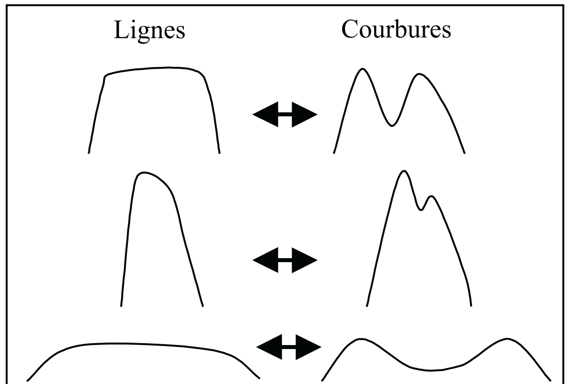


Figure 8 : 3 extrema de courbure

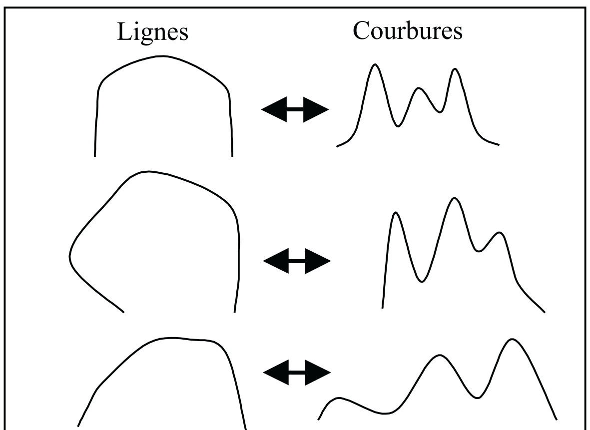


Figure 9 : 5 extrema de courbure et plus

En figure 9, le premier exemple se ramène à un virage pointu, le sommet du virage devant être placé au niveau du troisième extremum de courbure qui est le plus faible des trois maxima.

Le deuxième exemple ne se ramène facilement ni à un virage pointu ni à un virage émoussé; on a en fait un virage atypique.

Le troisième virage se ramène à un virage émoussé, le premier maximum étant à négliger par rapport aux deux autres.

Ces exemples montrent la nécessité de caractériser de façon plus précise ces extrema, aussi bien en abscisse qu'en ordonnée. Il va falloir comparer les intensités, l'étendue des extrema, comme on le voit sur la figure 10 où est reprise la fonction de coubure du premier virage ci-contre.

Une première expérimentation nous a permis de mettre au point quelques mesures pour déterminer le type de virage d'après la fonction de courbure et de fixer certains seuils : par exemple, selon les notations de la Figure 10, si les conditions $\{0.8 < E_1/E_5 < 1.25\}$ et $\{E_3/\min(E_1,E_5) < 0.4\}$ sont vérifiées, on peut penser qu'on a affaire à un virage *émoussé*.

Notons qu'aucune mesure à elle seule ne peut donner le type du virage; cependant, tel résultat donné par une mesure nous amènera à calculer une autre mesure pour le confirmer. Ceci implique de fixer un ordre de calcul des mesures.

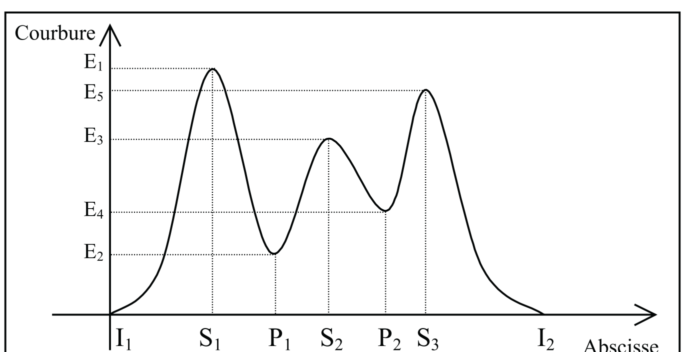


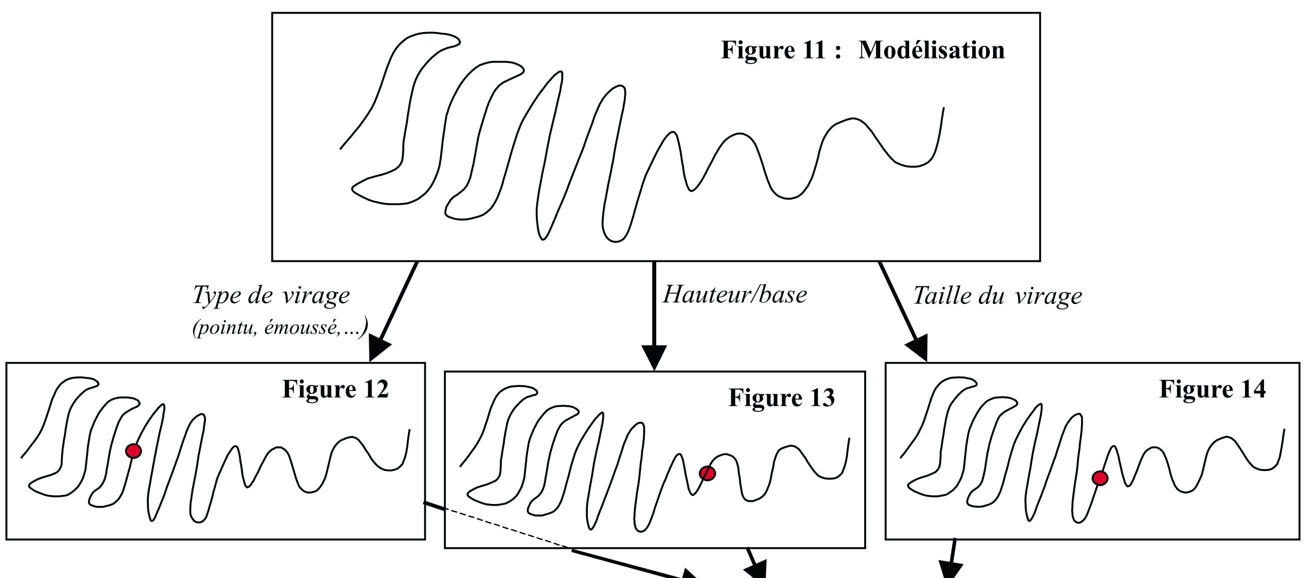
Figure 10 : Fonction de coubure d'un virage

Conclusion

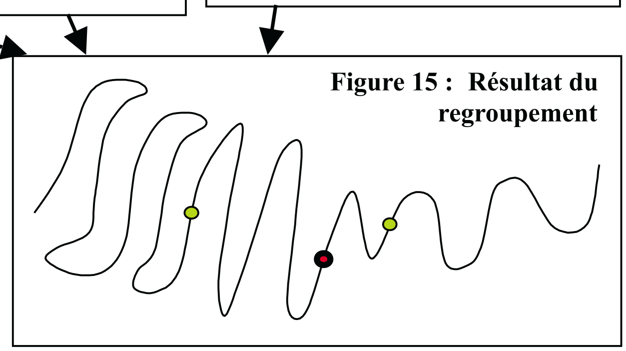
Ce travail est loin d’être terminé: il nous reste en particulier à regrouper les virages pour former des *tronçons homogènes*.

Deux principaux problèmes interviennent pour réaliser cette tâche: tout d’abord, il faut définir des seuils, c’est-à-dire trouver la valeur limite du résultat d’une mesure donnée à partir de laquelle il faut considérer qu’on change de type de virage. Ensuite, il faut choisir les mesures dont on va se servir pour regrouper les virages entre eux; en particulier, elles devront être classées par ordre d’importance pour pouvoir décider si une série de virages doit ou non être considérée comme homogène (cette hiérarchisation est complexe et consistera sans doute en un système de règles dépendant de la configuration). En effet, chaque mesure va entraîner un découpage différent et il est vraisemblable qu’il y aura des inclusions mais aussi des chevauchements entre tronçons comme on le constate dans les figures 11 à 15 ci-dessous: nous avons une série de virage (Figure 11) que l’on découpe selon trois critères:

- Le *type* du virage en 12 (pointu, émoussé, chaussette, …).
- La *taille* (hauteur) en 13.
- La *hauteur/base* en 14.



Si on met toutes les mesures au même niveau, rien ne permettra d’établir une hiérarchie entre tronçons. A partir de la modélisation de la figure 11 et en combinant trois mesures, nous obtenons quatre tronçons et à moins de disposer de règles d’ordonnancement, nous sommes incapables de trouver le découpage donnant les deux tronçons les plus homogènes possible. Dans cet exemple, il apparaît que c’est le critère *hauteur/base* qui est primordial, suivi du *type* de virage. Nous obtenons donc la hiérarchie donnée dans la figure 15 (en gris clair figure le découpage secondaire).



Cet exemple montre le genre de problème auquel nous sommes actuellement confontés : nous cherchons à établir des critères d’importance relative des mesures de forme ainsi que des autres types de mesures (orientation, taille, …), lorsqu’on regroupe les virages en tronçons homogènes.

Notre travail doit donc être poursuivi jusqu’à l’obtention d’une palette d’outils et de mesures de forme suffisamment riche; il restera ensuite à mieux les *comprendre* (notamment savoir comment instancier les seuils et relier chaque concept (symétrie, sinuosité, …) à un ensemble de mesures géométriques) et à appréhender leurs interactions, leur importance relative dans un contexte donné; une fois ce travail réalisé, les mesures en résultant seront intégrées dans un processus de généralisation basé sur des techniques d'apprentissage automatique (le but de cette technique d'apprentissage est de trouver pour chaque type de ligne la séquence d'algorithme de généralisation idéale, chaque ligne étant caractérisée par une série de mesures). Cette phase permettra alors de mieux définir les besoins: il conviendra d'améliorer les outils de caractérisation ou même éventuellement d'en créer d'autres. La phase suivante qui est la réalisation d'outils de comparaison entre ligne généralisée et ligne originale occasionnera certainement aussi des mises au point de notre travail.

Bibliographie

Barrault, M. (1998). Rapport de thèse, Université de Marne-la-Vallée/IGN, France.

Buttenfield, B. (1984). Line Structure in Graphic and Geographic Space. Rapport de thèse, Université de Washington, Etats-Unis.

Fritsch, E. (1999). Algorithmes de généralisation basés sur le lissage de la courbure. ACI, Ottawa, Canada.

McMaster, R.B. (1983). Mathematical measures for the evaluation of simplified lines on maps. Université du Kansas, Etats Unis.

Mustière, S. (1995). Mesures de la Qualité de la Généralisation du Linéaire. Université Paris I/ENSG, France.

Plazanet, C. (1996). Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes). Rapport de thèse, Université de Marne-la-Vallée/IGN, France.

Ruas, A. et Plazanet, C. (1996). Strategies for automated generalization. 7e International Symposium on Spatial Data Handling, vol. 1, Delft, Pays-Bas.

Ruas, A. (1999). Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. Rapport de thèse, Université de Marne-la-Vallée/IGN, France.

Session / Séance 39-B

A System for Automated Generalisation of Contour Lines

Zhilin Li, Haigang Sui *and* **Jianya Gong**

Dept. of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University
Kowloon, Hong Kong
e-mail: lszlli@polyu.edu.hk

Abstract

This paper describes a system for automated generalisation of contour lines. In this system, a line simplification algorithm based on a natural principle is employed for the generalisation of individual lines and a triangulation-based algorithm is developed for derivation of a new contour line from two original neighbouring contours. The algorithms are implemented in such a way that they guarantee (a) no self-intersection and cross-intersection; (b) very smooth resultant contours; and (c) very coherent relationship between resultant contour lines, with shape very faithful to the original contours (at a larger scale). The system is intensively tested using real data sets and its superiority has been revealed by the testing results.

1. Introduction

The use of contour lines for the representation of the variations of a 3-D surface was originated from France in the 18th century. It might be said that such a technique is the most widely used and the most effective means for the representation of a 3-D variations on a 2-D surface and has therefore been considered as being one of the most important inventions in cartographic history.

Contour representation has been used in topographic maps, thematic maps and other graphics. Indeed, it is contours that are the most fundamental element of topographic maps. When the generalisation of topographic maps is carried out for either derivation of maps at a smaller scale (in the case of cartography) or for "real-time" zooming (in the case of GIS), contour lines need to be generalised together other features.

Contour generalisation is important for a number of reasons. Firstly, contour is one of the most important types for natural features. Secondly, the generalisation of contour is a common, important, challenging and vexing problem in cartography and GIS. Thirdly, it can be regarded as the key step for the development of more comprehensive procedure for map generalisation, if the problem can be successfully solved.

This paper describes a new technique for the generalisation of contours, which is based on a natural principle. Section 2 will examine existing approaches for contour generalisation. Section 3 provides a brief description of the theoretical background – the natural principle of objective generalisation. Section 4 discusses the implementation of algorithms for contour generalisation. The new technique (algorithm) has been intensively tested and the results are reported in Section 5. Finally, some concluding remarks are made in Section 6.

2. Critical examination of contour generalisation approaches

In contour representation, the quality and/or effectiveness is determined by a few parameters, mainly the spacing between contour lines, faithfulness of individual contour lines to the 3-D surface and coherent relationship between contour lines.

Generalisation of contours means to transform contour representation from a larger scale to suit for the representation at a smaller scale while the quality and/or effectiveness of the representation is still compatible at that level.

In general, two approaches are possible, i.e. direct generalisation and indirect modelling. In the latter, a three-step procedure is used, i.e. (a) contour data are used to construct a DTM of the area; (b) the DTM is to be generalised; and (c) to new contours are to be produced from the generalised DTM. The resultant contours are supposed to be generalised. The issue now becomes how to generalise the DTM. The advantage of this approach is that there will be a guarantee of no intersection between contours if appropriate contour interpolation algorithms are employed. However, so far, no algorithm for the generalisation of DTM surfaces with comprehensive theoretical basis has been known to the authors. In most cases, a low-pass filter is applied to smoothed out the DTM (e.g. Weibel, 1987) and there is no theory behind the use of such filters. Another disadvantage of this approach is that a loss in accuracy and/or fidelity may be caused and noise introduced during the contour/DTM/contour conversion process.

In the direct approach, individual contour lines are simplified (generalised) to suit the representation at target (smaller) scale. The main problem with this approach is that, as pointed out by many researchers (e.g. Weibel, 1996), most of existing line simplification algorithms provides no guarantee of intersections (i.e. self-intersection and cross-intersection) except a few (Li and Openshaw, 1992, Wang and Müller, 1992, de Berg et. al., 1995). This could be the main reason why this direct approach is not very popular. This is the practical part of the problem. To avoid intersections, researchers have used some constraints. One of these approaches is to employ Voronoi diagrams for points along all contour lines as an extra constraint (Wu, 1987).

The theoretical part of the problem associated with this direct approach is that most of existing so-called line simplification algorithms employ a strategy of selective omission of points along the lines. The theoretical background of these algorithms is the discovery by Attneave (1954), i.e. some points on an object are richer in information than others and these points are sufficient to characterize the shape of the object. These algorithms can be grouped into three types (Li, 1995), i.e. corner detection, polygonal approximation and a hybrid technique (a combination of the first two). The one popularly used in GIS is the Douglas-Peucker (1973) algorithm, which is also known as Ramer algorithm (see Li, 1995). Attempts have also been made to improve this algorithm (e.g. Li, 1988). However, as has been pointed out by many researchers (e.g. Li, 1993), it is misleading to use these algorithms for line generalisation purpose because the original purpose of these algorithms was not for line generalisation but for curve approximation (to approximate a curve line using straight line segment). Although there are also other algorithms based spectral analysis (e.g. Boutoura, 1989) for line simplification (generalisation), the experience gained by the authors reveals that these algorithms are also not working well.

3. Theoretical basis of the line generalisation algorithm– a natural principle

To generalize contour line for the representation at target scale, the following conditions must be fulfilled, i.e. (a) The contour lines must be simplified in structure; (b) The resultant contours must be smooth enough in appearance, and (c) The natural characteristics of contour lines (e.g. being parallel, without self- and cross-intersections, geometrically similar to the shape of the 3-D surface) must be retained.

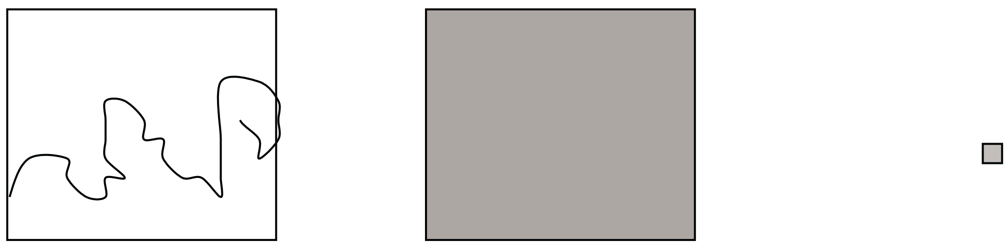
Among available algorithms, Weibel (1996) has bade a critical evaluation and found that the Li-Openshaw algorithm (Li and Openshaw, 1992) (see Weibel, 1997, p.124 for terminology) is able to guarantee no self-intersection. And later in this paper, it will be demonstrated by experimental testing that other conditions for contour generalisations can also be fulfilled by careful implementation of this natural principle.

The so-called Li-Openshaw algorithm is the line generalisation algorithm was developed by Li and Openshaw in 1992 and published *in International Journal of Geographical Information Systems*. The theoretical basis of these algorithms is the so-called a natural principle for objective generalisation, discovered by Li and Openshaw (1993). The natural principle states:

> "*for a given scale of interest, all details about the spatial variations of geographic objects beyond certain limitation are unable to be represented and can thus be neglected*".

From this principle, a simple corollary can be derived as follows: *by neglecting all spatial variations within a certain limitation, natural results can be obtained for the generalisation.*

This corollary can be easily implemented. This can be demonstrated by Figure 1. It shows that, for a given target scale, all spatial variations within certain limitations can be completely neglected and a point can be used to represent this limitation (area). The appropriate size of this limitation is about 0.6-0.7mm at map scale, as concluded by Li and Openshaw (1992) based on intensive experimental testing.



(a) Spatial variations within a certain limitation

(b) The spatial variations within this limitation can be completely neglected

(c) This area is then represented by a point at a smaller scale

Figure 1 The natural principle by Li and Openshaw (1993): A point or a raster cell can be used to represent the spatial variations within a certain limitation.

4. Implementation of algorithm for generalisation of individual contours

Based on this natural principle, Li and Openshaw (1992) have implemented three algorithms for line generalisation and they have recommended the "vector-raster" algorithm. In this algorithm, the average of the coordinates of the first and last intersections between the line and the grid cell is taken as the position of the new point (to represent the cell). Of course, the cells can also be overlapped to produce more realistic results and to avoid the dependency of starting points, as suggested by Li and Openshaw (1992).

However, as mentioned in the previous section, the generalisation of contour lines is more complicated than that of a single line. The most difficult part is to make sure that no contours will intersect each other. For this purpose, the simplest implementation of this natural principle of objective generalisation is to lay down a raster grid with the cell as the SVO (smallest visible object) and to take one point (at any position) within a raster cell as the representative of the spatial variations within this cell. The most appropriate value for the SVO size is about 0.6 to 0.7mm at target map scale (Li and Openshaw, 1992).

In addition, a number of technical issues also need to be solved. The first one is smallest loop (closed contour) to be retained. It is quite obvious that if the looped line is within a cell, it should be deleted. However, depending on the position of starting point of the grid, the same line may appear on 4 neighbouring cells. Therefore, it is still safe to delete closed contour lines occupying fewer than 4 cells after generalisation.

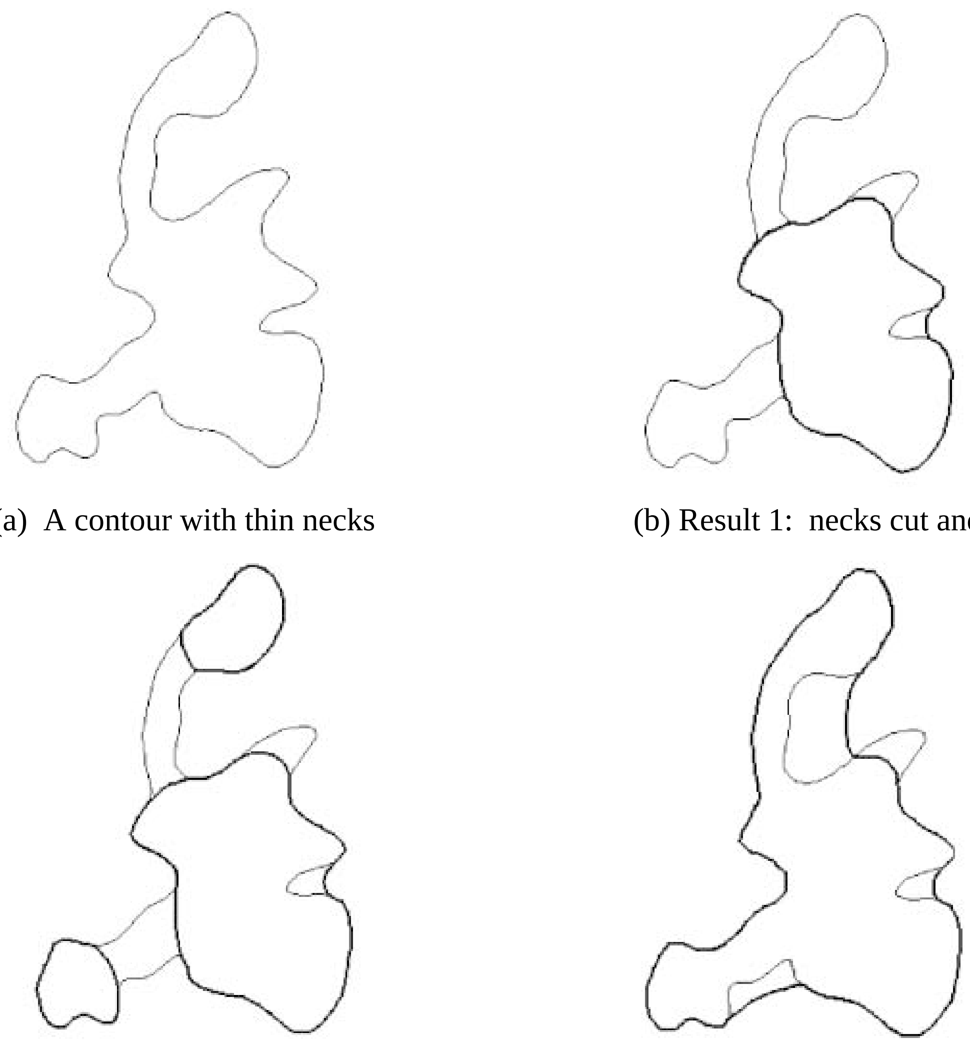
The second one is the treatment of thin necks of contour lines. Figure 2(a) shows an example. If the neck is too thin (thinner than 2 cells), there are three solutions, i.e. to throw the small convex parts away (Figure 2b), to form a close loop for the small convex parts (Figure 2c) or to exaggerate the thin necks (Figure 2d). However, some additional constraints must be imposed while the exaggerating the concave parts. In this implementation, this option is omitted. In fact, the first two options are not isolated but interconnected. If the convex parts are too small (i.e. occupying less than 4 cells), then the first option is taken, or else the second option is taken.

5. Algorithm for deriving a new contour from two neighbouring contours

When a large-scale map is generalised to a small-scale map, small variations of contour lines would be removed while the most important characteristics of contour lines should be kept. When the scale change is dramatic, then the spacing between two contour lines (i.e. planimetric contour interval) will be reduced to such a level that they will touch each other. As a result, there is need to remove some of the contour lines in order to retain the clarity of the maps. Indeed, a rough guideline for contour intervals at different scale is summarised in Table 1:

Scale	*Contour Interval*
1:200,000	25 to 100 m
1:100,000	10 to 40 m
1: 50,000	10 to 20 m
1: 25,000	5 to 20 m
1: 10,000	1 to 10m

Table 1. Contour intervals at different map scales



(a) A contour with thin necks

(b) Result 1: necks cut and small loops lost

(c) Result 2: necks cut and small loops separated

(d) Result 3: necks exaggerated

Figure 2. Various possible results for a contour line with thin necks

In the change of contour interval, there are two possible cases. The first case is very simple, i.e. to use a contour interval with a multiple of the original contour interval. For example, the original contour interval is 1m, but the new contour internal is 5m. In this case, the solution is to select the contour lines at the multiple of 5 metres, i.e. 5m, 10m, 15m, 20m etc., from original set of contour lines. However, sometimes, there is a need to change from 2m to 5m or from 20m to 50m. In this case, a new contour line needs to be derived from two original lines, i.e. 5m contour from 4m and 6m contours (Figure 3).

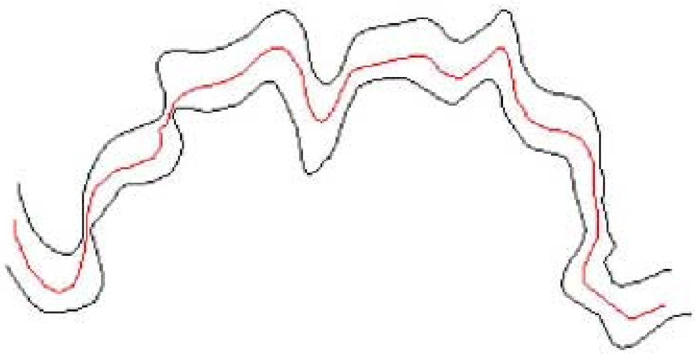


Figure 3 A new contour to be derived from two original neighbouring contours. (The middle one is the new one derived from the other 2)

One way to derive a new contour line from the original two lines is to derive the skeleton (e.g. Bookstein, 1979; Shapiro et al., 1980, Su et al., 1998) of the areas formed by the two original lines. Experience gained from experimental testing shows that many small unwanted branches may be produced by skeleton algorithms (Su et al., 1998) and therefore, alternative methods have been sought. The method used in this project is the triangulation-based algorithm.

The idea behind this algorithm is (a) to construct a triangular network using the points on the contour lines; (b) to interpolate points with the height of new contour; and (c) to joint these points to form a new contour line. In this implementation, the Delaunary triangulation algorithm is employed. Figure 4(b) shows an example of such a triangular network constructed from the contour map shown in Figure 4(a). The resultant new contour would be something as shown in Figure 4(c). As can be seen clearly, there is a problem with this network, i.e. there is an artificially flat area formed by a set of “flat” triangles on the left side of map. This will be a common problem if there are spike-like lines. To avoid this problem, one possible solution is to add some points along the ridgeline. The result is shown in Figure 4(d). The quality of the resultant contour will be dependent on the accuracy of those feature points added. Another solution is to set a constraint for the triangulation - no more than 2 points should be selected from a contour line in forming a triangle.

6. System implementation

In the previous sections, the theoretical background and algorithmic implementation of the new technique for contour generalisation have been discussed. In this section, a brief discussed on the system components will be given.

The system consists of the following core components:

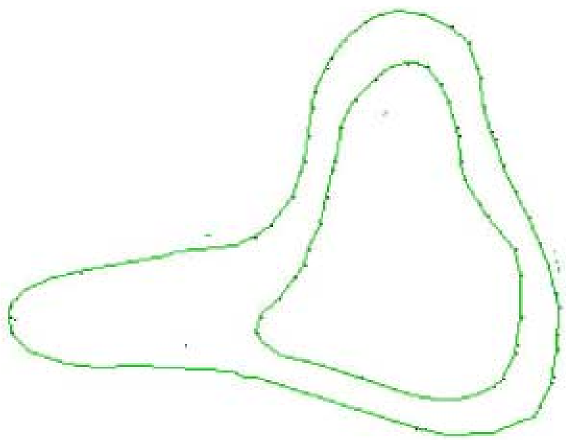
a) A pre-processing module to remove gross errors, to connect the broken index line, and so on;

b) An module to manipulate the change of map scale and consequently contour interval;

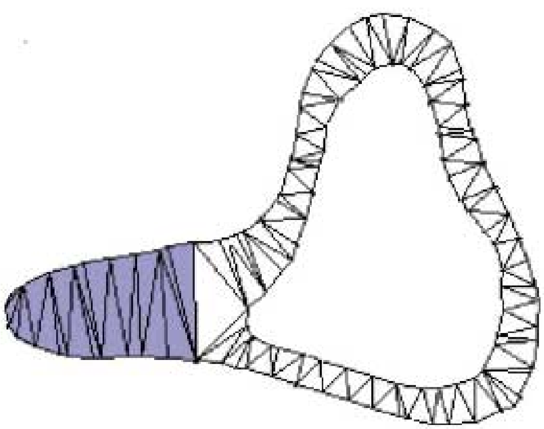
c) A basic module to generalise contour lines;

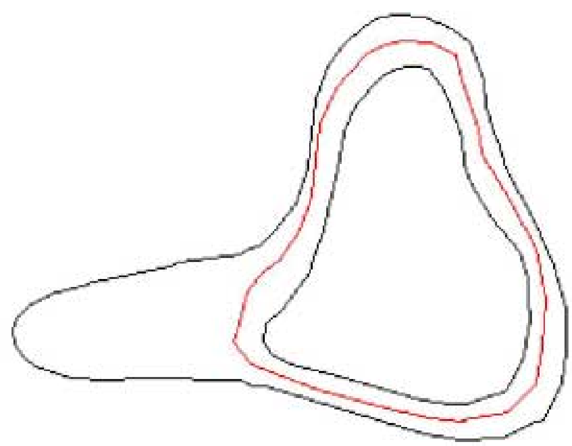
d) A post-processing module to remove small sloops and to do other tidy work.

At the moment, the system makes use of GeoStar as a platform, which is developed by the GIS Centre at the Wuhan Technical University of Surveying and Mapping. Therefore, many other functions such as format conversion, editing and so on are all there.

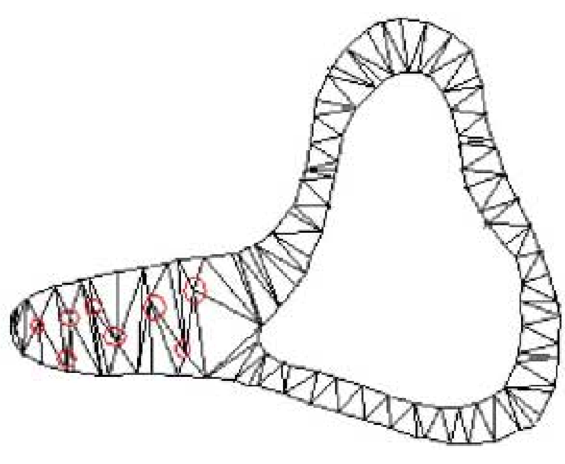


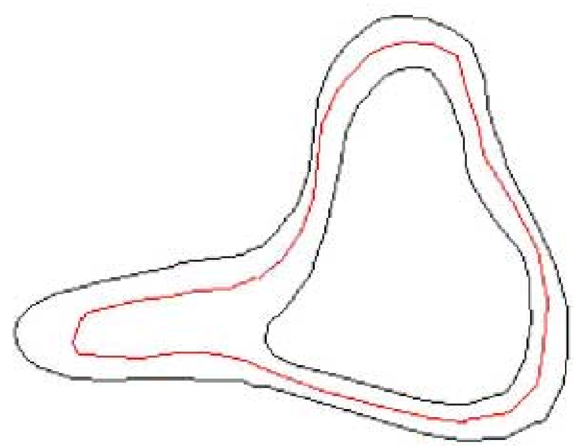
a) Two original neighbouring contours, from which a new contour need to be derived





(b) Two contours are triangulated, but with flat artifacts, (c) A new contour is derived from (b)





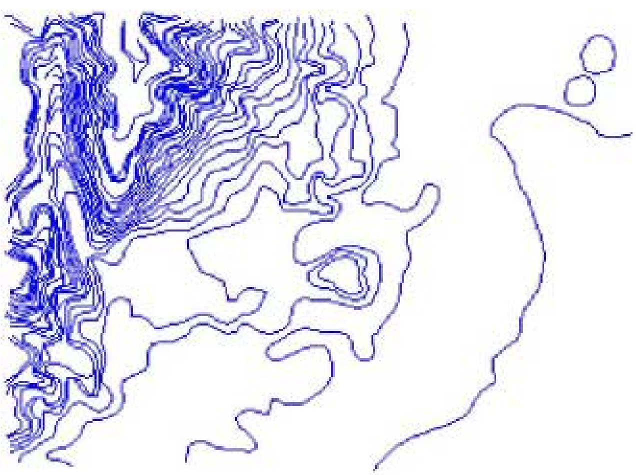
(d) Two contours are triangulated with feature points, (e) A new contour is derived from (d)

Figure 4 A triangulation-based technique for derivation of a new contour from two original neighbouring contours

7. Experimental evaluation

In other to test the performance of this system, a number of real contour maps have been used for testing. Figure 5(a) is a 1:10,000 scale topographic map, located in GuangZhou City of Southern China. In this example, a value of 0.6mm (at map scale) is used as the size of the raster SVO for the algorithms. The map is generalised to produce a resultant map as shown in Figure 5(b). It is clear that the result is very acceptable. However, on the other hand, if the same criterion is used for the popular algorithm (Douglas-Peucker algorithm), then the result will be something like Figure 5(c). It is clear that such a result is not acceptable.

(a) A contour map at 1:10,000 for experimental evaluation



(b) Result obtained from the system developed by the authors

(c) Result by Douglas-Peucker algorithm with intersection marked by circles

Figure 5 Experimental evaluation of new system developed by the authors

8. Conclusions

This paper describes the generalisation of contour lines using a line simplification algorithm, namely the Li-Openshaw algorithm. The theoretical background is given; the implementation of algorithm for generalisation of individual contour lines described; techniques for deriving a new contour from two original neighbouring contours discussed; and the main components of the system summarised.

From theoretical point of view, the technique described in this paper guarantees (a) no self-intersection and cross-intersection; (b) very smooth resultant contours; and (c) very coherent relationship between contour lines, with shape very faithful to the original contours (at a larger scale).

To test the performance of this system, real contour maps have been used. Experimental results show that the quality of the generalised contour is extremely good by visual inspection.

Acknowledgement

The work described in this paper was substantially supported by a grant from the Hong Kong Polytechnic University (Project No. A-P-091).

References

Attneave, F., 1954. Some informational aspects of visual perception. *Psychological Review*, 61(3): 183-193.

Bookstein, Fred.L., 1979. The Line-Skeleton .Computer Graphics and Image Processing.11,123-137.

Boutoura, C., 1989. Line generalisation using spectral techniques. *Cartographica*, 26(3/4): 33-48.

De Berg, M., van Kreveld, M. and Schirra, S., 1995. A new approach to subdivision simplification. *Proceedings of Auto-Carto* 12, 79-88.

Douglas, D. and Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digital line or its caricature. *The Canadian Cartographer*, 10(2): 112-122.

Li, Zhilin, 1988. An algorithm for compressing digital contour data. *The Cartographic Journal*, 25(2): 143-146.

Li, Zhilin, 1993. Some observations on the issue of line generalisation. *The Cartographic Journal*, 30(1): 68-71.

Li, Zhilin, 1995. An examination of algorithms for the detection of critical points on digital cartographic lines. . *The Cartographic Journal*, 32(2): 121-125.

Li, Zhilin, and Openshaw, S., 1992. Algorithms for automated line generalisation based on a natural principle of objective generalisation. *International Journal of Geographic Information Systems*, 6(5): 373-389.

Li, Zhilin, and Openshaw, S., 1992. A comparative study of the performance of manual and automated generalisations of line features. *AGI 1992-1993 Year Book*, 501-513.

Li, Zhilin and Openshaw, S., 1993. A Natural Principle for Objective Generalisation of Digital Map Data, *Cartography and Geographic Information System*, 20(1): 19-29.

Shapiro, B., Pisa, J. and Sklansky, J., 1981. Skeleton generation from x,y boundary sequences. *Computer Graphics and Image Processing*. 15,136-153.

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

Wang, Z.S. and Müller, J. C., 1992. Complex coastline generalisation. *Cartography and Geographic Information Systems*, 20(2): 96-106.

Weibel, R., 1987. An adaptive methodology for automated relief generalisation. *Auto Carto 8*, 42-49.

Weibel, R., 1996. A typology of constraints to line simplification. *Proceedings of 7th International Symposium on Spatial Data Handling (SDH'96)*, 9A.1-9A.14.

Weibel, R. 1997. Generalization of spatial data - Principles and selected algorithms. In: M. Kreveld, J. Nievergelt, T. Roos and P. Widmayer (eds.). *Algorithmic Foundations of Geographic Information Systems*. Springer.

Wu, Hehai, 1997. Structured approach to implementing automated cartographic generalisation. *Proceedings of ICC'97*, 1: 349-356.

Session / Séance 43-B

Integrating Vector and Raster-Based Techniques for the Generalization of Categorical Data

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. Comprehensive methodologies for the generalization of categorical data in both the vector and raster model are urgently needed. Initially, generic cartographic constraints governing the generalization of categorical data are specified. In the next step, these constraints are 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 possibilities 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. Categorical maps are commonly modeled as either vector data (i.e., as polygonal maps or polygonal subdivisions) or as raster data. 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. More sophisticated methods of preliminary nature have been proposed in the research literature for both raster data [Schylberg, 1993; Su et al., 1997; Jaakkola, 1998] and vector data [Wang and Muller, 1993; de Berg et al., 1998]. 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 in sections 2 to 4. The first element concerns the specification of so-called generalization constraints, that is, conditions that 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. Section 4 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 presents conclusions and directions for future research. An extended version of this paper that includes more examples and figures is available in PDF format from http://www.geo.unizh.ch/~beatp/publications.html.

2. Constraints to Categorical Data

2.1 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. Constraints originate from specific *map controls* applicable to a generalization problem. They are usually specified as something to *maintain* or to *avoid*. 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. *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. In general, constraints are not bound to a specific data model.

2.2 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].

2.3 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.

In principle, constraints express cartographic design specifications and are not associated with a specific data model. 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.

2.4 Classification of Constraints

For the classification of constraints the scheme of Weibel and Dutton is used [Weibel and Dutton, 1998]. 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:

- graphical (e.g., size, width, perceptibility)
- topological (e.g., connectivity, adjacency, containment, intersection)
- structural (e.g., shape, pattern, alignment, distribution)
- Gestalt (e.g., global characteristics, visual balance)

3. Translating Constraints to Measures and Generalization Algorithms

3.1 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.

3.2 Classification of Measures

Measures can be classified according to the main characteristic they represent. 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

3.3 Translating Constraints to Measures

The translation process of constraints to formal measures is by the system designer at R&D 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 since the underlying concepts are quite different. 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 their properties comprehensively. Translating constraints to measures is therefore also a *selection process*. The goal is to make the main properties of a spatial entity available to formal mathematical descriptions. The degree to which this can be achieved has a major influence on the results of the generalization process. Properties that have not been formalized cannot be dealt with by generalization algorithms 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.

3.4 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.

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 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 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.

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 have evaluated several methods for this operator [Bader and Weibel, 1997]. They propose a solution based on the computation of a *skeleton* for the polygon to be eliminated (see figure 1). The area of the polygon is distributed equally among its neighbors and introduction of topological error is avoided.

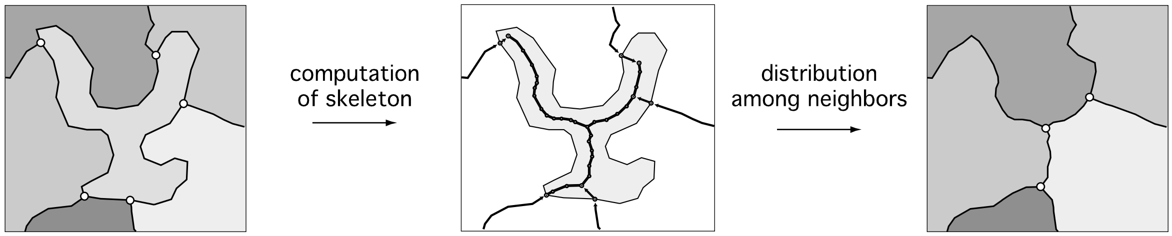


Figure 1. Elimination of a polygon using a skeleton algorithm [Bader and Weibel, 1997]

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 also extended version of the paper). Methods for raster data require less computational 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 patch 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]. This cell can serve as a basis for the actual amalgamation process as well as other for operators such as *displacement* (see also extended version) [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 [Bader, 1997] and Jones et al. [Jones et al., 1995] who use constrained Delaunay triangulations not only for the amalgamation operator (see figure 2) but also for polygon exaggeration and collapse.

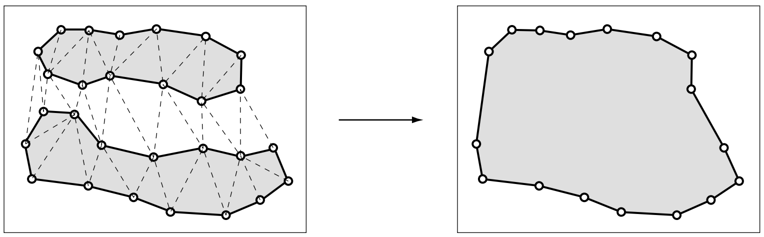


Figure 2. Amalgamation of polygons based on a Delaunay triangulation after [Jones et al., 1995]

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. Various algorithms for the amalgamation operator have been implemented for landuse/landcover raster data [Schylberg, 1993; Jaakkola, 1998; Peter, 1997]. Schylberg uses a simple *grow-and-shrink* algorithm (see also extended version). Objects overlapping or touching in grown state remain connected after re-shrinking by the same amount of cells [Schylberg, 1993]. Although very simple to implement, this method might not be adequate in situations where spatial variability is high. Peter has adapted and modified a method by proposed Brown et al. using cost-distances [Peter, 1997; Brown et al., 1996]. 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 in between provided the respective cells were given low costs (figure 3). 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).

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 depends to a large degree on the spatial resolution of the dataset.

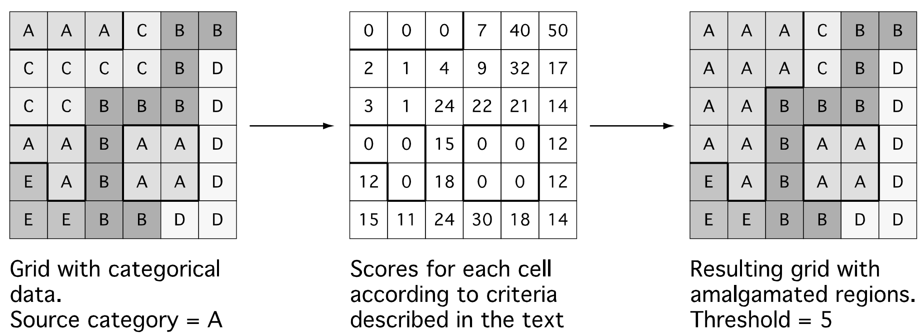


Figure 3. Amalgamation of regions based on cost-distance methods

4. Integration of Vector and Raster-Based Methods

4.1 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 Su et al., 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) [Su et al., 1998]. 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 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. have implemented several conversion algorithms and have evaluated them based on qualitative, quantitative and efficiency criteria [Piwowar et al., 1990]. 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 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 (raster) 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.

4.2 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 (see also extended version). This method can result in a considerable increase of the amount of data and stepped lines may still be visible. 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 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 [Peter, 1997]. A simple yet effective algorithm has been developed by Herzog et al. [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 4). 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.

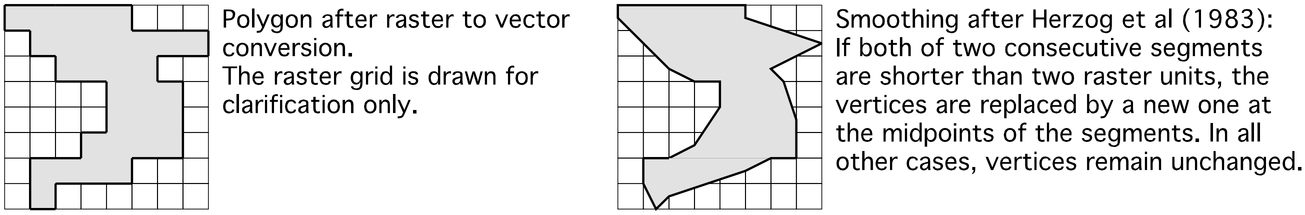


Figure 4. Smoothing of transformed raster regions in vector mode after [Herzog et al., 1983]

4.3 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. The newly integrated data provides valuable structural information that can easily be integrated in the generalization process. Rasterized linear features can, for instance, be given high costs if cost-distance methods (see section 3) are used to control the amalgamation operator while rasterized point features, or their containment information respectively, can be used to enforce topological constraints (see also extended version).

4.4 Local Data Model 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 5. 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.

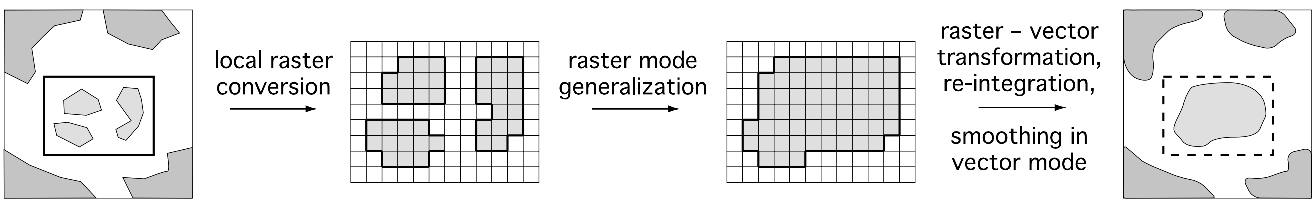


Figure 5. Generalization based on bi-directional local data conversion

5. 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. 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 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, 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.

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. *Doc. of Vers 2.0*. Oregon State Uni.

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.

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

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.

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., Lodwick, G. (1998): Morpholigical Models for the Collapse of Area Features in Digital Map Generalization. GeoInformatica, 2(4):359-382.

Wang, Z. and Muller, J.C. (1993): Complex Coastline Generalization. *Cartography and Geographic Information Sys.*, 20(2):96-106.

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.

Session / Séance 39-A

Raster-based generalisation of polygon data, with special reference to coastlines

Allan Brown

Division of Geoinformatics, Cartography and Visualization
International Institute for Aerospace Survey and Earth Sciences (ITC)
P.O. Box 6, 7500AA Enschede, The Netherlands
E-mail: brown@itc.nl

Abstract

Lines dominate most maps. A great deal of attention has been paid to the problem of automatically generalising these lines for smaller scale representation. A common type of line forms the boundaries between classified polygons e.g. land and water. For generalisation purposes these lines may simply be treated as lines or else the characteristics of the polygons themselves may be considered. The alternative raster approach treats an area as a collection of unit grid cells. The end result of a raster-based generalisation procedure can be kept in raster form, often at a coarser resolution, or it can be converted to line boundaries for further processing in a vector-based GIS, for example. The paper demonstrates that acceptable generalisations of area polygons, in this case a coastline with many islands, can be accomplished using processing techniques available in a typical raster-based GIS. A new algorithm, the 'moving aggregator' is also introduced. The products of the raster generalisations are vectorised and compared with manual generalisations and with the results of using line generalisation algorithms available in a commercial vector-based GIS.

Introduction

Many maps and much spatial data used in a Geographic Information System (GIS) consist of polygons. When a graphic overview of a large area is required, these polygons have to be generalised to retain visibility and the overall pattern. If this is the only aim, we can speak of cartographic generalisation. However, for the purposes of analysis procedures in a GIS we may want to generalise to a coarser or higher level of detail in order to reduce data quantity and file size, and to study broader patterns. This aspect of generalisation has been called model or model-oriented generalisation [Müller et al., 1995].

Both aspects of generalisation of digital spatial data have been the subject of a considerable amount of research, recently summarised by Weibel and Dutton [1999]. They state that the main aim of cartographic generalisation is to optimise visual communication, with a subsidiary aim being to derive data and maps for a variety of purposes. The main aims of model generalisation are to use resources economically and to increase/ensure data robustness. The study described below is mainly concerned with cartographic generalisation, although the results can be used, with care and taking into account the errors deliberately induced by the generalisation process, for further processing in a GIS. The dataset used in the study is coastline data for the complex coastline of southern Chile.

Within the aims outlined above and for the given data, there remain three basic approaches to the problem of generalising (land) polygons: treat the coastlines simply as lines, consider also the geometry and topology of the polygons themselves or adopt a raster (grid-cell) approach. Research to date has concentrated on the first two approaches, although with increasing computer power and storage capacity the raster approach is fast becoming more technically and economically feasible. The image processing community in particular is very familiar with raster data and with producing grid-based spatial data, and a well-developed digital image processing or grid-based GIS software package contains a wide variety of operators for handling gridded data. The typical users of these packages will want to view their data at small scales to get a broad overview, but they may not want to spend much time and effort developing intricate programs to perform the necessary generalisation. They would prefer just to use the operators already available, even if the resulting graphic image is not what a cartographer would call 'perfect'. This study attempts to show that these standard operators can indeed produce acceptable generalisations of polygon data, even for a complex coast, provided they are used with care. One operator, called here the 'moving aggregator' operator, is used as a generalisation operator, even though it may not be available as a standard in all packages. It is, however, straightforward to program.

Line and polygon generalisation

A very common operator used to generalise polylines is based on the Ramer-Douglas-Peucker (RDP) algorithm [Dutton, 1999]. This algorithm, however, is simply intended to reduce the number of points needed to describe a line. It does not produce good graphic results when it is used to *generalise* a line, particularly for large reductions. Wang and Müller [1993] have combined the use of the RDP algorithm with an approach based on finding a structural hierarchy based on geographical characteristic and meaning. Weibel and Dutton [1999] outline various other approaches which have been attempted for line generalisation.

A noteworthy development, since it is directly available to many GIS users, is the introduction of the BENDSIMPLIFY option within the ARC/INFO GENERALIZE command, in addition to POINTREMOVE (the RDP algorithm). The basic idea of this iterative method is to analyse a line on the basis of local shapes (bends) and determine how to treat them (delete, exaggerate, etc.) according to the characteristics of the bends and cartographic rules [Dan Lee, personal communication]. These two operators were applied to the digital coastline data.

Figure 1a shows ESRI Country 98 data of southern South America, plotted on the UTM projection, zone 18, at the scale 1:10 000 000. The coastline is quite strongly generalised and is presumably intended to be viewed at a fairly small scale. Figure 1b is of the same area at the same scale and projection, but using the Vector Map Level 0 data supplied by the National Imagery and Mapping Agency. These data were collected mainly from maps at

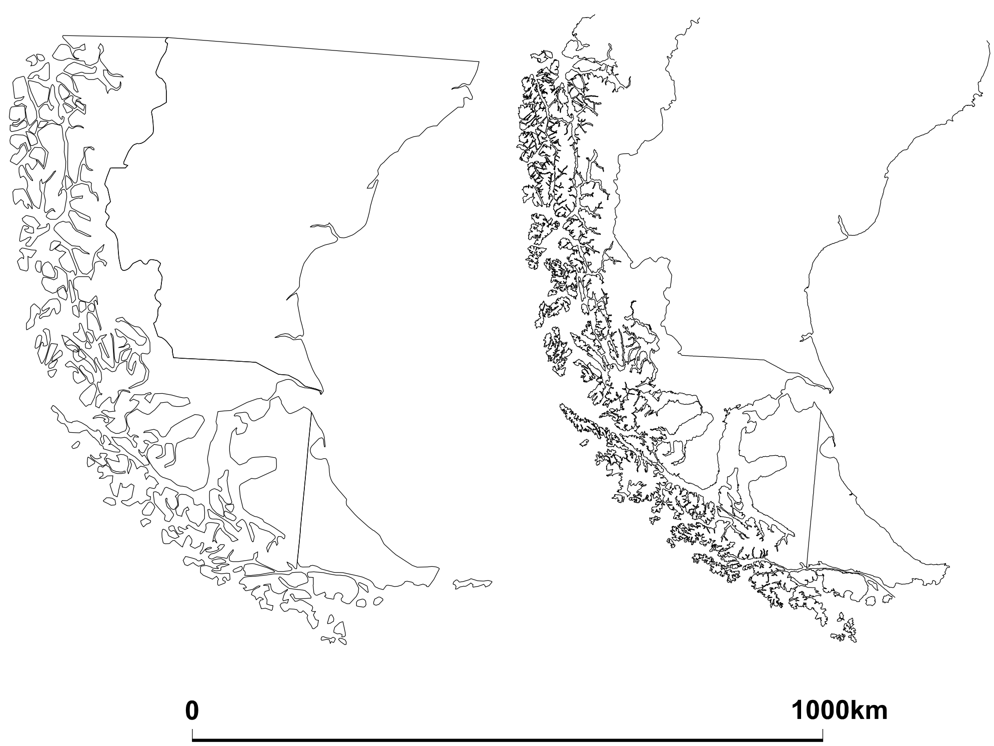


Figure 1. Two digital datasets at 1:10 000 000 scale. Left (a) ESRI Country 98, right (b) NIMA Vector Map Level 0

1:1 000 000 scale, so the 10 times reduction produces a too highly detailed result. Some generalisation is necessary to improve legibility. Generalisation of the NIMA data was carried out using the POINTREMOVE and BENDSIMPLIFY options, within the GENERALIZE command in ARC/INFO. In both cases the 'weed tolerance' was set at 5000 metres. The results were plotted at 1:10 000 000 scale for comparison (Figure 2).

The disadvantages of the RDP algorithm used in POINTREMOVE as a generalisation operator are obvious, when the weed tolerance chosen is relatively large. The coastline is reduced to a series of straight lines, lines sometimes cross and small islands are reduced to single lines. The BENDSIMPLIFY algorithm produces a better, more 'natural' result, but there is still too much detail for the scale. In both cases, of course, a simpler result would be obtained if small islands were to be deleted, but this also has disadvantages, since a cluster of small islands would not be represented at all. To deal with this kind of situation, it may be better to treat the data as polygons rather than as lines.

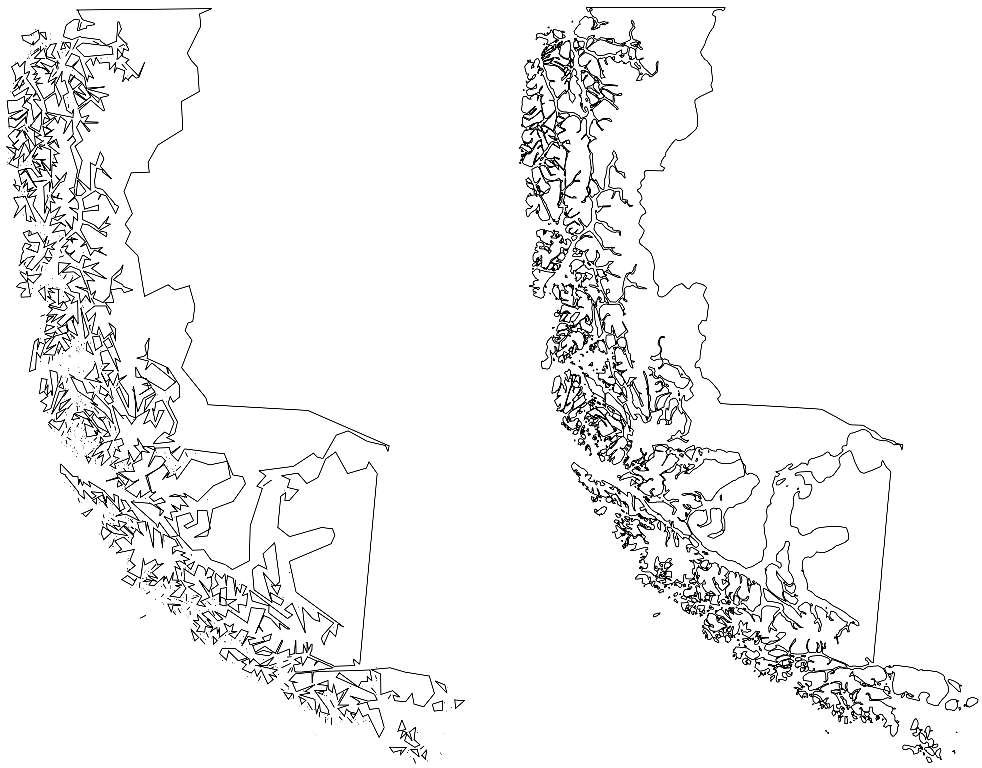


Figure 2. NIMA data for southern Chile generalised in ARC/INFO and plotted at 1:10 000 000 scale. Left using POINTREMOVE, right using BENDSIMPLIFY, with 5000m weed tolerance in both cases.

Several researchers have suggested methods for generalising polygon-type data. Probably the best known commercially available application of some of these techniques is Intergraph's MGE Map Generalizer, an interactive system involving the methods described by Dan Lee (1995). Weibel and Dutton (1999) point out that in most commercial GIS the generalisation functionality for polygon data is limited to polygon aggregation (merging) and simplification of their boundaries. Polygons may be eliminated if below a certain threshold area and merged with a neighbour according to some rules. Aggregation may be based on a hierarchy of attributes, as for example for a soil map. In the particular case of the land/water situation, it is of course not possible to make use of a classification hierarchy to merge or eliminate polygons: the only basis for generalisation is geometry.

Given that the techniques and approaches described in the literature for polygon generalisation are in general not yet implemented in commercial GIS, it may be worth while to consider an alternative, namely the use of raster operators available in (often inexpensive) commercial grid-based GIS packages.

Raster-based generalisation

The raster-based generalisation techniques have received less attention than the vector-based techniques. Yet for area data they would seem to be intrinsically suitable, since the basic raster unit, the grid cell, is itself an area. McMaster and Monmonier (1989) review the concepts involved for both quantitative and qualitative raster data. Several researchers (e.g. Bo Su et al., 1997) have investigated methods suitable to the binary situation typified by the land/water dichotomy. Let us, however, consider here only those operators generally already available in a raster-based GIS, on the principle that most users would prefer to use these rather than to program their own, more complex algorithms, provided they can achieve results acceptable to them.

The first problem is always to fix the size of the basic unit grid cell. If it is too large it may introduce an unacceptable level of generalisation into the initial data, if it is too small the data volumes become very large and processing times very long. The ideal original grid size is obviously related to the level of detail, which in turn is related to the ideal display scale of the data. For the example of the NIMA data for the southern coast of Chile, an initial 500m grid was selected. This will retain all except the smallest islands and narrowest inlets. At the data source scale of 1:1 000 000, this distance is represented by 0.5mm, acceptable considering that the main purpose is to produce smaller scale results.

Single operators

One of the most used 'generalisation' algorithms in a raster-based GIS involves producing a new, coarser grid which is a multiple of the original grid. The term 'aggregation' will be used here for the procedure, although this term also has other uses. Figure 3 shows the results of two aggregations of the original data, based on a 2500m and a 5000m grid respectively, on the basis that the new, larger grid cells take the predominant class of the smaller component cells. Visual inspection shows that the 5 times aggregation is

Figure 3. Two aggregations of the original 500m raster data using a 5 times (left) and a 10 times (right) aggregation. Plotted scale 1:10 000 000.

Figure 4. Majority filters applied to the original 500m raster data using a 5x5(left) and a 9x9 (right) filter.

still rather detailed, whereas the 10 times aggregation produces a 'blocky' appearance.

Another common technique involves the use of a square majority filter in which the central pixel of the filter, which can be of any size provided it is an odd number, takes the class of the majority of the component cells. This smoothing technique does not change the grid size. In figure 4 the results of a 5x5 and 9x9 filter are compared. The result of the 9x9 filter seems to be acceptable when plotted at 1:10 000 000 scale. There are some effects which a cartographer would consider undesirable, however, such as closing of narrow sea inlets. The grid cells involved could be manually edited to correct these errors.

Since the situation is a binary one, the 'shrink and expand' technique can be used. In this case a 3x3 shrink filter is applied three times to the land grid cells, followed by three times application of a 3x3 expand filter. The procedure is repeated for the sea grid cells. This technique is another of the ones in common use, but figure 5 shows that the result is biased depending on which of the binary classes is made to shrink and grow.

The moving aggregator algorithm

The grid aggregation algorithm can be used to count the number of cells of each class which fall in each cell of the coarser grid. This is illustrated in figure 6 on the basis of a small extract from the Chile data. Figure 6b shows the result of counting the land pixels for a 5 times aggregation. The result is a number between 0 and 25, represented as steps of a grey scale in the figure. This kind of visual presentation can serve as a useful reminder of the errors introduced by grid aggregation.

Figure 5. Shrink and expand filters applied three times each to the original 500m raster data using the land as basis (left) and the sea as basis (right).

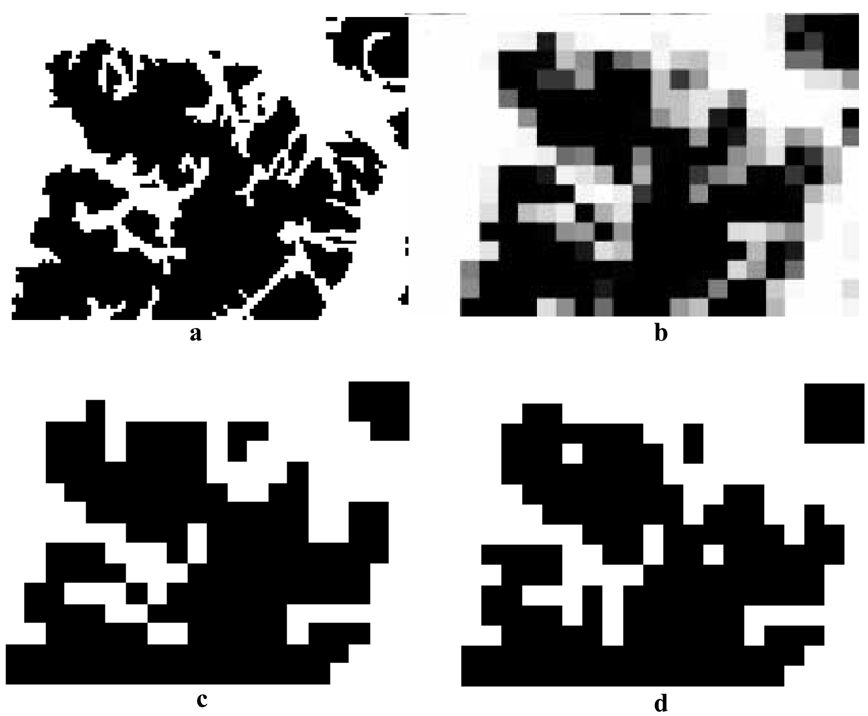


Figure 6. a) The original raster data plotted at 1:1 000 000 scale. b) 5 times aggregation with counting, to give values between 0 and 25. c) 5 times aggregation, taking the predominant class. d) As for c), but offset by one original grid cell in the x-direction.

Another way to illustrate this is to offset the coarser grid in relation to the original grid. In figures 6c and 6d the results of offsetting the (predominant) 5x5 grid aggregation by 500m, i.e. one original grid cell, are compared. 25 versions are possible in total. Presenting these in an animated sequence is one way of bringing aggregation errors to the attention of the user.

The offsetting of the coarse grid can be used as a generalisation technique in what may be called the 'moving aggregation' algorithm. In the case of a 5 times aggregation of the Chile data, all 25 possibilities were produced. For each *original* cell a count was made of how often it is included in an aggregated cell. This resulted in a number from 0 to 25. The result can be presented graphically (see Figure 7a). The technique automatically eliminates all islands and lakes which occupy fewer than 25 original grid cells. It also eliminates all very narrow areas of land or water. Recreating the generalised binary image is simply a matter of choosing a threshold value. In the case of Figure 7b, the threshold chosen was >13. This results in a slight deliberate bias in favour of the sea areas, so tending to 'open up' very narrow inlets.

Combination of methods

We have seen above that grid aggregation introduces errors. However, the result is a much smaller data set, which is an advantage for storage and processing. Furthermore, a small amount of aggregation, say 2 or 3 times, may produce negligible visible errors when plotted at a sufficiently small scale. Also, aggregation may be necessary in raster processing of different data sets to bring them all to the same grid size.

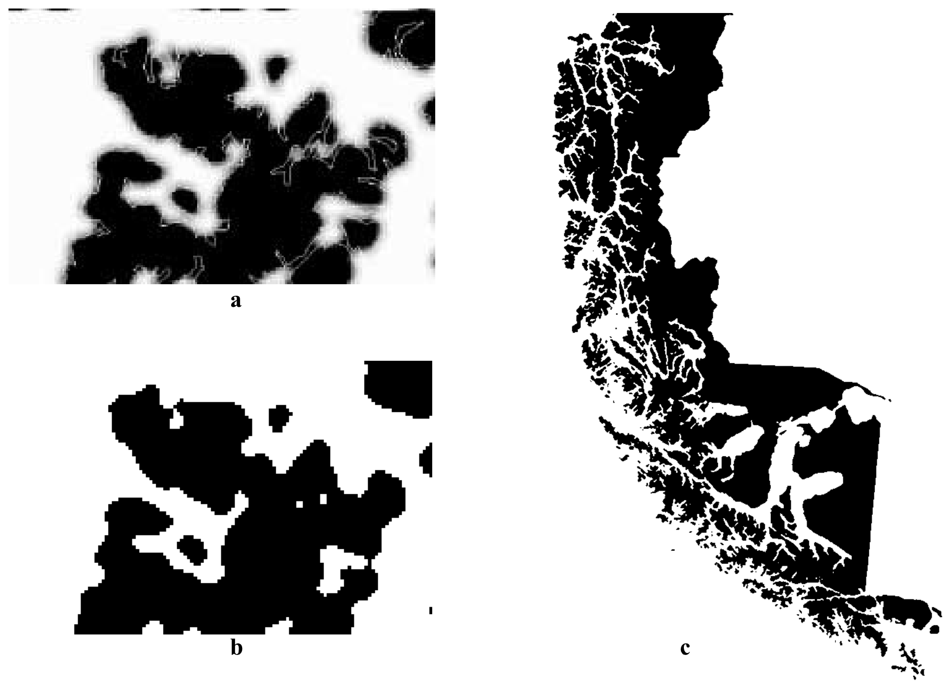


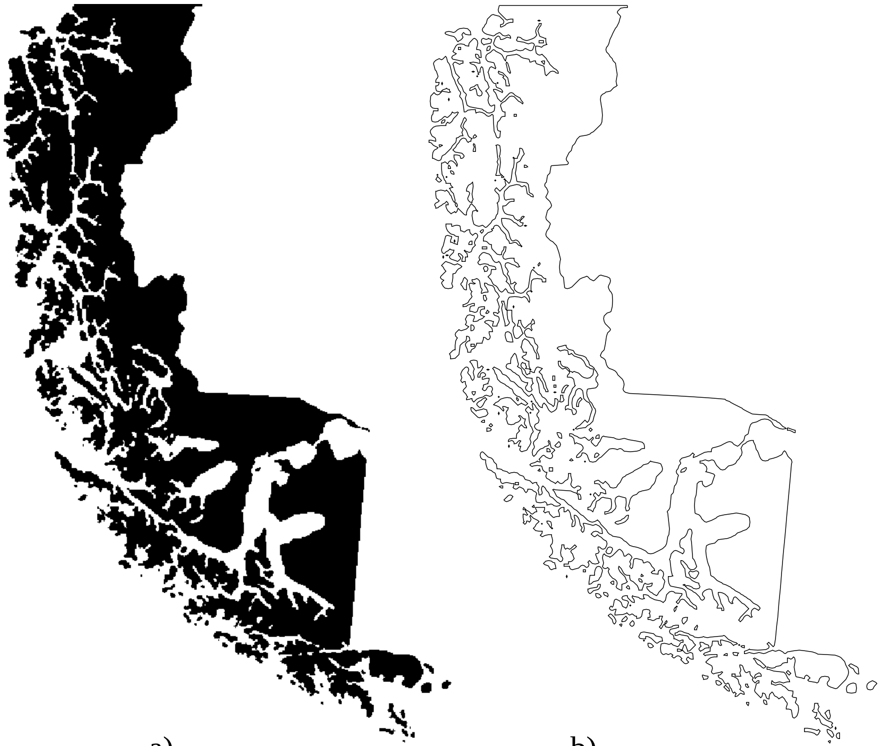
Figure 7. a) First counting result of a 5x5 moving aggregator, original coastline in white. b) Conversion to binary, using threshold >13. c) Result for the complete test dataset, plotted at 1:10 000 000 scale.

Aggregation in combination with other techniques gives acceptable results. The result of the moving aggregator plotted in Figure 7c can simply be aggregated 3 times, followed by the application of a 3x3 majority filter (see Figure 8a). If a vector result is required, the end product can be vectorised, with smoothing to avoid jagged lines (see Figure 8b). Combinations of smoothing and aggregation can be continued in sequence to give a highly generalised result suitable for plotting at small scales. Best results are obtained if the process begins and ends with the application of the majority (smoothing) filter (see Figure 9).

Discussion

All the raster-based generalisation procedures presented above are batch processes based on the use of simple, widely available operators, although the moving aggregator operator may require special programming in some GIS. Some of the operators and their combinations produce results which are cartographically reasonably acceptable, certainly during the data exploration and even the analysis stages of working with raster data. However, for final display purposes, some manual editing would normally be required, either of the raster or the vector result. A large majority filter produces quite good results for the binary Chile data. However, the original file size remains unchanged, therefore many users might prefer to create smaller files by using the majority filter and aggregation in sequence. In this case, the filter size and aggregation multiple are kept small. As a generalisation operator *per se*, the moving aggregator does not appear to produce better results than a large size majority filter. However, it does offer more flexibility in allowing the operator to choose threshold values depending on the use of the final generalisation. It also has uses for graphically displaying aggregation errors, for example by an animated sequence or by creating a boundary uncertainty zone. Further investigations are under way by the author to study the applicability of the technique to multi-category data such as land use.

In the investigation done for this paper, some attempt was made to measure quality, for example by finding the percentage of land cells in the Chile generalisations. For reasons of space, the detailed results are not included. Furthermore, the simple methods presented here need to be compared to more complex raster-based techniques, for example as used by Jaakkola (1997), and with the more promising of the vector-based polygon generalisation methods.



a) b)

Figure 8. a) The data of Figure 7c, aggregated 3 times followed by the application of a 3x3 majority filter. b) Vectorised version.

a) b) c) d)

Figure 9. The result of application of a 3x3 majority filter (M) and 3 times aggregation (A) in the sequence M-A-M-A-M, starting with the original Chile data. The final grid size is 4500m. The raster and vectorised results are plotted at 1:20 000 000 scale (a and b) and at 1 50 000 000 scale (c and d).

References

Bo Su, Zhilin Li, 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.

Dan Lee, (1995). Experiment on formalizing the generalization process. In J.C. Müller, J.P. Lagrange and R. Weibel (Eds.). GIS and Generalization, Methodology and Practice. Taylor and Francis, London.

Dutton, G. (1999). Scale, Sinuosity, and Point Selection in Digital Line Generalization. Cartography and Geographic Information Science, 26(1), 33-53.

Jaakkola, O. (1997). Multi-scale land cover databases by automatic generalisation. Proceedings of the 18th ICA/ICA International Cartographic Conference, vol. 2, 709-716.

McMaster, R.B., and Monmonier, M. (1989). A conceptual framework for quantitative and qualitative raster-mode generalization. Proceedings of GIS/LIS '89, ACSM/ASPRS, 390-403.

Müller, J.C., Weibel, R., Lagrange, J.P., and Salgé, F. (1995). Generalization: state of the art and issues. In J.C. Müller, J.P. Lagrange and R. Weibel (Eds.). GIS and Generalization, Methodology and Practice. Taylor and Francis, London.

Wang, Ze-Sheng, and Müller, J.C. (1993). Complex Coastline Generalization. Cartography and Geographic Information Systems, 20(2), 96-106.

Weibel, R., and Dutton, G. (1999). Generalising spatial data and dealing with multiple representations. In P.A. Langley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind (Eds.). Geographical information systems, Vol. 1 Principles and technical issues (2nd edtn.). Wiley and Sons, New York.

Session / Séance 05-C

Computational Methods for the Automated Symbolization of 1:24,000 and 1:100,000 USGS SDTS Data

Francisco Brazile

Department of Geography,
University of Zürich–Irchel,
Winterthurerstrasse 190,
8057 Zürich, Switzerland.
Fax: +41 +1 635 6848
fbrazile@geo.unizh.ch

Abstract

The U. S. Geological Survey currently releases digital data sets to the public in the Spatial Data Transfer Standard, digital line graph (DLG-3) format free of charge. This wealth of data initially seems well suited for certain branches of digital cartographic research, such as automated generalization, due to the varying nature of the geography covered in the U. S. and the depth of coding detail in the schema. Unfortunately, instructions or mechanisms to accurately symbolize all of the data as originally published on hardcopy maps or even to the screen do not seem to be widely available. Seeking to fill that gap, this paper provides selected algorithms, examples, and procedural insight into automatically recreating the separately published topographic paper map from the digital data base. Problems similar to other on-screen visualization projects such as multimedia atlases and cartography are also noted.

Introduction

Cartographic generalization research is dependent on the ability to measure, plan for, and allocate the exact space requirements of symbolized features. The critical ability to generate an authentic and well-defined representation is then needed to design experiments and analyze results in generalization. To meet these research needs, methods, pitfalls, and shortcuts for rendering some of the original United States Geological Survey (USGS) topographic map specifications are given. Through limited reverse engineering, consulting published guides and USGS cartographic application designers, and also indirect means, such as scanning and raster comparison, specifications were created which both yield good first results and highlight the complexity of symbolizing and rendering an image completely faithful to the specifications.

Methodology 1 — ARC/INFO

The goal of this exercise was to retrace the steps needed to re-create a digital representation of a published USGS 1:24,000topographic (topo) map and the same area at the progressively smaller scales of 1:50,000 and 1:100,000. In doing so, it was hoped that a higher level of familiarity with the exact nature of graphic problems for a known geographic location and modeled in a typical, predictable data encoding could be gained by the author. For this reason, the Austin East, Texas, 1:24,000 topo sheet was selected. The corresponding area is modeled by the USGS in a series of digital files for each theme, e.g. hydrography, boundaries, man-made structures, contours, roads, railroads, survey points, and vegetation, in the SDTS DLG-3 format, a vector-based format. These files were found available free of charge on the Internet (http://www.usgs.gov/).

It was hoped that the Laser-Scan Lamps2 system used for testing and research at the University of Zurich Geography lab would be able to read the raw binary data format of the SDTS DLG-3 files. Unfortunately, at this time, that specific format is not supported and so it was decided instead to use ARC/INFO to read and symbolize the data. Lamps2 would have made an interesting platform to test multi-scale symbolisation issues due both to the sophisticated multiple-scale representation support and the nature of data model storage, which is in an object-oriented format. However, ARC/INFO not only had a capability to import the SDTS DLG-3 format, but a script was provided by the USGS (sdts2cov.aml, http://mcmcweb.er.usgs.gov/sdts) specifically in AML, the ARC/INFO macro language, to facilitate such an import. The script helped to perform data tables joins with geographic features, which can be a complicated step in the non-SQL database, INFO.

Clarke explains the general SDTS terminology: the real-world phenomena to be shown are called cartographic entities; the digital representation of the entity is a cartographic object and the cartographic object is depicted or symbolized as a cartographic element on the map rendering [Clarke, 1995].

After the Austin data set was imported, a quick visualization on-screen was performed with single-pixel line types. This imaging indicated the presence of many geographic feature types (cartographic entities), though directly usable mechanisms for an accurate symbolization were not present in either the SDTS data or ARC/INFO. A more close visual inspection of the success and completeness of the data set after the import had to wait until a visual feature by feature comparison between the ARC coverages and the printed map could be performed.

There was no facility in the ARC/INFO application to indicate which unique entity types were present in the nine themes of the Austin data set. To determine the answer, a database command was issued to unload all records in the field, ENTITY_LABEL, to a file for both line and polygon coverages for each coverage. First, a Perl script was written to create a tab-separated file of key-value pairs for the DLG ENTITY_LABEL codes and their corresponding text descriptions, e.g. “1700201” - “Primary Route, class 1, symbol undivided” from the DLG User’s Guide (http://www.usgs.gov). Then, a small, FreeBSD shell script was used to sort all records, find the unique codes, and match the code with the Perl-generated file of DLG keys. For example, the Austin East data set contained the following entity types relating to roads (see Table 1).

The next task was to group the entities by symbol type. Even though each symbol used on USGS topo maps is indexed numerically in the technical instructions [Department of Interior – USGS, 1995], there seemed to be no way to identify exactly which symbol type belonged to which DLG-3 entity type, though good guesses could be made. The symbol numbering scheme predates digital production (a symbol type to entity type key was not found until after this exercise was performed). The only way to be sure of symbology type at the time seemed to be to visualize on the screen, each feature of a unique entity type, and visually match that location on the printed map, noting the symbology used on the printed map. In this tedious manner, each symbology type was identified for the entity types that existed in the test data set.

Next approximately 750 lines of AML code (not including comments) was written to select the entities that should be grouped for symbolization and have ARC render them to a postscript image at scale in a full page extent. The line symbol editor in ARC/INFO allowed the pre-associating of the appropriate line width, dashing pattern, and casing pattern (i.e. a centerline symbolised as parallel lines) to line symbol variables, which were then used, along with color specifications, to draw the lines or area fills. The correct line widths were found in the technical instructions [USGS Publications, 1995]. The color used on the printed map was taken by searching for the red green blue (RGB)-value with image processing software on the corresponding Austin East, TX, Digital Raster Graphic (DRG) for the color that seemed to match closest the printed map. A DRG is a USGS product that is essentially a scanned topo sheet in a raster GeoTIFF format. Then, experimentation revealed an appropriate stacking order: drawing in order where subsequent features are drawn on top of previous ones, urban area tints, wooded area tints, water areas, contour lines, water lines, lower class roads, higher class roads, railroads, building areas, airports, power stations, and substation power lines.

Table 1. Unique road entities found in the Austin East, 1:24,000 SDTS DLG-3 data set after importation to ARC/INFO.

ENTITY_LABEL	*Text description*
1700005	Cul-de-sac
1700201	Primary Route, class 1, symbol undivided
1700202	Primary Route, class 1, symbol divided by
1700203	Primary Route, class 1, divided, lanes separated
1700204	Primary Route, class 1, one-way, other than
1700205	Secondary Route, class 2, symbol undivided
1700206	Secondary Route, class 2, symbol divided by
1700209	Road or street, class 3
1700210	Road or street, class 4
1700211	Trail, class 5, other than four-wheel drive
1700213	Footbridge
1700215	Perimeter of parking area
1700217	Road or street, class 3, symbol divided by
1700222	Road in transition
1700402	Cloverleaf or interchange
1700405	Nonstandard section of road

Results 1

Upon creating the postscript image (not shown), it was clear some of the entity labels did not map one-to-one to a specific symbology type and it was cartographically unacceptable to force one symbology for these entities. For instance, entity type, “1700222 - a road in transition”, was symbolized on the published map as a Class 1, with a highway type symbol, elsewhere as a Class 2, boulevard type symbol, and sometimes as a dashed trail type symbol. The ARC/INFO AML written for this exercise never succeeded in characterizing those segments correctly in all cases. A consideration was made on computing the most predominant type the transitional segment represented, via some kind of interpolation, but that work was not executed. Without that ability, the only correct way to symbolize these features seems to be to mark them explicitly as a special case and symbolize each individually.

Another problem of omission was noted in that point features representing house clusters in newly developed neighborhoods were absent from the ARC/INFO data coverages. These features are prominent on the published map and would pose a unique challenge when generalizing those areas to a smaller scale. The author believes that the features are present in the original binary SDTS files, but that the USGS script used to manage the import process may omit them due to the nature of their geometry. Those buildings are thought to be line objects whose start and end points are the same and hence are weeded out during a clean or build process, but this has not been confirmed.

There were two much worse problems with the simplistic rendering with the first AML implementation, however. The first problem was the intersections of roads that were represented as double-line casings intersected in a manner that is unacceptable (Figure 1). (Examples of this situation have been found on maps of other scales and purposes, such as the Kümmerly Frey 1:23,000,000 Weltkarte. On the Kümmerly Frey 1:5,000,000 map of Europe, there even seems to be a possible inconsistency near Halmstad, Sweden; see Figure 1b).

The second problem was that certain entities represented with dashed symbols were not symbolized because the cartographic object, say a highway, was represented by disjoint line segments (Figure 2; problem appears also in Figure 3). Based on the simplified data model, the ARC symbolization engine was not able to space the dashed patterns consistently.

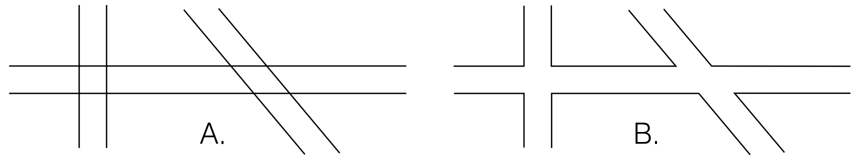


Figure 1a. Centerlines in the data base symbolized as parallel line casings intersect at road junctions, A. The desired impression for these casings is B.

There is a command in ARC that allows the rendering engine to symbolize a line with periodic variety correctly, LINEADJUSTMENT, and perhaps even if implemented by separate segments. This feature, however, did not respond in the manner intended, which could be the fault of the author. If the data to be symbolized were restructured via a network line, made up of multiple segments, it seems clear that the alternating dashed pattern would be correctly rendered.

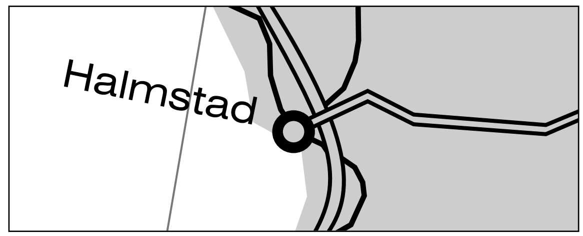


Figure 1b. A double-line casing road with no area fill intersects a higher priority double line casing highway with no area fill, redrawn from Map of Europe, 1:5,000,000, Copyright Kümmerly+Frey, Bern, Switzerland. Yet the lower class road is shown in higher priority figure ground. At this scale connectivity should be indicated as in Figure 1a, part B, regardless of road class, as was performed elsewhere in this Map of Europe. Digital systems would, in general, need the ability to handle this case.

Similarly, the road casing intersection problem could be solved with a restructuring of the data that allows a recursive traversal of the road network. The algorithm begins by selecting all roads to by symbolized and considering them at once. Starting at any road, find the perpendicular to the first road segment, and with a vector in the perpendicular direction with a magnitude of one half the desired symbolized casing width from the centerline, follow the road cluster along the outside cluster, drawing a casing line as the roads are traversed, until the perpendicular of the starting point in the negative vector direction is reached (the beginning). Then for those situations where interior partitions were created, a mechanism would be needed to visit every line, and on either side of the line, determine if a symbol was present. If a symbol was missing, traverse the interior partition, until all interiors were processed. This algorithm would also need a mechanism to determine when the symbolized casing line self-intersected during graph traversal, and those intersections eliminated, which would occur when two cartographic objects form a steep angle.

In essence, after considering this algorithm, it seems using the ARC BUFFER command with a parameter width, again, one-half the casing symbol width, would achieve the same objective. This buffering was performed and evaluated, but not integrated into the final postscript map image. It shows some promise, though it does add a complicated step in merely symbolizing the data.

It is believed by the author that an implementation in AML to symbolize the casings without using the buffer command or restructuring the data to add network lines made up of the underlying road segments, would be unduly complex. Therefore a decision was made to symbolize the data programmatically, in the Java programming language, so that complete control of the appearance could be obtained and measures of symbol extent could readily be computed.

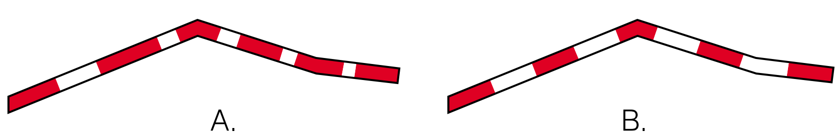


Figure 2. The dashing pattern required by some entities such as narrow highways was interrupted due the segmented nature of the underlying object. The desired pattern is shown, B.

Methodology 2 — Java

An implementation in Java of a mechanism to store the data by object oriented classes and symbolize the data to screen or postscript seemed like a good exercise for a few reasons. First, the DLG-3 format represents an older, legacy encoding and the USGS is proposing a new, experimental object oriented encoding, DLG-F. An object oriented implementation by the author would look more to the future of data dissemination by the USGS. Second, the Laser-Scan Lamps2 system stores features in an object oriented data base. Designing a class hierarchy in Java with the test data set would be valuable in determining how a data schema could be built in Lamps2 for the relational DLG-3 format and how the feature symbologies could be organized. Third, and unrelated to this symbolization exercise, is that programming a representation in Java would allow the author the ability to add and isolate only those computational methods needed to carry out on-going object primary partitioning and post-generalization evaluation work. This partitioning mechanism also relies on the ability to determine exact symbology requirements and measurements [Brazile, 1998; Brazile and Edwardes, 1999].

The implementation resulted in the creation of about 4000 lines of Java code. This number seems large if the aim were merely to draw the data to the screen. However, this size also represents a hope that the code can be extended to allow additional functionality for measures and analysis (note OpenMAP, mentioned below, seems to be about 35,000 lines of Java code). Though, the first aim was merely to display the data on screen, to scale. The first step to meet this aim was to determine how to represent the data model and how to import the data. The most simple procedure possible was chosen, namely the UNGENERATE format of ARC/INFO, which merely creates an ASCII file of records with IDs and X, Y coordinates. Using the same groupings as found early, the coverages were written to UNGENERATE files based on similar symbology types, e.g. all roads symbolised with red and the same line width were written to a one data file. A simple five state state-machine to parse this data file was written and then the main method read the UNGENERATE files and created table entries with the corresponding object feature classes.

This method of data transfer was quite primitive. All of the data attributes, e.g. elevation for the contours layer, were omitted. Narrow contours had to be written to a file separate from index contours, which are symbolized thicker. For this representation experiment, the loss of attributes was not a serious factor, though in some cases, with more attributes, the amount of programming code needed would be reduced. More sophisticated methods for data interoperability are currently the subject of research.

Cranston et al. have demonstrated an intelligent way to transfer DLG data to a common interface (as common objects) and back to a native format again using the OpenGIS paradigm [Cranston et al., 1999]. However, even with systems implementing this new interoperating paradigm, the needed ability to symbolize the data is missing from any interoperability specifications [Doyle et al., 1999]. Doyle et al. further note in their OpenGIS implementation, OpenMAP, that the graphical appearance didn't meet the requirements imposed on cadastral GIS applications in Germany, due to the omission of interchangeable portrayal specifications [Doyle et al., 1999]. Keller proposes a mechanism to address this deficiency called INTERLIS which would make it easier to exchange graphic views on the data [Keller and Thalmann, 1999], thereby logically completing the spirit of GIS data interoperability.

Rendering the feature classes to screen in Java was initially more complicated than desired. This was because at the time Java did not provide any symbol widths other than one pixel. To render lines to screen wider than one pixel, one was forced to redraw the line again, computing an offset each time. However, during this exercise, Sun Corporation, the sponsor of Java, had released a 2D programming interface (API) with a CAD/CAM audience in mind. This API allows more sophisticated graphics handling, such as flexible line widths, the ability to create dashing patterns, and the ability to apply affine transformations, such as scaling and rotation. These new features are welcome, though will replace and override much of the code and effort that had been undertaken by the author for normalized projection and scaling.

Results 2

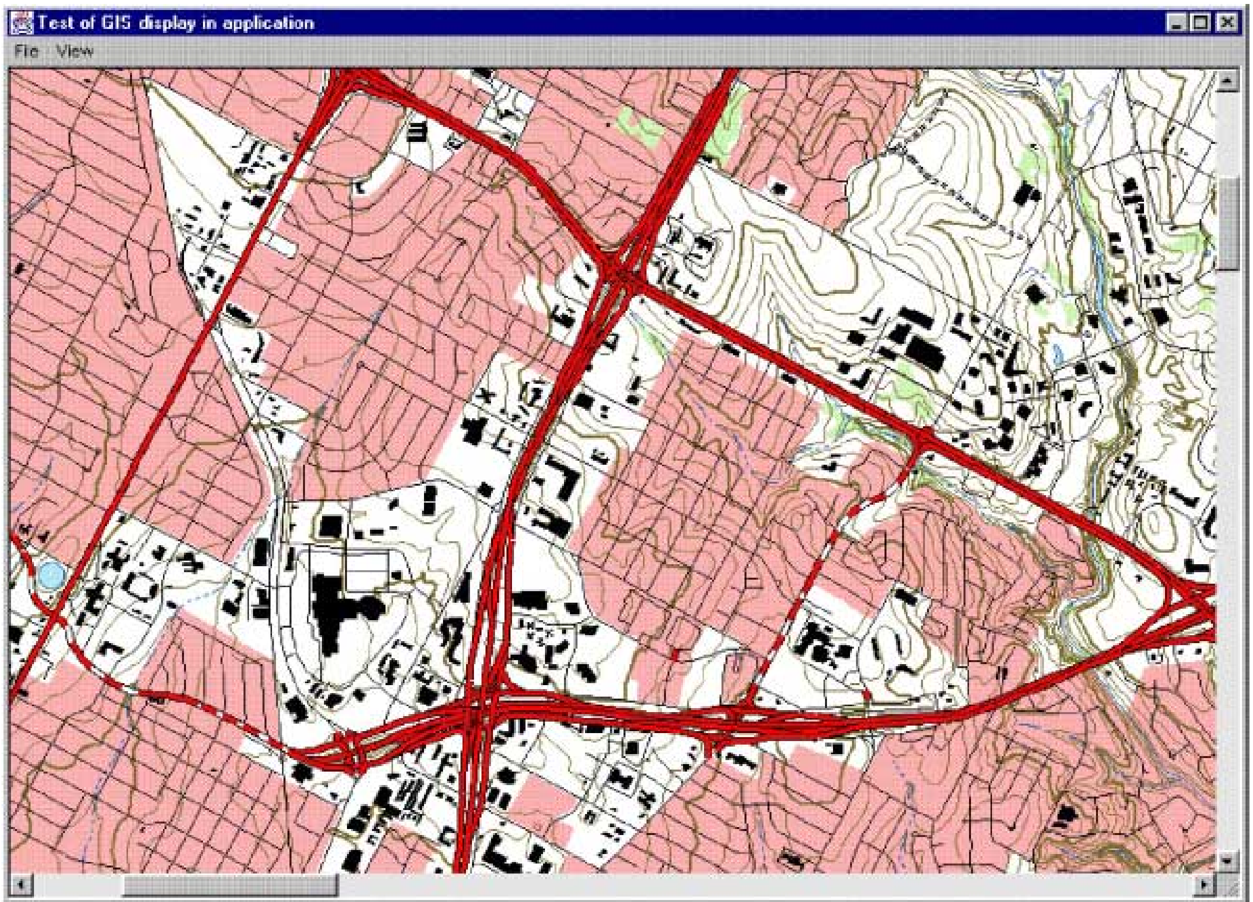


Figure 3. Java Application to symbolize the USGS SDTS files on-screen.

The representation to screen in Java closely approximated the moderately successful postscript image that had been generated in ARC/INFO and viewed with a postscript previewer (see Figure 3). Surprisingly, the performance of the display paint method was quite responsive, even though the author had not explicitly created an intelligent double buffering. Namely, the paint method merely iterated through each feature class in the features table, and again through each feature in that feature class. For each feature the feature's draw(Graphics2D g) method was invoked, where the graphics context g is the open graphics window in the user interface. The ability to encapsulate the draw methods in the feature class for any arbitrary feature class, whether they were roads, buildings, lines or areas was very powerful. For major highways, for instance, a black line with width five pixels wide could be drawn, and on top of that, a red line with width three pixels overlaid, giving the impression of a black road with a red fill.

Again, although this model for encapsulation and iteration over data implementing the draw method interface was powerful, it was not powerful enough to render the data completely faithfully when mixed with the way the data was stored, by road segment which had been earlier spilt for topological reasons. With that same black and red highway example, the following problem had arisen (see Figure 4).

In order to overcome this problem, it seems necessary, in this case, to draw all the black lines for the highways and then draw the red interior lines. This can be accomplished but no longer allows the simplistic iterating over all objects and calling their draw method. A class or engine would then need to be created which can draw each feature class in the table, based on multiple iterations over all features in that class. Likewise, it may not be a bad idea to have this engine contain a structure to make explicit the stacking order of feature classes.

Carrying the concept further and having a single class contain all of the possible representations, where a road object's draw method would merely call a static instance of a drawing engine class with a parameter for that road type is not advised. This idea was considered, in part because it seems it would make central all issues dealing with representation, which would be valuable during debugging and inheritance. However, it would also seem to violate the spirit of object-oriented design by making the data objects not self-contained. Generally object-oriented design issues, such as when to use factory patterns [Gamma et al., 1995] and how to design the most pure inheritance hierarchy, although considered, were mostly ignored due to the author's lack of expertise. Although using these patterns in GIS implementation seems promising and may be in use in proprietary systems, they are also complicated and seem to be strongly programming language dependent. The reference material available to the author at the time of this exercise did not provide examples in Java.

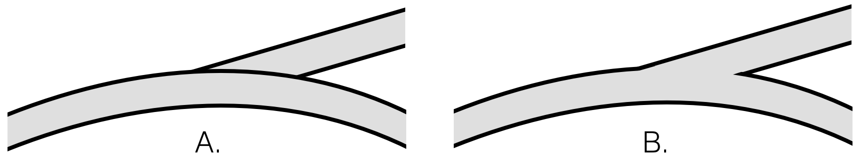


Figure 4. Road casings drawn in one shade and then overdrawn in another by line segment. The graphical representation of the roads do not join properly, A. The desired representation is B.

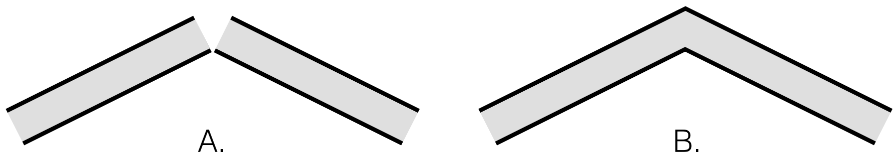


Figure 5. Contiguity problem stemming from disjoint line segments when a single polyline is expected.

Additional problems with representation were again artifacts of the data containing line segments when instead a single polyline line or network line grouping segments is desired so that spacing or contiguity problems are handled correctly (see Figure 5; problem appears also in Figure 3, highway in center of figure). Remaining problems have been noted with the representation on screen, which are similar to those experienced by [Ditz, 1997]. Namely, the pixel is the smallest addressable size, yet a pixel is already approximately .01" on a 17" monitor at 1024x768. Many line weights in the USGS specifications, intended for printing to paper from film, are half that size or less. Additionally, line symbols that must show variation, such as black outlines with a red interior casing, need at least three pixels—two for the black outer pixels and one for the inner red—an exaggeration of at least 50%. An on-screen specification will probably be necessary for future digital products.

Another concern in the Java implementation was how to compute the line widths interactively, based on the current resolution of the screen and the specified scale factor. There is a function call in Java that provides a resolution, though it was not the correct one because it could not know the size of the monitor. Regardless, this necessary capability was not yet implemented.

Another problem with Java was that the object class, BasicStroke, which implements different line styles cannot be serialized, or written directly to a file to be read in quickly later, due to potential security reasons because it has private variables. However, each feature class had a draw method and instantiated an object of BasicStroke to draw a symbolized line. That means that the mechanisms employed to save the objects persistently all broke when the BasicStroke functionality was used. That further means all the UNGENERATE data had to be read in, parsed, projected, and symbolized from scratch each time the program was run, slowing the time the program was started to the time all features had been drawn from under one minute with serialization to approximately seven minutes without serialization. This delay is painful and so methods to override the design of the BasicStroke are being investigated.

Methodology 3 — RevPG

During this exercise, it was later discovered by the author, that the USGS developed in-house a series of scripts and programs, called RevPG, completed by the early 90s, that symbolizes non SDTS-based DLG features and aids in their extraction from USGS Digital Orthophotos [RevPG User's Manual]. This legacy application contains approximately 150,000 lines of AML code to perform similar tasks that were explored in this exercise with the aim of supporting all possible features in a DLG data set. It is clear that accurate symbolization of the USGS data is no small task.

The USGS is currently developing a new computer application for feature extraction, called Framework Tools Interface (FTI). The FTI effort is replacing Product Generation E (PGE), which was meant for DLG-E data, though the symbolizing routines are being resurrected from RevPG [Cress, 1999]. It seems very likely that maintaining the procedures in the legacy AML code for symbolization will prove to be very difficult and costly. A smarter way to symbolize the data accurately and allow the information to have a longer shelf-life than ARC/INFO version 6.11 AML provides would probably be desired.

The RevPG system was installed, but the type of DLG data the program expected was not easily available for the study area and so further evaluation will be delayed. The source code was examined, however, and many unpublished symbology problems the USGS application designers encountered and provided solutions for were then uncovered.

Conclusion

These preliminary exercises results indicate the methodology served to represent a majority of the features found on the published map reasonably well. However, some non-trivial symbology issues still remained, such as rendering non-overlapping double line streets from lower class road centerlines found in the data base and also creating randomized icon-based area-fills that remain analogous to the original. However, for measuring purposes, the implemented system will probably yield sufficient information for determining symbology space usage, which would aid generalization methods. It is also noted that an intelligent model for storing and symbolizing the SDTS DLG-3 data and data which is similar in an object oriented model is still an open-problem.

As for the use of this data infrastructure to investigate generalization there were many items to consider. These items were, for the scales 1:24,000 and 1:100,000, digital vector and raster data (although the 1:100,000 vector data did not contain a complete list of feature classes), scanned maps, scanned orthophoto images, published specifications, and for 1:24,000, inexpensive printed maps (the 1:100,000 published map was discontinued). In certain spots on the maps, the data provided useful case studies, in situations where building footprints were shown, tightly clustered, and aligned (as in Figure 3). Experiments on generalizing these areas could be carried out and the results extended to situations in other countries in Europe or Asia. Otherwise, the geography and map specifications involved are predominately North American, such as streets that form an almost regular grid pattern and urban areas that are merely symbolized with a shade rather than retaining all buildings large enough to be shown, even at 1:24,000. At 1:100,000, there are generally no more than five buildings total in a 30' x 60' extent, yet the entire street network seems to be retained.

Another deficiency with the data infrastructure was the lack of maps or map products aimed at the 1:50,000 scale, which is popular in Europe but apparently absent in the US. Neither the USGS nor the National Imagery and Mapping Agency (NIMA) produced a map at 1:50,000 that covered the study area. According to [Beaulieu and Dohmann, 1997], NIMA has vector data at 1:50,000 and 1:100,000 for some areas, yet according to a recent inquiry with NIMA customer service, this data is restricted to U. S. Dept. of Defense personnel. Otherwise, this data might have proven interesting for generalization research.

And lastly, there were a few insights that were gained that could be applied to the domain of automated generalization. The road network seemed to be the dominant feature in this test area, and so the original USGS cartographic desire to delicately enhance some of the lower class roads on the 1:100,000 representation via displacement, when overlaid directly with the 1:24,000 representation, became clear. There were only a few situations where the 1:100,000 road network was altered, when more displacement had been falsely expected. Another matter to consider was that the 1:100,000 map was not derived from the 1:24,000 map. If this data were used in other ongoing work in calibration, a mechanism for matching would have to be implemented, which might be non-trivial. However, insight into designing experiments became clearer. The data situations on these real maps are complex and for algorithmic and measure testing purposes, a simplification of some of the case studies are then desired. However, by basing the experimental data on these case studies (as well as keeping mind solutions found on published maps) it is believed that the derived calibrating thresholds, especially to displacement, would be more useful.

Another question that arose from this exercise was what is the smallest set of feature classes that would demonstrate solutions to generalizing a sufficiently complicated environment and is such a question valid. Gower et al. note that the sequence in which processes are performed is an integral part of effective map generalization [Gower, 1997], yet too many feature classes would impede obtaining the research objective. It is difficult to commit an answer to paper at this time, yet in the study area, major roads were the most important structure imposed on the landscape, followed by second class (residential) roads, followed perhaps by water bodies, both lakes and rivers of all sizes, followed by selected buildings. Contours gave a heavy impression on the map due to the brown color, which weighted them more than is necessary. Railroads, transmission lines, and other features do add to the complexity of certain situations, but they could probably be omitted in order to prove if a system could treat cartographic data holistically. Yet, clearly, more research is needed.

Acknowledgments

The Swiss National Science Foundation (NF) Project No. 20-52759.97 and the European Union Information Technologies Programme ESPRIT No. 24939 (AGENT) supported this research. Special thanks go to D. Catts, USGS, NMD, for explaining unpublished methodology and work and providing the symbology publications and to M. Bader, U. Zurich and D. Ohe, USGS, WRD, for considering some aspects of the feasibility of this work. Thanks go to R. Weibel and M. Barrault for suggestions. Thanks also go to P. Wiese, J. Cress, S. Silch, B. Ailts, and S. Whitaker at the USGS, NMD for answering questions concerning USGS products and applications.

References

Beaulieu, P. and Dohmann, H. The Digital Geographic Information Exchange Standard and Military Mapping. In *Proceedings of the 18th ICC*, pp. 563-570, Stockholm, 1997. ICA.

Brazile, F. A Generalization Machine Design that Incorporates Quality Assessment. In T. Poiker and N. Chrisman, (Eds.), *Proceedings 8th International Symposium on Spatial Data Handling*, pp. 349-360, Vancouver, 1998. Simon Fraser University.

Brazile, F. and Edwardes, A. Organizing Map Space for Generalization through Object-primary Partitioning. This paper is under review, 1999.

Clarke, K.C. *Analytical and Computer Cartography.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Cranston, C.B., Brabec, F., Hjaltason, G., Nebert, D., and Samet, H. Adding an Interoperable Server Interface to a Spatial Database: Implementation Experiences with OpenMAP. In A. Vckovski, K. Brassel, and H.J. Schek, (Eds.). *Interoperating Geographic Information Systems*, pp. 115-128. Lecture Notes in Computer Science 1580, Springer, 1999.

Cress, J. USGS FTI project manager, personal correspondence, February 1999.

Department of Interior, U.S. Geological Survey, National Mapping Division (NMD). *Part 5 Publication Symbols. Standards for 1:24,000 and 1:25,000-Scale Quadrangle Maps*, updated 1994. Technical Instructions, referral number ST-365-4.

Department of Interior, U.S. Geological Survey, NMD. *Standards for 1:100,000-Scale Quadrangle Maps*, updated 1991.

Department of Interior, U. S. Geological Survey, National Mapping Division. *RevPG Procedures Manual and User's Guide*. 1995. Comes with the RevPG software, available from the USGS.

Ditz, R. An Interactive Cartographic Information System of Austria – Conceptual Design and Requirements for Visualization on Screen. In *Proceedings of the 18th ICC*, pp. 571-578, Stockholm, 1997. ICA.

Doyle, A. Dietrick, D., Ebbinghaus, J. and Hädstatter, P. Building a Prototype OpenGIS Demonstration from Interoperable GIS Components. In A. Vckovski, K. Brassel, and H.J. Schek, (Eds.). *Interoperating Geographic Information Systems*, pp. 115-128. Lecture Notes in Computer Science 1580, Springer, 1999.

Gamma, E. Helm, R., Johnson, R., and Vlissides, J. *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

Gower, R., Pepper, J., Edwardes, T. Landplan – Automated Generalization Comes of Age. In *Proceedings of the 18th ICC*, pp. 126-132, Stockholm, 1997. ICA.

Keller, S. and Thalmann, H. Modeling and Sharing Graphic Presentations of Geospatial Data. In A. Vckovski, K. Brassel, and H.J. Schek, (Eds.). *Interoperating Geographic Information Systems*, pp. 115-128. Lecture Notes in Computer Science 1580, Springer, 1999.

Session / Séance 39-C

Algorithmes de Généralisation basés sur le Lissage de la Courbure

Emmanuel Fritsch *and* **Sylvie Lamy**

Institut Géographique National
2-4 avenuePasteur, 94160 Saint Mandé - France
Emmanuel.Fritsch@ign.fr   Sylvie.Lamy@ign.fr

Abstract

Nous montrerons d'abord que le lissage gaussien de la courbure d'un arc routier est majoré par une constante inversement proportionnelle. Cette propriété permet de concevoir un algorithme opérant des élargissement de virages avec un rayon de courbure minimal garanti. Pour utiliser cet algorithme, il faut pouvoir recaler la courbe dont la courbure a été lissé. Après avoir considéré quelques transformations du plan, nous avons développé une procédure de recalage plus sophistiquée, qui réutilise la courbure pour détecter les virages les plus marqués. Ceux-ci sont replacés sur leur position initiale, mais dans leur conformation «élargie». Les segments intermédiaires sont ensuite replacés selon la méthode de l'homothétie.

Introduction

L'automatisation de la généralisation cartographique nécessite de développer des algorithmes de traitement spécifiques à certains problèmes. Nous allons ici nous intéresser au problème des routes très sinueuses, telles les routes de montagnes, dont la sinuosité induit de nombreux conflits, à l'intérieur même de l'arc, en particulier au niveau des virages trop serrés.

Les mauvais résultats des algorithmes couramment utilisés sont dus à l'utilisation exclusive de la représentation par liste de points, ou mode vecteur. Cette représentation ne permet pas de caractériser la forme locale des courbes étudiés. Au contraire, la courbure de la ligne est assez représentative de la forme locale, puisque les changements de direction, correspondant à des virages, se traduisent par une courbure élevée.

La courbure est largement utilisé en analyse d'image et reconnaissance des formes [Wuescher & Boyer, 91]. Certains algorithmes, tels celui de Lowe, [Lowe 88], s'appuient déjà sur la courbure, mais sans l'utiliser comme représentation de l'information. Nous montrerons dans cet article que représenter un arc routier par sa courbure est particulièrement adapté à la caricature des virages.

Caricature par lissage de la courbure

Dans cette partie, nous allons étendre la notion de courbure aux polylignes au moyen des distributions, distributions que l'on lissera ensuite au moyen de fonctions gaussiennes pour en assurer l'usage numérique.

La courbure d'une polyligne dans l'espace des distributions

La tangente à une polyligne étant une fonction constante par morceaux, elle n'est pas dérivable partout, et la courbure d'une polyligne n'est pas définie. Pour définir une courbure sur les polyligne, nous allons utiliser l'extension de la dérivation des fonctions aux distributions [Bony, 91]. Dans l'espace des distributions, la dérivation d'une fonction continue par morceaux est une somme de fonctions de Dirac, centrées sur chacune des discontinuités et pondérées des valeurs des sauts des discontinuités.

Ainsi, la courbure d'une polyligne, dérivée de l'orientation de la courbe au sens des distributions, est la somme de fonctions de Dirac centrées sur les sommets, pondérées des déviations angulaires au niveau de ces sommets.

La notion de courbure, non définie dans l'espace des fonctions, reçoit ainsi une extension dans l'espace des distributions.

Approximation numérique de la distribution.

Cette définition n'est pas numériquement utilisable, puisque les fonctions de Dirac n'ont pas d'expression numérique. Naturellement, il serait possible de procéder à du calcul formel sur ces expressions, mais le traitement cartographique nécessite des résolutions numériques. Pour cela, on peut procéder au lissage de la distribution-courbure. Ce lissage peut s'effectuer de façon très simple à partir de la courbure formelle et de la forme numérique de la fonction de lissage. En effet, la convolution d'une fonction avec un Dirac est la fonction de départ translatée de la valeur du point sur lequel le Dirac est centré.

Ainsi, si la courbure de la polyligne s'écrit : $\rho = \sum_{k=1}^{n} \alpha_k . \delta_{s_k}$

et que l'on veut la lisser avec une fonction gaussienne g_σ de paramètre σ,

alors : $$\rho_\sigma(s) = \sum_{k=1}^{n} \alpha_k . g_\sigma(s - s_k)$$

Nous avons ainsi une expression explicite de la courbure lissée en tout point de la polyligne.

Uilisation du lissage de la courbure

Le lissage gaussien a pour effet de diminuer la valeur absolue des extrema de courbure, et de le faire d'autant plus que le paramètre de la courbure est élevé [Babaud et al, 86]. Or la valeur absolue de la courbure est l'inverse du rayon de courbure de la courbe, et les extrema de cette courbure correspondent aux virages les plus serrés. Le lissage gaussien de la courbure va donc augmenter le rayon de courbure des virages les plus serrés, et cette augmentation du rayon de courbure sera d'autant plus importante que le paramètre de lissage sera élevé.

Ainsi le lissage gaussien de la courbure peut avoir deux applications différentes, selon la valeur du paramètre : dans le but initial de définir la courbure d'une polyligne, une faible valeur du paramètre est nécessaire. Pour obtenir une caricature des virages, un paramètre plus élevé sera utilisé. Dans le premier cas, il faut que l'ordre de grandeur du paramètre soit équivalent ou inférieur à la résolution de l'échelle initiale. Dans le second cas nous allons montrer maintenant que le choix du paramètre s contraint directement une valeur minimale du rayon de courbure de la courbe généralisée.

L'élargissement de virage - Le rayon de courbure minimal

Nous allons montrer maintenant que l'arc routier reconstruit à partir de la courbure lissée présente une courbure maximale, c'est à dire un rayon de courbure minimal, dépendant uniquement du paramètre de lissage. Cette propriété est intéressante, car en fixant le paramètre de lissage de telle sorte que le rayon de courbure minimal garanti soit égal ou proche de la demi-largeur du symbole, on assurera la lisibilité des virages les plus serrés.

Cette propriété de rayon minimal de la courbure n'est pas vérifiée pour un arc quelconque : on peut toujours trouver un arc géométrique donné dont la courbure, après lissage, sera supérieure à la valeur seuil de la courbure.

Toutefois, les arcs routiers présentent des caractéristiques spécifiques qui permettent d'assurer l'existence du seuil des courbures. Nous énonçons et justifions ici ces cette hypothèse restrictive :

Les déviations angulaires entre deux points de la route sont inférieures à 180º. En effet, une route, instrument de communication, tend à relier deux points par un chemin aussi simple que possible, et n'a donc pas tendance à faire des boucles. En fait, même en zone de montagne, on dépasse rarement les 180º. Certaines épingles à cheveux peuvent présenter un aspect saillant, et la valeur de 180º localement se trouvera dépassée, mais généralement de peu. Dans la suite, nous admettrons que les virages sur les routes sont au maximum de 180º.

D'autre part, nous considérerons que les virages sur un tracé routier sont indépendant les uns des autres, séparé par des distances importantes.

<u>Enoncé de la propriété</u> : Pour assurer un rayon de courbure supérieur à *r* à des arcs dont la variation de direction est inférieure à 180º, il suffit de pratiquer un lissage des courbures par une fonction gaussienne de paramètre σ, avec :

$$\sigma = r.\sqrt{\pi}.$$

<u>Démonstration</u>

Nous allons considéré ici l'effet du lissage gaussien sur un virage, dont la courbure ρ est modélisé oar une fonction gaussienne, de paramètre τ :

$$\rho(x) = \alpha \frac{e^{-x^2/\sigma^2}}{\tau.\sqrt{\pi}}$$

La déviation angulaire entraînée par ce virage vaut α, exprimé en radian,

et le virage atteint son maximum de courbure ρ_0 en x=0, et $\rho_0 = \frac{\alpha}{\tau.\sqrt{\pi}}$ (Eq1)

Soit ρ_s la courbure lissée par un gaussien de paramètre σ. Le produit de convolution de deux fonctions gaussiennes est une fonction gaussienne, de paramètre : $\tau_\sigma = \sqrt{(\tau^2 + \sigma^2)}$.

et la courbure maximale est là encore atteinte en 0, avec $\rho_{\sigma 0} = \frac{\alpha}{\sqrt{(\tau^2 + \sigma^2)\pi}}$ (Eq2)

$$\text{(Eq2)} \quad \Rightarrow \quad \rho_{\sigma 0} < \frac{\alpha}{\sigma\sqrt{\pi}}$$

Or par hypothèse, la déviation angulaire entre deux points est majorée : $\alpha < \pi$, d'où :

$$\rho_{\sigma 0} < \frac{\sqrt{\pi}}{\sigma}$$

Ainsi, pour un paramètre de lissage σ donné, si nous définissons : $R_{res} = \sigma / \sqrt{\pi}$

alors, pour toute valeur *x* et pour tous les arcs de la base, $\rho_\sigma(x) < \frac{1}{R_{res}}$

Ainsi pour assurer un rayon de courbure en tout point supérieur à R_{res}, il suffit d'appliquer à la courbure de l'arc un lissage gaussien de paramêtre σ, avec :

$$\sigma = R_{res}.\sqrt{\pi}.$$

Recalage de la courbure lissée par affinités

le recalage des courbures lissées

La courbe définie par sa courbure n'est connue qu'à un déplacement près. En effet, la courbure d'un artc est une information complètement délocalisée. Se pose alors le problème du recalage de la ligne lissée sur les données originale. Ce recalage est nécessaire, en particulier pour assurer la connectivité de l'arc au réseau. Pour cela, un simple déplacement n'est pas suffisant. En effet, comme nous le montrons sur la figure 1, il est impossible de faire coïncider par un déplacement les points extrémités des deux courbes, et donc de rétablir la topologie du réseau.

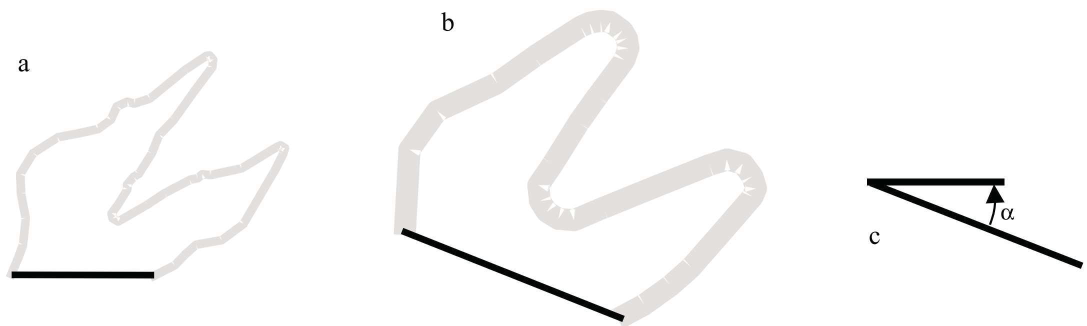


Figure 1. Présentation du problème de recalage d'un arc lissé.

La figure illustre le problème du recalage : la figure a présente un arc (en griset), et la figure b la reconstruction de cet arc après lissage de sa courbure. L'élargissement des virages induit une déformation de la courbe, et donc du segment de base. La différence entre les deux segments est visualisée sur la figure c.

Nous allons maintenant nous intéresser à des transformations du plan plus complexes que le déplacement, qui vont donc modifier la géométrie de la ligne généralisée pour en faire coïncider les extrémités avec celles de la courbe originale. Naturellement, nous souhaitons que ce recalage ne dégrade ni le travail de caricature réalisé par le lissage gaussien, ni les formes caractéristiques de l'arc.

Similitudes

La Figure 2 présente le recalage le plus simple qui soit, c'est à dire ls similitude. Cette similitude se décompose en une rotation et une homothétie centrée sur l'origine de l'arc. Une fois la base de l'arc généralisé ramenée sur celle de l'arc original, au moyen de la rotation, on assure la coïncidence des points terminaux par l'homothétie.

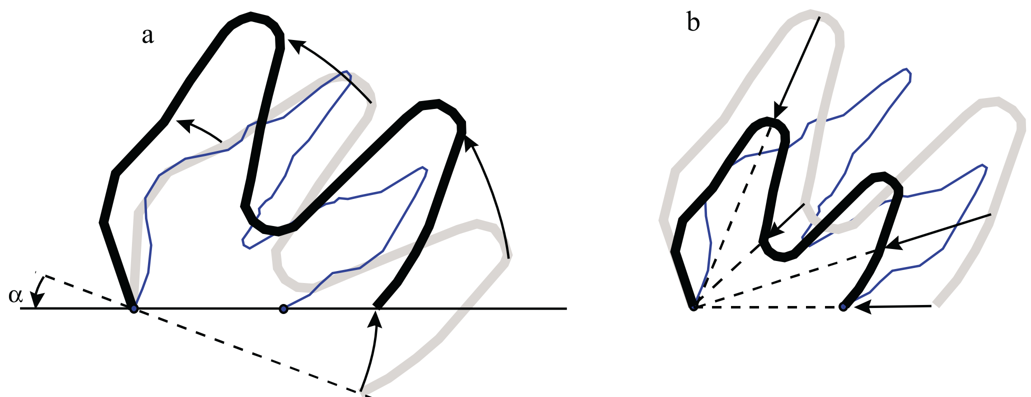


Figure 2. Recalage par similitude.

La similitude de recalage de l'arc de la Figure 2 est décomposée sur cette figure en une rotation (figure a) et une homothétie (figure b). Le trait fort représente la courbe produite par la transformation, le trait grisé habille la courbe avant transformation, et le trait fin est la trace de l'arc original, pour comparaison.

Cette figure montre les défauts attachés à ce type de recalage : comme le lissage de la courbure augment la taille des virages, l'ensemble de la courbe se trouve distendue. Le rapport de l'homothétie est donc inférieur à 1. Il en résulte :

- une mauvaise localisation des détails de la courbe
- une forte réduction de la taille de chaque épingle à cheveux.

Ces artefacts sont plus ou moins sensibles selon les arcs, mais dès que l'on traite des routes à fortes sinuosités, la similitude donne des résultats aberrants. Pour maintenir la taille des virages et la précision géométrique, il nous faut utiliser des modes de recalages plus complexes.

Dans les procédés de recalage que nous allons développer maintenant, nous réutiliserons toujours la rotation. Elle permet en effet de faire coïncider les bases des deux arcs, sans aucune modification du traitement de généralisation effectué par le lissage de la courbure.

La rotation n'ayant aucune incidence sur la géométrie de la ligne, c'est l'homothétie qui introduit les déformations inacceptables, et c'est elle que nous allons maintenant essayer de remplacer par d'autres transformations plus complexes.

Affinités orthogonale et oblique

L'affinité, présentation géométrique

Pour ramener le segment de base de la courbe généralisée sur le segment de la ligne originale, on a utilisé une similitude qui altère le message cartographique. L'action de la similitude étant uniforme selon les directions, il a semblé intéressant d'utiliser une autre classe de transformations géométriques simples qui permettent de comprimer selon une seule direction en en gardant une autre invariante : les affinités.

Une affinité du plan se définit par la donnée de deux axes séquents D_1 et D_2, affectés chacun d'un coefficient (k_1 et k_2). Si $\underline{u}$ est un vecteur du plan, se décomposant en $\underline{u} = \underline{u}_1 + \underline{u}_2$ sur les deux axes, alors l'image de $\underline{u}$ par l'affinité est : $\underline{v} = k_1.\underline{u}_1 + k_2.\underline{u}_2$.

L'intérêt des affinités sur les similitudes est ainsi de découpler la direction de compression, nécessaire au recalage, d'une autre direction, soit orthogonale, soit définie par la direction principale de la ligne. Le long de cette seconde direction, pour supprimer les déformations, on imposera un taux de compression nul, i.e. $k_2 = 1$.

Ainsi, l'on espère au moins maintenir les détails dont la plus grande extension ne se trouve pas parallèle au segment de base.

L'affinité orthogonale

Dans ce paragraphe, nous considérons pour direction invariante l'orthogonale au segment de base.

On constate sur les premiers tests que si les longueurs des virages sont dans l'ensemble mieux respectées avec cette méthode qu'avec la précédente, les orientations des épingles à cheveux sont parfois fortement modifiées.

En fait, les deux axes définissant l'affinité sont les deux seules directions invariantes de la transformation, dès que les coefficients de l'affinité sont différents. Il n'est donc pas étonnant de constater une altération des directions des épingles à cheveux. Mais cette altération est difficilement tolérable. En effet, les épingles à cheveux permettent à la route de gagner de l'altitude. La pente doit en être continue et relativement faible, ce qui se manifeste sur la carte par un angle d'incidence très faible entre les lignes de niveaux et la route. Si l'on change l'orientation de la route, celle-ci va couper les lignes de niveaux avec une incidence beaucoup plus importante, et l'altimétrie de la route semblera très étrange : un véhicule circulant sur une telle route monterait et descendrait alternativement entre chaque épingle à cheveux.

Ainsi l'orientation des virages est-elle une composante essentielle du message cartographique, et nous allons essayer de mieux tirer profit des possibilités des affinités afin d'améliorer la prise en compte de cet impératif. En effet, l'axe d'invariance de l'affinité a été choisi orthogonal au segment de base, parce qu'ainsi sont définies les affinités les plus utilisées, mais aussi parce que c'est ainsi qu'en théorie on assure une déformation géométrique minimale : plus les deux axes sont proches et plus les déformations sont importantes.

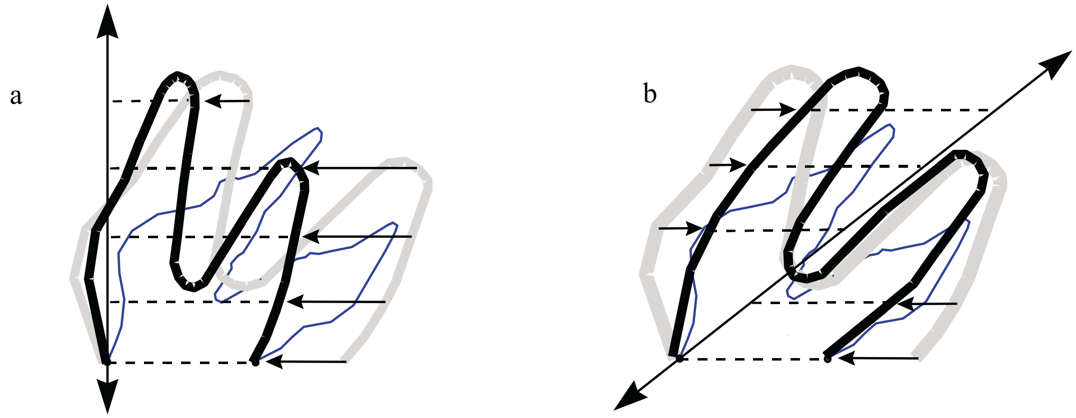


Figure 3. Principe de recalage par affinité orthoginale (a) ou oblique (b).

La courbe en grisé représente l'arc dont la courbure a été lissée, puis qui a subi la rotation définie plus haut. C'est sur cet arc que l'on va faire agir les affinités.

Les affinités obliques

Pour fixer une direction invariante plus intéressante que la simple direction orthogonale, il nous faut donc composer avec deux exigences :

· si la direction invariante est la direction orthogonale, les orientations sont faussées,

· mais si la direction invariante est trop proche de celle du segment de base, nos déformations seront très importantes.

La direction invariante que nous allons fixer doit donc être intermédiaire entre l'orthogonale au segment de base et la direction privilégiée des détails de la courbe. La difficulté qui apparaît maintenant est la définition de cette direction privilégiée.

On conçoit intuitivement ce que représente la direction privilégiée d'une épingle à cheveux : C'est l'axe de symétrie de ce virage. La direction d'un ensemble de virages inhomogènes est plus difficile à définir, mais reste sensible à l'intuition. Dans le cas d'un arc quelconque, cette direction ne sera pas toujours définie (certains arcs ont un tracé sans aucune organisation), par exemple un trajet bouclant sur un cercle n'a pas de direction privilégiée, alors que sur une ellipse, la direction privilégiée est le grand axe de l'ellipse, et ce d'autant plus que l'excentricité de l'ellipse est grande.

Pour définir la direction privilégiée d'un arc, nous pourrions utiliser diverses méthodes de la littérature, telle que la “distance direction matrix” proposée par [McMaster, 83], ou le code de Freeman ([Freeman, 74]). Néanmoins, ces deux types de méthodes donne des résultats discrets, et sont plutôt applicable à une représentation en mode raster. Pour obtenir une mesure continue, plus adaptée au mode vecteur, nous avons développé une mesure de direction non-orientée. La direction orientée est un angle entre 0 et 360°, c'est à dire, mathématiquement, une valeur de $R/2\pi Z$. Ainsi, une direction non orientée est un angle pris entre 0 et 180°. Pour travailler avec de tels angles, il faut travailler dans $R/\pi Z$.

Pour cela, le calcul de la direction non-orientée moyenne se fait par sommation de nombres complexes dont la représentation $z = r.e^{i\Theta}$ est remplacée par $z = r.e^{2i\Theta}$ avec q l'angle de chaque segment, et r sa longueur. Le résultat de cette sommation s'écrit $Z = R.e^{2i\Theta}$, et Θ est alors l'angle de la direction non-orientée moyenne de l'arc considéré, et donne donc l'axe invariant de l'affinité.

Comparés aux résultats du recalage par similitude, ceux de l'affinité oblique avec détection de la direction privilégiée sont généralement meilleurs. Le rétablissement des orientations des virages est bien effectué, et l'amplitude des virages est généralement mieux respectée. Néanmoins, plusieurs défauts demeurent. En particulier la localisation des détails cartographiés n'est pas vraiment assurée. Si les points extrémités sont replacés avec exactitude, les autres parties de la courbe peuvent subir des dérives importantes. On constate ainsi que certains virages sont très loin, après le recalage, de leur position nominale. Ce défaut, très important sur les résultats de recalage par affinités, est déjà présent sur les résultats de recalage par similitude.

Mais le principal défaut que l'on puisse trouver aux différentes méthodes de recalage présentées jusqu'ici est de ne pas respecter les rayons de courbure que la méthode des lissages avait permis de garantir. En effet, l'élargissement des virages induit systématiquement un allongement du segment de base. Les recalages uniformément répartis sur la ligne induisent donc une compression de cette dernière, ce qui diminue les rayons de courbure des virages, et mange en partie le gain du lissage. Cette diminution des rayons de courbure est relative, mais elle dépend fortement de la forme de la ligne, et n'est absolument pas contrôlable.

Algorithme plâtre

Nous avons vu que les recalages développés jusqu'ici ne permettent pas de conserver le rayon de courbure minimal théoriquement garanti par le lissage de la courbure. Les similitudes comme les affinités ont tendance à réduire le rayon de courbure des épingles à cheveux. Pour préserver ce rayon de courbure minimal, il fallait développer une transformation qui effectuât les amortissements sur les parties de faible courbure, sans déformer les parties de fortes courbures. D'autre part, les algorithmes précédents n'assuraient pas la localisation des points caractéristiques sur les arcs généralisés.

Description de l'algorithme

Pour remédier à ces deux défauts, nous avons mis au point un algorithme de déplacement par morceaux. Le principe est de déplacer dans un premier temps les parties de forte courbure qu'il faut recaler sans en modifier la forme, et ensuite seulement les parties de courbure plus faible, sur lesquelles on peut ainsi effectuer les amortissements de déplacement nécessaires au raccordement des arcs.

Plus précisément, notre algorithme isole sur la ligne les morceaux dont la courbure est supérieure à un seuil donné (proche de la limite théorique de courbure définie en première partie). Pour chacun des morceaux ainsi détectés, on calcule alors les barycentres, sur la courbe originale et sur la courbe reconstruite après lissage des courbures. Le vecteur défini par ces deux points détermine la translation que l'on applique à chacun des points du morceau considéré. Les parties de forte courbure sont ainsi recalées indépendamment les unes des autres sur leurs positions initiales, et leurs formes sont conservées par translation. Naturellement, les extrémités de l'arc, sont recalées lors de cette première phase, sous la forme d'éléments ponctuels en même temps que les parties de fortes courbures

Enfin, on effectue les déplacements des parties de courbure plus faible en amortissant les différences de déplacements entre les parties déjà placées. L'amortissement du déplacement se fait linéairement selon l'abscisse curviligne.

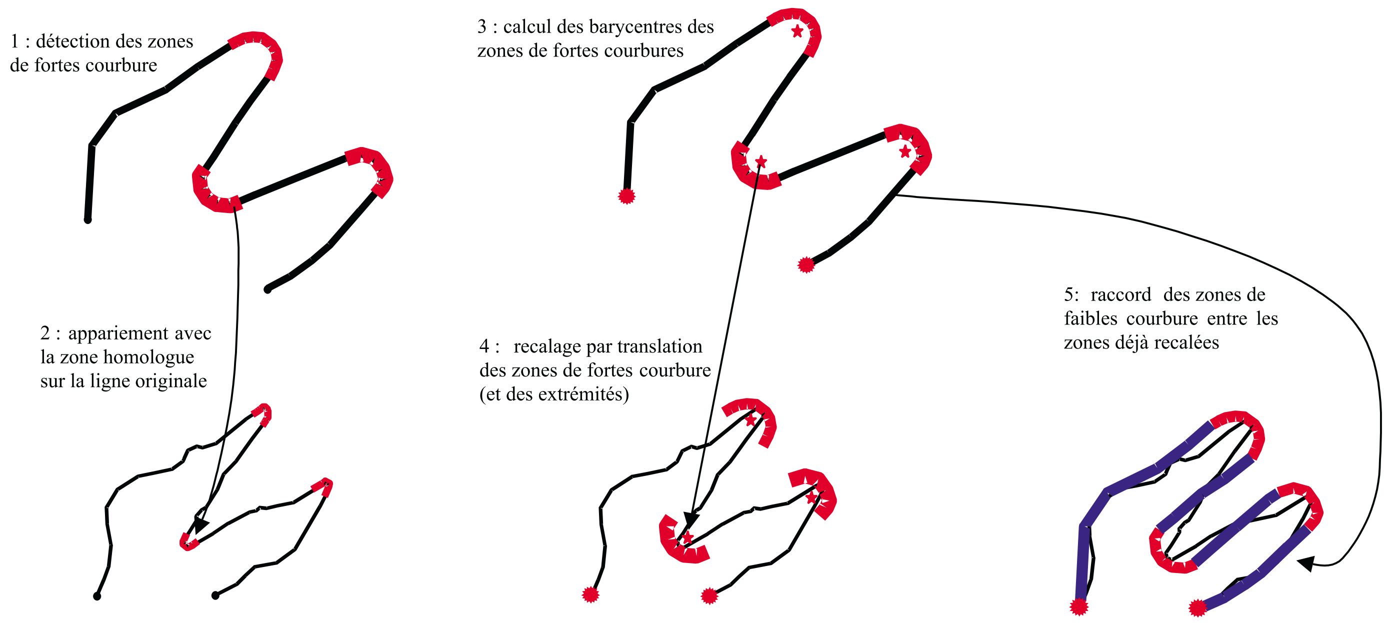


Figure 4. Principe du recalage plâtre.

Résultats

Nous avons appliqué le lissage de la courbure sur un échantillon issu de la BDCarto®, dont les données ont été constituées par saisie manuelle des tronçons routier de la carte au 1:50.000e. L'échantillon couvre ½ feuille de la zone de Bourg-Saint-Maurice et compte 559 arcs. Le tableau ci-dessous présente la proportion estimée de tronçons rectilignes, de tronçons avec sinuosités, et de tronçons de liaisons. On estime d'autre part, par comparaison avec la carte IGN série verte no 53, que la quasi totalité des arcs de l'échantillon doit figurer sur la carte au 1/100.000e.

arcs	sinueux	rectilignes	de liaison
proportion	40%	50%	10%

Figure 5. Exemples de résultats du recalage par affinité, et de l'algorithme plâtre, extrait de l'échantillon traité. En haut, les données d'origine, en bas après utilisation de l'algorithme. La lisibilité des virages est fortement améliorée, sans artéfact sur la précision géométrique. Les successions de virages serrés posent néanmoins des problèmes, surtout si les épingles à cheveux sont de faible amplitude.

Les résultats de cet algorithme sont relativement satisfaisants. Le contrôle obtenu tant sur les rayons des virages que sur la localisation et l'orientation des détails principaux permet de compter sur une généralisation plus sûre et plus homogène qu'avec les autres méthodes. Néanmoins, deux défauts apparaissent :

- Si deux épingles à cheveux se suivent sans laisser entre elle une branche suffisante, on voit apparaître sur les résultats un décrochement assez inesthétique. Les deux virages sont recalés indépendamment l'un de l'autre. Le déplacement des branches de l'un à l'autre ne peut pas être amorti, car il n'y a pas assez d'espace pour cela.
- L'algorithme n'effectue qu'un élargissement de chacun des virages. L'amélioration de lisibilité obtenue est relative uniquement à la courbure trop élevée des données initiales. En particulier les conflits par superposition de portions de route qui ne sont pas contiguës selon l'abscisse curviligne ne sont pas traités par l'algorithme. De fait, ces cas réclament une dilatation, et il faut utiliser un autre algorithme, par exemple l'accordéon [Plazanet, 96] ou encore un mécanisme de répulsion [Fritsch, 96]. Si vraiment la congestion de la zone est trop forte, il faudra procéder à l'élimination d'une partie des détails.

Figure 6. Exemple de décrochement sur de petits virages (agrandissement).

Malgré ces deux défauts, l'algorithme propose une opération de caricature des virages très intéressante. Menée dans le cadre des tests de généralisation de l'OEEPE [Mustière 97], une étude a montré que l'algorithme apportait une solution élégante et rapide sur de très nombreux arcs. D'autre part, l'algorithme développé à l'origine sur la plate-forme PlaGe [Fritsch, 98], a été porté sur Lamps2 dans le cadre du projet Agent.

Sur une paire de petits virages, le résultat de l'algorithme présente un décrochement, dû à la trop courte distance entre les deux virages qui rend impossible un amortissement harmonieux.

Conclusion

Validations des représentations basées sur la courbure

Les améliorations de lisibilité obtenues par les algorithmes développés ici montrent que le lissage de la courbure effectue une bonne opération de caricature sur les arcs routiers.

Développements envisagés

Si l'algorithme plâtre peut recevoir des améliorations, c'est avant tout dans la définition de ses zones de recalages sans déformation. La première amélioration devrait consister à vérifier qu'entre deux zones, une distance minimale est respectée. Si tel n'est pas le cas, les deux zones doivent être agrégées. Ensuite, par détection des superpositions, ou par des méthodes de calcul de proximité, on pourrait regrouper les virages d'une succession d'épingles à cheveux pour en faire une seule zone.

L'algorithme plâtre distingue sur un arc les zones de hautes courbures (les virages) des zones quelconques. Cette segmentation ne sert que lors du recalage. On pourrait envisager de faire subir des traitements différents à la courbure selon la zone dans laquelle on se trouve, par exemple de lisser normalement les zones de faibles courbures, et d'approximer par un cercle (effet de rayon de courbure seuil) les zones de trop forte courbure.

Applications possibles

Le principal avantage de la méthode "plâtre" est son ergonomie : la tâche qu'elle accomplit est claire (élargissement des virages), et le choix du paramètre unique est contraint par la largeur du symbole dont sera habillé l'arc dans la carte généralisée. C'est donc un algorithme relativement simple. On pourrait ainsi envisager de l’appliquer de façon systématique sur tous les arcs, en prétraitement, lors de la généralisation d’une zone de montagne. Pour rendre cette possibilité intéressante, il faudrait être capable de valider le résultat fourni, c’est à dire de détecter les cas d’erreurs pour proposer en aval une reprise à l’opérateur. D’autre part, il manque actuellement une gestion des carrefours et des croisements, sur lesquels apparaissent un grand nombre de conflits. Enfin, il faudrait pouvoir répercuter sur les autres thèmes les déplacements créés par le routier.

Les recalages fondés sur similitudes et affinités présentent des résultats beaucoup plus contrastés. On peut envisager néanmoins leur utilisation dans un processus de généralisation assisté.

Références

[Babaud et al, 86] Babaud, J. Witkin, A. & Baudin, M. & Duda, R. “Uniqueness of the Gaussian Kernel for Scale-Space Filtering”. Trans. on Pattern Analysis and Machine Intelligence 1986, Vol 8, No 1, pp 26-33.

[Bony, 91] J.M. Bony, 1991. “Cours d’Analyse”. Ecole Polytechnique.

[Freeman, 74] H. Freeman, 1974. «Computer processing of line drawing images». Comput surveys, vol 6.

[Fritsch, 96] E. Fritsch, 1996. «A mechanical approach for line generalisation». GISRUK’96, avril 1996 p 49-55

[Fritsch, 98] E. Fritsch, 1998. «Représentation de la Géométrie et des Contraintes Cartographiques pour la généralisation du Linéaire Routier». Thèse de doctorat de l’Université de Marne-la-Vallée, décembre 1998.

[McMaster, 83] McMaster R. “Mathematical measures for the evaluation of simplified lines on maps”. Rapport de thèse, Université du Kansas, 1983.

[Lowe, 88] Lowe D.G. “Organization of smooth image curves at multiple scales”. Proc. 2nd ICCV, 1988, p 558-567

[Müller et al, 95] J.C. Muller, R. Weibel, J.P. Lagrange & François Salgé (1995). Generalization: state of the art and issues, in GIS and Generalisation, edited by J.C. Müller, J.P. Lagrange et R. Weibel, GISDATA no1 - Taylor & Francis 1995, 3-17

[Mustière, 97] Mustière S. “Tests des algorithmes de PlaGe”. Rapport Technique, IGN - OEEPE, 1997.

[Plazanet, 96] Plazanet C. « Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes) «. Thèse de doctorat, Université de Marne-La-Vallée. 1996.

[Wuescher & Boyer, 91] D.M Wuescher and K.L. Boyer, 1991. “Robust contour Decomposition using a constant curvature criterion”. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol 13 pp 41-51.

Session / Séance 10-D

Dynamic Generalisation from Single Detailed Database to Support Web Based Interaction

Elizabeth Glover *and* **William A. Mackaness**

Department of Geography, The University of Edinburgh,
Drummond St, Edinburgh EH8 9XP, Scotland
email: wam@geo.ed.ac.uk

Abstract

It is believed that Web-based visualisation of spatial information could be greatly enhanced through the use of dynamic or 'on-the-fly' map generalisation. This involves dynamically deriving scale- and theme-dependent displays from a single detailed dataset, thus eliminating the need to maintain duplicate datasets at different resolutions. Dynamic generalisation also prevents the generation of cartographically-poor maps that result from the display of spatial information at a considerably different resolution from that at which it was captured. The implementation involves combining automated generalisation and symbolisation techniques together in a simple rule-base to produce a virtual map of central Edinburgh in which both the level of detail and the symbology is dynamically tailored to a user-specified scale and map theme (tourist or topographic maps). The paper explores the inherent differences between the generalisation of a virtual map and that of a paper map, and also considers issues such as those arising from the relationship between map scale, map theme and utility. The research demonstrates that it is possible to achieve effective results using a limited number of simple mechanisms, and argues that the true potential of dynamic generalisation lies in two main areas - firstly, in an ability to narrowly-define the map theme, thus enhancing map clarity and more closely matching user requirements; and secondly, in using control over the map scale as a gateway to other types of map use and other themes.

1.0 Introduction

Dynamic generalisation is the derivation of a temporary generalisation from a detailed geographic database, often for display on a computer screen (Van Oosterom, 1995). As every display is automatically derived on-the-fly from a single detailed dataset, database redundancy is avoided (Van Oosterom, 1995). Automated generalisation offers other advantages. Not only is the costly and time-consuming bottleneck of manual generalisation removed, but currency and consistency across the range of representations can be vastly improved (Gower et al, 1997). Secondly, the production of cartographically-poor maps can be prevented by dynamically generalising the map according to the requested display scale, thus producing smoother transitions over scale. Dynamic generalisation also enables a range of map themes to be produced through selection of the features displayed.

2.0 Modelling and Automating Map Generalisation

The representation of a cartographic feature becomes increasingly abstract as map scale decreases, as illustrated in Figure 1a. It is possible to model this transformation in terms of the generalisation operators used. Most generalisation at larger scales occurs through the transformation of the feature's geometry. The mechanisms of simplification, enlargement and displacement are utilised, and changes occur quite smoothly. Particular generalisation solutions can therefore be envisaged as operating over a particular scale band defined by an upper and lower threshold scale. As the map scale reaches a threshold, a change in the level of generalisation is invoked. This cut-off point is in part determined by the feature's geometry. Shape and size will dictate the scales at which firstly detail, and eventually the entire feature, become unidentifiable. Figure 1b shows the scale bands over which the representations in Figure 1a might operate.

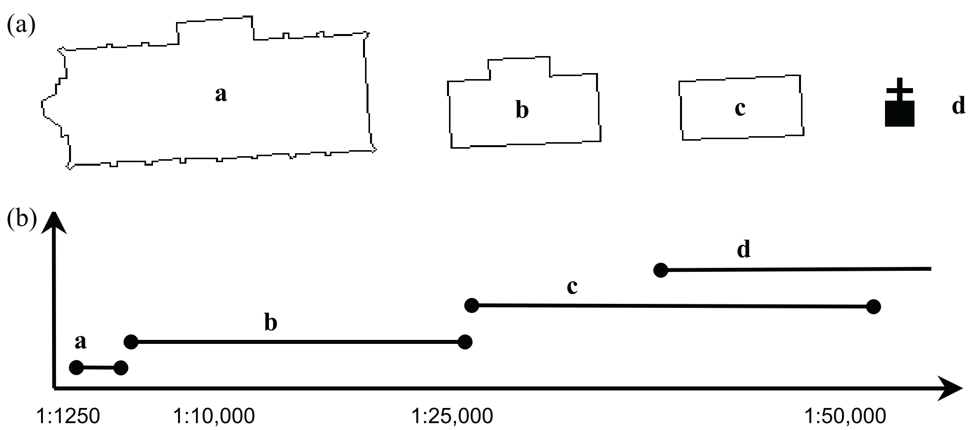


Figure 1: a) Transformations with decreasing map scale; b) Corresponding scale bands for a topographic map.

The generalisations used, and their associated scale bands, are however influenced by a number of other factors. As Ormsby and Mackaness (1999) have suggested in their phenomenological approach to generalisation, both map theme and object type play a part. Muller (1989:202) too points out that '... the threshold point which separates geometric and conceptual generalisation does not occur at the same scale for every cartographic feature and every thematic realm'. A further influence is the context of the object undergoing generalisation. For example, a small building in a city may be eliminated as map scale is reduced, but a similarly-sized building in a remote area would be retained.

The dynamic generalisation presented in this paper attempts to cross a broad range of scales. Most commonly, automated generalisation is undertaken over scale reductions of factor 10 or perhaps less. For example, the Ordnance Survey 1:10,000 LANDPLAN data is derived from 1:1250 source data (Gower et al, 1997) and the Institut Géographique National in France is automating generalisation from a source scale of 1:50,000 to a target scale of 1:250,000 (Lecordix et al, 1997). Here, however, displays at 1:50,000 are derived directly from a source dataset at 1:1250 - a scale reduction of factor 40.

Identification

The study of both Ordnance Survey and tourist maps of Edinburgh yielded a number of generalisation solutions that could feasibly be automated. Given the urban nature of the area, the process of knowledge acquisition focused on key buildings such as hospitals, stations, and museums, general buildings, and road features. Examples of building generalisation are given in Figure 2 for two broad groups of buildings: key (or special buildings) and general buildings.

3.0 Key buildings

Selection and simplification

Representation (b) of Figure 2 can be approximately derived by eliminating the smaller buildings and simplifying the remainder. Simplification is performed using a previously-implemented algorithm designed for the simplification of features with right-angled geometries. The algorithm works by flipping out the corners of the building. See Glover (1998) for a more detailed explanation of the algorithm.

Aggregation

Representation (c) in Figure 2 requires the aggregation of all the buildings making up the castle. In order for aggregation to occur, the component objects first need to be grouped together. The derivation of meaningful groups of objects is a critical element of map generalisation and as such has received considerable attention in the literature. Methods proposed include the use of minimum spanning trees (Regnauld, 1996), graph theory (Mackaness and Beard, 1993) and multi-variate cluster analysis (Ormsby and Mackaness, 1999). These methods are designed to produce clusters within a particular feature layer, for example buildings that have not been explicitly defined. The buildings are then aggregated using an existing algorithm that returns the convex hull of the set of geometries. The convex hull is simplified using a Douglas Peuker line simplification algorithm to give a less rounded appearance. A full account of the grouping procedure and aggregation algorithm and can be found in Glover (1998).

Collapse

Representation (d) shown in Figure 2 involves deriving a point geometry from the original area geometries. If the centre of a single building is required, the centre of the building is used as the point location. If the centre of a number of buildings is required, the centre of the minimum bounding rectangle of the group is used. A more representative location might be achieved by computing the centre of gravity of the group.

4.0 General buildings

Selection and simplification

Representation (b) in Figure 2 can be approximately derived by eliminating smaller buildings and simplifying the remainder.

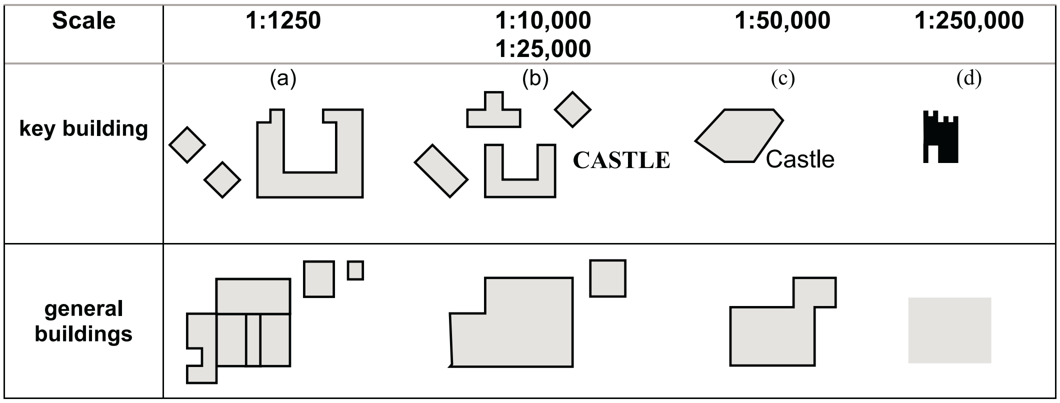


Figure 2. Examples drawn from paper maps of building generalisation at various scales

Area patch generalisation

Representation (c) of Figure 2 can be produced by using an approach based on Muller and Wang's (1992) area patch generalisation algorithm, designed to generalise sets of area features varying in shape and size but of the same type. Each feature is either expanded, contracted, or eliminated. Although the algorithm was originally developed for the generalisation of natural features, it has since been modified and successfully applied to buildings by Ormsby (1996). A simplified approach has been used here. Buildings below a minimum area are eliminated and those remaining are expanded through scaling. The enlargement causes the appearance of aggregation as buildings overlap to produce built-up areas.

Solid built-up area

A solid built-up area is represented by polygonising the areas in between the road network to form city blocks. These can then be plotted with a solid fill.

Roads

Roads are generalised simply by selection and the alteration of symbology. At the larger scales, roads are displayed as cased lines. As map scale decreases, minor roads are initially reduced to a single line and are finally eliminated.

5.0 Identification of scale bands

Paper maps provide just a few snapshots of the whole generalisation continuum. Interpolation was therefore required to identify generic scale bands over which each solution operates. The points at which one representation gives way to another could, theoretically, occur at any point on along the continuum. In this research, the scale of the printed map was taken to be the upper threshold for that particular representation. Scale bands were then interpolated back to the point at which a more detailed map had been produced. For example, the scale band for the representation used on a 1:50,000 map would be 1:25,000 - 1:50,000. The scale bands identified in this manner are not directly applicable to a virtual map displayed on a computer screen due to differences in resolution. While raster printers and plotters can achieve resolutions ranging from 200 to 2000 dots per inch, the resolution of a computer screen is typically only 72 dots per inch (Jones, 1997). A map displayed on a computer screen can therefore support far less detail than a printed map of the same scale. This means each threshold must be shifted towards the large-scale end of the continuum. Generic scale bands for the virtual maps were identified by displaying each representation at its upper threshold scale as identified from the printed map, and then zooming in until the resolution was comparable to that of the printed map. The two sets of scale bands are displayed in Table 1.

Table 1. Scale bands for printed and virtual maps

Scale band	Printed map		Virtual map	
	Lower threshold	Upper threshold	Lower threshold	Upper threshold
a	1:1	1:6,000	1:1	1:6,000
b	1:6,000	1:25,000	1:6,000	1:14,000
c	1:25,000	1:50,000	1:14,000	1:26,000
d	1:50,000	1:250,000	1:26,000	1:50,000

6.0 Influences on generalisation

A number of factors beyond just geometric influence the form of representation. These are map theme, object type and context.

Map theme

Not only does the generalisation solution depend on the map theme, but generalisation operators can also be used to control the thematic content of the map. In order to demonstrate these relationships, two map types with strongly contrasting styles (topographic and tourist) were studied and automatically derived from the source dataset. Table 2 outlines the main differences between the two map types. Bertin (1967) was perhaps the first author to propose that map theme could influence the generalisation process, demonstrating that very different results could be produced according to whether the end product was to be a road map, air chart or for an atlas. The different styles were achieved by manipulating the threshold scales between solutions.

Table 2. Differences between topographic and tourist maps.

	TOPOGRAPHIC	TOURIST
Content	Wide range of features of general interest	Features of touristic interest or importance only
Primary consideration	Shape, size and distribution of features	Location and relative size of features
Style	Full of detail Geometric icons used	Simple, clear Pictorial icons common

Map theme can also be controlled through the reclassification of features. Key buildings can be reclassified as general buildings if they are not ‘key’ to the particular map use. For example, schools need not be displayed on a tourist map so are displayed instead in the same way as the general buildings.

Object type

It is now commonly accepted that generalisations vary according to object type (see for example Mark, 1991). A buffering technique appropriate for the generalisation of natural features is, for instance, inappropriate for right-angled buildings as it does not preserve orthogonality (Ormsby and Mackaness, 1999). As demonstrated in Figure 2, generalisation solutions vary according to the type of building. They also vary between different key buildings; for example, the castle and station are collapsed to point symbols whereas university buildings and schools are eventually aggregated into a built-up area.

Context

Having discussed the importance of map theme and object type, it must now be acknowledged that generalisation needs to be sensitive to context. The solution must consider the neighbourhood of an object if it is to preserve the original relationships between objects. A number of mechanisms are introduced here that modify the generalisation in order to accommodate neighbouring objects. When generalising a key building object, it needs to be considered in the context of its surroundings. If it is surrounded by other buildings of the same type, the buildings are generalised as a group through aggregation. If the object is surrounded instead by general buildings, the building is instead generalised by using a previously-implemented algorithm that creates its minimum bounding rectangle, scales it to preserve the original building area, and rotates it to reflect the principle orientation of the original building.

At smaller scales, the density of the surrounding buildings should be assessed, to evaluate whether they would be better generalised as a solid built up area. A measure of the density of buildings within the city block is gained by calculating the total area of all buildings within the block and dividing by the total block area. This gives an index between 0 and 1. A cut-off density was evaluated through empirical observation. If the index is 0.35 or above, the block is plotted as a solid built-up area. Otherwise, a sub-set of the original buildings are plotted using the area patch option outlined above. As the scale reduces towards 1:50,000, only the blocks are plotted as any individual building detail becomes unidentifiable.

7.0 Implementation of a Cartographic Rule Base

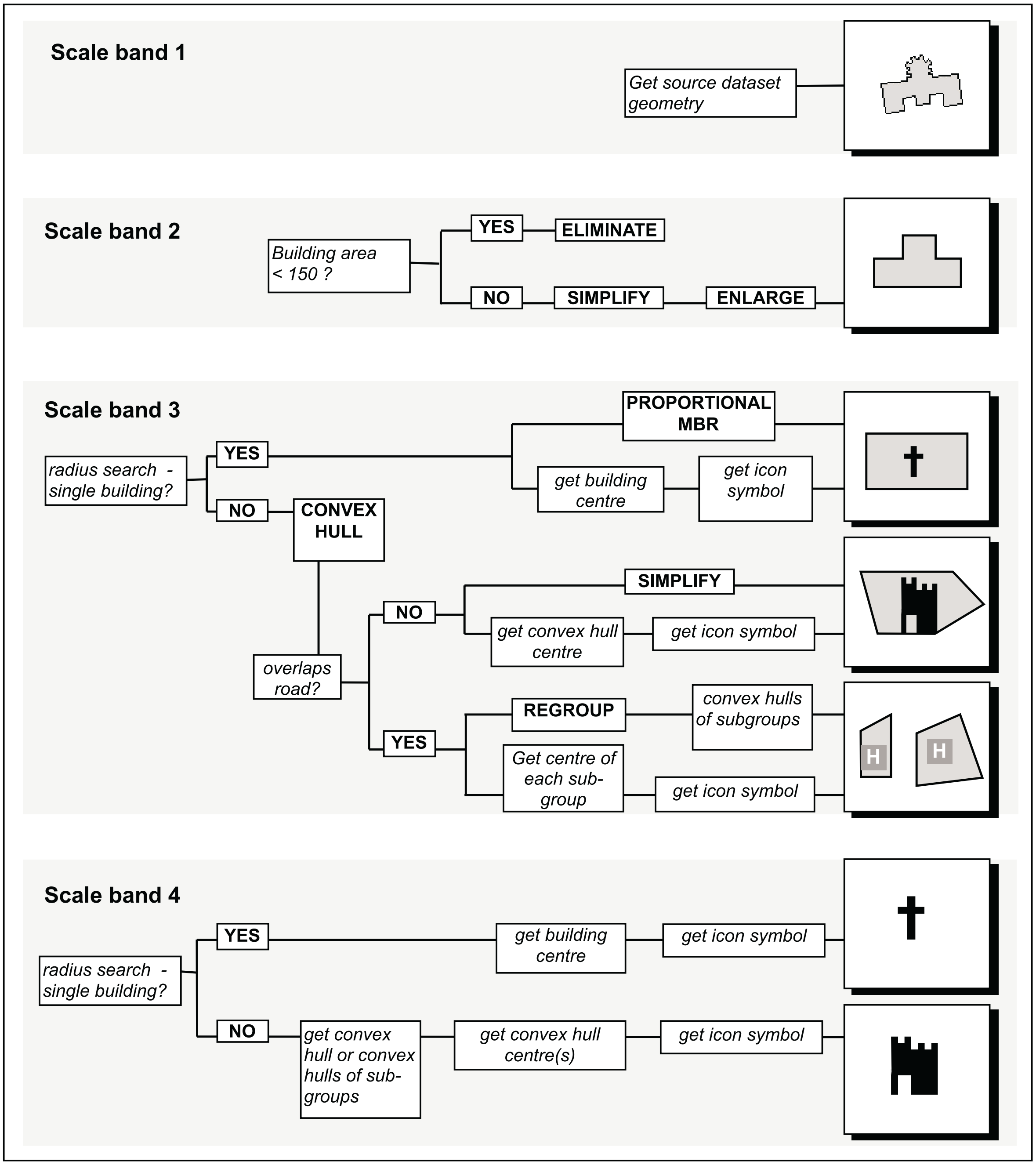


Figure 3. Decision tree for key buildings.

The creation of transitions in representation such as those outlined above requires generalisation to be performed at the object level. It also necessitates a series of decisions to be made on the basis of factors such as map type and display scale. These two requirements have meant that the implementation has been carried out using a powerful object-oriented GIS, and using a rule-based approach. An explanation of the object-oriented approach and the details of the implementation within an object-oriented environment, can be found in Glover (1998). A decision tree or simple rule-base determines which of the solutions is to be carried out, invokes the actual generalisation and plots the chosen representation. The rule-based approach to generalisation is one that has received considerable attention in the literature (see for example Nickerson and Freeman, 1986; Beard, 1991; Richardson and Muller 1991; Buttenfield 1991; Weibel 1991, Heisser et al 1995). Rules are constructed using the if-then syntax - if the condition is met, the action is triggered.

The rule base can be summarised as a three stage approach:

1. The display scale, map type and if necessary, the object class, are retrieved. Plot styles and threshold scales are set up accordingly.
2. Any other necessary analysis is then performed - for example, how large is the building? Is it detached? The appropriate generalisation solution is then retrieved.
3. The generalisation is plotted.

The decision tree used for plotting key buildings and general buildings are shown in Figures 3 and Figure 4.

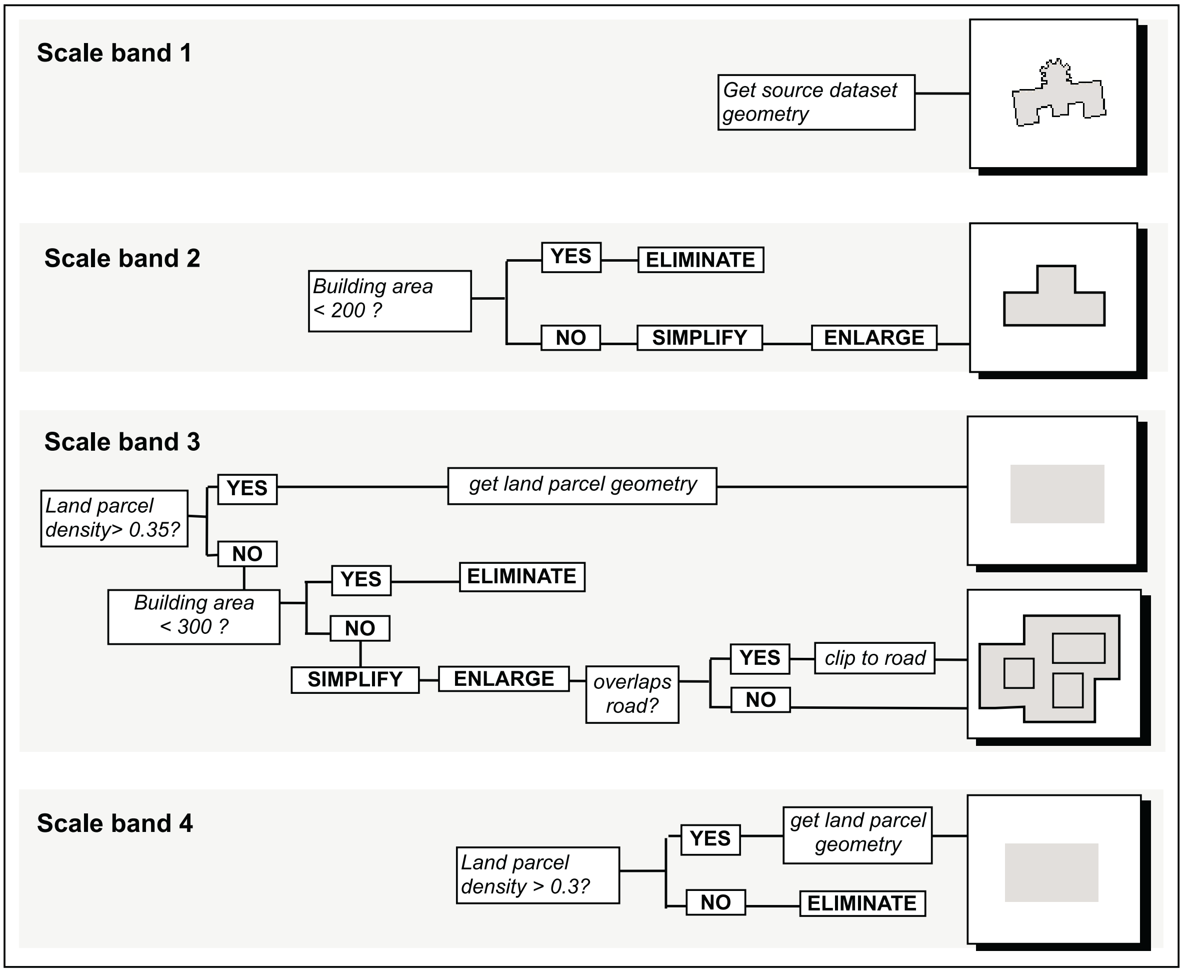


Figure 4. Decision tree for general buildings.

8.0 Results and Evaluation

Quality of output is commonly evaluated through comparison with manual generalisations. This is the approach adopted here, although it should be noted that some doubt exists as to whether automatically produced maps should attempt to imitate manual ones exactly. This is because manual solutions can be instinctive and subjective, with different cartographers producing very different results (Muller et al, 1995). Figure 5 shows derived products from a single detailed database, using the above decision trees, implemented as display methods in Laser Scan's Gothic GIS. Full descriptions of the implementation and evaluation are given in Glover (1998).

The results are encouraging, particularly those gained at greater abstraction . The majority of maps are clear and easy to read. There are however a number of areas where improvements can be made and are considered to be important components of future work.

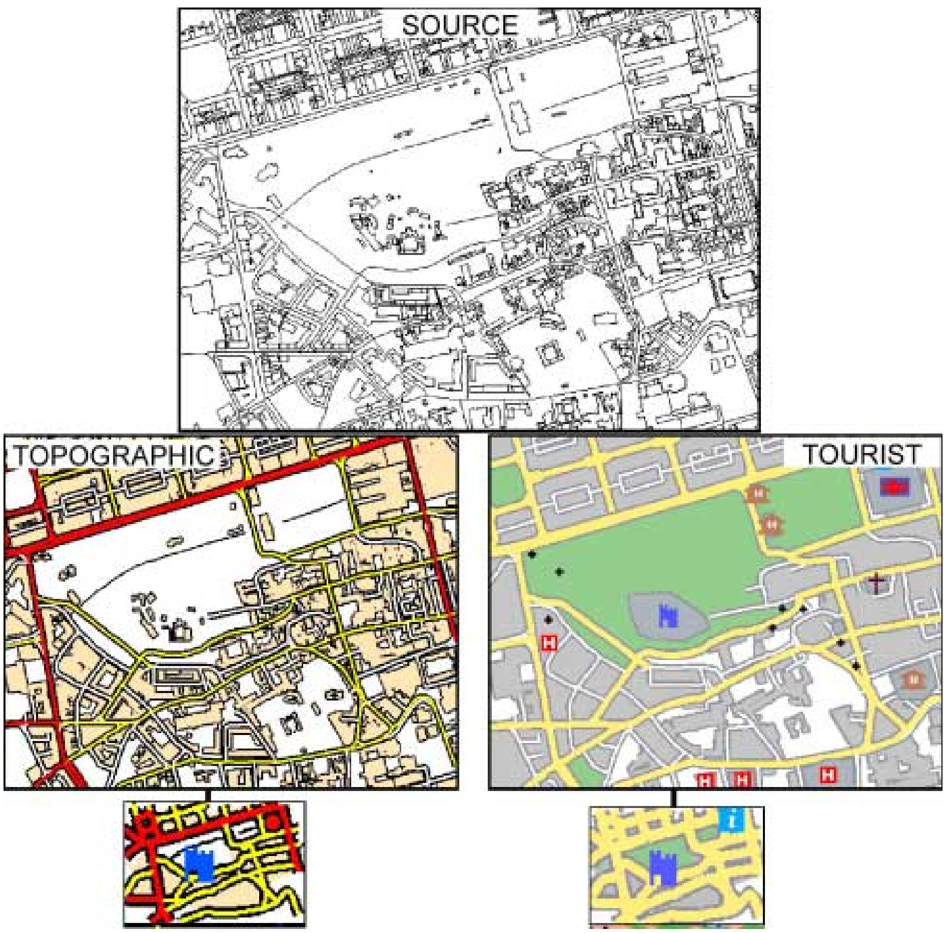


Figure 5. Different products according to theme and scale derived from the same source.

9.0 Conclusion

Through the implementation of dynamic generalisation, this study has demonstrated that effective scale- and theme- dependent virtual maps can be produced using a limited number of simple mechanisms. The most promising results are achieved at greater abstraction levels and for a more specific map theme.

In conclusion, the true potential of dynamic generalisation lies in two areas. Firstly, opportunities exist not only in its application to the derivation of scale-dependent displays, but also in its ability to produce a wide range of specific map themes, thus enabling the user to choose a map type closely matching their requirements. By narrowly defining the map theme, greater selectivity in the features displayed is possible. As the results demonstrate this will promote the production of clearer and more effective maps. Great potential also lies in the possibility of harnessing the implicit link between map theme, scale and utility, to develop a virtual map in which control over the map scale acts as a gateway to other types of map use and theme.

10.0 Acknowledgement

The authors are extremely grateful for access to source data from the National Mapping Agency of the UK, the Ordnance Survey, under the Chest Agreement and for their support for research in this field.

11.0 References

Beard, K. 1991. Constraints on rule formation, in Buttenfield, B.P. and McMaster, R.B., eds. Map generalization: making rules for knowledge representation, Longman, Essex, 121-135.Bertin, J. 1967. Semiologie graphique, Gauthier-Villars, Paris.

Buttenfield, B.P. 1991. A rule for describing line feature geometry, in Buttenfield, B.P. and McMaster, R.B., eds. Map generalization: making rules for knowledge representation, Longman, Essex, 150-171.

Glover, E. 1998. The application of dynamic generalisation to virtual map design. The University of Edinburgh MSc Dissertation Research Paper. Unpublished.

Gower, R., Pepper, J. and Edwards, T. 1997. Landplan - automated generalisation comes of age, ICA Proceedings, 126-133.

Heisser, M., Vickus, G. And Schoppmeyer, J. 1995. Rule-orientated definition of small area selection and combination steps of the generalization procedure, in Muller, J-C., Lagrange, J-P., Weibel, R. GIS and Generalization: Methodology and Practice. Taylor & Francis, London.

Jones, C.B. 1997. Geographical Information Systems and Computer Cartography, Longman, Harlow.

Lecordix, F., Plazanet, C., Lagrange, J-P. 1997. PlaGe: a platform for research in generalisation. Unpublished.

Mackaness, W.A. and Beard, K. 1993. Use of Graph theory to support map generalization. Cartography and Geographic Information Systems, 20(4), 210-221.

Mark, D.M. 1991. Object modelling and phenomenon-based generalization, in Buttenfield, B.P. and McMaster, R.B., eds. Map generalization: making rules for knowledge representation, Longman, Essex, 103-108.

Muller, J-C. 1989. Theoretical considerations for automated map generalization, ITC Journal, 3, 200-204.

Muller, J-C. and Wang, Z.S. 1992. Area-patch generalization: a competitive approach. The Cartographic Journal, 29, 137-144.

Muller, J-C., Lagrange, J-P., Weibel, R. 1995. GIS and Generalization: Methodology and Practice. Taylor & Francis, London.

Nickerson, B.G. and Freeman, H. 1986. Development of a rule-based system for automatic map generalisation, Proceedings, Second International Symposium on Spatial Data Handling, 537-56.

Ormsby, D. and Mackaness, W.A. 1999. The development of Phenomenological generalization within an object-oriented paradigm. (Unpublished).

Ormsby, D.W. 1996. The development of a phenomenological approach to cartographic generalization. The University of Edinburgh, MSc dissertation. Unpublished.

Regnauld, N. 1996. Recognition of building clusters for generalization. Proceedings of the 7th International Symposium on Spatial Data Handling, 1, 185-198.

Richardson, D.E. and Muller, J-C. 1991. Rule selection for small-scale map generalization, in Buttenfield, B.P. and McMaster, R.B., eds. Map generalization: making rules for knowledge representation, Longman, Essex, pp.

Van Oosterom, P. 1995. The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning, in Muller, J-C., Lagrange, J-P., Weibel, R. GIS and Generalization: Methodology and Practice. Taylor & Francis, London, 120-132.

Weibel, R., 1991, Amplified intelligence and rule-based systems, in McMaster, B. P. B. a. R. B., ed., Map generalization: Making rules for knowledge representation, Essex, Longman Scientific & Technical, 172-186.

Session / Séance 05-D

Exploiting parametric line description in the assessment of generalization quality

Lysandros Tsoulos

Cartography Laboratory, Department of Rural and Surveying Engineering
National Technical University of Athens
H. Polytechniou 9, 157 80 Zographou Campus, Athens, Greece
Tel: +30 +1+772-2730 Fax: +30+1+772-2734
lysandro@central.ntua.gr

Andriani Skopeliti

Cartography Laboratory, Department of Rural and Surveying Engineering
National Technical University of Athens
H. Polytechniou 9, 157 80 Zographou Campus, Athens, Greece
Tel: +30 +1+772-2639 Fax: +30+1+772-2734
askop@central.ntua.gr

Abstract

The problem of generalization quality assessment, is an important issue in current cartographic research. Quality assessment consists of the evaluation of geometry, topology, semantics and aesthetics of individual objects, locally and at the entire map level. Cartographers realize the need for quality measures, since the lack of techniques for controlling generalization, poses another obstacle towards its "automation". This can be achieved through the comparison of parameters describing the original and the corresponding generalized objects. This paper focuses on the geometric aspect of the maps and especially on the treatment of linear objects. Each linear object is being classified in a category according to its shape. This classification has been based on earlier work of the authors and it utilizes a number of measures describing linear objects' shape. The evaluation of parameters describing the linear objects shape, before and after the generalization, results not only to the selection of the proper generalization approach, but to the consistency of the generalization applied throughout a map or a map series. It also results to knowledge acquisition, which is indispensable for the automation of generalization within a hybrid environment, consisting of a cartographic and an expert system. Methods for better reporting the generalization results are elaborated along with the ways the reported information should be handled by the cartographer, towards the selection of the most suitable generalization solution.

Introduction

Positional accuracy is one of the most important components in spatial data quality assessment [Guptill. and Morrison, 1995]. The overall positional accuracy of the map, is the result of a number of transformations applied throughout the map production process (data collection, cartographic generalization, cartographic projections etc.). Each one of these transformations has its own influence on the accuracy of the map and if modeled and measured, would enable the cartographer to control the amount of error introduced.

Researches have pointed out that positional error due to data collection depends on the shape of the cartographic line [Amrhein and Griffith, 1991; Keeper et al., 1988; Openshaw and Brundson, 1993]. The successful classification of cartographic lines according to shape can lead to the retroactive assessment of positional data collection error for each feature. As a result, each feature will be stored in the database with metadata information describing its positional accuracy. At the same time, the cartographer's decisions during generalization are strongly influenced by line shape. As it has been pointed out, the cartographic generalization of linear features will be successfully implemented, if it is based on the knowledge of the line shape. This will also lead to the realistic estimation of generalization error [Buttenfield, 1991; Plazanet et al., 1995; Reichenbacher, 1995; Weibel et al., 1995]. If the identification and quantification of the parameters describing the cartographic line is achieved, an important step towards the estimation of the digitization and generalization error, will have been done.

Parametric description of line shape

The visual effect of the shape of the cartographic lines on the map reader along with the relevant line symbol, provoke the recognition of the entity which they represent. In addition, the map reader can characterize the line shape as "complex", "maiandric" or "straight". "Knowledge" about line features stored in geographic data bases, can be acquired through the evaluation of the parameters that control their generating processes (aspect, climate, geological factors etc.) [Ammstrong, 1991], or through an algorithmic approach with computations performed on the coordinates and the geometry describing those features [Ammstrong, 1991; Buttenfield 1991]. The assessment of the shape of a cartographic line, through computations performed on its building components, constitutes a basic research trend in contemporary Cartography. Results mainly focus on the identification of measures describing the line shape qualitatively. Numerous approaches to line shape description have been proposed [Affholder, 1993; Bernhardt, 1992; Boutoura, 1989; Buttenfield, 1991; Freeman, 1978; Fritsch and Lagrange, 1995; Jasinski, 1979; Jenks, 1979; Mandelbrot, 1967; McMaster, 1986; Plazanet et al.; 1995, Ruas and Lagrange, 1995; Thapa, 1988].

Research has been also conducted by the authors [Skopeliti & Tsoulos, 1999] for the development of a methodology for parametric description of line character. A wide range of parameters were cross-examined and evaluated. Amongst them the fractal dimension, the mean angularity and its variance, the mean magnitude angularity and its variance, the curvilinearity ratio for one level of resolution, the angularity and curvilinearity ratio plot at different vertex intervals (ranges), bandwidth, segmentation, error variance concurrence and the ratio of the line length and anchor line length were evaluated.

The fundamental concept of the methodology developed, is the calculation of measures at different resolution levels. This concept has been adopted by a considerable number of researchers following different approaches for line shape description [Bernhardt, 1992; Buttenfield, 1991; Mandelbrot, 1967; Moktharian and Mackworth, 1986; Plazanet et al.,1995]. This common approach implies an attempt to work independently from the scale factor and to reveal information about line character which cannot be derived by a comparison of parameters measured at a certain level of detail/resolution. In addition the lines are pre-processed in order to acquire a common resolution [Plazanet et. al., 1995; Thapa, 1988] before the calculation of angular parameters. When the lines are represented by equally spaced vertices, they are recorded in a consistent way for every data set. Results are comparable with no bias due to the equal spacing of vertices [Carstensen, 1990].

After analysis and experimentation, four parameters were selected: **the fractal dimension, the average angularity magnitude plot at different vertex intervals, the ratio of the line length and anchor line length and error variance**. Furthermore, it was proved that once these parameters are calculated for line segments, cluster analysis could be used for the classification of lines with similar characteristics. The results of the cluster analysis conducted in the experiment, confirm the ability of the selected measures to describe the line. The methodology for clustering any data set is implemented in three stages:

a. calculation of the proposed set of measures for all the lines in the data set
b. application of hierarchical cluster analysis, identification of the number of groups that exist in the data set based on the calculation of certain statistical indices [Sharma, 1996], calculation of the cluster centers
c. non-hierarchical clustering with the use of the identified cluster centers

It is noted that this approach will always provide a solution, assuming that line segments are homogeneous along their extent. Line clustering is performed on a specific dataset and as a result, lines are grouped with the more similar ones in order to be handled in the same way in the cartographic processes.

Line partitioning in homogeneous segments

Fractal dimension is considered most appropriate for line segmentation because it has certain advantages over the others. Along with its value, an indicator of its ability to describe line character, is provided. In addition to this, fractal dimension is a global parameter by definition (since it measures line length at different levels of resolution), whereas other parameters like average angularity, summarize local measures. Basic advantage of this method is that it always provides a solution to the problem of line segmentation in homogeneous parts, minimizing the cartographer's intervention. Another characteristic of the method is that it is based on the identification of self - similar segments for every vertex of the line, which ends up with a detailed and continuous observation of line character. In addition, any changes detected are not local, because they are derived from a global parameter.

Cartographic line segmentation to homogeneous segments, is based on the identification of all the self – similar segments constituting the cartographic line and the selection of those which partition the line in consecutive segments having different character. Self - similar segments are identified through the "Moving Endpoint Technique".

According to this technique, the first and the last vertex of the line segment are variable. The first vertex of the line is selected as a starting point and the goal is the identification of an endpoint that forms a self-similar segment. This is repeated starting with the second point and so on. The process is repeated for every line vertex. The locations of starting and ending points are concentrated along the line. In order to select a number of segments for line partitioning, only the most statistically reliable ones can be selected. This would result into distracting segment continuity. Since all calculations are statistically strong, it is more important to ensure that the selected segments will be continuous and will cover the whole extent of the cartographic line. The line segments resulting from this technique are numerous, something that complicates their management and increases the required computational burden.

An alternative solution would be the creation of a graph showing all segments. The x-axis represents the vertex code and depicts the starting and ending nodes of the self-similar segment. The y-axis depicts the angle of the regression line fitted to the Richardson plot. A self-similar segment is plotted in this graph as a line from (nstart, w) to (nend, w). This graph will be called from now on "Fractal segments plot" (see Figure 1). In order to achieve a homogeneous scale on both axes, regression line angle is multiplied by 1000. From this plot, self - similar segment distribution along the line and concentrations into position and regression line angle, can be observed. In order to identify areas where segments sharing a common fractal character (similar regression line angle values w) are located (starting code similar values), cluster analysis is utilized using as input line segments' starting point and regression lines angle values. At first place, hierarchical cluster analysis is applied, but the number of clusters in the data set has to be decided. Cluster analysis can be repeated to obtain a solution for a different number of clusters and use special statistics such as the overall R^2 and the ratio of the within RMSSTD to the total RMSSTD [Sharma, 1996] to evaluate the solution. The overall R^2 should be high, indicating that the clusters are homogeneous and well separated and the RMSSTD ratio should be low indicating that the resulting clusters are homogeneous. When the number of clusters is decided, the centers of the hierarchical clusters are calculated and non- hierarchical clustering is applied to the data. Non-hierarchical clustering leads to the final classification and its results are used for the identification of the segmentation vertices.

Initially, the segmentation vertices were considered as identical to the cluster centers. However, the plotting of the cluster centers on the "Fractal segments plot" shows that this is not an acceptable assumption and does dot lead to proper selection. A better solution is achieved when the nodes of the segment are calculated by subtracting the standard deviation of the starting vertices of the cluster from the average starting point and adding the standard deviation of the ending vertices of the cluster to the average ending point. These line segments are not continuous, since self - similar segments do not exist throughout the line extent. Thus intermediate segments are also added to the segment list in order to acquire a continuous segmentation.

Cartographic generalization and quality

It is evident that generalization introduces error, but it is important to investigate the magnitude of this error. Generalization may have unpredictable effects on the metrics, topological and semantic accuracy of map products. Each generalization operator influences one or more of data quality components. Displacement influences positional accuracy, completeness will be affected by selection and merging operations; some attributes may be lost through reclassification, consistency may be affected by uneven application of spatial or temporal abstraction [Muller et. al., 1995]. However, it is difficult to appreciate which generalization effects are important and which ones are not, as the importance of these errors depends to a certain degree on the intended application [Openshaw, 1989].

In the ICA workshop on Map Generalization in 1997, it was proposed to conduct quality assessment in different levels (individual objects, "situations" and entire map products) in terms of geometry-topology, semantics and aesthetics [Mackaness et al., 1997]. It was also stated that proper assessment techniques must focus on constraints associated with the content. Following these guidelines, a case study is formed with linear entities generalized for a topographic map, examined at the individual entities level. For a topographic map, geometric constraints take a high priority along with conservation of line character.

Constraint violation is related to the generalization operators used. For the generalization of individual linear features, generalization operators such as simplification, enhancement and smoothing are used, affecting metric and geometric characteristics of these features. They change the coordinates, which describe position and thus positional accuracy, the shape and basic characteristics such as length. The change in horizontal position can be measured by displacement measures like the ones introduced by McMaster (1987), Jenks (1989) etc. The change of line shape can be evaluated by calculating measures that describe line character. The methodology for parametric description of line character will be utilized for that cause. In addition to the above operators, displacement is also used to avoid coalescence of adjacent entities. Vector of displacement must be recorded since it is a pure measure of horizontal position change.

In the following experiment, the contribution of line segmentation in homogeneous parts in cartographic generalization results will be evaluated from two points of view: the preservation of line shape and positional accuracy.

Cartographer's goal is the minimization of generalization error. Automatic generalization can only be implemented with the utilization of cartographic knowledge. When the choice of algorithms and parameters leads to an acceptable solution, knowledge regarding "how to perform" cartographic generalization can be extracted by associating information regarding original line shape, generalized line shape, algorithm and parameter values used, influence in positional accuracy. Thus this experiment contributes in the identification of measures for reporting cartographic generalization quality and the development of a procedure for cartographic knowledge acquisition.

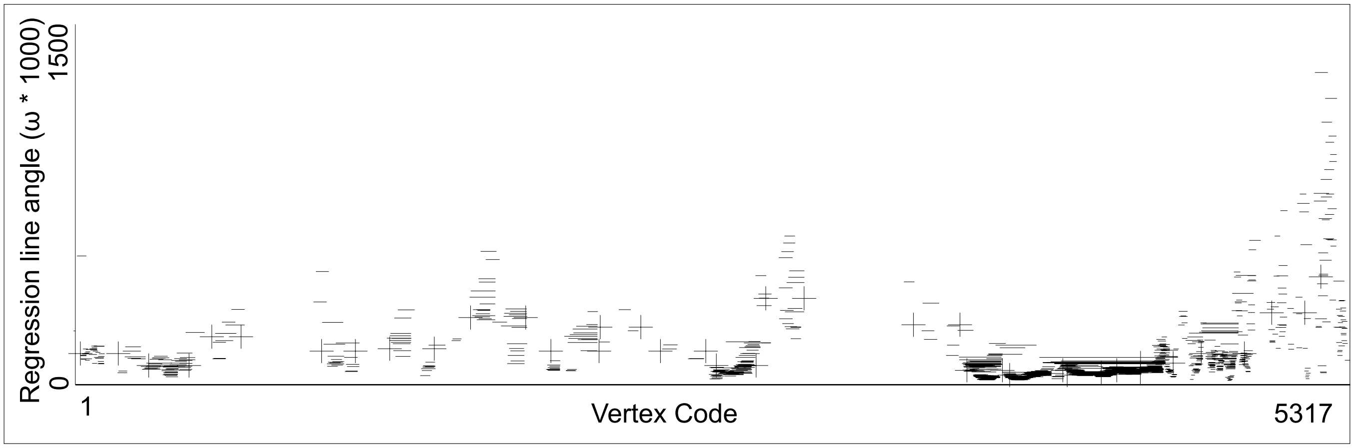


Figure 1. Self-similar segments identified on Lefkada island. Segments' nodes calculated by cluster analysis are also plotted

Experimental analysis

The coastline of the Greek island Lefkada, was used for the implementation of the above described methodology. It was selected because of the variability of its line character (see Figure 2). Data was digitized from a 1:100 000 scale map and the average distance between vertices is about 25 m.

The moving end point technique was utilized for the identification of all the self - similar segments on the coastline. The results were used in the hierarchical cluster analysis. Trials lead to a 20-cluster solution that satisfies the statistical criteria. The total R^2 (0.998) is high and the RMSSTD ratio (0.183) is quite low, suggesting that the clusters are quite homogeneous and well separated. Segmentation vertices were eliminated, if too small segments were formed.

Linear segments shape description and clustering

When the line is segmented, the resulting segments will be classified into groups according to their shape. Line shape can be described by four parameters: statistical measures of the average angularity magnitude plot at different vertex intervals, fractal dimension, the ratio of the line length and the anchor line length and error variance [Skopeliti & Tsoulos, 1999]. The values of these parameters are used to perform hierarchical cluster analysis. From the dendogram (see Figure 3) it is apparent that the line used in the experimental analysis can be grouped into three or seven clusters. Statistical values for the total R^2 (0.975) and the ratio of within RMSSTD and total RMSSTD (0.407), prove that the seven-cluster classification leads to a much better solution. When the number of clusters is identified, non-hierarchical clustering is performed utilizing the centers of hierarchical clustering. Alternative cluster solutions are presented in Table 1, where the common shape characteristics of the segments classified in the same group, are described. Although

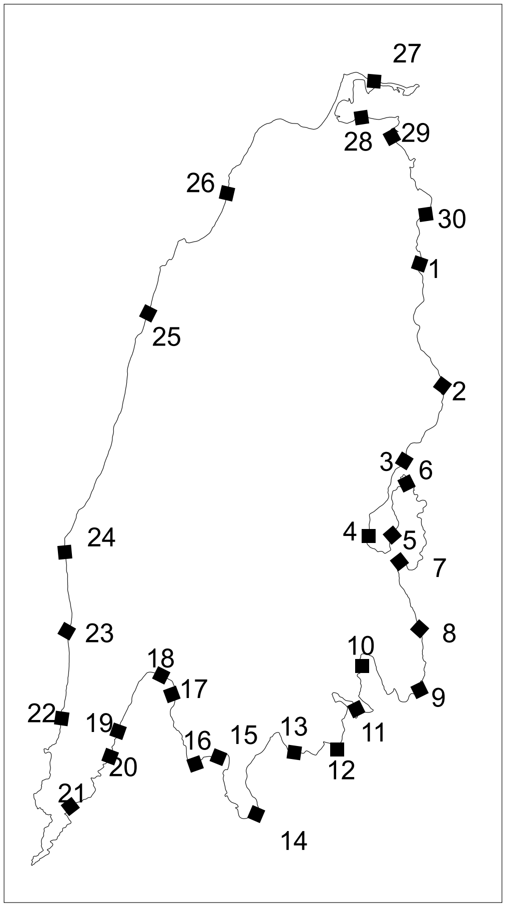


Figure 2. Segmentation results

the three-cluster solution is rejected, the way the seven-clusters are grouped to form the three clusters, indicates their inter-relation. This information is valuable when segments transition among clusters is observed after cartographic generalization. The same information can be derived through the calculation of distances among cluster centers.

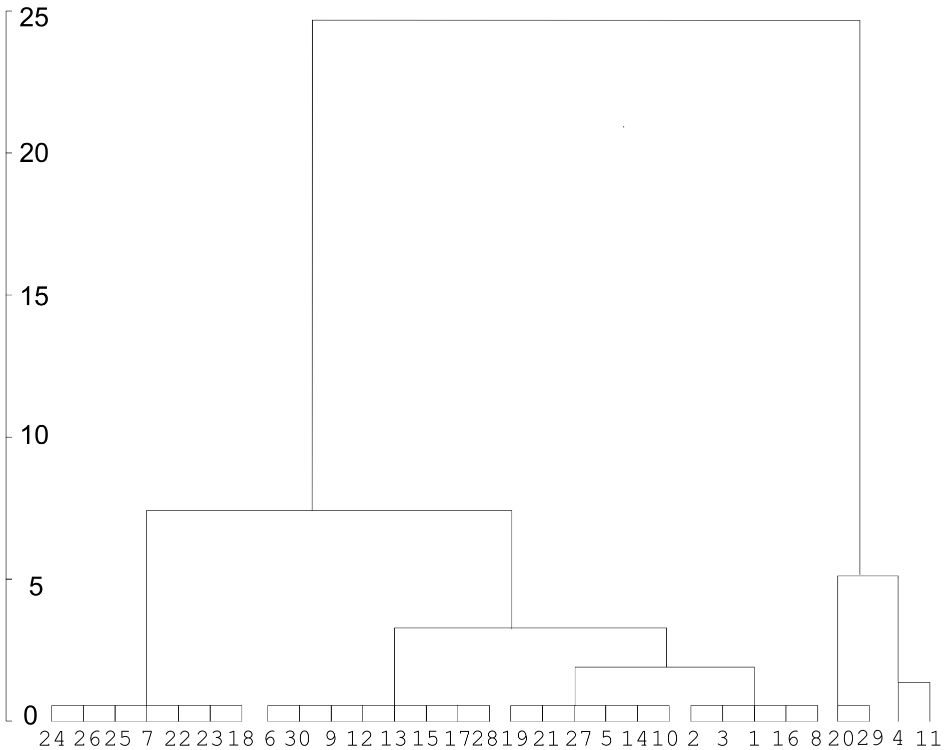


Figure 3. Hierarchical cluster analysis dendogram of linear segments

Table 1. Qualitative description of segments clusters

3 clusters	*7 clusters*	*Cluster members*	*Line shape characteristics*
A	1	1,2,3, 8, 16	Sinuous
	3	5, 10, 14, 19, 21, 27	Moderately sinuous
	4	6, 9, 12, 13, 15, 17, 28, 30	Very sinuous
B	2	4	Sinuous with a strong arc
	6	11	Moderately sinuous with a strong
	7	20, 29	Very sinuous with a strong arc
C	5	7, 18, 22, 23, 24, 25, 26	Smooth

Evaluation of generalization results

The dataset was generalized with the Douglas Peucker [Douglas and Peucker, 1973] algorithm and a module used in ARC/INFO, called "bend-simplify". This module applies shape recognition techniques that detect bends, analyze bend characteristics and eliminate insignificant bends from the original line. The resulting line resembles the original and displays better cartographic qualities. [ESRI, 1998]. In accordance with the cartographic practice, generalized data should have a resolution (threshold of separation - minimum allowable spacing between graphic elements) equal to 0.25 mm in the new scale. The original scale of the dataset used in the experimental analysis is 1:100000 and the scale of the generalized dataset is 1:250000. Parameters used by the generalization algorithms will be equal to 62.5 meters.

Generalized lines shape

Parameters describing line shape are calculated for the generalized versions of lines resulting from the above algorithms. The values of these parameters (statistical measures of the average angularity magnitude plot at different vertex intervals, fractal dimension, the ratio of the line length and the anchor line length and error variance) will be submitted to cluster analysis in order to examine changes in line shape. Table 2 shows the results of the cluster analysis of the generalized lines with the Douglas-Peucker algorithm. It is noted that line segments are again classified into seven groups and that changes at the membership composition appear only in two (3, 4) of them. This change is not considerable, because groups 3 and 4 belong into the same group of sinuous lines. The preservation of clusters combination means that line segments belonging to the same group, undergo the same shape modification, whereas changes in group 4 members, is an indication that these lines should have been generalized with a different parameter value. The same remarks are valid for the other generalization method although this method gives better results, since it is takes into consideration, line shape.

In order to estimate the influence on line shape, the centers of the clusters of the generalized lines are compared with those of the original segments. Calculating the difference for each parameter value or the distance between them, can compare two centers. As it is expected, distances for the lines generalized by the “bend simplify” method are very small since this algorithm preserves line shape. The different values of distance between clusters, indicate that different line groups should have been generalized with the application of different parameter values in order to acquire similar characteristics. These results can be utilized in the selection of the appropriate parameter values for a line with certain character, the scale and the algorithm. However, in order to get to such conclusions, experimentation with more lines, algorithms and scale variation is needed. It is therefore estimated that this method can contribute in the extraction of knowledge for cartographic generalization.

Table 2. Cluster analysis results (original line segments vs. generalized segments)

Original lines		*Douglas Peucker algorithm*		*Bend simplify algorithm*	
Cluster members		*Cluster members*	*Distance*	*Cluster members*	*Distance*
1	1, 2, 3, 8, 16	1, 2, 3, 8, 16,	3.54	1, 3, 8, 16	2.02
3	5, 10, 14, 19, 21, 27	5, 6, 9, 10, 12, 14, 15, 19, 21, 27	3.51	2, 5, 10, 14, 19, 21, 27	1.41
4	6, 9 ,12, 13, 15, 17, 28, 30	13, 17, 28	8.86	6, 9, 12, 13, 15, 17, 28,	0.05
2	4	4	4.55	4	0.02
6	11	11	6.60	11	0.01
7	20, 29	20, 29	6.08	20, 29	0.19
5	7, 18, 22, 23, 24, 25 ,26	7, 18, 22, 23, 24, 25, 26	2.57	7, 18, 22, 23, 24, 25, 26	0.03

Positional accuracy

The evaluation of positional accuracy is the second important factor in the estimation of generalization results quality. Positional accuracy can be evaluated by computing the displacement of generalized lines in relation with the original ones and the area between them. Since all lines are generalized with the same parameters and values, the displacements are similar. Mean displacement from the original lines is smaller than the threshold accuracy of 0.25 mm of the scale and is considered acceptable. This is the result of the selection of a proper value for the generalization taking into account the target scale. If each line segment were generalized with different parameter value, it is expected to have a different areal and vector displacement. In this way, information on positional accuracy can be stored for each segment. It will be examined whether line segmentation based on line shape, ameliorates positional accuracy compared to random line segmentation. Generalization is performed to the thirty (30) line segments resulting from segmentation according to line shape and to the two (2) line segments resulting from random line segmentation. Area and displacement value is calculated for the thirty (30) original line segments against the two sets of generalization results. Mean difference in area is smaller in the first case and mean displacement is almost the same. Area and displacement decreases for the 67% of the thirty (30) segments.

Table 3. Unknown switch argument.. Area difference and vector displacement.

Measure	*2 segments*	*30 segments*
Mean difference in area (m^2)	82533.16	75116.03
Mean vector displacement (m)	16.53	16.70

Conclusions

From the above experiment it is obvious that the methodology for parametric description of line character and line segmentation can be very useful in many ways. Cartographic knowledge on generalization in the form of rules can be extracted by the combination of parametric line character description and generalization algorithm parameter values. Different generalization parameter values are applied for each group of segments formed by cluster analysis. Lines with similar character are generalized in a consistent way. Through the comparison of the cluster analysis results of the original and the generalized segments, the changes in line character are identified. It is realized that there are not yet any criteria for the quantification of that change. However, the change of cluster centroids is an absolute measure of line shape change that may, in combination with positional generalization error, comprise a set of objective criteria for judging the suitability of an alternative generalization solution.

However, in order to predicate the above, extended experimental work should be conducted. The methodology must be applied to a wide dataset e.g. more coastlines, different linear entities etc. Different generalization algorithms can be utilized to study the changes in line shape and extract knowledge about their impact. Implementation of different algorithms should also cover a wide range of scales.

References

Affholder, J. G. (1993). Road modeling for generalization. *Proceedings of the NCGIA Initiative 8 specialist Meeting on Formalizing Cartographic Knowledge*, Buffalo, 23-36.

Ammstrong, M. P. (1991). Knowledge classification and organization. In B. Buttenfield and R. McMaster (Eds.). *Map Generalization: Making rules for knowledge representation*. Halow, Essex, U.K., Longman Scientific.

Amrhein, C. G., and Griffith, D. A. (1991). A statistical model for analyzing error in geographic data in an information system. Discussion Paper No 38. University of Toronto, Toronto, Ontario.

Bernhardt, M. C. (1992). Quantitative characterization of cartographic lines for generalization. Report No. 425, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio.

Boutoura, C. (1989). Line generalization using spectral techniques. *Cartographica*, 26(3&4), 33-48.

Buttenfield, B. (1991). A rule for describing line feature geometry. In B. Buttenfield and R. McMaster (Eds.). *Map Generalization: Making rules for knowledge representation*. Halow, Essex, U.K.: Longman Scientific.

Carstensen, L. W. (1990). Angularity and Capture of the Cartographic Line During Digital Data Entry. *Cartography and Geographic Information Systems*, 17(3), 209-24.

Douglas, D., and Peucker, T. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, 10, 112-22.

ESRI (1998). ARC Commands. Environmental Systems Research Institute, Inc., Redlands, California, USA

Freeman, H. (1978). Shape description via the use of critical points. *Pattern Recognition*, 10, 159-166

Fritsch, E., and Lagrange J. P. (1995). Spectral representations of linear features for generalization. *Proceedings of COSIT '95*, Semmering, Austria, 157-171.

Guptill, S. C. and . Morrison, J. L. (Eds). (1995). *Elements of spatial data quality*. International Cartographic Association, Elsevier Science: Oxford and Tarrytown NY

Jasinski, M. J. (1990). The comparison of complexity measures for cartographic lines. *NCGIA Report 90-1*, NCGIA, Santa Barbara, CA.

Jenks, G. F. (1979). Identification of characteristic points along naturally occurring lines: An empirical study. *The Canadian Cartographer* 16(1), 70-80.

Jenks, G. F. (1989). Geographic logic in line generalization. *Cartographica*, 26(1), 27-42.

Keefer, B. G., Smith, J. L., and Gregoire, T. G. (1988). Simulating manual digitizing error with statistical models. *Proceedings of GIS / LIS '88*, Sant Antonio, 475-483

Mackaness, W., Weibel, R., and Buttenfield, B. (1997). *Report on the ICA Workshop on Map Generalization*, Gavle, Sweeden

Mandelbrot, B. (1967). How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science*, 156, 636-638

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

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

Mokth arian, F., and Mackworth, A. K. (1986). Scale-based description and recognition of planar curves and two dimensional curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8), 34-43

Muller, J. C., Weibel, R., Lagrange, J. P., and Salge, F. (1995). Generalization: state of the art and issues. In Muller, J.C., Lagrange, J.-P., and Weibel, R. (Eds.). *GIS and generalization: Methodology and Practice*. London, Taylor and Francis.

Openshaw, S. (1989). Learning to live with errors in spatial databases, in Goodchild, M., Gopal, S. (Eds.). *Accuracy of spatial database*, London: Taylor & Francis.

Openshaw, S., and Brundson, C. F. (1993). Simulating the effect of error in GIS. In Mather, P. (Eds.). *Geographic Information Handling*. John Wiley and Sons.

Plazanet, C. (1995). Measurements, characterization and classification for automated linear features generalization. *Auto-Carto* 12, 59-68.

Plazanet, C., Affholder, J. G., and Fritsh, E. (1995). The importance of geometric modeling in linear features generalization. *Cartography and Geographic Information Systems*, 22(4), 291-305.

Reichenbacher, T. (1995). Knowledge acquisition in map generalization using interactive systems and machine learning. *Proceedings ICA '95*, Barcelona, Spain, 2221-2230.

Ruas, A., and Lagrange, J. P. (1995). Data knowledge modeling. In Muller, J.C., Lagrange, and J.-P., Weibel, R. (Eds.) *GIS and generalization: Methodology and Practice*. London, Taylor and Francis., 73-90.

Skopeliti, A., and Tsoulos, L. (1999). Parametric Description of the Shape of Cartographic Line, *Technika Chronika Scientific Journal Of the TCG I*, (in press).

Sharma, S. (1996). *Applied Multivariate Techniques*. John Wiley & Sons.

Thapa, K. (1988). Automatic Line Generalization Using Zero Crossings. *Photogrammetric Engineering and Remote Sensing*, 54(4), 511-517.

Weibel, R., Keller, S., and Reichenbacher, T. (1995). Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive systems and computational intelligence. *Proceedings of International Conference COSIT '95*, Semmering, Austria, 139-156.

Session / Séance 11-B

Generalization processes in numerical cartography : analysis of landscape patterns using tools of characterization

Valérie Caldairou

Laboratoire de Télédétection et de Cartographie Numérique
Ecole Supérieure d'Agriculture de Purpan
75 voie du TOEC, 31076 Toulouse cedex 3 - France

François Blasco

Laboratoire d'Ecologie Terrestre
BP 4403 - 13, Avenue du Colonel Roche, 31405 Toulouse cedex 4 - France
Email : blasco.let@cesbio.cnes.fr

Michel Gay

Laboratoire de Télédétection et de Cartographie Numérique
Ecole Supérieure d'Agriculture de Purpan
75 voie du TOEC, 31076 Toulouse cedex 3 - France
Email : gay@esa-purpan.fr

Abstract

Generalization processes are commonly applied in cartography in order to model the geographical space. The analysis of the loss of the detailed information during generalization processes remains an important, essential problem. The objective lies in preserving the sense of the initial cartographic message. The test site (the Lannemezan plateau – Northern side of the Pyrénées) has a particular distribution of the landuse classes : forests and crop areas related to elements of topography.

Both thematic approaches were taken into account : landuse classes (supervised classification SPOT/HRV) and landscape patterns (landuse classes and slope levels). From a certain level of generalization and whatever the used process in this study (structural generalization or categorical amalgamation), we realized that the cartographic concept of landuse class defined at the pixel detail has no significance anymore, then the limits of this type of analysis are reached. But the analysis of the spatial organization of a territory through the concept of landscape pattern includes an other type of perception taking into account the pixel in its neighbourhood, which seems more adapted.

1. Introduction: central problem

Numeric generalization processes can be defined by the focus on the essential information and, consequently, modify cartographically and geographically the information organization of the image. These transformations are necessary. The loss of the cartographic accuracy in the image is done for the benefit of a greater clarity in the reading of the map (Plazanet, 1996). The cartographic message contained in the map should not be distorted too much in order to be always decoded by the reader.

The methodology used in this study is based on the definition of the limits concerning the cartographic transformations after application of generalization processes. Statistical, cartographic and form index criteria have been taken into account for highlighting this loss of coherence of the spatial structure through both thematic approaches :

- landuse classes obtained by classification of SPOT/HRV images
- landscape patterns composed of landuse data and ancillary data with Digital Elevation Model (DEM).

2. Materials and used methodology

2.1 Generalization processes

2.1.1 Structural generalization (Table 1)

This process carries out mainly a change in the spatial resolution based on the modal value of the pixels taken into account (Woodcock & Strahler, 1987).

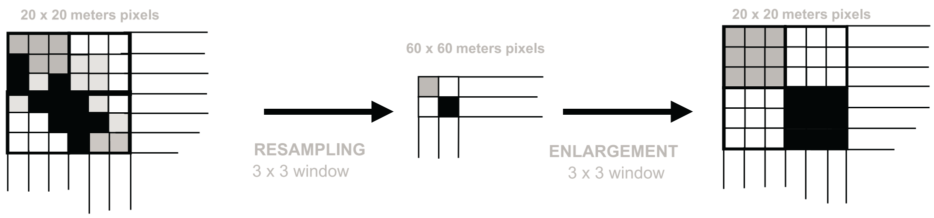


Table 1 : Description of the structural generalization process (example with 3 x 3 window)

- It results in the reduction of the number of pixels in the new images according to the factor selected (resampling step).
- In order to compare theses generalized images one to another, they have been enlarged to the original image size : 850 x 1020 pixels (enlargement step).

There are 12 generalized images successively from a spatial resolution of 20 meters (SPOT/HRV) to 300 meters. This process has been applied on the SPOT/HRV classified image.

2.1.2 Categorical amalgamation (Table 2)

This process is based on Wilkinson's work (1993). The algorithm he developed, keeps a constant statistical dispersion of the different landuse classes in a rasterized image. The objectives of this method are the preservation of this statistical distribution and at the same time the regrouping of classes in more homogeneous clusters in order to facilitate their vectorization. This algorithm is made up of both stages :

- iterative majority filter
- iterative reduced growing classes (IRGC).

12 images with a level of increasing value of regrouping were taken into account.

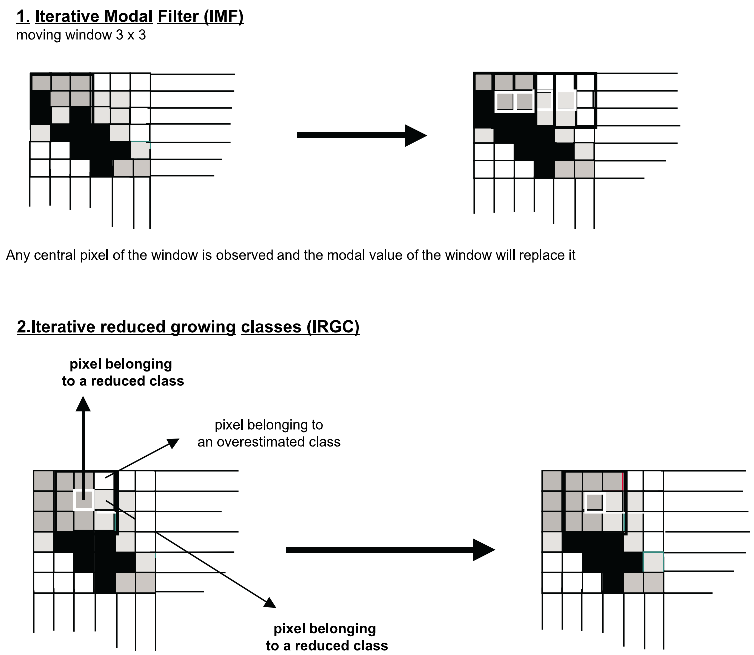


Table 2 : Description of the categorical amalgamation process

2.2 Presentation of the test site and available data

The site of the Lannemezan plateau is located on the northern side of the Pyrénées. The geomorphologic situation of the valleys in fan-shaped is very characteristic. Generally, the shallow forms of the relief (between 100 and 120 meters unlevelled) facilitate works of image processing and mapping analysis. The strip-shaped forests are mainly located on western and top hillsides when agricultural fields with regular and structured forms are located at the bottom and on eastern hillsides.

The SPOT/HRV data used are acquired in spring and summer 1989 (KJ 40/263 April 25th and KJ 41/263 August 9th). The test site is an extract of these SPOT/HRV images (850 X 1020 pixels, i.e. approximately 350 km²).

The Digital Elevation Model (DEM), based on IGN topographic maps was digitized at the digital mapping laboratory of Purpan. Its spatial resolution is 20 meters and the altitude step is 10 meters.

2.3 Description of the thematic classes

2.3.1 Classified image from SPOT/HRV images (landuse classes)

The supervised classification was established with 8 landuse classes with an unclassified pixel class (pixels belonging statistically to any definite class) (Table 3). A number of references was adapted for each class (Congalton, 1991).

For a better distinction, crops were gathered by type : **winter crops** with cereals (wheat, barley), oil seed rape and **summer crops** with corn, sunflower and soya bean. **Grasslands** represent any temporary or permanent grass surface. The **forest** class gathers deciduous and coniferous. And **fallows** are zones of natural vegetation at a shrubby stage with brambles (Rubus saxatilis), junipers (Juniperus communis) and ferns (Polypodium vulgare). Water bodies and urban classes have very small surfaces on the site and are not taken into consideration for the continuation of the image processing operations.

Table 3: statistical distribution of the landuse classes

	Landuse Classes	Pixel number	%
class 1	Winter crops	51 514	5,94
class 2	Summer crops	127 490	14,71
class 3	Grasslands	312 148	36,00
class 4	Forests	273 597	31,56
class 5	Fallows	54 695	6,31
class 6	Urban	242	0,03
class 7	Water bodies	19	0,00
class 8	Non classified	47 295	5,45
	TOTAL	867 000	100,00

2.3.2 The characterization of the landscape patterns and their classification with CLAPAS (Robbez-Masson, 1994)

The spatial structure of the considered landscape patterns represents the arrangement of biotic and abiotic elements the ones to the others. A certain organization can be detected in the distribution of the landscape for which seeks to understand the spatial order (Baudry, 1986 ; Deffontaines, 1986 ; Claval, 1993 ; Robbez-Masson & Al, 1996). This design of landscape patterns and their composition can be more or less heterogeneous according to parameters such as density or diversity.

The analysis of these patterns is composed by three principles (Men, 1996):

a) **the extraction, the qualification** of the landscape patterns by ground survey criteria,

b) **the cartographic characterization** of the patterns through the combinations of the digital thematic maps,

c) **the classification** of the landscape patterns with CLAPAS (Landscape classification and segmentation) (Robbez-Masson, 1994).

a) With ground surveys, seven landscape patterns were identified :

- pattern 1: crops with some grassland parcels on flat to weak slopes (0 to 5°)
- pattern 2: crops and grasslands on flat to weak slopes (0 to 5°)
- pattern 3: grasslands with some crop parcels on flat to weak slopes (0 to 5°)
- pattern 4: grasslands with some forests on middle to steep slopes (6 to 37°)
- pattern 5: fallows and forests on middle to steep slopes (6 to 37°)
- pattern 6: forest clumps on flat to weak slopes (0 to 5°)
- pattern 7: forest clumps on middle to steep slopes (6 to 37°)

b) The landuse image (6 classes without urban and water bodies classes) and the slope image (4 levels: from flat to steep slope (up to 37°)) have been crossed. Then, the 7 reference landscape patterns detailed above were cartographically characterized. The reference landscapes are defined by outlining typical training areas directly on the crossed image. These values are translated into **composition vectors,** i.e. a histogram with frequencies of the training areas values. There is a single composition vector for each landscape pattern.

c) Finally CLAPAS carries out the classification of this crossed image by taking into account any pixel in its neighbourhood (with windows of odd variable sizes (e.g. : 3 x 3, 5 x 5, etc…). CLAPAS always uses the 7 composition vectors (stored in a file) for comparing them to each pixel it is processing and the composition of its neighbourhood (Table 4). The operation of classification with CLAPAS is a distance calculation (Manhattan distance) between both vectors of composition :

- the first one is defining the pixel to be classified and its neighbourhood
- and the second one is characterizing successively the histograms of the reference landscape patterns (Robbez-Masson, 1994).

The choice between the seven reference landscape patterns is done by the calculation of the smallest distance. The smallest the distance is, the most likely the pixel with its neighbourhood belongs to this particular landscape pattern.

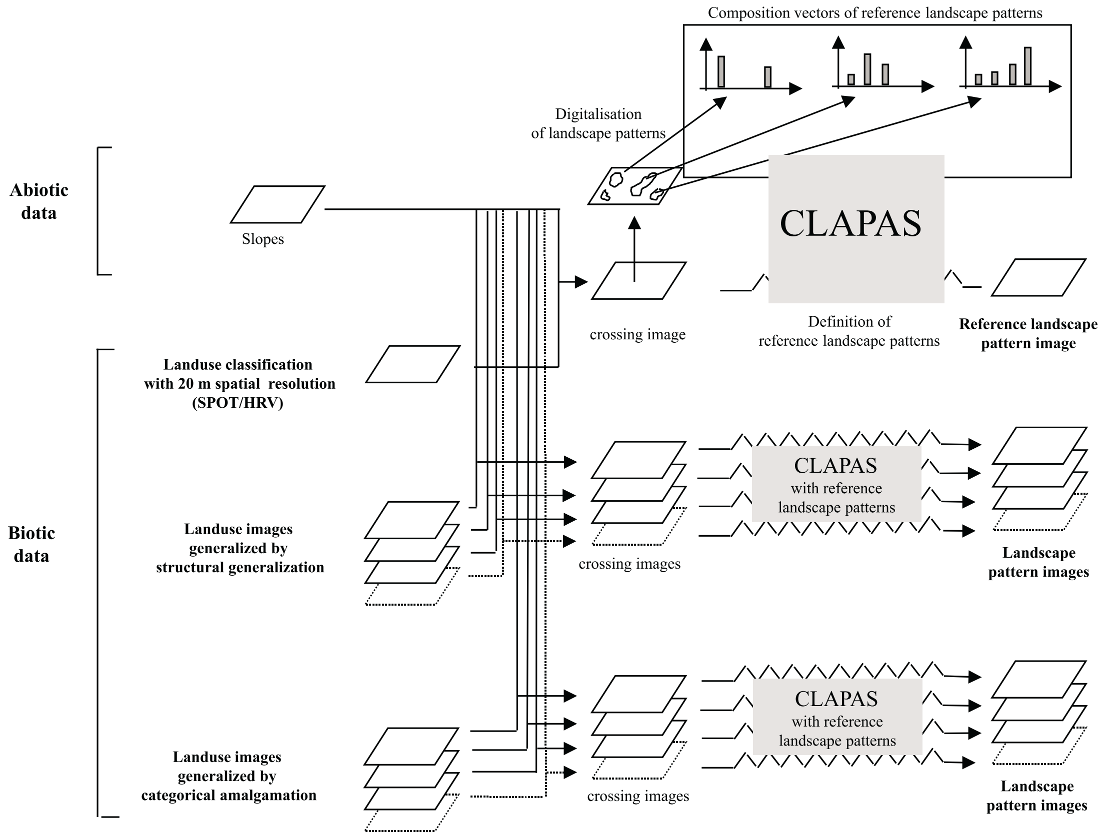
The crossing between the landuse image and the slope image is done for any generalization level of the landuse image and for the both processes (structural generalization and categorical amalgamation).

3. Methods

3.1 Measurements for the thematic generalized images (landuse and landscape patterns)

The measurements of the detail loss in the cartographic information are done through the comparisons of the landuse reference image or the reference landscape pattern image and successively the images generalized by the both used processes (structural generalization and categorical amalgamation). The confusion matrices have been effected over the totality of the images. Consequently, statistical and cartographic criteria have been described for any level of generalization and for any class or any landscape pattern (Caldairou and al., 1997).

Table 4: The different stages of the characterization and the classification of Lannemezan landscape patterns with CLAPAS.



The statistical criterion is based on the pixel population of the both reference images. The new statistical distributions of the landuse classes and the landscape patterns are recomputed at any step of generalization. This analysis emphazises underestimated or over-estimated classes.

The cartographic criterion is especially based on the elimination, the emphasis and the transformation of the forms, i.e. all the modifications of the local organization of the image.

In addition to these both criteria, a more qualitative approach of the problem has to be analyzed. Calculation of a **form index criterion** illustrates that. The form index used in this work was defined by Forman & Godron (1986) :

$$FI = \frac{P}{2 \times \sqrt{\pi \times S}}$$

FI = Form Index

P = Perimeter of the cartographic entity
S = Surface of the cartographic entity

This form index FI has the interest to be without dimension, when an index like P/S (perimeter/surface) is not. Its values are always greater or equal to 1. It is worth 1 when the form is a circle, and tends towards $+\infty$ when the form tends towards a line. In the situation of square, IF takes a value of $2/\sqrt{\pi}$, approximately 1,13 which is the smallest value for rasterized images (Robbez-Masson, 1994).

Robbez-Masson used the FI index to parameterize the curves of the diagram of Feeny (1988), defining the forms of the cartographic classes by type (Table 5) : from small to large and from compact to lengthened. Consequently, he gives value-limits to any form type of cartographic class :

Table 5: Value-limits of the forms of cartographic classes (ROBBEZ-MASSON, 1994).

FI values	Limits	Example
1,13	Smallest value for FI	Square form (pixel)
1,89	limit compact/compact lengthened	rectangle 10 times longer than wider
3,80	limit compact-lengthened/lengthened	rectangle 50 times longer than wider

These value-limits have been used to characterize the form of the different cartographic classes and their transformation after the application of the both generalization processes. The number of polygons for each type is an indicative element. Thus, the form index FI and the number of polygons are calculated for any landuse class and for any landscape pattern.

4. Evaluation of the results and discussion (see Illustration 1)

4.1.Calculation of the information transformations for the landuse images

4.1.1.Statistical criterion

The statistical attitude of the landuse classes is different according to the generalization process applied.

- the structural generalization *largely supports the dominant classes like **grasslands** and **forests** through an over-estimate of their final population. The other less favoured classes like crops and fallows are consequently underestimated and more particularly **winter crops** and **fallows** with statistical value fewer than 80%.*
- the categorical amalgamation *preserves the best statistical reality of the image. Thus, the statistical value of index is practically constant and almost perfect for the whole landuse classes : 100%.*

4.1.2.Cartographic criterion

Whatever the generalization process used, the whole landuse classes have cartographic indices which decrease, especially for classes with modest population spread all over the image through small parcels. Thus, **fallows** are the most affected and the cartographic reality is rapidly lost from the first steps of generalization processes, followed closely by the **winter crops**. Forming part of the dominant classes, **grasslands** and **forests** undergo weak cartographic modifications. **Summer crops** represent an average attitude of these both limits and its cartographic reality remains sufficient up to strong levels of generalization.

4.1.3.Form index criterion

For images generalized by structural generalization, the values of form index show a strong trend to a homogenisation towards the square form (IF = 1,13). The proportion of the polygons having this form index will represent more than 90% of the polygons of **fallows** class for example, and approximately 50% of **summer crops**, **grasslands** and **forests** ones. Moreover, this trend towards a square form gives the impression of a too large zoom on the image. Then, these coarse details keep the reader's attention and make interference to the real message of the map.

For images generalized by categorical amalgamation, the trend is a homogenisation towards square/compact type ($1,13 < IF < 1,89$). This form index characterizes nearly 90% of the polygons of **fallows**, **winter crops** and even **summer crops**. For **grasslands** and **forests**, this trend is less noticeable because they are particularly characterized by broad cartographic entities whose form indices are compact/ lengthened and lengthened. In spite of the application of the processes of generalization, this type of entities is preserved. This tendency to

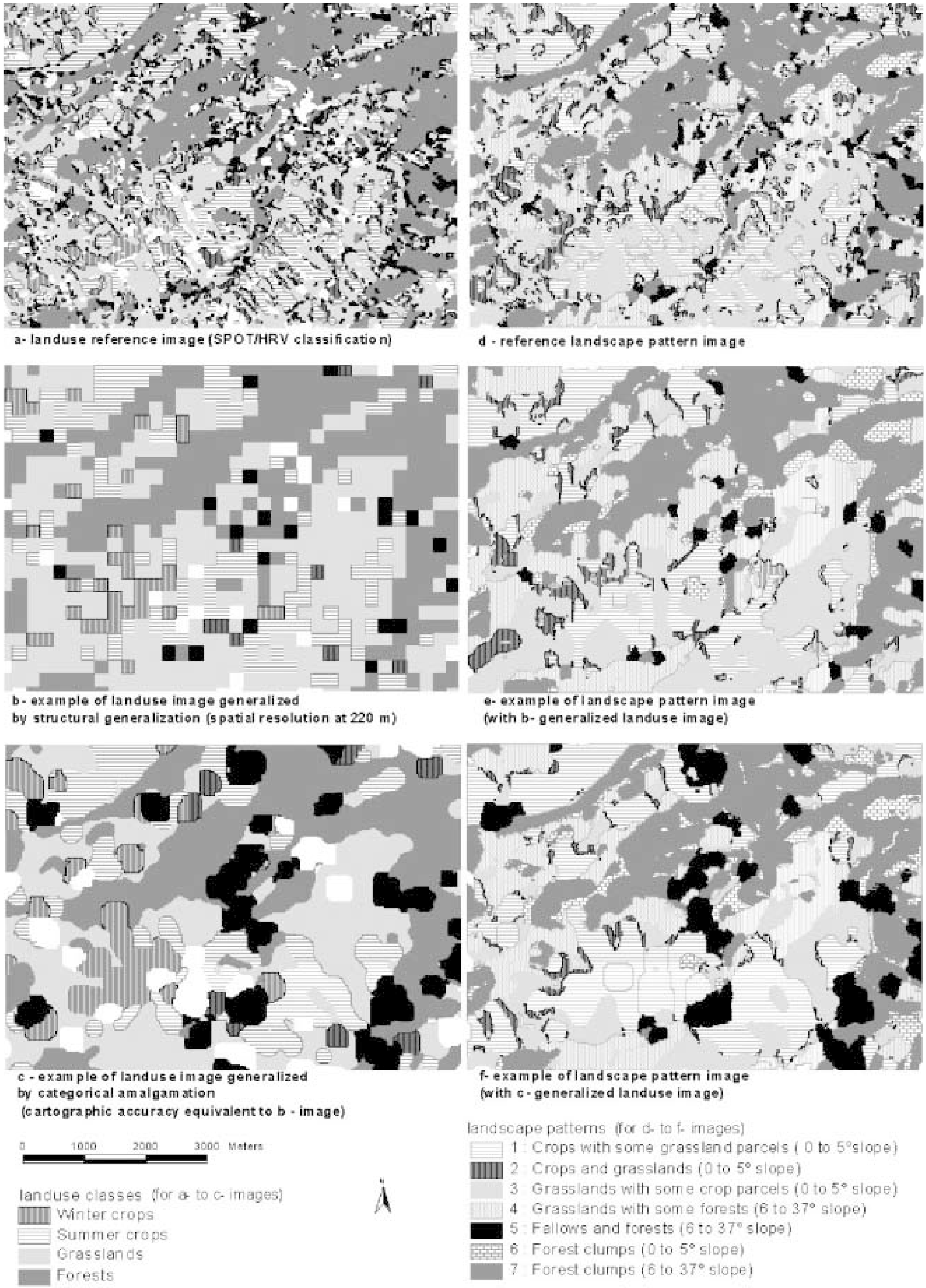


Illustration 1 : reference images and examples of generalized images for the both thematic approaches (landuse and landscape patterns).

compaction gives smoothed cartographic entities with harmonious forms. But these polygons too much gathered and not often correctly localised geographically make interference to the message of the map and consequently to the reading.

Finally, the results of generalization are unsatisfactory. Whatever the generalization process, geometrical transformation and homogenisation are too important and do not take into account of the form characteristics of any class. Moreover, the modification of class outlines is not without consequence on the perception of the spatial organization of the concerned territory. Interferences are then introduced into the decoding of the map message.

4.2 Calculation of the information transformations for images landscape patterns

Statistical and cartographic criteria values are obtained through confusion matrices between the reference landscape pattern image and successively landscape patterns images whose landuse image was transformed by both generalization processes (structural generalization and categorical amalgamation) (Table 4).

4.2.1.Statistical criterion

The strongest variations in values of statistical index are related to the composition vectors of these particular landscape patterns (Table 6). With the used generalization processes that smooth the images, the variability of classes composing certain vectors does not exist so often and then these landscape patterns tend to disappear (underestimation) when others are consequently overestimated.

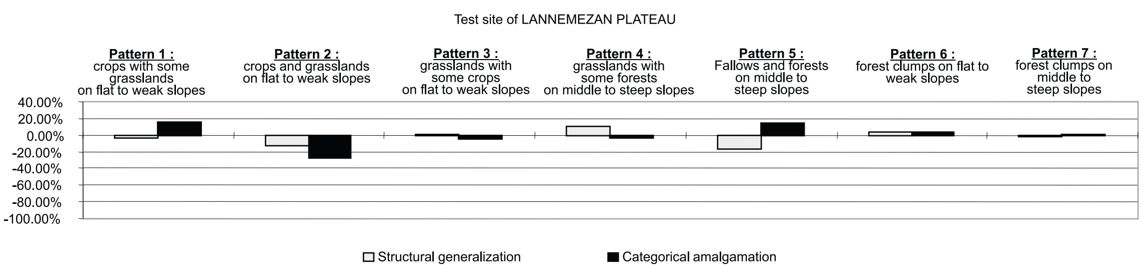


Table 6 : Average of the variation of the statistical index for landscape pattern images

4.2.2.Cartographic criterion

The results show a more important reduction in the cartographic values for landscape pattern images generalized by structural generalization than by categorical amalgamation. The comment made in the previous paragraph remains valid for this cartographic criterion when the disappearance of certain patterns replaced by others means that such landscape maps are not representative any more of the initial cartographic reality.

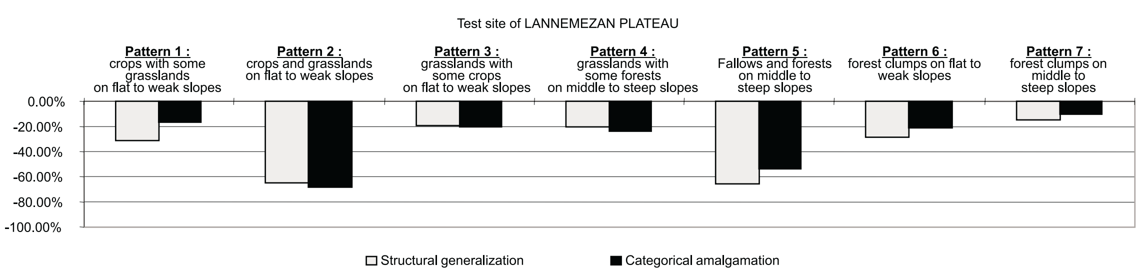


Table 7 : Average of the variation of the cartographic index for landscape pattern images

4.2.2.Form index criterion

Whatever the generalization process used for the landuse image included in landscape patterns, the calculation of form index does not show any trend for a certain form type. The number of polygons decreases during generalization processes, but the proportions within form types for any pattern remain roughly the sames.

The concept of landscape pattern added to ancillary data that do no undergo generalization processes allows a satisfying global preservation of the global forms of the cartographic entities.

4.3 Difficulties in preserving ecologically important pattern

Fallows on the plateau of Lannemezan are ecologically important areas. They have strongly decreased during generalization processes, even with the thematic approach of landscape pattern. But fallows are essentially characterized by anthropic elements as parcels away from the farming, lack of labour,…and steep slopes. With such ancillary anthropic database, this landscape pattern could have been cartographically recomposed and then preserved.

5. Conclusion

It is important to remark that a successful cartographic generalization means a better knowledge concerning the studied territory. Cartographic generalization is not a simple computer problem, territory functions have to be known in order to define what must be preserved.

Going from a simple landuse classification to the concept of landscape pattern forces to a real and ground perception of the spatial organization of this territory. Beyond a succession of information layers, necessary elements for a thematic synthesis, the concept of landscape pattern is more complex than a traditional classification. This concept tends towards an expertise on the spatial organization of the concerned territory with decision rules for conserving particular entities whose causes are known and defined.

This capacity to identify causes (system aspect) of what is belonging to landuse classes and the capacity to identify which types of databases are necessary and how to obtain them form the basis of Geographical Information Systems (GIS), with the successions of data layers used for vertical cartographic syntheses. The particular case of fallows of Lannemezan forces to know the ancillary elements necessary to highlight particular landscape patterns that can be eliminated or modified until an inconsistency with geographical reality by the generalization processes.

6. Bibliography

Baudry J. (1986). Approche écologique du paysage dans *Lectures de paysages. INRAP. Editions Foucher* : 23-31.

Caldairou V., F. Blasco, and M. Gay (1997). Monitoring of information loss in satellite images during cartographic generalization processes. *Proceedings 2, 18th International Cartographic Conference,. 23-27 June, 1997, Stockholm*, 701-708.

Claval P. (1993). Initiation à la géographie régionale. *Fac géographie, édition Nathan Université*, 288 p.

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37 : 35-46.

Deffontaines J.P. (1986). Un point de vue d'agronome sur le paysage dans *Lectures de paysages. INRAP. Editions Foucher*, 33-39.

Feeny V. (1988). Méthode d'analyse du contenant des unités cartographiques. Application au transfert d'échelle. *DEA Pédologie, Grignon*, 52 p.

Forman, R.T.T. and M.Godron (1986). *Landscape Ecology*. Wiley & Son Edition, New York.

Le Men, H. (1996). Cartographies semi-automatiques de l'évolution de l'occupation des sols par télédétection. Thèse d'habilitation à diriger la recherche; *Université Descartes* : 3-15.

Plazanet C. (1996). Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes). *Thèse de doctorat, Université de Marne la Vallée*, 210 p.

Robbez-Masson, J.M. (1994). Reconnaissance et délimitation de motifs d'organisation spatiale. Application à la cartographie des pédopaysages. *Thèse de doctorat*, *ENSA Montpellier*, 161 p.

Robbez-Masson J.M., and L. Cabello (1996). Critères pour une prise en compte de la diversité spatiale dans l'instrumentation de la notion de paysage. Application à une segmentation géographique assistée. *Journées de l'Ecologie du Paysage*, Aussois 9-11 Oct.

Wilkinson G.G.(1993). The generalisation of satellite-derived raster thematic maps for GIS input. *Geographic Information Systems,* 5 : 24-29.

Woodcock C.E., and A.H Strahler (1987). The factor of scale in remote sensing. *Remote Sensing of Environment*, 21 : 311-332.

Session / Séance 47-A

Optimal Map Generalization: saving time with appropriate measures of imperfection.

Michael E. Lonergan, Christopher B. Jones, *and* **J. Mark Ware**

University of Glamorgan

meloner1@glam.ac.uk cbjones@glam.ac.uk jmware@glam.ac.uk

Abstract

Successful automation of map generalization requires effective implementation both of individual generalization operators and of a scheme for controlling the interaction between multiple operations. This paper addresses the latter issue by regarding map generalization as a search problem: the identification of an appropriate realisation of the map from a set of candidate solutions. A means of choosing between the potential maps is required, either by using a numerical measure of quality to identify the 'best' one or through evaluating an explicit set of constraints that any acceptable solution must satisfy. Unfortunately the search space of possible maps is often very large, preventing a straightforward examination of all possible solutions. Iterative improvement methods such as hill-climbing or steepest descent algorithms attempt to speed the process up, but can get trapped in local extrema. Simulated annealing may be used to reduce this problem, but is non-deterministic and can be slow. More sophisticated search strategies such as genetic algorithms trade off speed and solubility in various ways, but can require more complicated formulations and representations of the constraints and candidate solutions.

Simple hill-climbing algorithms search a solution space by moving from an element to the 'best' of its neighbours. Other algorithms that select candidate maps for investigation in an order determined by the results of previous investigations can be considered as producing an ordering on the search space, such that elements that are likely to be investigated consecutively will tend to be close together. As a map quality metric has to take a single value for each candidate solution (while perhaps being a function of many constraints), it can be thought of as producing a simple surface over this space. The topography of this surface is determined by the behaviour of the metric, and in turn determines the speed and solubility of the problem for each type of algorithm. Choice of a metric that reduces the number and size of extrema and discontinuities in this surface could assist the solution of problems.

This paper compares simple metrics of 'map quality' for tackling problems of conflicts between rigid objects through object displacement. In particular it contrasts the use of metrics based on the magnitudes of conflicts with simple counts of the number of conflicts between objects. The first approach allows the use of a simple gradient descent method while the second requires a simulated annealing algorithm to escape local minima. Their ability to contribute to the solution of problems intractable to displacement is also assessed and their extensibility to include other generalisation operators and additional, more complex, constraints is considered.

Introduction

People have been making maps for thousands of years. Over this time, cartography has evolved from arranging sketches and descriptions of features, to producing much more abstract representations of reality [Keates, 1973]. This enables modern maps to convey large amounts of information to their users. Cartographic generalisation is the process of adapting the content of a map to suit its scale, purpose and display medium. It involves the selection of which information to include plus decisions about how and where to represent it on the page [Robinson et al, 1984]. The purpose of generalisation is to produce a good map, balancing the requirements of accuracy, information content and legibility.

Ideally, the generalisation process should produce the best possible map from the information available. There are two reasons why this is seldom possible: cartographers do not always agree as to what constitutes the best map of an area [Robinson et al, 1984], and even if they could, it would take too long to produce a perfect map [Muller et al, 1995] . The first problem has been partially solved, over the years each country and mapping agency has developed a 'house style', these are based on its assessment of 'best' and determine how its maps look. Unfortunately each agency has come to different conclusions, so that a French map may appear very different from a British map of a similar area. In practice the second problem, of the time taken to find the best map, has been sidestepped. If cartographers didn't finish work on a map once its appearance was acceptable, but continued until every other possible solution had been shown to be worse, no maps would ever get made.

Traditionally maps have been generalised manually, with skilled cartographers making decisions about each individual feature. This is a very slow and expensive process [Nickerson, 1991]. It limits the variety of maps available and how often they can be amended. Some advances have been made through the use of computerised tools, but these generally still require the operator to make most of the decisions about process control. Fully automated generalising systems could enable more specialised maps to be produced and all maps to be modified more frequently [Muller et al, 1995].

Automating any process involves a machine acquiring the ability to make appropriate decisions [Russell and Norvig, 1995]. In the case of cartographic generalisation, this means deciding whether a potential map is acceptable and, if not, how it should be modified. Plotting out all the information available often results in an illegible map. Simultaneously solving all of the large number of problems in such a map would involve evaluating all possible interactions between all combinations of objects. This is only to be practicable for very simple situations, usually the sheer number of interactions make the calculations involved infeasible. The problem can be simplified by taking an iterative approach, dividing the problem up and looking at each part in turn. It is necessary to find a series of intermediate solutions, each one of which is in some sense better than the preceding ones, that lead to the desired overall solution. This is basically the same approach as that used by traditional, manual cartography [Keates, 1973].

One major difference between people and computers is in how they process information. The human brain is good at picking out patterns and 'guessing' answers without explicit rules to follow. Conventional computers 'do sums' well, following rules and calculating solutions quickly and accurately. Some Artificial Intelligence approaches, such as Artificial Neural Networks, attempt to take a more 'human' approach to solving problems. These tend to carry a large overhead, needing to be given the ability to 'learn' and enough appropriate examples to build up their experience before being able to deal with the intended problems, but can succeed where more conventional approaches fail [Barr and Feigenbaum, 1982; Russell and Norvig, 1995]. The differences in abilities suggest that copying the strategies used by human cartographers may not produce the best automated systems.

This paper looks at how map generalisation can be formulated to ease its solution through conventional computing approaches. It compares some simple formulations of the problem and evaluates their impact on the problem's tractability.

Automated map generalisers

Requirements

Optimising the problem-solving process can take two forms, either producing the best possible result with a fixed amount of resources, or an acceptable result with the least resources possible. For computer systems the limiting resources include processing time and memory. Real time and safety critical systems are obliged to use the first definition, as late results are useless. For many other applications, including most map generalisation situations, the second definition is more relevant. The object of the system is to produce an acceptable final map as quickly as possible, subject to the limits on the hardware available [Robinson et al, 1984; Muller et al, 1995].

Any automated map generaliser needs to be able to do two things; to produce an acceptable map and recognise that it has achieved this. For an iterative system this means that it must find a way to improve on any map it considers unacceptable. Showing that a generalising system is optimal is likely to be even harder than proving a particular map is. The goal is therefore to produce an acceptable map generaliser. To be acceptable a system would need to produce acceptable maps of a range of situations in an acceptable amount of time. What constitutes 'acceptable' and how wide a range of situations must be handled will depend on the system's intended use. Eventually these become largely subjective, human decisions.

Final maps – what is good enough?

Defining a complete set of criteria that an adequate map has to satisfy is hard and may not be possible [Shea, 1991]. There are some simple rules, such as the requirement that all detail needs to be large enough to be printed and read. Other constraints are less clearly defined [Nygerges, 1991]. For example, the arrangement of houses within a town carries information, but preserving this while reducing the number of buildings shown is not straightforward. Cartographers handle such problems as much by experience and intuition as through following rules [Robinson et al, 1984; Keates 1973]. To fully automate map generalisation a complete and explicit set of rules is required. It may never be possible for these to permit a computer to produce exactly the same map of a situation as a human cartographer would. Even if it were to be possible, it might not be sensible. Map designs have always been a compromise between the users' requirements and the cartographer's convenience. The differences between different agencies' maps show that there is no immediately apparent best way to generalise a map. Maps produced by any new method will inevitably seem strange to those used to the existing ones, but this is likely to be a temporary effect. New approaches to generalisation could even result in more convenient and comprehensible designs [Muller et al, 1995].

The iterative approach could be applied to the process of producing acceptability criteria for maps. Starting from a simple set of rules, sample maps can be produced. Examining these may reveal their limitations and inadequacies. The rule base can then be extended to deal with the observed problems and the process repeated. This paper can be considered as an early iteration of this process, comparing the use of some simple criteria to investigate their adequacy.

The overall system needs to produce a binary result, either the map is acceptable or it is not. Which maps fall into each class depend on the quality criteria used. These criteria attempt to capture the characteristics that determine how well each map satisfies its users' needs. While simple, and adequate for determining when to finish the generalising process, a binary metric cannot help assess how an unacceptable map can be improved. For this a more sophisticated measure of map quality is required. This needs to either indicate the better of any pair of maps or what changes would result in a better map, while still identifying the correct group of maps as acceptable.

Intermediate maps – how to get there

While the criteria for an acceptable final map are externally determined, there is more flexibility in the evaluation of intermediate maps. The intermediate maps produced within an iterative process are stepping stones to a complete solution, and how they are evaluated is largely a matter of computational convenience. It can have some effect on which of the acceptable solutions is found, but its main impact is on the speed of the generalisation process.

An iterative strategy can be seen as a search through the set of possible maps of an area. Some of these maps satisfy all the criteria for acceptability and form the set of satisfactory solutions. Most of the maps are unsatisfactory in some way, some features may be too close together or shown too far from their true locations. How long it takes to find an acceptable map depends on four major factors: the size of the search space; the size of the target set; the number of intermediate maps that need to be considered; and how long each iteration takes. The target size is fixed by the acceptability criteria for the final map, but all the rest depend on how the search is formulated. It is possible to reduce the size of the search space by using some of the criteria of acceptability as constraints and considering only those candidate solutions that satisfy them. This has the side effect of reducing the number of paths that may be followed to the solution, so depending on which criteria are used for this it could either speed up or slow down the search. How many intermediate maps need to be investigated and how long each one takes to consider depend on the interaction of the particular situation being worked on and the search strategy used.

Given some measure of the quality of each candidate solution, an ‘error surface’ can be produced over the search space. A smooth surface is easier to search through. The highest point of a simple surface that rises gradually to a maximum can easily be found by using hill-climbing algorithms. These just look at the neighbourhood of the present location and move towards its highest point. The presence of large numbers of discontinuities and local extrema require more complicated strategies. To escape a local maximum it is necessary to move downhill, to a worse solution, in order to later on reach a better result. Simulated Annealing is one approach to this, allowing bad moves in early iterations, and gradually reducing the probability that such moves are made as the process continues. Using this approach can solve some problems hill-climbing cannot, but it is not guaranteed to produce correct solutions to the problem, and its results are non-deterministic [Russell and Norvig, 1995].

Search strategies can be seen as producing an ordering of the space they search through. This places each candidate solution near to those investigated before or after it. The rearrangement of the search space will affect the smoothness of the ‘error surface’. The choice of an appropriate combination of quality measure and search strategy may have a large effect on the overall speed of the system. The actual values of the error function do not have to be calculated explicitly each time as, unless the problem has been solved, all that is needed is to identify the next step in the process each time.

Redrawing features is time consuming in traditional, manual, cartography. Generalisation techniques have therefore attempted to minimise the number of times objects are displaced or modified [Robinson et al, 1984]. This also fits well with intuitive measures of map quality. Computerised systems can quickly recalculate object locations, making this requirement less important. It may be worth repositioning features many times if this reduces the complexity of the calculations involved in producing the final map. The differences between human and computerised approaches to pattern handling suggest that standard cartographic measures of intermediate map quality are unlikely to be optimal for computerised systems.

The example of conflict resolution

Displacing features

One requirement for most maps is that all features shown are some pre-defined minimum distance apart. (A common exception is that they may be permitted to be in direct contact, though not very close together. Here this will be considered to involve the individual features merging into a composite structure.) Accuracy is another basic requirement, so a second possible rule is that each feature on the map must be placed within a given distance of its 'true' location.

There are several ways of tackling this reduced problem. The direct approach, producing a measure of quality that considers both locational accuracy and separation distances requires searching a very large space. This can be greatly reduced in size by treating one of the criteria as a constraint and using the other to define a measure of quality within the reduced space.

Treating the minimum separation distance as a constraint, and allowing features to move freely subject to this, produces a system within which the rules of intermediate map improvement are complicated. Individual features have to find ways to slide past each other. Further constraining the system so that initially neighbouring features are considered as nodes linked by a network of semi-rigid edges (permitted to stretch but not shrink below the minimum acceptable length) slightly improves the situation. This produces a surface, generally likely to be much larger than the available space, which then needs to be compressed to fit. Allowing this process of distortion to take place in three dimensions might ease its solution, effectively fixing the corners of the map in place, allowing the middle of the sheet to balloon up and gradually finding ways to flatten it back down.

While they may be theoretically feasible, the above formulations are complicated. A much more straightforward approach to reducing the search space is to constrain every object to lie within the acceptable distance of its true position. Map quality can then be evaluated in terms of the distances between neighbouring objects. This formulation is also more intuitively acceptable, though this is no indication of its efficiency. Even having partitioned the requirements, there are still several possible approaches to searching for an acceptable map. The aim is to produce a map with no features in conflict (all are separated by at least the minimum acceptable distance). Starting from a map containing several conflicts, the most obvious strategy is to attempt to reduce the number of conflicts to zero. This approach has been used in previous work in the area [Ware and Jones 1998]. However, there is also an alternative, slightly less straightforward approach possible, that of attempting to reduce the severity of every conflict to nothing [Lonergan and Jones, unpublished manuscript]. Applying physical Finite Element techniques to the problem effectively implements this strategy [Hojholt 1998]. Both approaches finish up at the same point, producing acceptable maps, but these final maps may differ and the paths taken to them certainly will.

A simple example of an area to be generalised is shown in Figure 1. Within a fixed boundary, there are a total of 13 objects, each permitted to move up to a maximum of 120 units and required to be displayed at least 80 units apart. Similar, though not identical results, can be produced using the two approaches. Strategies based on minimising the number of conflicts cannot solve this example through hill climbing, as some conflicts necessitate the movement of several features. Incorporating Simulated Annealing cures this, but can involve large numbers of object displacements [Ware and Jones, 1998]. Maximising nearest neighbour distances reduces the number of moves required (from 70 to 4), though this is partly offset by the greater complexity of the calculations involved. (For the first approach it is enough to establish that there is a conflict between a pair of objects, for the second it is necessary to find the shortest vector between them.) In this case the results of the two approaches are very similar.

The strategy used for the reduction in the severity of the conflicts in the example above is based on equalising the conflicts on each side of an object at each step. This is equivalent to defining the measure of map quality as a list of the conflicts in order of severity. The worse of two maps is then the one with the single worst conflict, with ties broken by comparing subsequent pairs of conflicts. Weighting the magnitude of each conflict by a factor based on its position in the ordering and summing the results can produces a very similar Real-valued metric of map quality. However, calculating the value of this metric in full would be time-consuming, and is unnecessary in practise.

Figure 1. A simple example of a region that contains several objects in conflict (top), the results of the nearest-neighbour distance maximising method (below left), and the Simulated Annealing approach (below right). Bars indicate conflicts between objects

Moving on from displacement

Depending on the dataset and the limits set on displacement and locational accuracy, object displacement alone is not always sufficient to produce an acceptable map of a situation. It is often necessary to carry out other operations, either merging or changing the representation of features or eliminating some entirely. The methods described above can be used to identify the features requiring modification. In this case they produce very different results. Taking the same example as above, but with the required separation increased to 100, it is not possible to solve the problem entirely through displacement. Using the Simulated Annealing method [Ware and Jones, 1998] reduced the initial 9 conflicts to 2 (Figure 2a). During this process some of the objects were pushed closer together, implicitly suggesting that these should be merged or eliminated. Re-running the program could produce different suggestions, because of its random nature. The second approach shares out the problems and results in a cluster of objects in less severe conflict (Figure 2b). Rather than identifying one specific way forward it indicates the range of features whose modification could aid in the problem's solution. Some other method is then required to choose how to proceed from there. This could involve strategies such as eliminating the smallest feature or merging the 'most similar' ones.

Figure 2. A region containing several objects whose conflicts cannot all be solved by displacement (top), the results of the nearest-neighbour distance maximising method (below left), and the Simulated Annealing approach (below right). Bars mark conflicts between objects.

While the separation distance requirements are important in map-making, they are not the only ones [Nygerges, 1991]. Methods are required for reducing the amount of detail in an area, while preserving its character. Displaying a row of four houses may result in a more acceptable representation of a row of eight houses than showing all eight displaced into an irregular pattern. It is therefore necessary to control the effects of displacement-based methods of generalisation. There are two obvious ways of handling this problem. Where there are non-overlapping areas with distinctive patterns, these could be used as the basic units for displacement, and their internal rearrangement controlled by other strategies. This approach could also be used where patterns overlap, but it may be more appropriate to allow feature displacement in the normal way, using the nearest neighbour approach, and base any necessary decisions on feature modification on the structure of the patterns.

Clustering and Space partitioning

Partitioning the map surface can reduce the number of interactions between objects to be considered. This may simplify the overall problem and speed its solution. One common approach to space partitioning is to use networks of linear features, such as roads or railways, though these may need to be generalised as well. Splitting up the map's surface into independent areas could be more useful.

One side effect of maximising nearest neighbour distances is that the clusters of features it produces partition the map surface. Once the nearest neighbour distances have been maximised, clusters of features can be defined such that any two features in conflict are in the same cluster. Any subsequent elimination or merging of features can then split clusters, but is very unlikely to result in features in different clusters being brought into conflict. Using this partitioning of the map surface may allow each cluster to be worked on independently. This is likely to speed up the generalisation process and may permit its parallelisation. It may be that significant links between clusters can only occur in unusual and pathological cases, but this needs to be investigated further.

Conclusions

The acceptability of the results of an automated map generaliser depend on the map quality criteria it is based on. The choice of appropriate measures of intermediate map quality and strategies for moving between them can improve the speed of iterative improvement techniques in map generalisation. Partitioning the acceptability criteria, so that some limit the size of the space to be searched, while others provide the function to be optimised, can simplify the solution of the problem. The partitioning needs to reduce the size of the search space but still leave enough flexibility to reach an acceptable solution. Treating accuracy requirements as constraints on intermediate maps and optimising separation distances appears a division for conflict resolution through displacement.

The approach described above treats all maps whose components are within a pre-defined distance of their true location as equally good. A 'better' measure of map quality might attempt to minimise some function of these distances. It is not immediately apparent how different the results and speed of such a method would be. A post-processing step to draw individual features back towards their true positions after the elimination or merging of features might produce equally acceptable results. If there is no group of equally acceptable solutions, the problem returns to that of finding the single best map. It is still possible to use some criteria to reduce the space of candidate spaces to be searched, but the optimisation function is likely to be more complicated and the search much slower.

Simple generalisation techniques may produce adequate maps for some purposes. Where speed of production is more important than maximising the information content of a map, simple geometric approaches may suffice. The methods described here result in legible maps. For other purposes more sophisticated measures of map quality may be required. In some cases extending the set of map quality criteria used might enable this approach to produce adequate results. Whether this would be possible in all cases is unclear. Even if this method doesn't produce a complete solution to the problem of automating map generalisation, it may allow the control of basic operations and reduce the size of the remaining problem, either enabling other methods to solve it or increasing the efficiency of its users.

The time taken by any generalisation system will increase with problem size. This can be defined in two ways, either the number of constraints to be satisfied, or the number of objects to be handled. As the number and magnitude of local extrema and discontinuities increase, stochastic approaches such as Simulated Annealing have to make many 'good guesses' to succeed. This increases the time taken to solve problems. Reformulating the problem to allow a hill-climbing approach to produce a solution avoids this effect. In the case of interobject conflict, maximising nearest neighbour distances allows this. Adding in extra criteria may require changes to the formulation.

Equivalent formulations for non-rigid map features, both areal and linear, are being worked on to enable the extension of the methods described above. Using such an extended system to produce sample maps of areas and problematic situations should indicate its limitations. From these it is hoped to identify the modifications to the quality criteria required. How far such an iterative refinement approach can go towards producing an adequate system may eventually become clear.

References

Barr, A., and Feigenbaum, E.A. (1982) *The handbook of Artificial Intelligence – volume 1*. Addison_Wesley, Reading, Massachusetts.

Hojholt, P. (1998) Solving Local and Global Space Conflicts in Map Generalisation Using a Finite Element Method Adapted from *Structural Mechanics.* Proceedings of the 8th International Symposium on Spatial Data Handling.

Keates, J.S. (1973) *Cartographic Design and Production*. John Wiley, New York.

Keller, S.F. (1995) Potentials and limitations ofartificial intelligence techniques applied to generalisation. In Muller, J.-C., Lagrange, J.-P., and Weibel, R. (eds) *GIS and Generalization: Methodology and Practice*. Taylor and Francis,London.

Lonergan, M.E., and Jones, C.B. (unpublished manuscript) A dynamic displacement method for conflict resolution in map generalisation. School of Computing, University of Glamorgan, Wales, UK.

Muller, J.-C., Weibel, R., Lagrange, J.-P., and Salge, F. (1995) Generalization: state of the art and issues. In Muller, J.-C., Lagrange, J.-P., and Weibel, R. (eds) *GIS and Generalization: Methodology and Practice*. Taylor and Francis,London.

Nickerson, B.G. (1991) Knowledge engineering for generalization. In Buttenfield B P, McMaster R B (eds) *Map Generalization: Making rules for Knowledge Representation*. Longman, London.

Nygerges, T.L. (1991) Representing geographical meaning. In Buttenfield B P, McMaster R B (eds) *Map Generalization: Making rules for Knowledge Representation*. Longman, London.

Robinson, A.H., Sale, R.D., Morrison, J.L., and Muehrcke, P.C. (1984) *Elements of Cartography* 5th edn John Wiley, New York.

Russell, S., and Norvig, P. (1995) *Artificial Intelligence A Modern Approach*. Prentice-Hall, New Jersey.

Shea, K.S. (1991) Design considerations for an artificially intelligent system. In Buttenfield B P, McMaster R B (eds) *Map Generalization: Making rules for Knowledge Representation*. Longman, London.

Ware, J.M., and Jones, C.B. (1998) Conflict Reduction in Map Generalisation Using Iterative Improvement. *GeoInformatica* 2(4),383-407.

Session / Séance 47-B

The ‘Good Continuation’ Principle of Perceptual Organization applied to the Generalization of Road Networks

Robert C. Thomson

Prologic Systems Ltd.
Suite 206, 75 Albert Street
Ottawa, Ontario K1P 5E7
rcthomson@hotmail.com

Dianne E. Richardson

Canada Centre for Remote Sensing
Geomatics Canada
588 Booth Street
Ottawa, Ontario K1A 0Y7
Dianne.Richardson@geocan.nrcan.gc.ca

Abstract

Perceptual organization (or grouping) principles play a vital role in the understanding of images and maps, and their importance in map generalization has also long been recognized. This paper presents a brief review of the application of the principles in map generalization, and then examines their particular relevance to the generalization of road networks. It is shown how the ‘good continuation’ grouping principle can serve as the basis for analyzing a road network into a set of linear elements, here termed ‘strokes’. Further analysis allows the strokes to be ordered, to reflect their relative importance in the network. The deletion of the elements according to this sequence provides a simple and effective method of generalizing (attenuating) the network. This technique has been implemented, as part of the ‘GenSystem’ generalization software package developed at the Canada Centre for Remote Sensing. The implementation is outlined, and the effectiveness of the technique demonstrated.

Introduction

Perceptual organization (or perceptual grouping) describes the phenomena whereby the human visual system spontaneously organizes elements of the visual field. Perceptual organization principles play a vital role in the understanding of two-dimensional images of three-dimensional scenes [Witkin and Tenenbaum, 1983; Biederman *et al*., 1991; Lowe, 1985]. These principles are also crucial in the interpretation of other forms of images, such as sectional images [Thomson and Claridge, 1989; Thomson, 1999], and in the understanding of maps [Bertin, 1983; MacEachren, 1995]. The important role of these principles in map generalization has also long been recognized. Notably, DeLucia and Black [1987, p.175] stated that “… there are direct analogies between what is required for successful map generalization and … the principles of perceptual organization enunciated long ago by the Gestalt psychologists”; further, they expressed their bafflement concerning the limited presence of the Gestalt brand of perceptual psychology in the generalization literature.

This paper presents a brief review of the general principles of perceptual grouping, and considers their relevance to the generalization of cartographic data. The application of the 'good continuation' principle of perceptual organization to the generalization of networks, and of road networks in particular, is then considered in more detail. Road networks have proven more difficult subjects for automated generalization than hydrographic networks, reasons for which include the ubiquity of loops or cyclic paths, richer road attribute sets, and the lack of a natural direction for individual segments.

It will be shown that the good continuation principle can serve as the basis for partitioning a road network into a set of linear elements or 'strokes', which are chains of network arcs. Various attributes, such as length, representative class, or 'connectivity', can be derived for these elements. On the basis of these measures the set of strokes can be ordered to reflect the strokes' relative importance in the network. The deletion of the strokes according to this sequence then provides a relatively simple and effective method of generalizing (i.e. attenuating) the network. This technique has been implemented as part of the 'GenSystem' generalization software package developed at Canada Centre for Remote Sensing [Richardson 1993; 1994].

Perceptual grouping principles

Even with no high level or semantic knowledge available, the human visual system spontaneously organizes elements of the visual field. These phenomena have been studied by the Gestalt psychologists; they observed how some arrangements of picture elements tend to be seen as 'belonging together', forming natural groups. Often these may appear to stand out from the surrounding elements, i.e. as 'figures' against 'grounds'. Many perceptual grouping principles have been identified, such as proximity, similarity, symmetry, closure, parallelism, collinearity, co-termination and continuity. Some illustrative examples are given in figure 1.

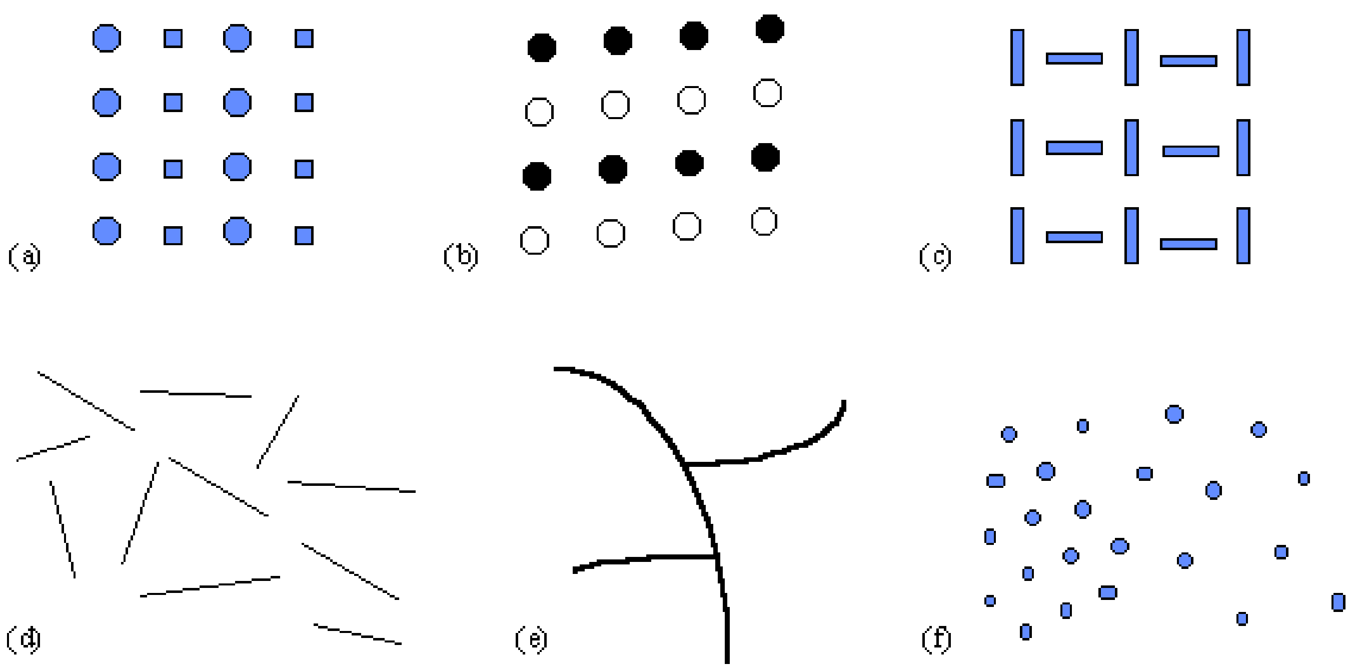


Figure 1. Examples of perceptual grouping

Figures 1(a) to 1(c) show grouping according to different forms of similarity – similarity of shape, colour (or grayscale) and orientation. Thus, for example, figure 1(b) is perceived as two groups of white circles and two groups of black circles, even though the vertical separation of the circles is slightly less than the horizontal separation. Figure 1(d) shows the effect of collinearity: the three collinear elements are perceived as a figure. Figure 1(e) shows the importance of good continuation: this figure is perceived as one smooth curve with two smaller curves incident on it, rather than as five connected linear elements, for example. Figure 1(f) shows the effect of uniform density: the collection of dots is perceived as two groups, where the dot spacing within each group is relatively uniform.

The perceptual grouping principles are now widely recognized as the basis for parsing the visual world into surfaces and objects according to relatively simple visual characteristics, in a process that operates independently of the domain being represented [Witkin and Tenenbaum, 1983; Palmer, 1983]. As cited above, the importance of these principles in the analysis and interpretation of different forms of images has been established. Further, direct analogies have been identified between the Gestalt perceptual grouping principles and the procedures required for successful map generalization.

DeLucia and Black [1987] used the Gestalt grouping principles as a resource of fundamental knowledge to guide decision-making in automating generalization. The Gestalt principles provided a means of predicting what the result of a generalization process should look like in order to be readily comprehended by the map user. The generalization processes they considered included simplification, aggregation, agglomeration, feature collapse, and distribution refinement. ‘Distribution refinement’ refers to the controlled density reduction or attenuation of a pattern – which may consist of lines, points and areas – while retaining those elements that are most important in the given context and, moreover, preserving the character of the pattern where possible. DeLucia and Black, however, did not attempt the generalization of ‘open irregular networks’, such as road or drainage networks.

One approach to road network generalization (distribution refinement) will be described below: this method employs the ‘good continuation’ principle described above and illustrated in figure 1(e). The ‘radical law’ of selection [Töpfer and Pillewizer, 1966] can serve to specify *how many* elements to remove with a reduction of map scale; the good continuation principle will provide a means of defining *which* elements to remove.

Road network generalization

The basic topological structure of a road network is captured by a graph – a set of edges or arcs connecting pairs of vertices or nodes. Clearly graphs represent the topology of road networks in a natural way, with nodes corresponding to road junctions, intersections, dead-ends and locations of special interest on the roads, and arcs corresponding to road segments between such points. When road class information is available the restriction may be imposed that each arc must be uniform in its attributes. This can be achieved by inserting nodes in the network at points where a road segment’s attributes change.

The automatic generalization or abstraction of road networks can be viewed, at its simplest, as a process of controlled attenuation – i.e. the progressive removal of road segments. This requires a means of establishing the relative importance of segments for the given context, which defines the sequence of segment removal to be applied. The context for the generalization may take the form of a set of locations of special importance on the network, such as major towns. In such a situation, it is possible to derive measures of the relative importance of the road segments, for example by way of considering optimal paths between the designated locations [Thomson and Richardson, 1985; Richardson and Thomson, 1996], and so proceed to generalize the network.

Defining a set of important network locations should not, however, be a compulsory stage in network generalization. A consideration of specimen road networks makes it clear that, even without such a site selection process to provide contextual information, they can usually still be generalized (by a human cartographer) purely on the basis of their geometric, topological, and thematic properties (i.e. road classification information). It even appears possible to infer the relative importance of road segments in a network in the absence of all thematic information.

It is proposed that this generalization process is possible because the human visual system naturally discerns the relative perceptual significance of the network elements and, further, that the relative perceptual significance of network elements corresponds closely to their functional importance in the network. Thus perceptual salience can serve as a means of quantifying relative importance of individual network elements, so supporting generalization of the network (and in situations where the correspondence between functional importance and perceptual salience is weaker appropriate adjustments can be made). These propositions will be expanded upon below, and a description given of how this translates into a practical method for automatic road network generalization.

Good continuation and 'strokes'

The perceptually salient features of the network are salient because of the action of one or more perceptual grouping principles. The principle of good continuation can be expected to be the dominant principle, but if thematic information is available then the principle of similarity (of thematic attributes) can also play an important role. The perceptually salient features of the network can therefore be identified by applying the same principles to group (i.e. concatenate) the arcs into chains. These chains, which are here termed 'strokes', do not branch, although they may cross (unlike their constituent arcs). The term 'stroke' is prompted by the idea of a curvilinear segment that can be drawn in one smooth movement and without a dramatic change in style.

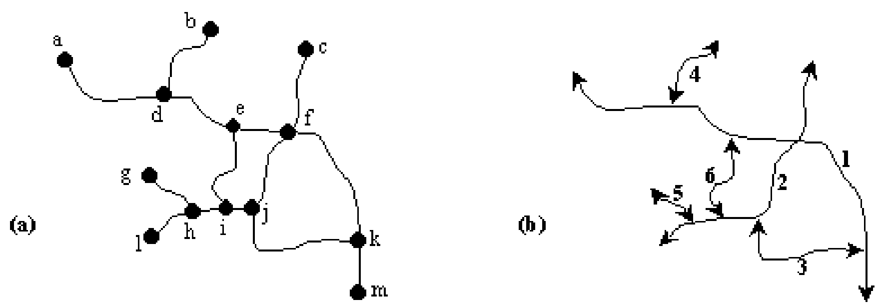


Figure 2. An example network and its analysis into strokes

Figure 2 provides a simple example of a network and the set of constituent strokes found by such an analysis. Figure 2(a) represents a simple road network, with no thematic attributes. This network, considered as a graph, has 14 arcs and 13 nodes; the nodes are highlighted and labelled *a* through *m*. Figure 2(b) shows the six strokes that were identified in the network; these have been labelled 1 through 6, and their endpoints marked with arrows. The values of the numbers used as labels have no special significance. As detailed in table 1, two strokes found are the concatenation of five arcs and each remaining stroke contains one arc. Strokes 1 and 2 cross at node *f*; node *f* is not a terminal node for any stroke.

Table 1. Description of the strokes found in figure 2

stroke	*terminal nodes*		*# arcs*
1	a	m	5
2	c	l	5
3	j	k	1
4	b	d	1
5	g	h	1
6	e	i	1

Table 2. Description of the strokes found in figure 3

stroke	*terminal nodes*		*# arcs*
1	d	m	4
2	c	l	5
3	j	k	3
4	b	d	1
5	g	h	1
6	e	i	1
7	a	d	1

Figure 3 shows the same network, but with additional thematic information: three categories of roads are now represented. In such a situation the process of arc concatenation to build the strokes is further constrained by imposing limits on the compatibility of arc thematic categories. For this example it will be assumed that it is allowable to concatenate an arc of category 1 with an arc of category 2, and similarly arcs of categories 2 and 3, but that arcs of categories 1 and 3 cannot abut in a stroke.

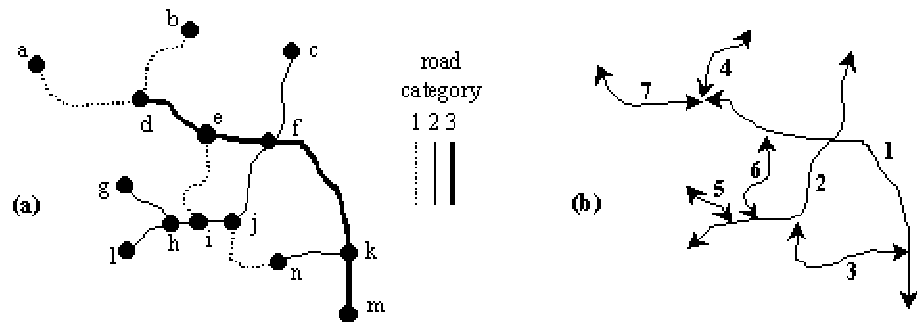
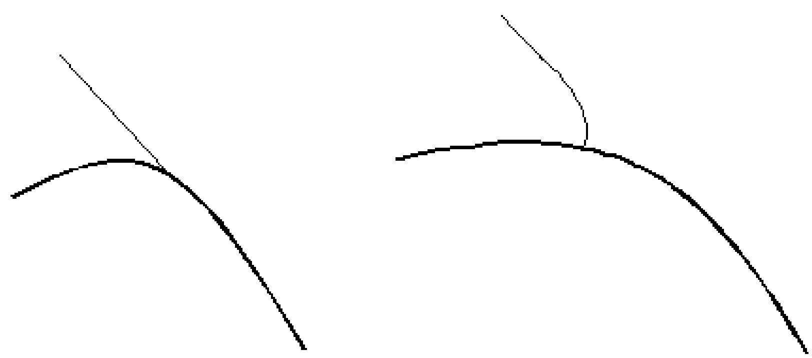


Figure 3. An example network with three road categories and its analysis into strokes

With the introduction of road categories a new node must be inserted on the arc between nodes *j* and *k*, so that the resulting arcs represent one category only. Comparing figure 2(a) with 3(a) we see that stroke 1 now terminates at node *d* since the arc linking nodes *d* and *a* is of category 1, which cannot be concatenated directly with an arc of category 3 – the grouping by similarity principle in this case rejects the grouping, overriding the good continuation of shape. The arc between nodes *a* and *d* and the arc between nodes *b* and *d* are not concatenated into a stroke even though they are both of category 1, because of the poor continuity of slope between the two arcs at node *d*. In contrast, the arc between nodes *j* and *n* and the arc between nodes *n* and *k* are concatenated into a stroke, even though they are of different categories – the categories are considered similar enough not to prevent grouping, and the continuity of slope between the two arcs is perfect. (A greater difference in categories or a poor continuity of slope could have prevented their concatenation into the same stroke.)

Figure 4 represents a portion of a road map where a minor road joins a major road on a bend. Figure 4(a), which represents a relatively small scale view, shows a good continuity of slope between the major and minor road at the junction; figure 4(b) shows a larger scale view, where the apparent smooth continuation is no longer present and a T-junction configuration is revealed. This shows the effect of scale in determining the continuities of slopes at junctions. It also reflects a general rule: the geometry of a road junction is usually such that relatively major roads passing through will show a smoother continuation than the relatively minor. Thus the rule of good continuation will usually succeed in correctly pairing up the incident components of a road at a junction.



a) **b)**

Figure 4. A road junction showing different form at different scales

Stroke attributes

Once the set of strokes has been built up for the road network, attribute values must be derived for each stroke in order to support the subsequent generalization of the network. One important stroke attribute is clearly its length. A second is the road category (or categories) represented by the arcs that constitute the stroke. Use of this attribute requires a method for ranking road categories in order of importance: suitable values can be established using empirical knowledge. Other measures for quantitatively characterizing strokes are possible, as will be considered below.

Table 3 shows how the road category and length information can be used to rank the strokes in order of importance. This table refers to the simple example network of figure 3, also described in table 2: here there are three road categories present, with category 1 the least important and category 3 the most important. In general, road category is more important than length and so the strokes are first ranked by category. Where a stroke contains arcs of more than one category different strategies are possible: one could use, for example, the best category, the worst, or the category representing the greatest constituent arc length. In table 3 a stroke's representative road category is taken to be the best category found for any of its constituent arcs, thus stroke 3 is given the representative category 2. Within each cluster of strokes with the same representative category the strokes are ordered by length. In table 3 the final column shows the final ranking derived for each stroke, with stroke 1 found to be the most important and stroke 4 the least.

stroke number	*representative road category*	*stroke length*	*importance ranking*
1	3	50	1
2	2	45	2
3	2	22	3
4	1	10	7
5	2	7	4
6	1	16	6
7	1	17	5

Table 3. Ranking the strokes of figure 3

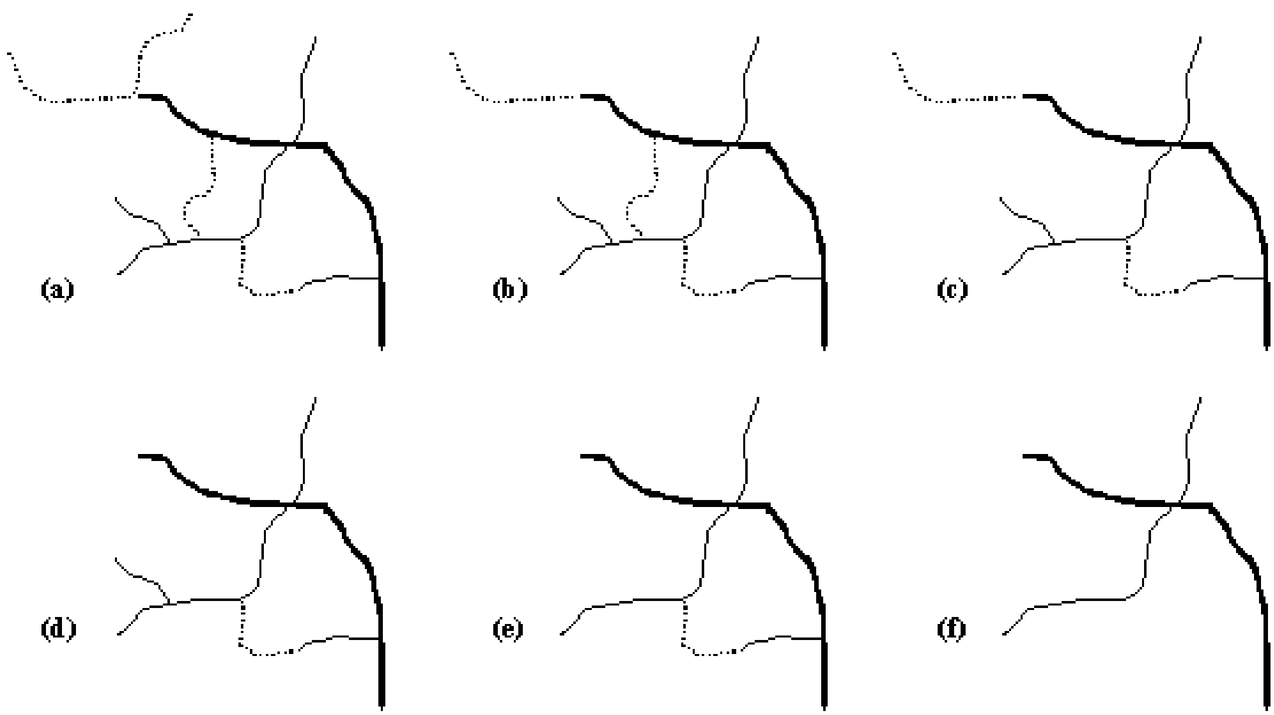


Figure 5. Road network attenuation by progressive stroke removal

Figure 5 shows the progressive attenuation of the network that would follow from this ranking of strokes. Figure 5(a) shows the original network, also depicted in figure 3. In figures 5(b) through 5(f) the least important stroke of the preceding figure has been deleted. This produces a progressive attenuation of the network; the final attenuation would be stroke 1 alone.

This approach can be modified for the common situation where it is considered important that the network should not become disconnected during attenuation. In order to avoid fragmentation of the generalized network, the strokes are considered in ascending order of importance and each stroke is checked to see if its removal would lead to the set of strokes with greater importance ranking being disconnected. If so, the stroke's importance ranking is raised (to the minimum importance ranking found for the strokes which intersect it).

It is the need to avoid fragmentation of the road network, and so preserve the network character where possible, that rules out the simplistic approach of generalizing simply on the basis of road category (with arc length used as a secondary criterion). Tests showed that such an approach commonly led to unwanted disruption of road continuity. Introducing the good continuation principle, allowing concatenation of arcs of different category, minimizes this problem.

Related work

It is interesting to compare this approach to road network generalization with that described by Mackaness [1995] for urban road generalization. The latter scheme is similar in that 'axial lines' are derived for a network, quantitative indices are derived for these elements, and these then provide the basis for the generalization. The derivation of axial lines, however, is based on lines of sight, following a concept drawn from urban space pattern analysis [Penn, 1993; Hillier and Hanson, 1984]. The indices presented by Mackaness for the characterization of axial lines are axial connectivity, depth, control, and relative asymmetry. Analogous indices could be derived for strokes and the methods presented by Mackaness for network analysis could similarly be adapted; this will be an area for future investigation.

Compared to axial lines, the 'strokes' described above are far simpler computationally to derive, they appear to generate results that are at least as useful, and, being based on more generally applicable principles, they can be used for a wide range of networks, not simply urban networks. Indeed, the 'good continuation' method can be applied to hydrographic networks, where it can be seen to be closely related to the use of Horton stream order. Horton's [1945] method of assigning orders to stream segments takes into account the shape of junctions, with the upstream segment that joins the downstream segment at least angle considered to be a continuation of the downstream segment (and given its order). Horton stream order has been established to be a good basis for hydrographic network generalization [Rusak Masur and Castner, 1990; Richardson, 1993].

Implementation

Road network generalization *via* 'stroke building' has been implemented at Canada Centre for Remote Sensing. Compared with network generalization methods that rely on the computation of optimal paths, the approach based on strokes is computationally much less intensive, and is not threatened by 'combinatorial explosion'. The implementation makes use of a set of tables or matrices that contain the rules defining aspects of the stroke creation. These are the 'data translation tables', the 'transition table' and the 'angular constraint table'.

Since the road data for generalization may come from a variety of sources, using different schemes for describing thematic attributes, a necessary preliminary stage is to re-classify the thematic road data into a set of standard categories. Subsequent processes need only operate in terms of the standard categories. This re-classification will refer to a data translation 'look-up' table previously compiled for this particular data source.

The transition table and angular constraint table together control the stroke-building process in which arcs are concatenated into strokes; these tables effectively encapsulate the perceptual grouping rules. During stroke-building, at each network node of degree 2 or more, a decision must be made for each incident arc as to which, if any, of the other incident arcs it should be concatenated with. The transition table controls the 'grouping by similarity': for each possible road category this table lists the road categories which are considered similar enough to allow concatenation of arcs to take place, and gives an order of preference for the alternatives. The angular constraint table specifies the maximum allowable turn between arcs that will be allowed during a concatenation; this limit angle can be varied according to the difference in arc road categories involved. Both these tables can be extended to provide different styles of stroke-building in urban and non-urban contexts. This requires that each network node be classified urban or non-urban, which can be inferred from node density [Borchert, 1961].

Given the information on the incident arc categories and turn angles (and urban/non-urban context, if required), the rules set down in these two tables determine the way in which arcs will be paired up at any network node during concatenation. It is convenient to establish and record this information for each node in a preliminary procedure. This makes the subsequent stroke-building process very simple to implement. This approach has been used to implement the method within the GenSystem experimental generalization software package. The implementation includes the enhancement to prevent unwanted fragmentation during generalization, and has additional options for incorporating information about important network locations. Further development is being undertaken, although tests show the current method to be very effective.

Figure 6 shows an example of the generalization of a road network using the method outlined above. In this example the input data was at a scale of 1:20 000 and the level of network attenuation was set to be appropriate for 1:250 000 scale mapping.

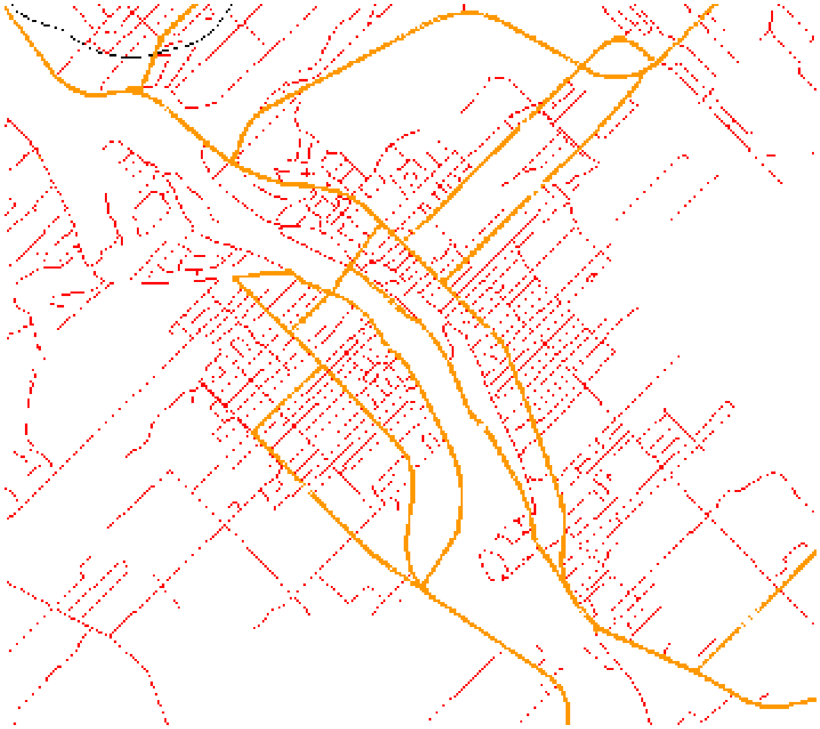


Figure 6. Urban road network (1:20K data) with the road elements selected for a generalization to scale 1:250K highlighted

Conclusions

The application of perceptual grouping principles in map generalization has been considered, with particular reference to the generalization of road networks. It has been shown that the 'good continuation' grouping principle, and the principle of grouping by similarity, can provide a means of analyzing a road network into a set of linear elements (i.e. strokes) and whose relative salience can then be established. Because of the general correspondence between the perceptual salience of strokes and their functional importance in the network the relative importance of the network arcs can be derived. In situations where this correspondence is weak the salience measures can be adjusted to better reflect the functional importance of road segments. The deletion of the strokes according to the resulting sequence provides a simple and effective method of generalizing the network to the required level of attenuation.

This technique has been implemented as part of the 'GenSystem' generalization software package developed at Canada Centre for Remote Sensing. The implementation has been outlined and the effectiveness of the technique demonstrated.

Acknowledgments

The authors wish to acknowledge the contribution of Mr. C. Kushigbor in implementing the basic ‘stroke-building’ algorithm and of Dr Y. Kettal in generating the results of figure 6.

References

Bertin, J. (1983). Semiology of Graphics, University of Wisconsin Press, Madison, WI.

Biederman, I., Hilton, H. J., and Hummel, J. E. (1991). Pattern goodness and pattern recognition. In: J. R. Pomerantz and G.R. Lockhead (Eds.), *The Perception of Structure*, American Psychological Association, Washington DC.

Borchert, J. R. (1961). The Twin Cities urbanized area: past, present, future, *The Geographical Review*, 51(1), 47-70.

DeLucia, A., and Black, T. (1987). A comprehensive approach to automatic feature generalization, *Proceedings of the 13th International Cartographic Conference* (Morelia, Mexico), 168-191.

Hillier, B. and Hanson, J. (1984). *The Social Logic of Space*, Cambridge University Press, Cambridge.

Horton, H. (1945). Erosional development of streams and their drainage basins, *Bulletin of the Geological Society of America*, 56, 275-370.

Lowe, D. G. (1985). *Perceptual Organisation and Visual Recognition*, Kluwer Academic, Boston.

MacEachren, A. M. (1995). *How Maps Work: Representation, Visualization, and Design*, Guilford Press, New York.

Mackaness, W. A. (1995). Analysis of urban road networks to support cartographic generalization, *Cartography and Geographic Information Systems*, 22(4), 306-316.

Palmer, S. E. (1983). The psychology of perceptual organisation: a transformational approach. In: J. Beck *et al.* (Eds.), *Human and Machine Vision*, Academic Press, New York, 269-339.

Penn, A. (1993). Intelligent analysis of urban space patterns: graphical interfaces to precedent databases for urban design, *Proceedings of Auto Carto 11*, 53-62.

Richardson, D. E. (1993). *Automatic Spatial and Thematic Generalization using a Context Transformation Model*, Doctoral Dissertation, Wageningen Agricultural University, Netherlands.

Richardson, D. E. (1994). Generalization of spatial and thematic data using inheritance and classification and aggregation hierarchies. In: T. C. Waugh and R. G. Healey (Eds.), *Advances in GIS Research 2*, Taylor and Francis, London, 957-972.

Richardson, D. E., and Thomson, R. C. (1996). Integrating thematic, geometric and topological information in the generalization of road networks. *Cartographica*, 33(1), 75–83.

Rusak Mazur, E., and Castner, H. W. (1990). Horton’s ordering scheme and the generalisation of river networks, *The Cartographic Journal*, 27, 104-112.

Thomson, R. C. (1999). Interpreting temporal relations in rock micrograph and geological section images, *Proceedings of the 1st International Symposium on Imaging Applications in Geology* (Liège, Belgium).

Thomson, R. C., and Claridge, E. (1989). A ‘computer vision’ approach to the analysis of crystal profiles in rock sections, *Proceedings of the 6th Scandinavian Conference on Image Analysis* (Oulu, Finland, July), 1208-1215.

Thomson, R. C., and Richardson, D. E. (1995). A graph theory approach to road network generalization, *Proceedings of the 17th International Cartographic Conference* (Barcelona, Spain), 1871-1880.

Töpfer, F., and Pillewizer, W. (1966). The principles of selection: a means of cartographic generalization, *The Cartographic Journal*, 3(1), 10-16.

Witkin, A. P., and Tenenbaum, J. M. (1983). What is perceptual organization for?, *Proceedings of the 8th International Joint Conference on Artificial Intelligence* (Karlsruhe, Germany), 1023-1026.

Session / Séance 47-C

The Application of Agents in Automated Map Generalisation

Sylvie Lamy *and* **Anne Ruas**

Institut Geographique National, Laboratoire Cogit, 2 Av Pasteur B.P. 68, 94 160 Saint-Mande, France
Lamy@cogit.ign.fr http://agent.ign.fr/

Yves Demazeau

INPG, LEIBNIZ/IMAG 46,Avenue Felix Viallet F-38031 GRENOBLE, France

Mike Jackson

Laser Scan, Cambridge Science Park, Milton Rd, Cambridge CB4 4FY, UK

William A. Mackaness

Department of Geography, University of Edinburgh, Drummond St, Edinburgh, EH8 9XP, Scotland

Robert Weibel

Department of Geography, University of Zurich, Winterthurerstrasse 190, Zurich, Switzerland

Abstract

This paper reports on current research utilising agent based methodologies in order to provide solutions in autonomous map generalisation. The research is in pursuit of systems able to support the derivation of multi scaled products from a single detailed database with minimal human intervention in the map compilation process. Such research has important implications for automated conflation (multiple database integration),and is in response to the huge growth in provision of digital map data over the Internet, coupled with a broadening community of map users who wish to visualise information in a variety of ways but who have little cartographic skill.

Introduction

This research is driven by a desire to be able to automatically derive multi-scaled multi-themed maps from a single detailed database. The research is of direct relevance to National Mapping Agencies (in the management and value added reselling of national series) and in access and retrieval of information over the Internet (Buttenfield 1997; Davies 1997). The derivation of smaller scaled maps from a detailed source is figuratively shown in Figure 1. The idea being that Figure 1b and 1c could be automatically derived from Figure 1a. In theory this can be achieved using a mix of generalisation techniques (such as selection, merging, displacing, symbolising, simplification).

Getting the right 'mix' of methods has revealed map design to be a complex spatial decision making process that operates at a number of design levels - coupling broad scale objectives such as the overall homogeneity of the map with fine detail refinement such as the small displacement of two objects to improve their legibility. The challenges in providing autonomous generalisation systems are many. We require methods to analyse map

Figure 1: 1:25 000 1: 100 000 1: 250 000 (Copyright of the IGN).

content and to measure the patterns inherent among a set of geographic objects (analysis). We require generalisation methods that enable us to manipulate objects in the map space in order to create candidate solutions (synthesis), and we require methods to evaluate the solutions in order to refine and measure the success of the design (evaluation). Most critically we require a framework that enables us to model design at various levels of granularity, from the map as a whole (the macro level) through to the fine detail (the micro level). It is the absence of this framework that has stymied progress in autonomous systems and why existing systems typically require intensive interaction with the user. This paper discusses the use of multi-agent systems for providing such a framework. In simple terms an agent is a self contained program capable of controlling its own decision making and acting, based on its perception of its environment and itself, in pursuit of one or more objectives. With respect to cartography, the map is the environment, and the objective is to resolve design problems at a number of scales and resolutions.

The paper discusses various qualities of agents that provide a better way of modelling the complex decision making process of design and goes on to consider the information requirements in order for agents to act autonomously. The paper begins by exploring the current context of research in map generalisation before providing an overview of agents and multi-agent systems. Agents are not considered to be a utopia that obviates the need to tackle many of the problems identified by recent research in map generalisation. However they do offer a more transparent means by which we can model the complexities of map generalisation, in particular the often competing goals of map design and the complexities of grouping phenomena in a meaningful way.

Why Agents?

Generalisation in its epistemological sense, is a process that attempts to establish the universality of a statement (Hawkins 1983). In other words, generalisation is all about answering the question - ‘how may the phenomena being studied be ordered or grouped?’ (Harvey 1967, 82). The objectives of ‘Map generalisation’ precisely mirror this question but with the added challenge of visualising these generalised phenomena in an effective and efficient manner. Whilst we can point to many interesting developments in automated cartography, it is the failure of vendors and researchers to acknowledge the importance of strategy in design that has led to the development of map production systems that are complex and tedious to use, and require user intensive interaction and guidance during the design process. In short these systems have attempted to automate the movement of the cartographic hand, they have done nothing to model the thinking behind the movement of that hand. This research on multi-agent systems in cartography is driven by a desire to address such shortcomings whilst offering a framework in which to address a number of critical issues in map generalisation research (Table 1).

Table 1: ‘Needs’ in generalisation research.

Specifically contemporary research in map generalisation has highlighted the need:

- to understand the underlying philosophy and objectives of map generalisation
- to view the map as a system of relationships rather than points, lines and areas
- to model the interdependence that exist naturally between geographic objects
- to model the sequence and degree of application of generalisation methods
- to define goal states and model the competing nature of goal states
- to understand the links between goal states and the application of a toolbox of generalisation methods
- to understand the context (spatial and thematic) in which generalisation takes

Where did the idea of agents come from?

Ant colonies are an example of a large society with apparently co-ordinated and co-operative behaviour that results in rather complex space time events, namely the building and maintenance of an ant hill, gathering of food, defence, the survival of the community and its colonisation of new sites. That such simple folk should be capable of such sophisticated co-operative events is intriguing. It is this concept of a society of co-operation among simple folk within a shared environment that has led to the study of ‘agents’ - a collection of simple operations, operating in a co-ordinated manner to achieve a cohesive collective goal. Their use is driven by a motivation to improve computer systems and to make them easier to design and implement, more robust, and less error prone. There is no precise agreement on what constitutes an agent, but one definition proposed by Luck is that an agent is ‘a self contained program capable of controlling its own decision making and acting, based on its perception of its environment, in pursuit of one or more objectives.’ (Luck 1997, 309). Where more than one agent exists, we can define what are called multi-agent systems (MAS): Multi-agent systems are ones in which several computational entities, called agents, interact with one another. The concept of an agent implies a problem solving entity that both perceives and acts upon the environment in which it is situated, applying its individual knowledge, skills, and other resources to accomplish high-level goals. Agents thus integrate many of the algorithms and processes that have been independently studied by researchers in artificial intelligence and more widely in computer science. Much of the conceptual power of this exciting new paradigm arises from the flexibility and sophistication of the interactions and organisations in which agents participate. Because an agent is relatively self-contained, it has a considerable degree of freedom in how it interacts with other computational and human agents. The study of multi-agent systems concentrates on the opportunities and pitfalls afforded by this freedom. Agents can communicate, co-operate, co-ordinate, and negotiate with one another, to advance both their individual goals and the good (or otherwise) of the overall system in which they are situated. Agent societies can be structured and mechanisms instituted to encourage particular kinds of interactions among the agents. Populations of agents acting on their individual perspectives can converge to systemic properties. Teams of agents, each providing a particular suite of capabilities needed by one another, can be constructed and deployed to collectively solve problems that are beyond their individual abilities. This teaming can even be done on the fly, and can include humans as well as heterogeneous computational agents (Demazeau 98).

With respect to cartography, this is translated into the goal of wishing to resolve design problems at both the local and community (or global) level. The various compromises between the local and global elements of a society of agents means that a sub-optimal but acceptable solution can often be reached. Using a multi-agent approach enables us to model: the roles of autonomy, communication, operation, co-ordination, and negotiation.

A Brief Comment On the Complexities of Map Design

Before discussing the application of agents to the cartographic domain, it is worth reminding ourselves of the essential qualities of design. Map design is essentially a decision making process and broadly includes three stages: intelligence gathering (analysis), design of solutions (synthesis), and choice and review of solution (evaluation). The human achieves this collectively/cohesively through an encompassing strategy that involves working at multiple scales/resolutions (localized design and broad overview design), manipulating complex object types that have multiple, scale dependent geometries, in order to reach an acceptable design solution. The objective of map generalization is to both conserve and convey the essence of the relationships among a set of geographic phenomena. This process takes place in a dynamic environment - at any instant there may be a large number of possible solutions, the chosen solution influencing consequent choices and actions. At any one stage during the design, there exists a large number of candidate solutions which can be created by applying generalization operators to a mix of objects, in vary degree and in varying sequence. Each solution is constrained by a desire to achieve certain goals. During generalisation we wish to achieve a set of goals:

- maintain clarity and legibility (defined as a minimum separation between objects, a minimum size, a minimum difference in symbology utilizing the 6 variables defined by Bertin(1983))
- to retain the quality of the objects (their defining characteristics in terms of location, shape/distribution/ homogeneity, and defining qualities such as location, connectivity, orthogonality, association)
- to retain a level of information content commensurate with scale

Why apply agents to cartography?

When viewed in this manner, it is clear that map design lends itself to the application of agent methodologies. Indeed this research is premised on the idea that there is something in the process of map design that is analogous to our ant community. Similar to the ants, there appears to be a one to one mapping between the description of agents and the objectives of automated cartography. The agent paradigm in artificial intelligence is based upon the notion of reactive, autonomous, internally motivated entities embedded in changing, uncertain worlds which they perceive and in which they act. With respect to automated cartography, that world is the evolving map space to which objects are added, merged, symbolised and taken away. We have

1) a (hierarchical) set of competing goals or tasks (defined above),

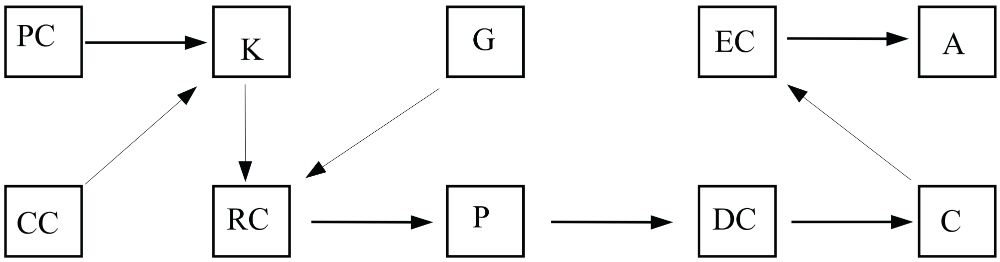
2) we understand the importance of sequence and believe there are heuristics (rules of thumb) governing sequence (Ruas and Mackaness 1997)

3) the need for compromise across scales - resolving localised / autonomous solutions whilst at the same time considering the map as a whole.

For this we can define a set of agents whereby each agent is capable of performing a specific task pertinent to map design. The operations of each agent are constrained by the protocols of what is ‘acceptable design’. Acceptable design born from the idea of a ‘design policy’ at a number of conceptual levels, constraining/ modifying the activities of the individual - defining what is acceptable. The agents work together, collectively, sharing in their successes and failures, the goal is a distributed set of activities that results in the construction of a map, having specified scale and theme. The essential components to support this process are 1) a capacity to perceive and communicate between agents, 2) a knowledge based on which to draw heuristic design information, 3) reasoning and design capabilities, 4)a capacity to create a set of choices, achievable through a set of plans, 4) driven by a desire to achieve a set of goals. These essential elements are summarised in Figure 2.

The sequence by which the agents act is summarised in Figure 3, which encompasses the processes of analysis, synthesis (proposals for solving a given design problem), and evaluation in assessing the success of the chosen proposal.

A large number of techniques are now being developed to support the analysis phase and include measures of shape, pattern, topology, and distribution. Implicit in the development of these techniques is the idea that if you wish to preserve some quality of the map or any object in the map, you first need to characterise it. Various research has highlighted the need to characterise the phenomena in order to 1) drive the solution and 2) to ensure that the solution is recognisable as being a generalised form of the source data. Research has focused on modelling qualities such as connectivity, sinuosity, alignment, relative size, and compactness (Regnauld1996). Cartometric techniques will also be required to model distributions in order to maintain the homogeneity of the map content. Such information will also play an important role in the provision of information for co-operation among agents.



PC	perception capabilities (first hand)	DC	decision capabilities
CC	communication capabilities (second hand)	EC	executive capabilities
K	Knowledge	G	Goals
RC	Reasoning capabilities	P	Possible plans
A	Actions	C	Choices
INPUT	perception stimuli	OUPUT	action on agents or environments

Figure 2: The various components of an agent based methodology (Demazeau 1990, 6).

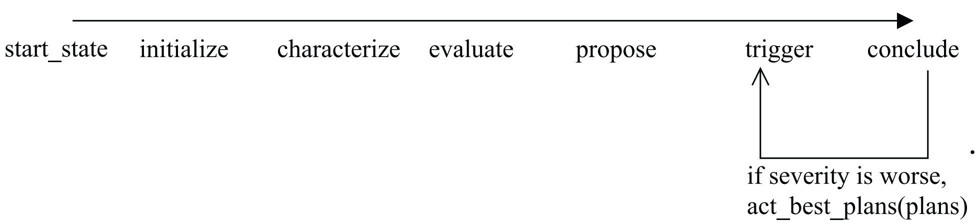


Figure 3: Agent state diagram (simplified) (Ruas 1999)

Development of Methods For Generalising Geographic Phenomena

The classification and description of geographic phenomena is central to the generalisation process. From a pragmatic point of view, we require meaningful ways of generalising phenomena whilst retaining their distinguishing characteristics and their interdependencies with other phenomena. We know that it is necessary to prioritise certain qualities and characteristics that define the phenomenon being represented. Their description is a prerequisite to this abstraction process. Furthermore if we are to observe notions of homogeneity, then by definition we need to prescribe the regions we intend to compare. It is therefore apparent that we need to define agents in terms of their overall tasks and the scale dependent nature of their activities. It is proposed that micro agents be created to manage the generalisation of individual geographic phenomenon. That meso agents be devised to manage groups of objects, and that macro objects are at a coarser scale still, involved in the broadscale issues of map design. The challenge is in deciding the most appropriate level at which to group phenomena together.

It is important that when considering the grouping of phenomena, we not only consider it at the geometric level but at the semantic and topological level (Ormsby and Mackaness 1999). For example a residential suburb is a geographic phenomenon. It is made up of houses, of relatively high density, roads, small shops, and is away from a city centre. And by way of a further example, a city is a phenomenon made up of suburbs, industrial sectors, shopping precincts, schools and transportation infrastructure. These examples clearly illustrate that phenomena can easily be complex collections of other phenomena. It is important to stress that the composition of these phenomenon may vary - perhaps driven by the thematic intent, or the intended scale transition and that one element or phenomenon might contribute/be part of more than one other phenomenon. Precisely how these phenomena might be formalised or prescribed is an important part of the research and is critical to the success of applying the agent paradigm in the map generalisation process. There is a close link between the generalisation of such phenomena and the way in which we partition the map space. The partitioning of the map space is required in order to allocate tasks and responsibilities between the different types of agents. One could partition based on some geographical distinction, such as the rural/urban divide, or a mountain/ valley divide. You could partition based on some geographic function such as the river catchment zone that defines the region into which a river flows. Alternatively one could partition on the basis of some anthropogenic feature. Popular among these has been the use of road networks to partition the map space. For the agent project it is likely that partition will occur using a mix of these partitioning mechanisms, depending on the task in hand. For example some generalisation techniques are more appropriately applied to urban regions that rural ones, and being able to partition the map space along some urban/rural divide might therefore be required (Mackaness 1995). Figure 4 shows one such (hierarchical) arrangement of agents across the building/district/ town divide, and Figure 5 shows how the roads can be used to partition the map space into meaningful chunks.

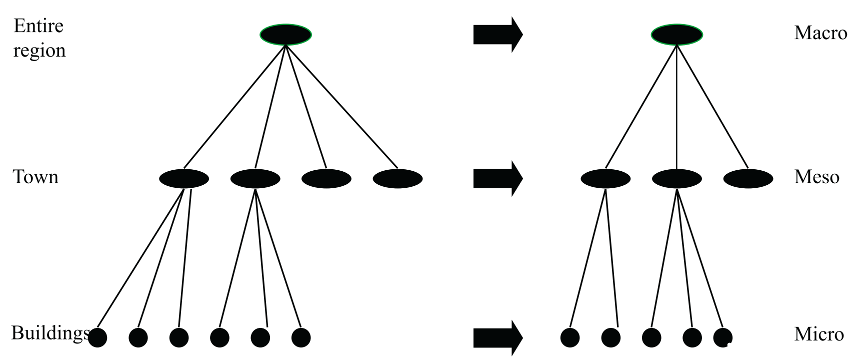


Figure 4: A Hierarchical structure of micro, meso and macro agents (adapted from Ruas 1999).

Figure 5: Using the road network to partition the map space in terms of responsibilities and activities between meso agents. Each 'smiley' represents one meso agent.

A Worked Example

Having devised a structure for agents, and identified the tasks associated with each agent type, we require a way of modelling the competing goals of design at each of these three levels. In the example below, we consider the activities of a micro agent, and show how a constraint based approach can be used to find compromise between a set of (competing) goals.

The methodology is centred around a constraint based approach to generalisation (Harrie 1999; Ruas 1998; Weibel 1996).

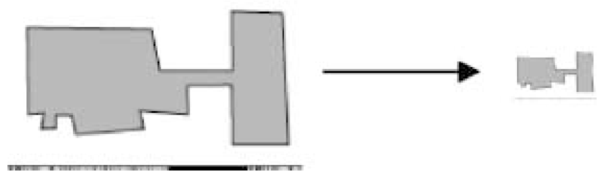


Figure 6: Scaling a building generates noise in the detail, and narrow sections are indiscernible.

In the following example we consider the constraints associated with a micro agent which represents an individual building. In the analysis phase, various qualities of the building are measures. These measures include size, minimum width, how square it is, its orientation, position, compactness. All these measures will influence how the object is generalised. Certain characteristics we wish to conserve (such as overall shape, its angular nature, location, and size relative to other buildings), and other characteristics we wish to alter in order to maintain legibility, and to support ease of interpretation. Some of these tasks are handled at the meso level (such as separation, or common/relative orientation). Collectively the result is a compromise among these constraints, at the micro, meso and macro level. In the figure below, various characteristics have been measured and an assessment is made as to whether the object will be discernible at the target scale. The narrow width will become illegible at reduced scale, and the fine detail in the boundary will not be visible at coarser scales. These qualities therefore need to be altered by applying methods to the agent building which will alter its form.

Table 2: A set of measures associated with the building agent.

Goal	Required value	Measure	Current value	State
Size	> 300 m^2	Poly_area	318m^2	Goal satisfied
Minimum width	> 20 m	Min_width	11m	**Goal unsatisfied**
Square_Angle_Dev	< 3 deg	Angle_Deviation	5.2 deg	**Goal unsatisfied**
D_Orientation	< 0.1 rad	Orientation_MBR	0.0 rad	Goal satisfied
D_Position	< 20m	Hausdorff_Distance	0.0m	Goal satisfied
………..				

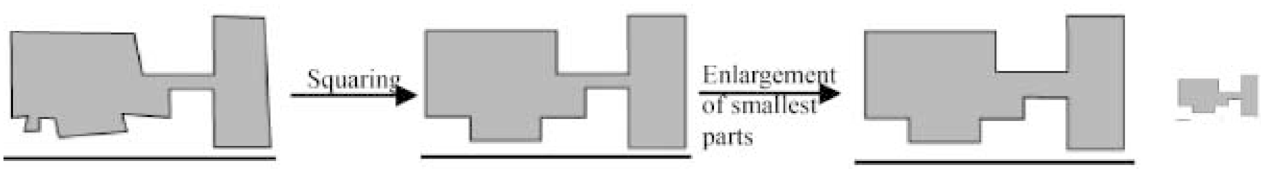


Figure 7: Squaring and selective enlargement leading to changes in constraint values (adapted from Ruas 1999).

From table 2 we see that two goals are unsatisfied. The narrow section of the building is indiscernible and the fine detail cannot be preserved - the building has lost its anthropogenic feel (squareness). The resulting generalisation methods are first to square the building, and then to enlarge the width of the narrow section. This results in changes in the total area of the footprint of the building. These changes are figuratively illustrated in Figure 8, in which just three of the goal states have been normalised against each other. Provided any changes don't have an untoward effect on other constraints (raising a bar of the histogram above the line) then we can essentially define 'an acceptable solution' whereby the changes are sufficient for the object to be legible (Figure 7), but not sufficiently great to alter the general image of the footprint of the building.

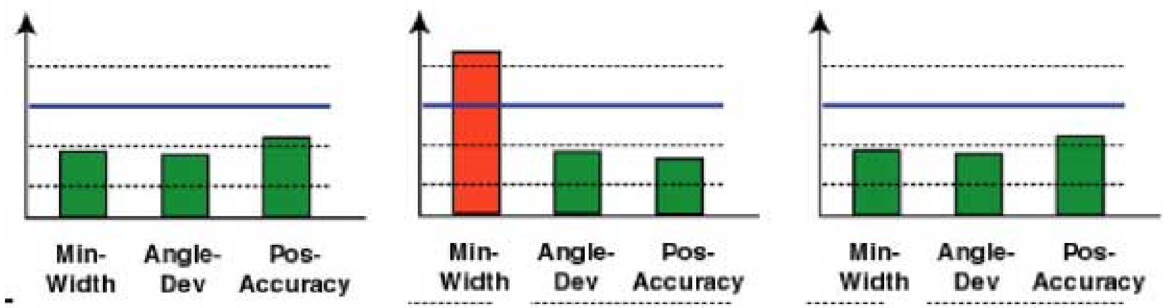


Figure 8: Modelling compromise between a competing set of design constraints (adapted from Ruas 1999).

The idea proposed for a micro agent is equally applicable for meso and macro agents, though different criteria would form the bars of the graphs.

Issues in the Use of Agents

Agent methodologies can be viewed as a natural progression to solving issues of automated map design (Baeijs et al. 1996). The development of object oriented techniques and the ideas of reactive databases are symptomatic of attempts to build greater intelligence into databases and to provide the functionality to model explicitly the relationships that exists between map objects. But there are many issues that still need to be addressed. Werner () warns us of believing that by throwing together a few ants, we can build anthills and by analogy that by throwing together a few cartographic agents that we can build maps. In particular current research is trying to understand:

- What are the levels of cooperation achievable betweeen a set of agents?
- What is the finest level of detail at which agents are defined?
- What information is shared between agents?
- How do we model sequence in activities of agents?
- How do we model an agent’s autonomy?

Such questions and more will need to be addressed during the lifetime of the project. The emphasis of these research questions is reflected in the composition of the agent consortium. The five institutions comprising the consortium have expertise ranging from a knowledge of the map user community, the agent methodology, research in map generalisation and R&D in commercial OO based GIS. The AGENT project is eighteen months into a three year research contract. The lead institution is the IGN, the national mapping agency of France. The collaborators are: INPG, Grenoble; Laser Scan in Cambridge; the Department of Geography, University of Zurich; and the Department of Geography, University of Edinburgh, Scotland. Collectively the teams are working in five critical areas:

- development of cartometric techniques
- definition of behavioural constraints of agents
- implementation/ prototyping of generalisation methods
- methods for partitioning of the map space for workflow decomposition
- methods for the meaningful grouping of phenomena

Conclusion

Agents are all about managing complexity and provide a fundamentally new way of considering complex distributed systems, containing societies of autonomous cooperating components. One should not infer that by utilising multi-agent systems (MAS) that the current impediments to automated map generalisation will easily be addressed or that MAS is a better approach than procedural approaches, OO, XS, neural networks or other approaches previously adopted (Muller 1993). Indeed the use of the MAS paradigm has highlighted the fact that these needs must be addressed. What MAS does offer is a new perspective on the problem - the 'right' framework in which to understand and model the generalisation process.

Mistakenly map generalisation has, in the past, been seen simply as a set of geometric manipulations. It is true that generalization manifests itself as the manipulation of geometry, but it is fundamentally driven by the need to convey specific meaning with respect to a particular map purpose. In reality the process of design has been shown to be complex, necessitating the modelling of geographic phenomena sufficient to support both the generalisation of that phenomenon, exploratory design, and effective visualisation (conveying meaning).

Acknowledgement

The authors are very grateful for funding for this research from the European Union via the Esprit funded AGENT project, 24 939 LTR. This paper reflects the combined effort of all researchers within the consortium. Our thanks for comments from all researchers involved in the AGENT project including Mats Bader in the provision of figure 5, and to the IGN in the provision of data.

References

Baeijs, C., Y. Demazeau and L. Alvares. 1996. SIGMA: Application of Multi-Agent Systems to Cartographic Generalization. In7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAA-MAW'96, vol. pp. 163-176.

Bertin, J., 1983, Semiology of Graphics: Diagrams, Networks, Maps: Madison, University of Wisconsin Press.

Brazile, F. and Edwardes, A. (submitted) Organizing Map Space for Generalization through Object-primary Partitioning, COSIT99.

Buttenfield, B. P. 1997. Delivering Maps to the Information Society: A Digital Library for Cartographic Data. InProceedings of the 18th ICA/ACI International Cartographic Conference, vol. 3, pp. 1409- 1416.

Davies, C. 1997. Organisational Influences on the University Electronic Library. Information Processing and Management, vol. 33, no. 3, pp. 377-392.

Demazeau Yves, Durfee Edmund, Georgeff Michael, and Jennings Nick, (1998) Preface, in Demazeau, Y., ed., Proceedings of the 3rd International Conference on Multi-Agent Systems, ICMAS'98, pp. xiii-xiv, IEEE, July 1998.

Demazeau Yves, and Müller Jean-Pierre, (1990) ³Decentralized Artificial Intelligence², in Demazeau, Y., and Müller, J.-P., eds., Decentralized A.I., 1st European Workshop on Modelling Autonomous Agents in Multi-Agent World, MAAMAW'89, pp. 3-13, North-Holland, Elsevier, August 1990.

Harrie, L. E., (1999), The Constraint Method for Solving Spatial Conflicts in Cartographic Generalisation: Cartography and Geographic Information Systems. v. 26, p. 55-69.

Harvey, D. 1967. Explanation in Geography. London: Edward Arnold.

Hawkins, J. M. 1983. Oxford Paperback Dictionary. Oxford: Oxford University Press.

Lamy, S., 1998, The AGENT Project: Automated Generalisation New Technology: Proceedings of the 4th EC-GIS Workshop (Published by JRC).

Luck, M. 1997. Foundations of Multi-Agent Systems: issues and directions. Knowledge engineering Review, vol. 12, no. 3, pp. 307-318.

Mackaness, W. A., 1995, Analysis of Urban Road Networks to Support Cartographic Generalization: Cartograpy and Geographic Information Systems, v. 22, p. 306-316.

Muller, J. C., Peng, W., and Wang, Z., 1993, Procedural logical and neural nets tools for map generalization: 16 International Cartographic Conference, vol 1, p. 181-191.

Ormsby, D., and Mackaness, W. A., 1999, The Development of Phenomenonological Generalisation Within an Object Oriented Paradigm: Cartography and Geographical Information Systems, v. 26, p. 70-80.

Regnauld, N., 1996, Recognition of Building Cluster for Generalization: Proceedings of the 7th International Symposium on Spatial Data Handling, p. 185-198.

Ruas, A., 1995, Multiple Paradigms for Automating Map Generalization: Geometry, Topology, Hierarchical Partitioning and Local Triangulation: Proceedings of Auto Carto 12, p. 69-78.

Ruas, A., and Mackaness, W. A., 1997, Strategies for Urban Map Generalization: Proceedings of the 18th ICA/ACI International Cartographic Conference, p. 1387-1394.

Ruas, A., 1999, Modèle de géneralisation de données géographiques à base de constraintes et d'autonomie, Thèse de doctorat de L'université de Marne La Vallée, Paris, Unpublished PhD Thesis.

Weibel, R., 1996, A Typology of Constraints to Line Simplification: Proceedings of the 7th International Conference on Spatial Data Handling, p. 533-546.

Werner E 1996 What Ants Cannot Do Proceedings of the 6th European Conference on Modelling Autonomous Agents in Multi-Agent World : Distributed Software Agents and Applications, LNAI, Vol. 1069, pp. 19-39, Springer Verlag, August 1996.

Session / Séance 47-D

New Cartographic Generalization Tools

Dan Lee

Product Specialist, Cartographic Product Development
Environmental Systems Research Institute Inc.
380 New York Street, Redlands, CA 92373
Phone: (909)793-2853 Email: dlee@esri.com

Abstract

Cartographic generalization can be a part of data conversion for creating a master database or deriving a smaller scale database from that. Although the result of generalization is a new set of data, it is usually not obtained by well-defined straightforward queries as in typical data extraction, but requires more complicated decision-making The rules guiding generalization vary from case to case and ways to avoid and resolve conflicts dynamically remain under research.

Given the "classic" ARC/INFO GIS software environment at ESRI Inc., our effort of developing generalization tools has been based on the existing data model and software structure of the system. The greater part of a master database is usually made up of buildings and roads in vector form. This paper presents our experience in deriving solutions for simplifying buildings and collapsing road casings to centerlines.

Building simplification and centerline creation are complicated generalization processes that require explicitly defined rules to cover all possible cases. A complete set of rules has yet to be devised. But in the meantime, an automated process can handle majority of the problems and leave the remaining ones for post-processing. The practical solutions, therefore, have become the combinations of automated processes, Macro-program procedures, and interactive editing. This approach has simplified many existing procedures and significantly reduced production time and cost.

Strategies of deriving generalization solutions

To provide computer-assisted solutions for generalization, we need to pursue the use of advanced technology in the mapping industry, reducing or replacing the tedious manual work in production environment. Our success is measured by how much time is saved and how much consistency is generated in comparison to the traditional manual generalization on the user's end, and by the efficiency in development. Generalization tasks are comprehensive and interrelated. The following strategies are very important in defining doable tasks and producing practical solutions: breaking complicated problems down to simpler solvable problems, automating the process as much as possible and enriching the output with marked areas and process status, supporting interactive editing, and defining an overall workflow.

Defining solvable problems

In manual generalization, a cartographer draws the generalized result in context with all mapped features. Many actions, such as elimination, simplification, aggregation, displacement, and so on, can happen at the same time as drawing a feature. Attempting such comprehensive human decisions and actions in full automation would be unrealistic in today's digital cartography and drag the development into an endless effort.

Defining a solvable problem in our development means to find a particular area in a bigger problem in which certain rules can be applied and a decent result can be accomplished automatically. The automation would result in significant time saving in real production. Also quite importantly, the implementation must be constrained to a reachable time frame, for example, a product release cycle.

To distinguish the components of generalization and define solvable problems, we have divided generalization into nine categories of operation, including simplification, aggregation, collapse, exaggeration, displacement (Lee, 1996). Each may then be solved automatically by unique operators along with human interaction. An operator is a program that utilizes a special technique or an algorithm to produce the best possible results for a particular generalization task, for example, the operator BENDSIMPLIFY for simplifying lines (ARC/INFO 7.2.1 release). Unresolved issues are anticipated and will be revisited and resolved later. Examples of new generalization operators will be given in later sections.

Automating the process and enriching the output data

A computer program can do no more than what human being tells it to do. Since we lack full knowledge about generalization cases and rules, we can only program what can be clearly defined and is technically solvable. The generalization operators usually work according to user-input parameters and a set of rules. In most cases these globally set parameters and rules can be adjusted to suit local circumstances, and the generalized results are satisfactory. However, there can be some areas where the global parameters don't apply and the user's inspections and decisions are necessary or where the computation and development become too expensive to do. The way of handling these situations is either making the program interactive for user's instructions at the question areas during the process, or marking these areas for post-process.

Enriching the output data is a new concept that comes along with computer-assisted generalization. It means flagging the remaining problems and reporting information about the automated process that would help further editing. This approach has the advantage of not interrupting the automated process, not requiring user's attention during the automated process, and giving flexibility to post-processing. The examples in later sections demonstrate the use of this approach in more detail.

Supporting interactive editing

As part of our current solution to generalization tasks, interactive editing programs, such as AML (the ARC/INFO Macro Language) scripts, are being made available to help solve the remaining problems and refine the results. Each of the editing programs supports certain types of editing associated with a generalization operator and its enriched output. It invokes the ARCEDIT-based editing environment with a menu-driven interface and allows the user to queue through each flagged area and apply a unique method to modify features. Some of these methods are made by putting a series of existing commands into one menu choice, which simplifies and minimizes user's involvement, others by AML programs that compensate what is not offered directly by the existing tools.

The interactive environment also allows other features to be displayed as references to help deal with spatial conflicts, topology, and the overall balance of mapped information. All other editing tools are accessible as well, if needed.

Defining a workflow

A generalization workflow puts all automated or interactive steps into a logical sequence to accomplish a result and is usually data and scale dependent. For a particular data set, our experience in supporting a generalization benchmark has set an example of using such a workflow. The project required that we derive a 1:5000 scale topographic mapping database from a 1:1000 scale database. Each of the 17 feature layers (building, road, boundary, relief, and so on) was processed separately. Within each layer a specific sequence was set to generalize point, linear, and areal features. The results were then combined into 14 output layers. The post-editing needed was described to the client. With some modifications, this framework can be applied to other data sets.

A generalization workflow will also need to be tailored to fit various scales. The more the scale reduces, the more the generalization technique shifts from making local modifications to making global representations. For example, buildings are represented individually at large scales. They can be simplified or exaggerated up to certain extent, but still remain individual. As scale reduces, more and more buildings will be excluded, collapsed, and displaced; some will become part of urban areas. At very small scales, buildings, even urban areas will completely disappear; they can only be represented by location symbols. As a map producer (Pla, 1998) described (summarized by the author):

In generalizing buildings from 1:5000 to 1:10,000 or 1:25,000 scales, the following options are used:

Extraction of small buildings and replacing them with a minimum symbol

Simplification of the rest of buildings

Exaggeration of special buildings

Other actions, including elimination, and aggregation

When generalizing buildings to smaller scales, 1:50,000, 1:100,000 or 1:250,000, typically in urban areas, more streets disappear and aggregation of buildings is more used, besides other options.

A generalization workflow, therefore, is a result of analyzing and understanding the overall requirements (the theme, resolution, feature relationships and priorities, and so on), and putting a cartographer's thinking into the most efficient and logical operations. It is critical in the completion of a comprehensive generalization task.

In the next two sections, the two newly developed generalization operators, BUILDINGSIMPLIFY (that reduces details from building boundaries) and CENTERLINE (that collapses road casings to centerlines) are introduced. The discussion is focused on defining the solvable parts of the tasks, the enriched output, and the necessary post-processing.

Simplifying Buildings

Simplifying buildings (footprints) means finding a simpler representation of the original buildings by reducing details in their boundaries, while maintaining the essential shape and size of the buildings (Figure 1). Buildings are generally orthogonal areas, therefore simplification will preserve and enhance orthogonality (Lee, 1998).

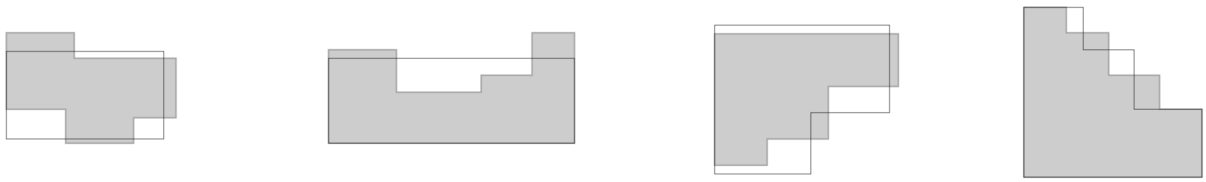


Figure 1. Buildings simplified to simpler forms.

Simplifying buildings is for applications at large scales, where buildings are still represented individually. This section discusses the complications related to this task and their solution.

Identifying the solvable task among the complications

Without worrying about spatial relationships, each building can be easily reduced to a simpler form based on a set of rules. An iterative approach can be implemented to evaluate the pre-defined conditions and rules one at a time and make modifications to the geometry accordingly (Lee, 1998a).

However, it is unavoidable that any changes in geometry will cause spatial conflicts, overlapping or crossing with neighbor features, or being too close to them. With the existing data model, it is quite difficult to detect and resolve conflicts during the process to guarantee a conflict-free result. Although in theory a conflict can be resolved within its solution space, or local region (Peng, 1997), and the displacement of involved features should gradually descend outwards from the center of the conflict, the implementation has not been proved to be easy with the current data model. Therefore, we had to define the limitations and rely on some post-editing.

Additional complications are caused by groups of adjacent buildings. If each building connected in a group is simplified separately, the shared boundaries may end up mismatched in the results. If only the outer boundary of a building group is simplified as a single building, recovering the interior walls is technically quite challenging.

It then became clear that the automatically solvable task is the simplification, with limited conflict detection, of individual buildings and buildings connected in the simplest ways. The simplification status will be recorded for later examination and post-processing.

Performing simplification and recording status

A building simplification operator, BUILDINGSIMPLIFY as an ARC command, has been implemented for the upcoming ARC/INFO release. It reduces extraneous details from building footprints according to two user-input parameters, simplification-tolerance and minimum-area. All attributes on building polygons are transferred to the output.

This operator recognizes buildings as topologically disjoint, connected with straight lines near parallel to each other, and connected in more complicated ways. Each separate building is simplified by itself. Buildings connected with straight lines are simplified as a group. Buildings connected in more complicated ways are not simplified (Figure 2).

The boundaries of disjoint buildings or buildings connected with straight lines will be enhanced so that all near-90-degree angles become exactly 90 degrees. Based on the simplification-tolerance, a building can be simplified by, for example, filling up, cutting off, or widening isolated small spaces (intrusions or extrusions), straightening a side, but keeping the measured area roughly the same as the original (Swiss Society of Cartography, 1987). Any building or group of connected buildings with a total area smaller than the minimum-area will be excluded. The maximum degree of simplification is reached when a building is reduced to a rectangle (Figure 3). Since this operator does not detect and avoid all spatial conflicts, the special REGION feature class is used as the output that allows overlapping topology in buildings.

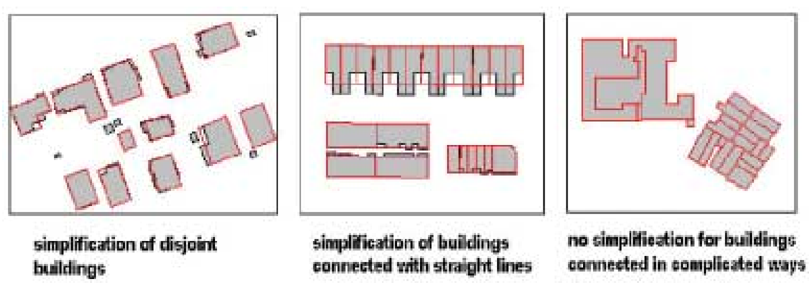


Figure 2. Buildings in three types of appearances.

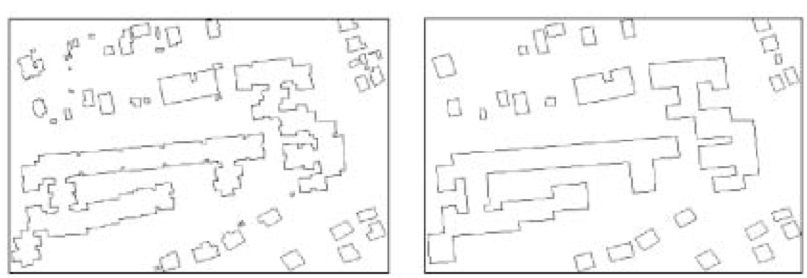


Figure 3. Before (left) and after (right) simplification.

The program records the status of each building in the output by an attribute item, BDS-STATUS. A separate building will have a BDS-STATUS value of 1 if it is completely simplified. If a spatial conflict is found during the iterative process, the building will not be simplified further and will receive a BDS-STATUS value of 2.

If the simplification_tolerance is relatively large compared to the size of the building, the building will be simplified directly to a rectangle centered at its own center of gravity. The area will remain the same. The sides of the resulting rectangle will be the same ratio as the sides of the bounding box aligned to the longest side of the original building (Figure. 4). If the resulting rectangle contains a side smaller than the simplification tolerance, the building will have a BDS-STATUS value of 3.

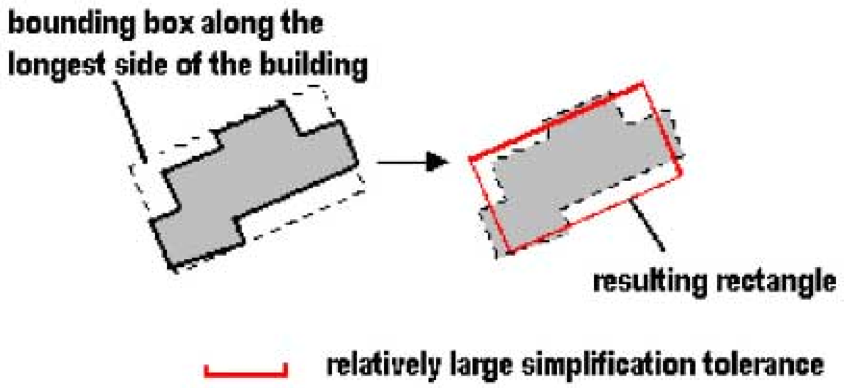


Figure 4. Building simplified directly to a rectangle.

For buildings connected with straight lines, simplification will be limited to simple rules only. These buildings will have a BDS-STATUS value of 4. And finally, buildings connected in a complicated way will have a BDS-STATUS value of 5.

The output will contain another new item, BDS-GROUP. This item stores a unique value for each group of connected buildings. A single building will receive a BDS-GROUP value of zero. This item is used in checking conflicts among buildings and groups of buildings.

Locating spatial conflicts

Once buildings are simplified, an overall conflict detection among them can be done automatically by another new ARC command, FINDCONFLICTS. This program takes the simplified buildings as input and finds where they overlap or are too close to each other based on a specified distance and on the BDS-GROUP information.

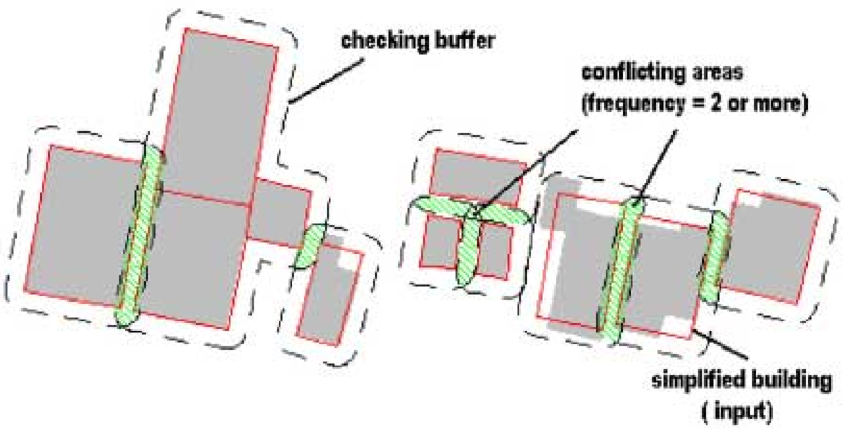


Figure 5. Finding spatial conflicts by buffering.

To find the spatial conflicts, region-buffers are created around each building or group of connected buildings. Overlapping buffers indicate a conflict. An output will then be produced, storing these region-buffers with an item FREQUENCY for polygons. A polygon gets a FREQUENCY value of 2 or more according to how many region-buffers overlap (Figure 5). All non-conflicting areas receive a FREQUENCY value of 1. Buildings connected in one group are not considered as conflicting with each other. Only the outer boundary of such a group is checked with neighboring buildings or groups of buildings.

Interactively resolving conflicts and refining the results

Given the simplified buildings with information about the status and conflicting areas, the post-editing is easier. An interactive program has been developed to invoke the ARCEDIT environment with a menu-driven interface. It allows the user to queue through each conflicting area and resolve it interactively. Two unique tools, move and delete as menu choices, were created that enable moving and deleting buildings as regions and maintain the region topology. The user can also easily select buildings according to their BDS-STATUS values and refine the result.

Collapsing Road Casings to Centerlines

Collapsing road casings to centerlines means finding the single-line representation of the casings based on specified width tolerances, while preserving proper connectivity (Lee, 1998b). It is another typical generalization problem that different algorithms (Christensen, 1999; Thomas, 1998) can be applied to solve the essential task, but a fully automated solution for dealing with real world data can be a great challenge. This section describes our approach, again including an automated process and post-editing.

Analyzing the nature of the problem and defining the solvable task

Road casings are collected and extracted in many different ways depending on user's data model and standards. What make the centerline problem more complicated than it seems include some confusing situations in road casings, difficult decisions on complex intersections, and single-line roads and wide areas mixed in data.

The first thing to do in deriving centerlines is to differentiate the inside of the casings, where the centerlines should be, from the spaces between them. If the casings representing a connected road network form a closed polygon, then it would be easy to use the inside of the polygon to guide the centerlines. But casings are not normally collected that way and it is not easy to either make such polygons or find the inside areas without such polygons or other references automatically. It is especially confusing when a number of casings run parallel with their widths within the specified width range or when casings have similar widths as the spaces between them.

Simple road intersections, such as a "T"-shaped or a "+"-shaped intersection, are easy to recognize and to solve automatically, while arbitrarily shaped intersections (Figure 6) need more rules to guide the way to connect them. The derived centerlines may not always connect nicely at one point. They need to follow priorities and connect in certain order. For example, at a three-way intersection, the two centerlines that are almost on a straight line or near perpendicular are connected first and then the third centerline can be projected or extended to the first connecting line. Applying the similar logic, it is possible that at a multiple-way intersection, all centerlines get connected, but the connection can be unsatisfactory because some lines can be projected too far or too many connecting points are produced. To certain extent, rules can be made to further refine and adjust these details, but some extremely complicated intersections may need human inspection and decision to help solve them.

Figure 6. Arbitrarily shaped intersections.

Ideally, casings represent the edges of roads. But quite often the casings also follow the edges of wide areas, such as parking lots, open areas, or unusually shaped cul-de-sacs, and single-line roads may be included in the same data as well. These features may not be attributed differently and it is not a trivial job to automatically recognize them so that centerlines can be produced and connected to them properly.

Our goal was to provide a generic solution without restrictions on the input data. We defined an approach that uses an automated program to create centerlines and simple intersections (up to four-way intersections), and yet requires post-editing to resolve the remaining intersections, to remove unintended centerlines, and to make corrections in other areas marked by the program.

Creating centerlines and necessary attributes

A new operator, CENTERLINE as an ARC command, has been implemented also for the upcoming ARC/INFO release. It produces centerlines according to two user-input parameters, maximum-width and minimum-width.

The CENTERLINE program scans and separates the data in two directions, horizontal and vertical, and creates centerlines where casing width is within the specified range. It then evaluates each intersection and connects those that fit the rules. Casings with the width beyond the specified range and single lines not used to create centerlines are copied to the output data such that the road network won't be broken.

The output data contains an item LTYPE (line-type) that differentiates centerlines from unresolved areas. If the width of the input data is relatively constant and the intersections are simple, a complete centerline result can be produced, that is, LTYPE = 1 for all resulting lines. Otherwise, unused lines (such as a single casing or casings with a width beyond the specified range) and outlines around complicated intersections will be flagged with a LTYPE value of 2 for editing them further (Figure 7).

Four other new items, LL# (left casing record number), RL# (right casing record number), L-ID (left casing ID), and R-ID (right casing ID), also come with the output data to relate the centerlines to their source casings for attribute transfer.

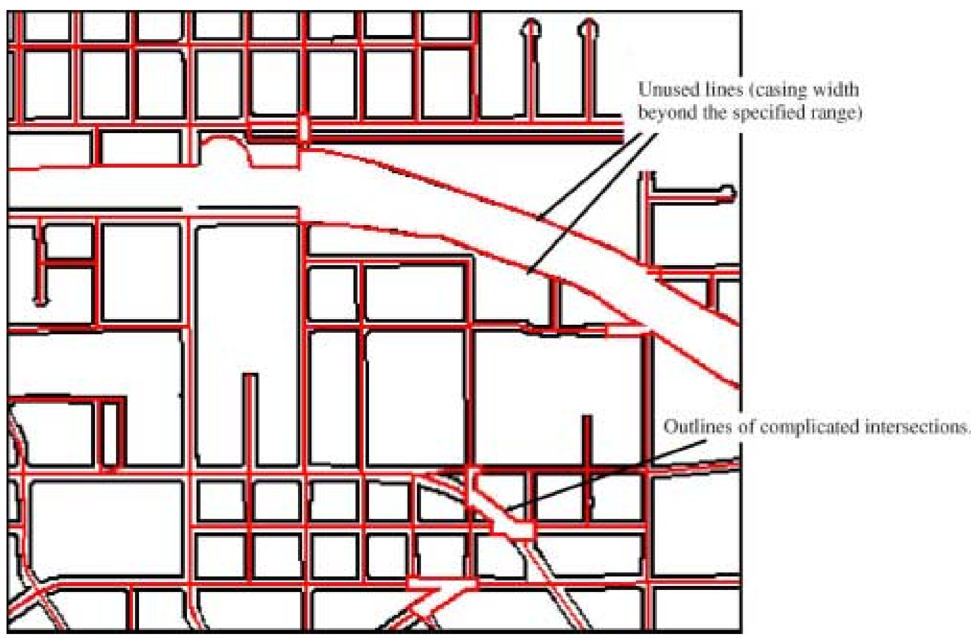


Figure 7. Results of CENTERLINE with unresolved intersections and wide roads flagged.

Interactively resolving intersections and refining the results

An AML program has been developed to invoke the ARCEDIT environment with a menu-driven interface. It allows the user to queue through each LTYPE 2 area and resolve the problem interactively. A few most needed editing tools, such as remove an arc (line), extend an arc, and add vertices, are included in the menu choices. A unique tool was created to automatically join a set of selected lines at a user-specified intersecting point. Other special tools may be added in the future to incorporate more advanced rules and make the editing even easier.

Conclusions

Our experience in deriving solutions for building simplification and creating road centerlines has led us to believe that properly balancing the automated process and interactive editing is a practical way of solving complex generalization problems reducing labor work significantly. For example, as already proved by two evaluation tests (Litton, 1998 and Oxenstierna, 1998), the CENTERLINE solution can replace existing procedures and generate 70-80% time saving in production.

We are also defining and implementing a number of other generalization operators, including aggregation of general area features, such as vegetation, and buildings in specific, collapse of buildings to points with orientations, and collapse of area features, such as lake, to single-line presentation. I would like to present our preliminary approaches in solving these problems at the Workshop and share discussions with others. As more and more of these generalization operators become available, the overall generalization workflow will continue being simplified, more labor work replaced, and more consistent results produced. Since the object-oriented technology is being used in defining our future products, it certainly will benefit generalization processes. We are looking forward to developing new tools based on stronger and richer data model and filling up more blanks in the area of digital map generalization.

Reference

Christensen, Albert H. J., (1999). Cartographic Line Generalization with Waterlines and Medial-Axes. *Cartography and Geographic Information Science*, Vol. 26, No. 1, p. 19-32.

Lee, Dan, (1998a), “Simplification of Buildings”, ESRI internal design document.

Lee, Dan, (1998b), “Creating Centerlines from Road Casings”, ESRI internal design document.

Litton, Adrien, (March 3, 1998), “On CENTERLINE approach”, email notes.

Oxenstierna, Andreas, (March 2, 1998), “On Centerlines”, email notes.

Peng, Wanning, (1997), *Automated Generalization in GIS*. ITC Publication Series, No. 50, p. 42.

Pla, Maria, (Jan. 7, 1999), “On building generalization”, email notes.

Swiss Society of Cartography, (1987), *Cartographic Generalization – Topographic Maps*. Second Edition.

Thomas, Federico, 1998, Generating Street Center-Lines from Inaccurate Vector City Maps. *Cartography and Geographic Information Systems*, Vol. 25, No. 4, p. 221-230.