Multiple Representations for Cartographic Objects in a Multi-Scale Tree - An Intelligent Graphical Zoom[1]

Andrew U. Frank and Sabine Timpf

Dept. for Geo-Information E127.1

Technical University Vienna

Gusshausstr. 27-29

A-1040 Vienna Austria

{frank, stimpf}@geoinfo.tuwien.ac.at

ABSTRACT

In Geographic Information Systems a function to draw cartographic sketches quickly and in arbitrary scales is needed. This calls for cartographic generalization, a notoriously difficult problem. Efforts to achieve automatic cartographic generalization were successful for specific aspects, but no complete solution is known, nor are there any expected within the immediate future. In practical applications, a base map is stored and its scale is changed. Without major distortions, only changes to twice or half the original scale are feasible by simple numeric scale change. Everything beyond this requires adaptation of symbols, selection of objects, placements of labels etc.

Extending ideas of hierarchies or pyramids, where representations of the same objects at different scales are stored, a multi-scale, hierarchical spatial model is proposed. Objects with increasing detail are stored in levels and can be used to compose a map at a particular scale. Applied to the particular problem of cartographic mapping, this results in a **multi-scale cartographic tree**. The same concept can be used equally well for other applications, which require rendering of objects at different levels of detail.

The structure of the multi-scale tree is explained. It is based on a trade-off between storage and computation, replacing all steps which are difficult to automate by essentially redundant storage. The dominant operation is 'zoom', which moves towards a more detailed level, intelligently replacing the current graphical representation with the more detailed one, appropriate for the selected new scale. Methods to select objects for rendering are based on the **principle of equal information density.** Principles of possible implementations are presented.

[1] Frank, A.U., & Timpf, S. 1994. "Multiple Representations for cartographic objects in a multi-scale tree - an intelligent graphical zoom". In *Computers and Graphics Special Issue on Modelling and Visualization of Spatial Data in GIS*, **18 (6)**, pp: 823-829.

Multiple Representations for Cartographic Objects in a Multi-Scale Tree - An Intelligent Graphical Zoom[2]

Andrew U. Frank and Sabine Timpf
Dept. for Geo-Information E127.1
Technical University Vienna
Gusshausstr. 27-29
A-1040 Vienna Austria
{frank, stimpf}@geoinfo.tuwien.ac.at

1. Introduction

1.1 GIS

Geographic Information Systems are a widely used tool to collect, manage and present information about the world in which we live [1]. They are used in science, for example to collect and analyze environmental data for small areas and for research in global change. But they are also used in administration for example to maintain property registers or dispatch emergency vehicles. Geographic Information Systems manage data with respect to a spatial location and the data are presented graphically as a map. There are a number of similar applications, where a database of objects with some geometric properties is used to render these objects graphically for different tasks.

Typically the tasks require graphical presentations at different levels of detail, ranging from overview screens to detailed views [2]. It becomes immediately clear that just changing the scale of the representation is not sufficient to produce useful graphical output. Cartographic tradition has developed over the past 400 years a large set of useful methods which are applied by human cartographers. They render a spatial situation such that a human observer can ’see the whole picture’ at the desired levels of detail.

1.2 Cartographic generalization

The usual approach considers the cartographic generalization as a filter, where the data from many detailed maps are filtered to produce the overview maps. This follows the traditional cartographic work flow, where overview maps are deduced from town plans and large scale topographic maps. Human cartographers use their understanding of the situation and in particular the recognition of spatial patterns to perform map generalizations.

[2] Funding from the Austrian Science Foundation (FWF) for the ESPRIT-AMUSING project and from Intergraph Corporation is gratefully acknowledged.

Cartography has developed over hundreds of years rules to communicate geographic information graphically. A skilled cartographer considers both application domain concepts and the rules of graphic communication [3, 4]. She optimizes the graphics to communicate spatial information. Buttenfield and McMaster give a comprehensive account of the current research trends [5]. Impressive results for special cases are known [6, 7, 8], but no complete formalization of the generalization rules has been achieved, nor is a satisfactory automatic solution available.

Today computers cannot produce automatically maps of the quality drawn by human cartographers. Also, improvements are not likely in the next few years, as the process requires human understanding and is therefore on par with other difficult AI problems, e.g. natural language translation. Nevertheless, an approximative method for automatic production of cartographic sketches is necessary, because there are many situation where spatial data must be rendered quickly. The effort of employing a human cartographer to produce a high quality output is often not economically justified or the time is not available for this laborious human process. This is most pressing for the graphical output production in spatial query language processors [9].

1.3 Approach

The approach selected here is to construct a multi-scale tree, where renderings for objects are stored at different level of detail. This avoids the difficulty of automated generalization at the expense of storage. The output map is constructed as a top-down selection of pre-generalized cartographic objects, till sufficient level of detail is achieved. It is assumed that the multiple representations of the same object are either produced automatically, with a computer assisted tool [10] or are collected from manually drawn maps at different scales.

The multi-scale tree has been presented in a cartographic context [11], but the method is broader in its range of application. It seems to be generally useful in computer graphics, e.g. for the visualization of non spatial data [2]. It is related to quad trees [12, 13] and strip trees [14] but tries to generalize the hierarchical concept [15, 16] to include object representations of different dimensionality.

Practical applications exist in mapping agencies and map production companies, where maps covering the same area but with different scales are maintained. The few companies that use extended automation still treat each map as separate file [17] but merging maps which differ only in scale in a single database would result in substantial savings during maintenance [11]. These mapping agencies could produce multi-scale trees and provide them to users to be used in conjunction with spatial query languages.

This paper is structured as follows: The next section casts the problem of graphical rendering at different levels of detail in an abstract model. Then different concepts for zooming are analyzed and the principle of constant information density deduced. The following sections describe the hierarchical structure and its use for intelligent zoom. The paper concludes with a performance assessment and points to particular problems requiring further study.

2. THE CONCEPT OF A MULTI-SCALE TREE

Mapping agencies have extensive collections of cartographic files, which are prepared for graphical output. Such collections are to be separated from GIS databases which model reality for other application purposes (e.g. tracking real estate property rights, modeling traffic flow).

The traditional view of the cartographic process considers three different models and two transformations which lead to the production of a visual map. There is first the model of the world, consisting of objects with descriptive data. From this model, a subset of the objects is selected to be included in a map, resulting in the set of display objects. These objects are then transformed from a geometrical description to a graphical form, applying rules for symbolization and other aspects of graphical encoding producing the set of cartographic objects. This viewpoint includes usually a strong feedback from the graphical rendering and its limitations to the selection process.

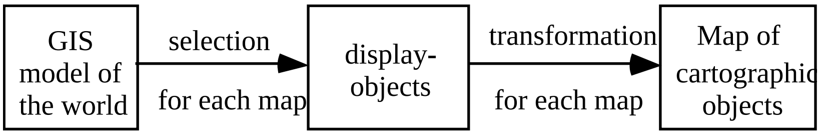

Figure 1: Traditional view of the cartographic process

The two steps of this process are reasonably understood, but despite some efforts, only very limited parts are automated. A few topics have found attention, especially line generalization [18,19], label placements for point features, [6] and generalization of built up areas [8].

The traditional view assumes that the database contains the objects at highest resolution and that procedures exist to reduce them in scale and correspondingly in detail etc. The selection step is first applied and the resulting objects then generalized to the desired level. This is - as outlined above - not practical, as automatic generalization algorithms are not available.

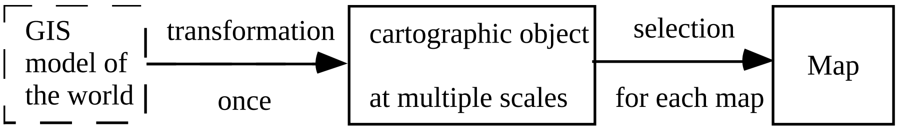

Figure 2: Selection process

This proposal stores object representations for multiple scales in the database and applies only the selection step automatically. The database will not be much larger than the most detailed database assumed in current proposals (assume that every more generalized representation is a quarter the size of the previous one, then the total storage requires only one third more capacity than the most detailed data set). Generalized representations can be collected from existing maps or for some cases produced automatically.

The idea is that every object is 'multiple represented' [10] in a tree, i.e. there are for every object multiple graphical renderings, organized in increasing graphical detail. This includes that an object may split in subobjects, each with its own graphical rendering (figure 3).

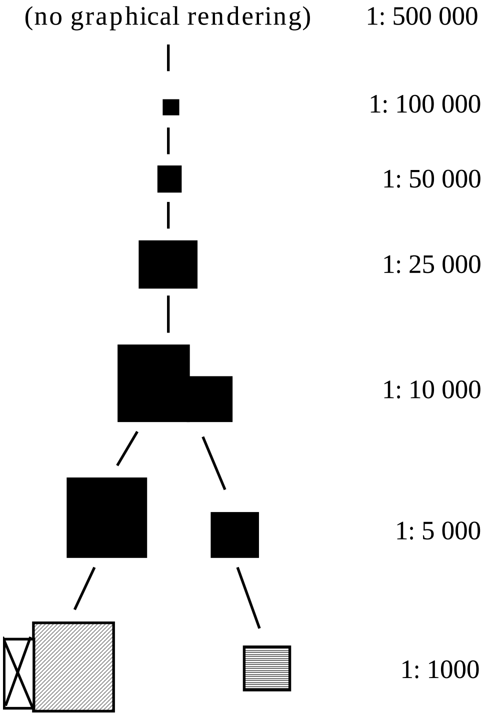

Figure 3: An example for a multi-scale tree of an object group

The novel approach taken here is to view the cartographic map change not as an operation which filters from a detailed map a generalized one, but as an intelligent zoom on a tree structure, containing all maps for the same area at different scales in a single conceptual schema. Representations of the same object in different level of detail are linked together, using different links describing the semantics of the relations.

The general rules regarding hierarchical spatial processing apply also to the intelligent zoom operation. Assuming that a rendering operation applicable to the 'flat' (non-hierarchically structured) problem is given, two points must be explained:

- a hierarchical structure and a method to transform a flat data space in an equivalent hierarchical one (see section 4),
- a rule how to apply the operation on this structure (section 5), in particular, how to switch between the levels (section 6), and
- a comparison of the results regarding correctness and performance gain (section 7).

3. Intelligent Graphical Zoom

A cartographic database consists of spatial objects which should be cartographically rendered. The user is interested in some subset of objects shown at a level of detail appropriate for the task at hand. Two operations are typically provided to the user to navigate space graphically, a *zoom* operation to get more detail and a *pan* operation to move the field of vision. These operations are fundamental as they parallel the human experience of focusing (and possibly moving closer) on an object and moving one’s glance to see other objects. Jackson has studied different solutions for the user interface [20, 21]. Here the focus is on the design principle for a data structure to support such zoom operations.

GRAPHICAL ZOOM

The zoom operation is graphical as it affects the objects and their graphical rendering. After zooming, objects are larger but less objects are in the field of view. A purely graphical zoom is simply a change of scale factor. Relative position, relative size and all other geometric relations are preserved.

A purely graphical zoom, as it is provided with most computer graphic systems is just ‘blowing up’ the image, all parts by the same factor. The level of detail remains the same. It can be implemented as pixel replication in image processing system and makes objects larger, but does not provide more detail. The same is true for scale change in CAD systems. A purely graphical zoom leads to artifacts as letters and symbols are enlarged the same amount as the content (figure 4).

Figure 4: A graphical zoom

Changing all graphical elements proportionally is not desirable. Hand drawn maps at different scales are not just blow ups, but the increase in size of objects is selectively applied. Symbols and labels for example are drawn larger, but the increase is much less than proportional.

CONTENT ZOOM

Graphical zoom must be differentiated from content zoom which affects the classification method applied. A content zoom leads to more or less differentiation between object properties (figure 5). For example a detailed schema of differentiating several dozens of soil classes is collapsed into a differentiation of only three major classes [22]. The scale to render objects at different levels of refinement of a classification schema must be adapted. For content zoom the independent variable the user selects is the differentiation level in the classification scheme, for a graphical zoom the users selects the size of his field of vision and indirectly the scale.

Figure 5: A content zoom

INTELLIGENT ZOOM

The graphical operations of ’zooming’ mimics the approach of the viewer to an object. The field of vision becomes smaller, and more detail about the objects in the field of vision appear (figure 6). This can be expressed as *the Principle of Constant Information Density.* A graphical zoom violates this principle, as it produces the same information content on a larger area, i.e. information content per area unit is reduced.

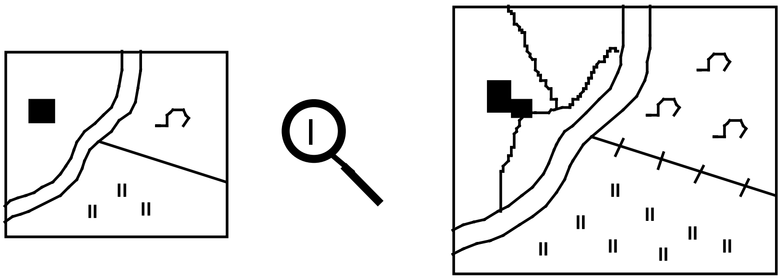

Figure 6: An intelligent zoom

In cartography the principle of constant information density is known as Töpfer's radix law [23]:

$$n_f = n_a * \_(m_a/m_f)$$

where **m_a** is the given scale,

m_f the following scale,

n_a the number of objects at the given scale, and

n_f the number of objects at the following scale

Reformulating this law in terms of viewing area and not in terms of the map scale, results in

number of objects/area = constant.

By intelligent zoom we understand a zoom operation, which respects the principle of constant information density. It implies that more detail about objects become visible as the field of vision is restricted and the scale is increased. This leads immediately to an hierarchical data structure, where objects are gradually subdivided in more details. This hierarchical structure is applied to all geometric objects, not only to lines as in strip trees [14], to a pixel representation of an area as in a quadtree [12, 13] or the pyramid structures used in image processing [16].

4. Hierarchical structure

4.1 Maps as structures

A map consists of clearly distinct features. One approach for structuring cartographic data is to consider cartography as a language with its own syntax and vocabulary [24, 25, 26]. These features are combined from a graphical vocabulary, which provides the atoms for graphical communication [4, 27, 28, 29]. Highly simplified, cartographic features can be differentiated by dimension (points, lines, areas) and the cartographic variations (object drawn as symbol, object representing a scaled representation, a feature associated with text, text without a delimited graphical feature). This results in roughly 12 categories [30, p.24]. The national standards for exchange of digital cartographic data (e.g. STDS [30], ATKIS [31]) provide elaborate description of graphical features.

4.2 hierarchization

Hierarchical subdivision of special object classes has been dealt with and was studied extensively. Strip trees [14] or a very similar design could be used to deal with lines which remain lines over multiple levels. Areal features can be represented by quad-trees as long as they remain areal features. But the multi-scale tree must also include more dramatic changes.

Considering existing map series, where the same objects are mapped at different scales, a strategy for hierarchization follows (table 1).

continuous changes	discrete changes		
	slight change	complete change	disappears
no change in appearance change according to scale	change in symbol reduction of detail deletion of label	change in dimension shift to symbol shift to geometric form splitting in several objects	

Table 1: Types of changes in the multi-scale tree

The list demonstrates that objects may change their spatial dimension in the generalization hierarchy. A particular problem is posed by objects which are not represented at small scale and seem to appear as one zooms in. Studies to identify these classes must be based on observations of hand generalized map series [32].

5. SELECTION PROCESS

The operation applicable to every node of the tree is a 'rendering' operation, which transforms the geometric data into a graphical picture. The problem to address is the selection of the objects in the tree which must be rendered. Two aspects can be separated, namely, the selection of objects which geometrically extend into a window and the selection of objects to achieve a constant information density.

The selection of objects which extend into the window is based on a minimal bounding rectangle for each object and a refined decision that can be made based on object geometry. In order to assure fast processing in the multi-scale tree, the minimal bounding rectangles must be associated with the tree and tree branching, such that complete sub-trees can be excluded based on window limits. This is well known and the base for all data structures which support fast spatial access [12].

The interesting questions is, how the depth of descent into the tree is controlled to achieve an equal information density. In data structures for spatial access, an 'importance' characteristic has been proposed [33]. It places objects which are statically assessed as important higher in the tree and they are then found more quickly. The method relies on an assessment of the 'importance' of each object, which is done once, when the object is entered into the cartographic database. When a cartographic sketch is desired, from this ordered list the most important objects are selected for rendering.

The usability of this idea is currently studied for a particular case, namely the selection of human dwellings (cities) for inclusion in a map [34]. A method based on an ordering of objects is not sufficient for the general case. It lacks provisions to deal with multiple representations of objects and must be extended for a multi-scale tree, which is designed to deal with multiple representations of the same objects. Nevertheless, substantial contribution to our understanding of the cartographic selection process is expected from the study of selection based on ordered (single-represented) objects.

A *perfect method* to achieve uniform information density requires a method to measure the information an object contributes to the map. The algorithm would then descend the tree, within the limits of the window, and refine objects till the desired level is achieved. Note the difference to a rank order selection: all objects within the window are selected initially, and the process is refining or expanding objects till the desired amount of detail is achieved (this requires that all objects are initially included in a highly generalized fashion, which may be a zero rendering).

This method is idealized and methods to measure information content of a cartographic object are currently unknown. A simplistic application of information theory [35] cannot deal with this particular case, where the information content of an individual object must be measured. This idealized method is also not desirable, because the cartographic process does not lead exactly to uniform *information density,* but for practical reasons, is approximated with a selection to reach uniform *graphical density*. The cartographic process is fundamentally constrained by the limit, that *one* piece of paper can carry only *one* graphical message. The cartographic selection process is mostly dealing with the management of the resource map space and how it is allocated. This is well accounted for in the 'name placement' algorithm [6].

A relatively simple practical method is to measure *'ink'*, i.e. pixels which are black. (The impact of color still has to be studied.) One assumes then that there is a given ratio of ink to paper. This ratio must be experimentally determined, measuring manually produced good maps (e.g. 1 inked cell per every 10 cells of paper). The expansion of the tree is progressing from the top to the bottom, accumulating ink content and stopping when the preset value for graphical density is reached. The ink content could be measured not for the full window, but the window can be subdivided and ink for each subdivision optimized.

This selection principle does not avoid that two objects should be rendered at the same location. It requires afterwards a placement process (similar in kind to the name placement algorithm known) to assign a position to each object. An alternative is to associate each graphical object with an approximate description of the map space it will occupy (i.e. a quadtree of the inked cells). The algorithm then descends the multi-scale tree and expands each object till no object can be further expanded without overlapping with an already expanded object.

This does not take care of what cartographers like to advance, namely an interpreted rendering, where more important features are put forward and lesser features are left out, according to an intuitive value scale of the cartographer, his understanding of the situation and the task a future map user needs to solve but this is exactly what makes the cartographic generalization an AI complete task.

6. Zoom on a Multi-Scale Tree

The operation considered is the rendering of a set of objects from a database for a given viewing area (the 'window' in computer graphics). The traditional method would select all objects which fall within the window and then select from those sufficient objects to achieve the desired information density. Our approach is slightly different:

We assume here, that a simple rendering method is given, which transforms the cartographic features in the multi-scale tree into visible objects on paper or a screen. This may be similar to the traversal of a linear display list but could be more sophisticated and contain particular cartographic rules. The hierarchical algorithm further requires two simple inquiry functions (minimal bounding rectangle and 'ink' quantity). For the principle described here, no specifics of how the objects are internally structured are necessary.

The intelligent zoom implies two operations, of which the algorithms will be outlined here. The operations require the calculation of two of four interdependent variables describing the overall aspects of the desired map. They are

- area of the window in the world (e.g. 2 qkm),
- viewport area (e.g. 20 qcm),
- scale, and
- number of objects rendered.

The interdependency is created by the principle of constant information density.

Searching the tree top down, the appropriate objects determined through their minimum bounding rectangle to belong to the object area are taken from the database. For each object rendered the amount of ink it contributes to the map is counted (or computed). Once the amount of ink reaches a predetermined percentage of the total map area, the search stops. The objects retrieved are rendered. The optimal percentage of ink is not only dependent on the map type, but also on the method how ink is measured (empirical determination in well done manual maps is necessary, see section 5).

To zoom in, previously retrieved objects are tested against a reduced window and only objects falling into the new window are retained. Computing the total remaining ink indicates how much additional objects from a more detailed level of the tree can be added. To zoom out, the tree is searched anew from the top with a larger window.

From the user point of view, zoom appears continuous, as previously visible objects change slowly (if at all, see table 1) and new objects appear as space for them becomes available. The data access pattern of the algorithm is strictly local and retrieval from disk fast; thus buffering yields smooth operation.

7. Performance

The proposed multi-scale tree is a method to produce maps of different scales from a single database [36]. It avoids all the known problems of cartographic generalization, which cannot be fully automated today, using redundancy. Objects are stored in different levels of generalization, assuming that at least for the difficult cases, the generalization is done by humans, but only once. Building a multi-scale tree is probably a semi-automated process where automated processes are directed by a human cartographer. All operations where valuable human time is necessary are done once only and the results are stored.

Retrieving a map at a given scale requires a top down search of the tree. This search is guided by comparing spatial locations of the objects with the window of interest. Such comparisons are fast (linear in the number of objects compared), and the tree can be structured such that only the relevant part must be searched. The depth of the search in the relevant part of the tree is bounded by the amount of graphical objects which can be shown. The test if an object can be shown is based on testing that the required space is free; a test requiring constant time per object. Most of the objects tested are also included in the output. Therefore the search process is linear in the number of objects included in the output, which is - following the principle of constant information density - constant.

8. CONCLUSIONS

A number of applications requires graphical presentations of varying scale, from overview sketches to detailed drawings. A simple scale change is not sufficient to produce drawings which humans can easily understand. The problem is most visible in cartographic mapping, where map scales vary from 1:1 000 to 1:100 000 000, covering a range of 10^6. Cartography has developed over the last five centuries useful methods to produce overview maps from more detailed ones. Several complex filtering rules are used to delete what is less important, simplify the objects retained etc. Unfortunately these rules have not been formalized and it was not possible, despite great efforts and partial solutions, to produce a fully automated system.

The approach used here assumes that generalized versions of a map for an area are available and could be stored in a multi-scale tree, where features which represent the same object, are linked. From such a multi-scale tree, a map of arbitrary scale can be deduced by searching in the tree till sufficient objects within the window of interest are found to produce a map of sufficient information density.

The concept is based on a trade-off between computation and storage, replacing all steps which are difficult to automate with storage. These steps are performed initially, while the remaining steps, which can be easily automated, are performed each time a query asks for graphical output.

The resulting tree structure is more complex, than spatial hierarchical structures proposed in the literature so far [12], as objects may change their geometric appearance considerably. For example they change their spatial dimension, or change from a single object to a group of objects. Special attention requires the case that objects seemingly appear as we zoom in. The proposed tree is related to quad-tree or strip-tree structures, but generalized: objects of all dimensions can be stored and the dimension of an object between steps of generalization can change.

REFERENCES

1. R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. The Apic Series 37. Academic Press (1992).
2. C. F. Herot. A Prototype Spatial Data Management System. *Computer Graphics*, vol. 14, No. 3, (1980).
3. J. Bertin. *La Graphique et le Traitement Graphique de l'Information*. Flammarion, Paris. (1977).
4. J. Bertin. *Semiology of Graphics*. The University of Wisconsin Press, Madison, WI. (1983).
5. B. Buttenfield and R. McMaster (Ed.). *Rule based cartographic generalization*. Longman, London (1991).
6. H. Freeman and J. Ahn. On the Problem of Placing Names in a Geographic Map. *International Journal of Pattern Recognition and Artificial Intelligence* 1 (1): 121-140 (1987).
7. B. M. Powitz. *Zur Automatisierung der kartographischen Generalisierung topographischer Daten in Geo-Informationssystemen.* PhD, Hannover. (1993).
8. W. Staufenbiel. *Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung grossmasstäbiger Gebäudedarstellungen*. Kartographisches Intitut Hannover (1973).
9. M. J. Egenhofer. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No.1. (1994).
10. B. P. Buttenfield and J. S. DeLotto. *Multiple Representations: Initiative 3 Specialist Meeting Report*. NCGIA. Report 89-3. (1989)
11. A. U. Frank. Design of Cartographic Databases. In *Advances in Cartography*, ed. Muller, J.C. 15-45. Elsevier, (1991).
12. H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Publishing Company, Reading, MA. (1989).
13. H. Samet. *Applications of Spatial Data Structures. Computer Graphics, Image Processing and GIS*. Addison-Wesley Publishing Co., Reading, MA. (1989).
14. D. H. Ballard. Strip Trees: A Hierarchical Representation for Curves. *ACM Communications.* 24 (5), 310-321 (1981).
15. G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 189-194. (1991).
16. A. Rosenfeld and others. *Applications of hierarchical data structures to geographical information systems*. Computer Vision Laboratory, Computer Science Center, University of Maryland, (1982).
17. M. W. Dobson. Digital Cartography in the World of Commercial Publishing. In *Third International Symposium on Spatial Data Handling in Sydney, Australia*, edited by D. Marble, International Geographical Union, 1-8, (1988).
18. K. Beard. *Multiple representations from a detailed database: a scheme for automated generalization.* Ph.D. Thesis, University of Wisconsin - Madison, (1988).
19. D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. C*anadian Cartographer* 10 (2): 112 - 122, (1973).
20. J. P. Jackson. Developing an effective human interface for geographical information systems using metaphors. In *ACSM/ASPRS Annual Convention in Denver, CO; March 18-23, 1990*, 117-125, (1990).

21. J. P. Jackson. *Visualization of Metaphors for Interaction With Geographic Information Systems*. Masters of Science Thesis, University of Maine, Department of Surveying Engineering, ME. (1990).

22. G. Volta. *Interaction with attribute data in Geographic Information Systems: A model for categorical coverages*. Master of Science, University of Maine, ME, (1992).

23. F. Töpfer and W. Pillewitzer. Das Auswahlgesetz, ein Mittel zur kartographischen Generalisierung. *Kartographische Nachrichten* 14: 117-121. (1964).

24. A. H. Robinson and B. B. Petchenik. *The Nature of Maps: Essays toward Understanding Maps and Mapping*. The University of Chicago Press, Chicago. (1976).

25. C. Youngmann. A linguistic approach to map description. In *First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems in Cambridge, Mass.*, edited by Dutton, Geoffrey, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, 1-17, (1977).

26. B. L. Palmer and A. U. Frank. Spatial Languages. In *Third International Symposium on Spatial Data Handling in Sydney, Australia,* edited by Marble, D.F., International Geographical Union, 201 - 209, (1988).

27. J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *TOG* 5 (2): 110 - 141, (1986).

28. H. Schlichtmann. Characteristic traits of the semiotic system 'map symbolism'. *Cartographic J.* 23-30. (1985).

29. C. G. Head. Mapping as language or semiotic system: review and comment. In *Spatial Languages*, ed. Mark, D. M., and Frank, Andrew U. Kluver, (1990).

30. J. Morrison. The proposed standard for digital cartographic data. *American Cartographer* 15 (1): 9-140, (1988).

31. AdV. Amtliches Topographisch-Kartographisches Informationssystem (ATKIS). Gesamtdokumentation. Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland, (1989).

32. A. U. Frank and S. Timpf. Rendering of subdivision hierarchies. Internal Report, Department of Geoinformation, TU Vienna, (1994).

33. P. Van Oosterom. A reactive data structure for geographic information systems. In *Auto-Carto 9 in Baltimore, MA*, edited by Anderson, E., ASPRS & ACSM, 665-674, (1989).

34. D. M. Flewelling and M. J. Egenhofer. Formalizing Importance: Parameters for Settlement Selection. In: *11th International Conference on Automated Cartography*, Minneapolis, MN. ACSM, (1993).

35. C. E. Shannon. *The Mathematical Theory of Communication.* Bell System Technical Journal, July and October 1948, reprint: Illini Books 1963: private, (1948).

36. K. Beard. How to survive on a single detailed database. In *Auto-Carto 8 in Baltimore, MA*, edited by Chrisman, N. R., ASPRS & ACSM, 211-220. (1987).