A MULTI-SCALE DATA STRUCTURE FOR CARTOGRAPHIC OBJECTS

Sabine Timpf and Andrew U. Frank
Technical University of Vienna
Department for Geoinformation E127.1
Gusshausstr. 27-29
1040 Vienna, Austria
{timpf,frank}@geoinfo.tuwien.ac.at

ABSTRACT

Geographic Information Systems manage data with respect to spatial location and that data is presented graphically as a map or sketch. Typically GIS tasks require graphical presentations at different levels of detail, ranging from overview scenes to detailed views. This is a general problem of visualization of information. Map generalization would solve this problem and efforts to achieve automated cartographic generalization were successful for specific aspects, but no complete solution is known.

In this paper we propose to store renderings of objects at different levels of detail in a database, which can then be used to compose a map. This is a trade-off between storage and computation.

1. WHY A MULTI-SCALE DATA STRUCTURE FOR CARTOGRAPHIC OBJECTS

Graphical sketches have become increasingly important as ad-hoc means of visual output from Geographic Information Systems (GIS). The requirements for these graphics are low in cartographic terms, namely fast visualization and presentation of the pertinent data only. In contrast, traditional maps have always been very sophisticated multi-purpose instruments.

Typically GIS tasks require graphical representations at different levels of detail, ranging from overview scene to detailed views [12]. They also call for a single database as basis for the tasks to perform [3]. Automated cartographic generalization would fulfill these requirements. Unfortunately no complete solution is known, although efforts were successful for specific aspects [6]. Generalization in cartography is a complex set of operations on graphical representations of objects, not on the object itself. The result is a commonly accepted or standardized representation of the object in a certain scale.

In this paper we propose to store renderings of objects at different levels of detail in a database. The proposed structure is a directed acyclic graph (dag). The renderings can then be used to compose a map. We restrict ourselves to the task of zooming and to the topographic scales (1:5000 - 1:100 000). The method is a trade-off between storage and computation.

When zooming, we discern two different operations or sub-tasks, namely selection for inclusion in a map and rendering:

Selection(of all objects from the dag according to constraints)-> SelectFrom (dag)

Rendering(of one graphical object) -> RenderGO (graphicalObject)

The approach extends ideas of hierarchies or pyramids and is related to quad-tress [21, 22] and strip-trees [2], but tries to generalize the hierarchical concept [18, 20] to include object representations of different dimension. The resulting dag structure is more complex than the hierarchical structures proposed in the literature so far, as objects may change their appearance completely. For example they change their spatial dimension, or change from a single object to a group of objects. Special attention requires the case that new objects appear when we zoom in.

This paper is structured as follows: after the introduction (chapter 1) the proposed data structure is explained (chapter 2). In chapter 3 the process of zooming in is examined. Chapter 4 summarizes the paper and gives an overview on future work.

2. THE PROPOSED DATA STRUCTURE

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 [16, 19, 30]. These features are combined from a graphical vocabulary, which provides the atoms for graphical communication [5, 11, 14, 23]. 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 [15].

The approach selected here is to construct a multi-scale directed acyclic graph (dag) for each object group that exists in a map and to specify rules for their interaction. Object groups are waterbodies, railroads, roads, settlements, labels, and symbols [10, 25]. They represent the first stage in a characterization of object features. The second stage will be realized through the 12 categories mentioned above.

In figure 1, a multi-scale DAG for houses is shown. At a very high level (meaning small-scale) the graphical object 'house' is not rendered at all, at a lower level it is rendered as a symbol, then as a generalized geometrical representation, and as a geometrical description. Between each of these renderings a jump in representation method is made.

In the next lower level, the geometry is depicted more clearly and it is shown, that the object is in fact an aggregated object, but the representation method remains the same. The most detailed rendering is again made possible through a jump in representation method.

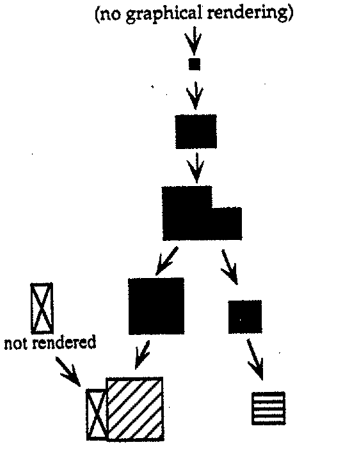

Figure 1: An example for a multi-scale dag

2.1 WHY A DAG?

A dag is a well known and documented structure in graph theory [17] and it has many applications in the database area [9]. The logical data structure suited for zooming would be one that links object representations of one level or scale with object representations of another level or scale in a hierarchically fashion. In our structure the nodes of the dag contain the object representations and therefore the information necessary to render the objects. The directed links determine the direction of the zooming-in process.

2.2 MORE THAN ONE TREE MAKE A FOREST

In graph theory a forest is a collection of trees while trees are acyclic graphs [1, 17]. In our data structure each object group (waterbodies, railroads, roads, settlements, labels, and symbols) is represented by a forest (fig. 2). The whole collection of forests is again a forest.

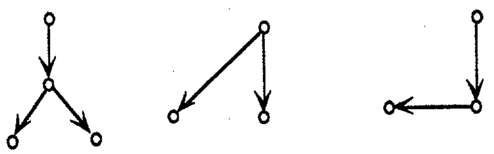

Figure 2: An example for a forest

When implementing the structure we use an approach similar to the Reactive Tree [28]. The difference is that we first structure or classify the graphical objects and construct several trees that are spatially intertwined. They may then be represented in a reactive tree. There is still research to be done to define the relationships between the different trees or forests in order to preserve topology between objects from different groups.

3. MOVING ABOUT IN THE STRUCTURE

Zooming is a concept that originates from the metaphor of the sound of an airplane flying towards the earth. This means getting nearer to the object of interest, seeing more detail. In computer graphics this has been partly realized as getting nearer to the focus of the window of interest while enlarging the information contained in the window (fig. 3a). Volta [29] studied a content zoom in which the categories of the window of interest are shown in more detail (fig. 3b). For example three major soil classes are differentiated into a detailed schema of several dozen classes. We identified the need for an intelligent zoom [8], that realizes both requirements for zooming (fig. 3c).

Figure 3a: A graphical zoom

Figure 3b: A content zoom

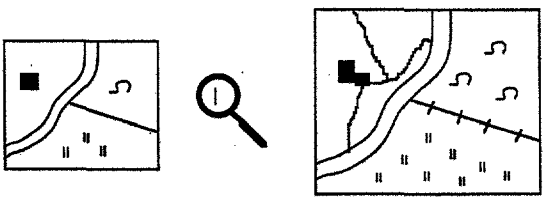

Figure 3c: An intelligent zoom

By intelligent zoom we understand a zoom operation, which respects the known principle of constant information density [4, 27]. 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 [2], to a pixel representation of an area as in a quadtree [21, 22] or the pyramid structures used in image processing [20].

3.1 OCCURRING CHANGES

The objects need to continually change their representation in order to allow for a continuous zooming. We examined the changes that objects undergo during the process of zooming [26].

Most of the changes (1-4, 6,7 in table 1) are not reflected in the data structure.

continuous changes	discrete changes		
	slight change	complete change	appears
1. no change in appearance 2. increase of scale	3. change in symbol 4. increase of detail 5. appearance of label	6. change in dimension 7. shift to geometric form 8. split into several objects	9. appears

Table 1: Types of changes of object representations for smaller to larger scale

However, changes 5, 8, and 9 affect the data structure. The changes 'appearance of a label' and 'object appears' require a new link to come into the existing dag from the outside (fig. 4a).

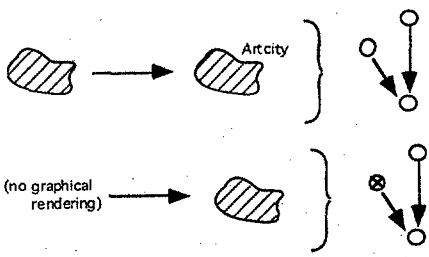

Fig. 4a: Examples for the change 'appearance of label' and 'object appears'

The change 'split into several objects' requires that more than one link is leading from a single node (fig. 4b). Of these two structural changes, namely new link and several links from one node, the first is the more important one and also the most difficult to handle.

Fig. 4b: Example for the change 'split into several objects'

3.2 ZOOM-IN

There are two different operations when moving about in the data structure i.e. ZoomIn and ZoomOut. When zooming in we work our way in the direction of the links. This is the operation we want to consider here. Zooming-in is a function dependent on the window area, the number of graphical objects in the window, the current scale, the current level of descent in the data structure and the ZoomIn percentage.

ZoomIn = F(WA, NoGO, Percent, Scale, Level)

where	WA	window area
	NoGO	number of objects in the window
	Percent	ZoomIn percentage
	Scale	current scale
	Level	current level of descent in the data structure

The number of objects multiplied with the area of the object should be constant over the same map area [8]. This is the principle of constant information density. In our case, the map area corresponds to the window area, the follow-up window area can be calculated with the current scale, the ZoomIn percentage, and the current window area, and the area of the objects rendered is approximated by the area of the minimum bounding rectangle of that object.

4. WHAT'S LEFT TO BE DONE?

This paper has presented a multi-scale data structure to enable 'intelligent' zooming in Geographic Information Systems. The problem reaches into the still unsolved problem of automated generalization. Our solution attempts to shed a light from a different angle on this problem, and by solving one problem learn more about the other.

One of the problems encountered in this study is the small body of literature on cartographic formalization. We think, that more studies are needed in the formalization of cartographic knowledge and in cartographic theories. Displays should be more adaptable to users needs and their tasks instead of creating and presenting a static view of the data [13].

When continuously zooming, the jumps in representations could be smoothed by using a morphing algorithm. Literature on 2D-morphing of geometric features is abundant and well researched (see for example [24]).

The proposed data structure is very complex and still resists to be implemented. It has been shown that the graph concept can be implemented in the functional programming language GOFER [7]. The work is still in progress and comments and discussions are highly appreciated.

REFERENCES

[1] Ahuja, R.K., Thomas L. Magnanti, and J.B. Orlin, 1993. Network Flows: Thoery, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice Hall.

[2] Ballard, D.H., 1981. Strip Trees: A Hierarchical Representation for Curves. ACM Communications. 24(5): p. 310-321.

[3] Beard, K., 1987. How to survive on a single detailed database. in Auto-Carto 8. Baltimore, MA: ASPRS & ACSM.

[4] Beck, W., 1971. Generalisierung und automatische Kartenherstellung. AVN. 78(6): p. 197-209.

[5] Bertin, J., 1967. Sémiologie Graphique. Gauthier-Villars: .

[6] Buttenfield, B. and R. McMaster, ed. Rule based cartographic generalization. 1991, Longman: London.

[7] Car, A. and A.U. Frank, to appear. Formalization of Conceptual Models for GIS using GOFER. in GIS/LIS '95 Central Europe. Budapest, Hungary:

[8] Frank, A.U. and S. Timpf, 1994. Multiple representations for cartographic objects in a multi-scale tree- An intelligent graphical zoom. Computers and Graphics Special issue: Modelling and Visualisation of Spatial Data in Geographical Information Systems. 18(6).

[9] Güting, R.H., 1994. GraphDB: A Data Model and Query Language for Graphs in Databases. FernUniversität Hagen:

[10] Hake, G., 1975. Kartographie. Sammlung Göschen//de Gruyter: orig.

[11] Head, C.G., 1991. Mapping as language or semiotic system: review and comment, in Cognitive and Linguistic aspects of space, D.M.a.F. Mark A.U., Editor. Kluwer Academic Publishers: Dordrecht. p. 237-262.

[12] Herot, C.F., et al., 1980. A Prototype Spatial Data Management System. Computer Graphics. 14(3, July 1980).

[13] Lindholm, M. and T. Sarjakoski, 1994. Designing a visualization user interface, in Visualization in modern cartography, A. MacEachren and D.R.F. Taylor, Editor. Pergamon, Elsevier Science Ltd.: Oxford. p. 167-184.

[14] Mackinlay, J., 1986. Automating the Design of Graphical Presentations of Relational Information. TOG. 5(2): p. 110-141.

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

[16] Palmer, B. and A. Frank, 1988. Spatial Languages. in Third International Symposium on Spatial Data Handling. Sydney, Australia:

[17] Perl, J., 1981. Graph Theory (in german). Wiesbaden (FRG): Akademische Verlagsgesellschaft.

[18] Robertson, G.G., J.D. Mackinlay, and S.K. Card, 1991. Cone Trees: Animated 3D Visualizations of Hierarchical Information, in Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems. p. 189-194.

[19] Robinson, A.H. and B.B. Petchenik, 1976. The Nature of Maps: Essays toward Understanding Maps and Mapping. Chicago: The University of Chicago Press. 138.

[20] Rosenfeld, A., et al., 1982. Applications of hierachical data structures to geographical information systems. Computer Vision Laboratory, Computer Science Center, University of Maryland:

[21] Samet, H., 1989. Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS. Reading, MA: Addison-Wesley. 507.

[22] Samet, H., 1989. The Design and Analysis of Spatial Data Structures. Reading, MA: Addison-Wesley. 493.

[23] Schlichtmann, H., 1985. Characteristic traits of the semiotic system 'map symbolism'. Cartographic J. : p. 23-30.

[24] Sederberg, T.W. and E. Greenwood, 1992. A physically shaped approach to 2-D shape blending. in Computer Graphics Proceedings SIGGRAPH'92. Addison-Wesley.

[25] Staufenbiel, W., 1973. Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung grossmasstäbiger Gebäudedarstellungen. Kartographisches Intitut Hannover:

[26] Timpf, S. and A.U. Frank, 1995. A Multi-Scale Dag for Cartographic Objects. in ACSM/ASPRS. Charlotte, NC:

[27] Töpfer, F. and W. Pillewitzer, 1966. The principes of selection. Cartographic Journal. 1966(3): p. 10-16.

[28] Van Oosterom, P., 1989. A reactive data structure for geographic information systems. in Auto-Carto 9. Baltimore, MA: ASPRS & ACSM.

[29] Volta, G., 1992. Interaction with attribute data in Geographic Information Systems: A model for categorical coverages. University of Maine, USA:

[30] Youngmann, C., 1977. A linguistic approach to map description. in First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems. Cambridge, Mass.: Laboratory for Computer Graphics and Spatial Analysis, Harvard University.