Cartographic Data Structures

Thomas K. Peucker and Nicholas Chrisman

ABSTRACT. Efficient and flexible data structures are important to the development of computer mapping. Most current data banks are characterized by 1) structures which are convenient at the input stage rather than at the stages of use within computer programs, 2) separate and uncoordinated files for different types of geographic features, and 3) a lack of information about neighboring entities. The term "neighborhood function" may be used to indicate the relative location of a geographic entity and is a concept which is involved in all three of these characteristics. Ongoing research on data structures had led to work on the GEOGRAF system for encoding planar data and the GDS ("Geographic Data Structure") for encoding three-dimensional surfaces. Both involve data manipulation between the digitizing stage and the actual use of the data within computer mapping programs.

INTRODUCTION

A series of ongoing research projects are concerned with efficient and flexible data structures for geographic and cartographic analysis. The three main points of concern in the research can be summarized as follows:

1) In most cartographic data banks, the arrangement of the data is guided by the input stage. In other words, little manipulation of the data is performed after the data have been input into the system from maps.

2) Cartographers and computer scientists have made few attempts to combine different types of cartographic information, for example, height with other cartographic features. Therefore, the different types of cartographic entities are stored in different files and it is usually extremely time-consuming to combine them.

3) The data structure is usually very simple and lacks one facet in particular which is essential for much geographic and cartographic analysis—an indication of the relative location of a geographic entity, i.e., the position of a geographic entity with respect to its neighboring entities.

These three points may be abbreviated with the terms flexibility, comparability, and topology. This paper will characterize types of existing geographic and cartographic data systems for planar and three-dimensional surfaces, especially with respect to these three points. The paper will also describe attempts which have been made by the authors to produce data systems which eliminate some of the problems of existing ones. The term "neighborhood function" will play a major role throughout the paper and will therefore be explained in more detail in the following section.

NEIGHBORHOOD FUNCTION

When asked for the location of a city, we will give the location with respect to a river, a seacoast, a pass, a neighboring larger city, or other feature. Rarely will we use the geographic coordinates of longitude or latitude, nor will we use map coordinates. We are taught in elementary geography that the geographic coordinates will tell us little about either the large-scale (site) or small-scale (situation) characteristics of a place. Similarly, if I de-

Thomas K. Peucker is associate professor, Simon Fraser University, Burnaby, Canada. Nicholas Chrisman is senior programmer, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Massachusetts. The research for this paper has been supported by ONR contract N00014-73-C-0109 and the Laboratory for Computer Graphics and Spatial Analysis, Harvard University.

scribe my position on a piece of terrain, I will not use my map to determine my location within the UTM-grid; rather I will look for nearby relief features (peaks, rivers, slopes, roads) as orientation characteristics.

In contrast, when a geographic data bank of any kind is created, it utilizes some kind of absolute coordinate system. Usually neither the geographic evaluation at the time of digitizing nor the mapping system used with the data allow the inclusion of such cartographic features as streets, rivers, and roads which would give us an indication of relative location.

While the human user can be aided in his orientation on a map through overlays or map comparison, the computer has difficulty in determining relative location. If the relative location in terms of the closest points for each of, say, 5,000 points is to be calculated, the program literally has to compute every point's distance to every other point. Indeed, some widely-used programs do this computation several times within one program run.

Some indication of the relative location of a geographic feature can be very useful. This neighborhood relationship will be referred to as a "neighborhood function." It can be expressed in different ways: as an *explicit* or *implicit function,* or as a discrete function in the form of a *table.*

The *explicit function* can be a polynomial or trigonometric equation set for a discrete grid of surface patches which give the form of the surface at each point within the patch. Typical for this approach is the work of Junkins, et al. (1973). Two-dimensional spline functions also fall into this category (Holroyd and Bhattacharyya, 1970). A much more frequent way of defining a neighborhood is by the explicit function in the form of a sort routine which finds the closest neighbors. This is done in various interpolation algorithms to produce a regular grid of points (Shepard, 1968; Heiskanen and Moritz, 1967). The computations increase close to the square of the increase in the number of points, since the search has to be repeated for every point and all points, or at least a large number of them, must be processed each time.

This search procedure also applies in the case of planar surfaces where neighboring polygons must be found. For example, when contiguity constraints are imposed in problems of factorial ecology and other regional correlations, all polygon points must be searched to find those which are in common for a pair of polygons. Again, the problem increases in complexity according to the square of the number of the items being searched.

The *implicit function* expressing neighborhood relationships is usually a function that describes the coding structure of the geographic entities. One very good case is described in Rosenfeld (1969) for different types of neighborhood relationships within a regular grid in which the point P_{ij} has the four neighbors $(i+1, j)$, $(i, j+1)$, $(i-l, j)$, $(i, j-1)$ and it has the eight neighbors $(i+1, j)$, $(i+1, j+1)$, $(i, j+1)$, $(i-1, j+1)$, $(i-1, j)$, $(i-1, j-1)$, $(i, j-1)$, $(i+1, j-1)$.

Neighborhood relationships in the form of *tables* are very rarely used. This type records the neighborhood function by "pointers" indicating neighboring geographic entities. For example, a structure which is built on the basis of Thiessen polygons could have such a structure by simply having the labels of the neighboring points accompany the record of each point. The most widely-known structure of this type is the DIME file of the U.S. Census which encodes line segments, the names of the polygons to the left and right of each line segment, and the names of the two nodes at either end. The neighborhood relationships used in the DIME development are derived from the discipline of topology (Cooke and Maxfield, 1967).

Neighborhood relationships will be discussed in greater detail in the analysis of various existing data structures. Two types of geographic data bases will be discussed: those defining planar surfaces and those defining three-dimensional surfaces. For both types, a summary of their historical development will be presented, and it will be shown that, although presently at different stages of development, these two types can be treated as special cases of one topological data structure.

DATA STRUCTURES FOR PLANAR SURFACES

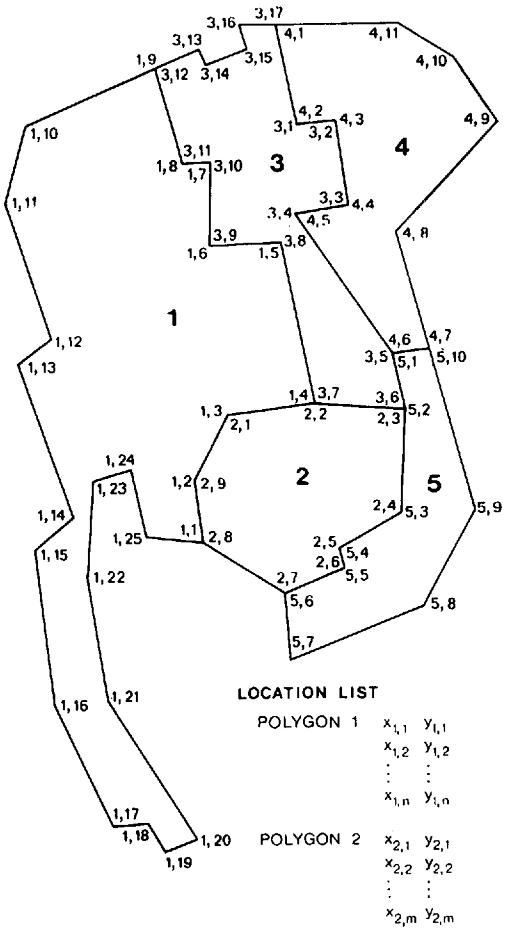
Types of Structures

The types of geographic entities on planar surfaces are points, lines, and area-enclosing lines or polygons. The latter are perhaps the most frequently encoded feature in geographic data systems.

The simplest data base system for planar surfaces is that of encoding *entity by entity* with little or no regard for entity overlaps or adjacencies (Fig. 1). In other words, every polygon in a polygon system is encoded and stored without any regard for contiguous polygons, and lines are encoded without regard for the fact that they may intersect or merge with other lines. The results of such an encoding are "sliver lines" (duplication of lines in slightly different positions). These sliver lines are confusing and unaesthetic and, hence, it is virtually impossible to do anything directly with such a data base except an extremely coarse graphic image.

To go beyond the use of such data for the production of coarse images, editing must be performed. This alternative has been attempted in several cases, one being the MAP-MODEL system (Arms, 1970). The editing in this system is guided by the assumption that every segment has to be represented twice except for segments on the outer boundary. For each segment, the editing program sorts through all remaining segments to find its complement (the identical line of the neighboring polygon). Those segments for which it has not found a complement are tagged to indicate potential errors to the user.

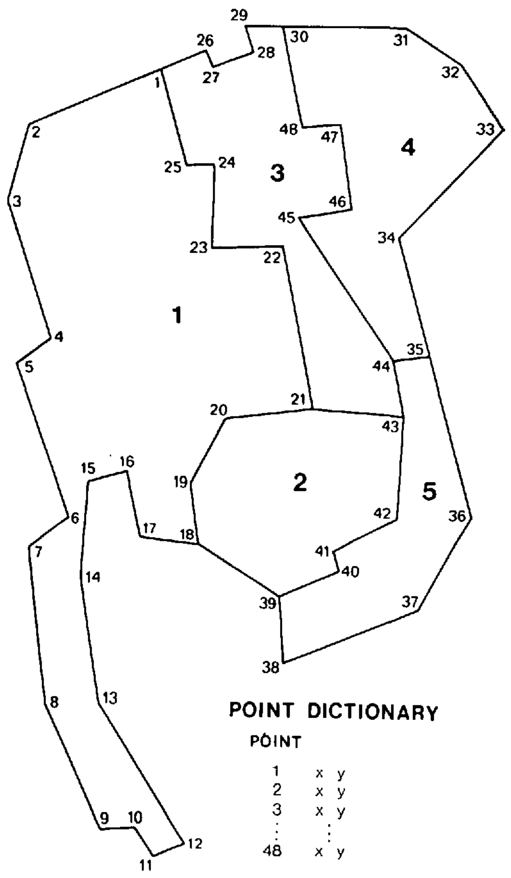
To overcome some of the limitations of independently encoded entities, systems have been developed based on a common *location dictionary*. This dictionary contains the coordinates of every boundary point on the map. Polygon boundary lists are then compiled which consist of the labels (location numbers) of these boundary points (Fig. 2). Line information can



POLYGON 1	$x_{1,1}$	$y_{1,1}$
	$x_{1,2}$	$y_{1,2}$
	⋮	⋮
	$x_{1,n}$	$y_{1,n}$
POLYGON 2	$x_{2,1}$	$y_{2,1}$
	$x_{2,2}$	$y_{2,2}$
	⋮	⋮
	$x_{2,m}$	$y_{2,m}$

Fig. 1. The simplest encoding is areal entity by areal entity. In this system, most points are recorded twice and some (e.g., $P_{1,4}$; $P_{2,2}$; $P_{3,7}$) three times. A point does not necessarily have identical coordinates in all recordings.

be handled in the same way. Programs based on this structure include CALFORM from the Laboratory of Computer Graphics and Spatial Analysis. Other programs have subroutines to convert this point dictionary structure to the simple entity-by-entity line list described above. The data can then be used in programs such as SYMAP (Laboratory for Computer Graphics and Spatial Analysis) which are compatible with the entity-by entity structure. Programs have also been developed to simplify data input through automated polygon identification (Douglas, 1973).



POINT DICTIONARY

POINT

1	x	y
2	x	y
3	x	y
⋮		⋮
48	x	y

POLYGON

3	1, 25, 24, 23, 22, 21, 43, 44, 45, 46, 47, 48, 30, 29, 28, 27, 26, 1
5	44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 44

Fig. 2. In the second type of polygon system, the individual points are encoded only once and are stored in the Point Dictionary. For every polygon, a Polygon Boundary List is then established.

The point dictionary data base has the advantage that sliver lines do not occur. However, the problem of neighborhood relationships is not handled any better than with the entity-by-entity approach. The search for common lines is no longer according to the coordinates of points but by their labels; this brings us closer to a solution only by a little less computer time. It also creates difficulties. A point dictionary can and will be accessed in an arbitrary order, since there are no restrictions regarding point placement. The standard response to this problem is to make the dictionary core resident. Unfortunately, this will limit the complexity of the map that can be handled in this manner, since all points must be stored throughout the operation of the program. This shortcoming of such sharing of data is augmented by the continued independence of the entities created by the dictionary; instead of n points with their x and y coordinates, there are simply n references to points.

Some of the objections to the ordinary point dictionary approach can be eliminated by formulating an intermediate object between the entity and the points used as an addressing scheme (Nake and Peucker, 1972; Peucker (ed.) 1973). A geographic entity can be created from a *list of line segments;* these segments are, in turn, created from references to the point dictionary. This system allows for easy definition of the entities with a minimum of pointers, but each entity is still independent in the sense that its neighbors are not known. The direction of access is still from entity to location but not the reverse.

All of the data structures described thus far are of limited flexibility and utility because neighborhood relationships are not known. By adding the topological neighborhood function of each element to a data structure, large improvements in flexibility and scope of applications can be realized.

If one is concerned about the memory capacity which is needed for the storage of explicit neighborhood relationships, one might consider a system with implicit neighborhood functions by *modifying the entity form* in the encoding stage. Many existing geographic information systems store land-use data in grids of rectangular cells (Hsu, 1975). However, a serious problem is encountered with such a regular discrete encoding of planar surfaces. According to the sampling theorem, the sampling interval has to be half the size of the smallest features to be encoded (e.g., Tobler, 1969). Hence, either the size of the cells has to be very small to enable encoding of detailed variations (e.g., urban land uses) or a grosser cell size can be used for encoding only the very slowly changing variations (generalized land use). In the case of the small units, a very large volume of redundant information is created in an area of uniform land use and matrix reduction techniques such as run-length encoding (recording in each row only those cells differing from an immediately previous cell) create only physical and not logical compaction (Amidon and Akin, 1970). In the case of the grosser cell size, highly varying land use features are aggregated to a degree which reduces the usefulness of the whole system.

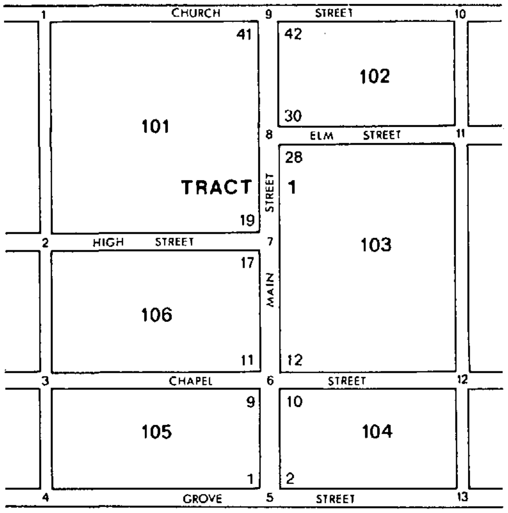
Beyond sampling problems, a grid structure imposes a bias toward specific orientations of features. Diagonals, although physically longer, are given the apparent cell relationships of unit distances along the major axes. The grid structure also creates the illusion of working with a discrete point space, rather than a regular areal partitioning.

Recent studies have developed parameters which determine the degree of inaccuracy of a given sampling mesh (Switzer 1975). In cases where some types of geographic features vary over small areas and others over very large areas, such as the case of statewide land-use patterns, the error is averaged over all features and will affect features with a high-frequency variation more than others.

The variation of the mesh size within the information system would be of only little help. The data management and the manipulation and display routines would become more complex, although probably less than many researchers imagine. Exact figures about the differences cannot be given since no literature about such a system is known to the authors.

Structures Using Explicit Topological Relationships

One of the first known attempts to incorporate explicit topological structure into a geographic data base is the Dual Independent Map Encoding (DIME) system of the U.S. Bureau of the Census (Fig. 3). The DIME files were originally developed as an automated topological error detection system for the Address Coding effort of the 1970 census.



CENSUS ADDRESS CODING GUIDE RECORDS

STREET	TRACT	BLOCK	LOW ADDRESS	HIGH ADDRESS
Main	1	102	30	42
Main	1	103	12	28
Main	1	104	2	10
Main	1	105	1	9
Main	1	106	11	17
Main	1	101	19	41

DIME STREET SEGMENT RECORDS

STREET	NODE START	NODE END	TRACT LEFT	BLOCK LEFT	TRACT RIGHT	BLOCK RIGHT	LOW ADDR.	HIGH ADDR.
Main	5	6	1	105	1	104	1	10
Main	6	7	1	106	1	103	11	17
Main	7	8	1	101	1	103	19	28
Main	8	9	1	101	1	102	30	42

Fig. 3. The DIME-file base. Represented is a portion of Tract 1 in a hypothetical city (Cooke and Maxfield, 1967). The small numbers at the street intersections are the nodes, the larger numbers on Main Street are the addresses. Two types of records are created as illustrated in the tables.

The basic element of the DIME file is a line segment defined by two end points. It is assumed that the segment is straight and not crossed by any other line. The metropolitan files usually define this unit as a street block face. Complex lines are represented by a series of segments approximating the line. The segment has two "node" identifiers, along with the coordinates of its two end points and codes for the polygon on each side of the segment.

While DIME topology makes much information accessible to urban researchers, neighborhood relationships are not made explicit. Segments sharing a node, for example, must be found by laborious search procedures. Search is also required to assemble the outline of a polygon. More importantly the DIME structure is cumbersome to use for many cartographic applications involving areas made up of complex lines. For procedures in a one-time checking effort, as is the case of Address Coding and Address Matching in metropolitan areas, it is quite adequate. For efficient computer storage and retrieval and for many applications, however, improvements are desirable. For example, the reliance on the individual line segment makes the reduction of detail for display purposes difficult since line segments cannot be simply deleted without correcting the reference codes for the affected nodes.

At the Laboratory for Computer Graphics and Spatial Analysis, the junior author has developed a data structure, POLYVRT, that is designed to contain all the information needed to construct any of the previously enumerated planar structures. The basic object of POLYVRT is the "chain." Like a DIME segment, a chain has nodes at its two ends, separates two areas, and is assumed to be uncrossed. It differs in that the POLYVRT chain may be made up of many points whereas the basic DIME unit has only two points. A boundary between two polygons can be referenced by a single chain no matter how complicated, because line detail is topologically unimportant (Laboratory for Computer Graphics and Spatial Analysis, 1974).

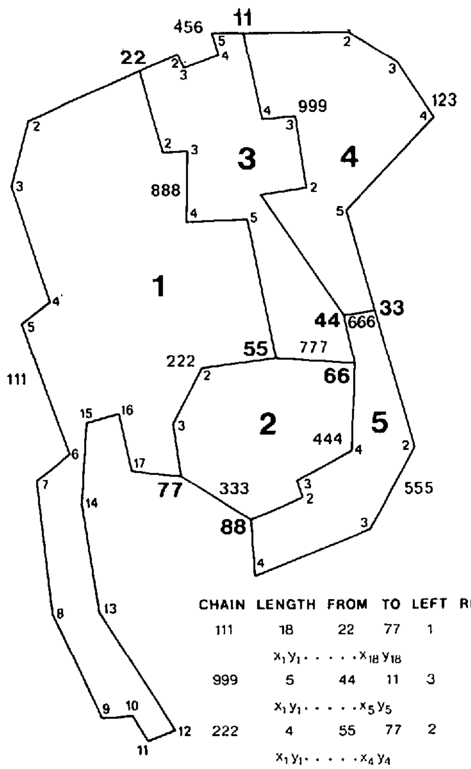
The coding of a complicated boundary as a unit is not unique to POLYVRT. The data bank used in the project "The Interactive Map in Urban Research" (Nake and Peucker, 1972; Peucker (ed.) 1973) as well as the World Data Bank I (Schmidt, 1969) are composed of "lines." In the latter case some of them contain over 4,000 points. The chain based system of POLYVRT, however, is a different type of structure because of the topological role assigned to the chain and the subsequent construction of a list data structure. Based upon this assignment, the topological information about a chain resembles the information on a DIME record except that the distinction between nodes (i.e., points used for more than one chain) and the points internal to a chain allows internal points to be eliminated without influencing the neighborhood relationships. The main innovation in developing a chain representation is that areas of significant line detail may be efficiently handled. Topological checking is reduced from dependency on the number of points to dependency on the number of boundaries.

In addition to the indication of the relative location of the chain with respect to its neighboring polygons, POLYVRT information is stored in separate lists assembling the bounding chains for every polygon. Thus, searches can take place in two directions, from the chain to the polygon and from the polygon to the chain. This is very important for any type of neighborhood manipulation, since neighboring entities can be found through their "bounding" or "bounded" complements. In other words, to follow along a group of chains one flips through chain to polygon to the next chain, etc., whereas to traverse a series of polygons one tests for adjacent polygons by going through the chain directory for each polygon.

The POLYVRT program places point information in secondary storage. The three higher level objects (chains, nodes, and polygons) are core resident. Only the chain refers directly to the point file in the secondary storage. In addition to indicating the locations of the points in the point file, the chain record incorporates the name of the chain, the labels of the starting and ending nodes, and the left and right polygons. Conversely, the polygon list consists of the bounding chains in proper sequence (Figs. 4 and 5). A list linking nodes to chains could easily be constructed.

DATA STRUCTURES FOR THREE-DIMENSIONAL SURFACES

Boehm (1967) describes in a very detailed analysis the advantages and disad-



CHAIN	LENGTH	FROM	TO	LEFT	RIGHT
111	18	22	77	1	0
	$x_1y_1 \cdot\cdot\cdot\cdot\cdot x_{18}y_{18}$				
999	5	44	11	3	4
	$x_1y_1 \cdot\cdot\cdot\cdot\cdot x_5y_5$				
222	4	55	77	2	1
	$x_1y_1 \cdot\cdot\cdot\cdot\cdot x_4y_4$				

2	POLYGON	**.22**	NODE
888	CHAIN	3	POINT WITHIN CHAIN

Fig. 4. The external representation of the POLYVRT chain-file. Every chain has a "name," the number of inner points (length), the two limiting nodes, the two boundary polygons, and a series of coordinates for the points.

vantages of different ways of encoding surfaces. He comes to the conclusion that the encoding of surfaces by contours minimizes the storage capacity, whereas a regular grid of surface points minimizes the computing time necessary for several types of manipulations. Boehm did not include in his study a data system with an irregular distribution of points but only contour encodings and regular grid structures of constant and variable mesh width. He did not reflect on the reasons why contoured data minimize storage capacity whereas a regular grid minimizes computing time. If he had done so it is quite possible that the development of geographic data bases would have taken different routes. Surprisingly this topic has produced little discussion despite the fact that extremely large data banks of terrain (digital terrain models) have been developed.

When encoding surfaces, it is necessary to adapt the density of points to the variation in the local terrain. The question of how dense the points have to be can be answered by again using the philosophy of the sampling theorem. For the terrain within a typical map, the variation can change considerably, resulting in a need for frequent adjustments of the sampling interval.

In a contour map the density of contour lines changes with the density of relief variation. Therefore, it fulfills the requirements set by the sampling theorem for a "non-stationary surface," i.e., a surface with changing terrain. For the regular grid, on the other hand, if the smallest object one wishes to detect anywhere within a study area is of size ("wave length") S, then the grid spacing everywhere must be $S/2$ or less (note, for example, Mark, 1974). The regular grid again tends toward redundancy since smooth areas within the study area will contain far more points than are needed to accurately portray their form. To improve the "resolution" of a grid by a factor of f, the grid spacing must be decreased by

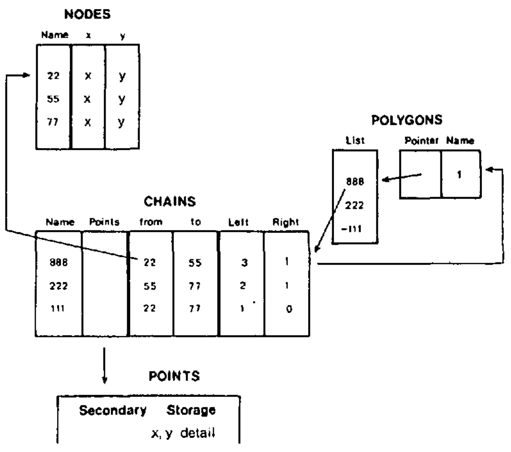


Fig. 5. The internal representation of the POLYVRT chain-file. Reference to polygon chains are positive or negative according to the direction of the chain used.

the reciprocal of this factor and the total number of points is increased by f^2.

The question has been raised many times in photogrammetry whether contours are the best representatives of topographic surfaces. It has been stated that contours do not detect many types of breaks which are frequent on terrain (e.g., Brandstatter, 1957). Therefore, the encoding of surfaces by vertical profiles has been attempted many times in photogrammetry in recent years. Points are encoded only when the slope of the surface changes (e.g., Kraus, 1973). In another case the break lines are encoded in addition to a regular grid (Sima, 1972), and in another approach only the break lines or only the ridges and channel lines of a surface are encoded (Grist, 1972). The amount of detail with these approaches depends on the scale used.

When performing numerical computations on the basis of these digital terrain models, the quantity of data involved will be only one determining factor for the amount of programming and computations needed. Most of the numerical computations on surfaces require some type of neighborhood function, either to compute some surface behavior, such as slope or local variation of relief, or to find the next unit for the drawing of a contour or a vertical profile. For a set of contours it is relatively easy to create a directory which indicates a sequence of contours in a type of tree in which the surrounding contour is the base and the other contours are the branches (Morse, 1968). To find the neighboring points on the two adjacent contours for a given point on an intermediate contour, however, one must search through all the points on the two adjacent contours and compute the distances to the point in question. This procedure is time-consuming if the contour lines are represented by very many points. A regular grid on the other hand has an implicit neighborhood function and finding a neighbor does not involve search nor extra computer time. For a set of irregularly-distributed points as they are represented in very simple data structures (e.g., SYMAP) the creation of a neighborhood function is usually performed by finding the closest small number of points where the number varies around six.

THE GENERAL CASE

It has been noted that although plane surfaces and three-dimensional surfaces look rather different it takes only a few assumptions to treat one as the subset of the other. While three-dimensional surfaces are always based on interval or ratio data, planar surfaces often involve ordinal and nominal data. However, it has been shown that ordinal and nominal data may be treated as interval data (Nordbeck and Rystedt, 1970; Rosenfeld, 1969) and even without this conversion we can combine the two types into one general case by simply using different assumptions about neighborhood.

Given a set of n data points for which x, y, and z coordinates are known, continuously defined surfaces (i.e., surfaces for which one and only one value exists at every point) can be created, using any of three different assumptions about the surface behavior (Peucker, 1972). The first assumption is that of a stepped surface, which says that the surface retains the value of a data point within the neighborhood of that data point, where neighborhood is defined either by a given polygon (the choropleth approach) or by the fact that the area is closer to one data point than to any other one in the neighborhood (the proximal approach). The second assumption is that each data point represents a sample of a single value on a constantly changing surface. Neighborhood is then a number of closest neighboring data points, and intervening values are interpolated with different types of interpolation procedures. The third assumption is that the data point is a sample from a constantly changing surface that may contain errors; thus the data point is not necessarily located on the surface, but close to it. This approach implies the further assumption that the actual surface is smoother than the surface constructed through the sample data points.

With these assumptions about surface behavior, any point or areal distribution of a variable can be treated as a continuous function $z = f(x, y)$, and planar and three-dimensional surfaces can be combined into one type. This has already been done in some computer programs, the most notable being SYMAP. Many cartographic applications must treat both types of surfaces and, therefore, data structures must be developed which can handle both types at one time.

THE PROPOSED DATA STRUCTURES

Planar Surfaces

The need to incorporate different types, or hierarchies, of polygons uncovered one of the limiting assumptions made in POLYVRT. A chain plays a dual role: first, it is the boundary of two areal entities, and secondly, it is the unbroken unit of point retrieval. This is not a problem if one is limited to a single nonoverlapping set of polygons. The following proposed new system, to bear the name GEOGRAF, is an attempt at greater flexibility.

Because of the addition of many layers of complexity involving multiple polygon sets, the chain cannot remain, to the same degree, the controlling object of the data structure. Just as the notion of an unbroken line is important, so is the notion of an unpartitioned space. In a system which must handle overlapping polygon networks, there is a need for a root object which is defined as an area uncut by any further partitioning. This object is termed the Least Common Geographic Unit (LCGU). The LCGU's are constructed as a POLYVRT polygon directly from chains. The relationship of the LCGU's to all other polygon types is hierarchical (Fig. 6).

In turn, the existence of the LCGU allows for the creation of each class of polygon. In order to allow simple coding of the boundary relationships at these higher levels in the structure, the "chain group" was devised. A chain group is a set of chains which form a boundary of two areal units for a given polygon class. These polygons are constructed of chain

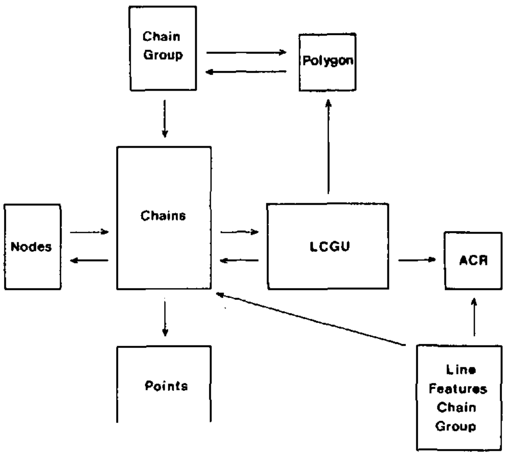


Fig. 6. The relationship between the various parts of the data structure.

groups which, in turn, are constructed from chains.

Line features can be built up of chains in the same manner as chain groups. Note that the chain group listing for each level of polygon and each listing for a line feature reference only the chains themselves. This allows each system to be considered as a separate directory which is core-resident only when that class of objects is retrieved.

The LCGU has other implications and applications that are useful because of the topological data structure. The LCGU, with its coding for each of the polygon sets, can be combined with contiguity information of linear feature types to produce an Attribute Cross Reference (ACR). The ACR is a table in which all objects (in polygon and linear systems) are cross-referenced to each other to determine nesting. By using attributes of chains (lengths) and LCGU's (areas, population densities, etc.) this cross-referencing capability can assign a string of data, collected for one polygon type, to a string of a second type.

Topological manipulation routines are central to the success of this structure. The intersection of geographic features will rely on topological knowledge to realize economies of scale in processing large files. All operations with lines will actually work with bands, built with endpoints of the line, and the furthest deviants to both sides (if the bands become too wide the lines are split, etc.). With this approach, the windowing process uses nonlinear windows (which is often the case with geographic coordinates and map projections) and becomes quite elegant. Similarly, intersection procedures such as point-in-polygon, line-across-polygon, and polygon-over-polygon determination allow for gainful application of the topological principle. For point-in-polygon searches, chain groups are constructed which bisect the universe into parts to sort the points (or nodes of polygons) into three groups: left, within the band, and right. Only the second group needs more detailed treatment. The point set is then recursively partitioned until it reaches the level of the LCGU's. For line-oriented problems, the topological connections of the two sets compared would allow intersections to be limited to immediate neighbors.

Another important procedure will be a nested chain-intersection routine. Here, again, the chain band and its recursive segmentation is used. The number of points which define a chain is constantly increased until the intersection test can be determined. The search of a line through a set of polygons will use a graph-search algorithm developed for the GDS (Geographic Data Structure) project discussed in the next section. The neighborhood search routines create records of neighbors for every point or line or polygon at any specified depth of neighborhood.

Three-Dimensional Surfaces

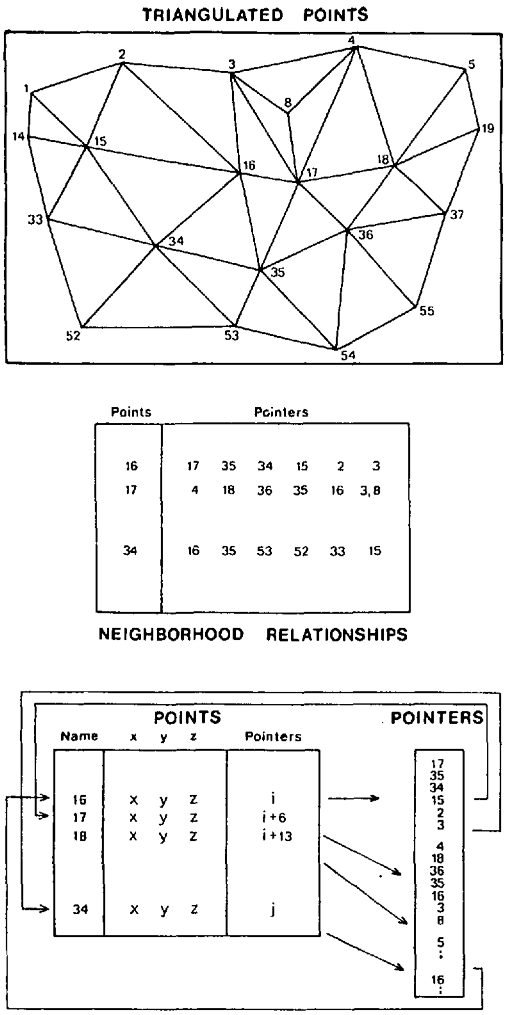
To implement the ideas presented here, the senior author is developing a geographic information system for three-dimensional surfaces. Both systems, the three-dimensional and the planar, are based on data structures with explicit topological neighborhood relationships.

The basic philosophy of both approaches is to separate the data base from the application programs. In the early days of computer cartography, a data set was generally tied to the application program that used it. Now the available data have become so voluminous and the application programs so varied that extra efforts in the preparation of data for more efficient computations seem to be justified. We are well in step with modern computer science to separate the data base from the application programs with the data structure becoming the link between the two.

The creation of a structured data base is nothing new. The interpolation of an irregular grid to a regular grid of height-points provides a data structure through the implicit neighborhood function of the grid. And in the case where polygons are independently defined by a series of points, the neighborhood relationship is replaced by a search algorithm which finds the neighbors for every polygon by searching for matching segments in the boundary files of the other polygons. The idea is to spend computing time before any application has been performed in the anticipation of heavy uses of the data base. An efficient structure of the data base should not only speed up computations considerably, but should also simplify the production of application programs.

The data structure developed under the working title "Geographic Data Structure" (GDS) is based on irregularly distributed points which are assumed to be sample points without sampling errors from a single-valued surface. Two types of structures form the core of the data bases. The first creates neighborhood relationships by "triangulating" the data set and storing for every point the labels of all points which are linked with the point by a triangle edge (Fig. 7). The second structure is produced by selecting those points of the surfaces which lie along lines of high information content, such as ridges and channel lines, and defining them by their nodes, which are peaks, passes, and pits (Fig. 8). This second data structure serves two purposes: First, it is a general representation of the surface for rough computations; second, it is a "directory" into the more detailed first structure.

In the first structure, the creation of the neighborhood relationship is based on the assumption that data-sets are usually of



Points	Pointers					
16	17	35	34	15	2	3
17	4	18	36	35	16	3,8
34	16	35	53	52	33	15

Fig. 7. The GDS-first data structure. The illustration shows the points on the surface with their links to neighbors (edges). The external representation is shown by the neighborhood relationships. The internal representation is composed of the point-file and the pointer-file.

two types: (a) sets of irregularly-distributed points which were digitized with the understanding that every point is significant, and (b) sets of regularly or irregularly-distributed points where it is known that a number of points are redundant and can be eliminated from the set, e.g., regular grids of points and encoded contours.

The first type of data set is linked by some type of triangulation. At least two approaches exist. The first (Dueppe and Gottschalk, 1970) creates all possible links, chooses the shortest, and eliminates all links which intersect the shortest. This procedure is repeated with the next shortest links until no links intersect. The result is the set of links with the minimum cumulative distance between neighboring points.

The procedure has one disadvantage: since $\binom{n}{2}$ links have to be created, the number of points is therefore limited to only several hundred. The first step in our approach therefore limits the links to a number of "potential neighbors," among which the shortest link is chosen and intersected with all other links originating in these potential neighbors. This procedure limits the number of tests for intersections of links to less than $\frac{n \cdot m^2}{4}$ where m is the number of potential neighbors, an arbitrary number between 8 and 14 depending on the density variation of these points. The procedure does not guarantee, however, that only triangles are constructed; polygons with more than three sides can result, although they are relatively rare. The check for such polygons and their elimination is very easy and fast.

The second possibility is to create a triangulated structure through use of Thiessen polygons. A published solution (Rhynsburger, 1973) intersects for every point the links to every other point midway and chooses the smallest polygon created

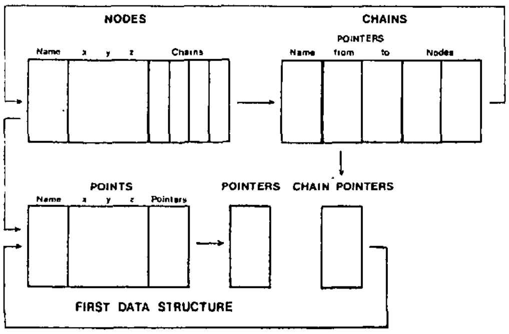


Fig. 8. The GDS-second data structure. Both the node-file and the chain-file have access to the first data structure, the node-file directly and the chain-file through a chain-pointer-file.

by the perpendiculars. Every point which contributes to the Thiessen polygon is a Thiessen neighbor. This procedure can again be simplified by the assumption of a limited set of "potential neighbors." The same checking routines as above have to be applied.

Another approach which limits the number of necessary tests, but is mathematically correct at the same time, has been developed within the project by Kurt Brassel at Harvard University. The procedure is based on "fields of potential neighbors" which converge very rapidly.

The alternative to triangulation is three-dimensional generalization, i.e., to select from a set of points those which define the structure with the least deviation from the original surface. The basic concept, developed by Randolph Franklin of Harvard University and T. K. Peucker, is to approximate the surface of a series of triangles through a selected set of points, where each additional point included into the set is the one which deviates the most from the approximated triangles until the deviations are below a given value (see Peucker, 1974).

In both cases, the triangulation and generalization of the surface, the result is a "linked list" of surface points. The term linked list means that points are linked with one another through pointers. In other words, a point is not only identified by its x, y, z coordinates, but also by a list of the labels of the points which form edges of triangles with the point. In our case each record consists of the x, y, z coordinates of a point and a reference to the start of the pointers to the neighbors in a pointer list. The reason for not having the neighborhood pointers with the point record is that the number of points varies considerably (Mark, 1974). Since the record has to be long enough to include all possible numbers of neighbors, large parts of the pointer sections would be empty most of the time. The pointers are sorted, starting with the pointer the least East of North of a point (Fig. 7).

The use of this type of data-structure is very simple and efficient. For every search (profile, contour, etc.) a criterion for edge-intersection is developed. For contouring, for example, the critical question is whether one point of the edge is above the contour level and the other below. A start is found and one point of the edge is considered a reference point and the other a subpoint. The next subpoint is found by looking up the next neighbor in the pointer list. If the test is positive, the intersection is performed and the process repeated. If the test is negative, the reference and subpoints are switched and the process repeated.

Other procedures are equally simple. To find a triangle, for example, one has only to have a reference and a subpoint. The third point is the next label in the pointer list of the reference point after the subpoint. To find all triangles one goes through the total pointer list leaving out all those edges connecting reference points with subpoints with a smaller label since they would create triangles which had been treated when the subpoint was a reference point.

Although the first data structure, as presented above, seems to be efficient in terms of storage capacity, it does not provide easy access to the data base which is often very large. It is for this reason that we are developing the second data structure to represent the general structure of the surface and to serve as a directory to reach into the first data base (Pfaltz, 1975) (Fig. 8).

The first step in the creation of the second data structure is to find the ridge, channel, and break lines on the surface. If one labels the highest point for every triangle, the unlabeled points are members of the channel line, at least on smooth surfaces. A subsequent search routine deals with the irregularities. Points along ridges are found by eliminating the lowest point of every triangle, using a routine developed for a regular grid by D. M. Douglas. The detection of break lines is somewhat more difficult.

Some theoretical studies of surfaces by Warntz (1966) show that ridge lines and channel lines cross at passes, a useful point of information relative to the development of the second data structure. Practical considerations suggest that in terrain and other surfaces, this regularity is not always present.

Once the topological structure is at hand, it can be used as a directory into the first structure. The line is therefore treated as a chain similar to the chain of the POLYVRT system. The nodes of the chain are the peaks, passes, pits, and other endpoints of chains on the surfaces. These points are stored with their coordinates and the names of the chains which terminate at the nodes. The chains are stored with the labels of the nodes and pointers into the chain lists which consist of labels of points in the first data structure (note that here the chain structure differs from that of POLYVRT).

A third component of the "Geographic Data Structure" should be mentioned since it illustrates very well the logical adaptation of a computer problem solution to geographical data. The problem at hand is the partitioning of the data set. Since with large data sets only portions can be kept in fast memory, the data base is segmented into "pages" which are brought into memory as units. For the "Geographic Data Structure" the paging system can solve several problems inherent in a complex geographic information system.

The boundaries of "patches," as we call the areal extension of a "page," are chains already defined for the second structure. Since detail along the chain is of no topological interest, the density of points along the chain can differ for its two sides. In other words, the density of triangles can change from patch to patch. This allows for very efficient data encoding even in terrain with sudden changes in the surface behavior as at the change from a mountainous area into a plain (Peucker, 1972).

Another advantage of the paging-system is the ease of including topographic and planar information. Linking point, line, and areal data to the triangulated points would lead to high definitional redundancy. The secondary structure could lead to ambiguities where the terrain is very elongated. Since an attempt has been made to keep the shape of the patches as compact as possible, the combination of non-terrain data with patch boundaries seems to be most appropriate.

Since the patch boundaries are again chains, another virtue comes to light: The patches can be treated as polygons of the POLYVRT system with little difficulty. This link between the two systems lends hope that eventually they may be merged.

It is an appropriate question to ask what such a data structure as the GDS will be able to accomplish. A number of display routines have already been developed (Cochrane, 1974) and a series of procedures for surface analysis based on heuristic searches are underway (Fig. 9). Since both levels of data structure are graphs, we will be able to rely on many of the developments connected with operations research, specifically network analysis, for the manipulative treatment of the data.

As both systems, GDS and GEOGRAF, have topological structures, it is possible to merge the two. The creation of polygons from points is the major link from the GDS project to GEOGRAF. The creation of a set of centroids for polygons allows the conversion in the opposite direction. This way, surfaces can be treated as polygonal sets and can be displayed and manipulated by the routines of GEOGRAF. Conversely, polygonal data can be treated as surfaces for GDS. The neighborhood routines are what make the project useful in quantitative geography and planning. Neighborhood searches are extremely expensive without the topological data structure, but they are usually a most important part of urban and environmental analyses once a general overview is obtained from the data.

Although basic research and application development are two sides of one coin and must go together to obtain lasting results, this paper has concentrated on the theoretical parts of the project since their development is ahead of the application routines, a fact which should be expected.

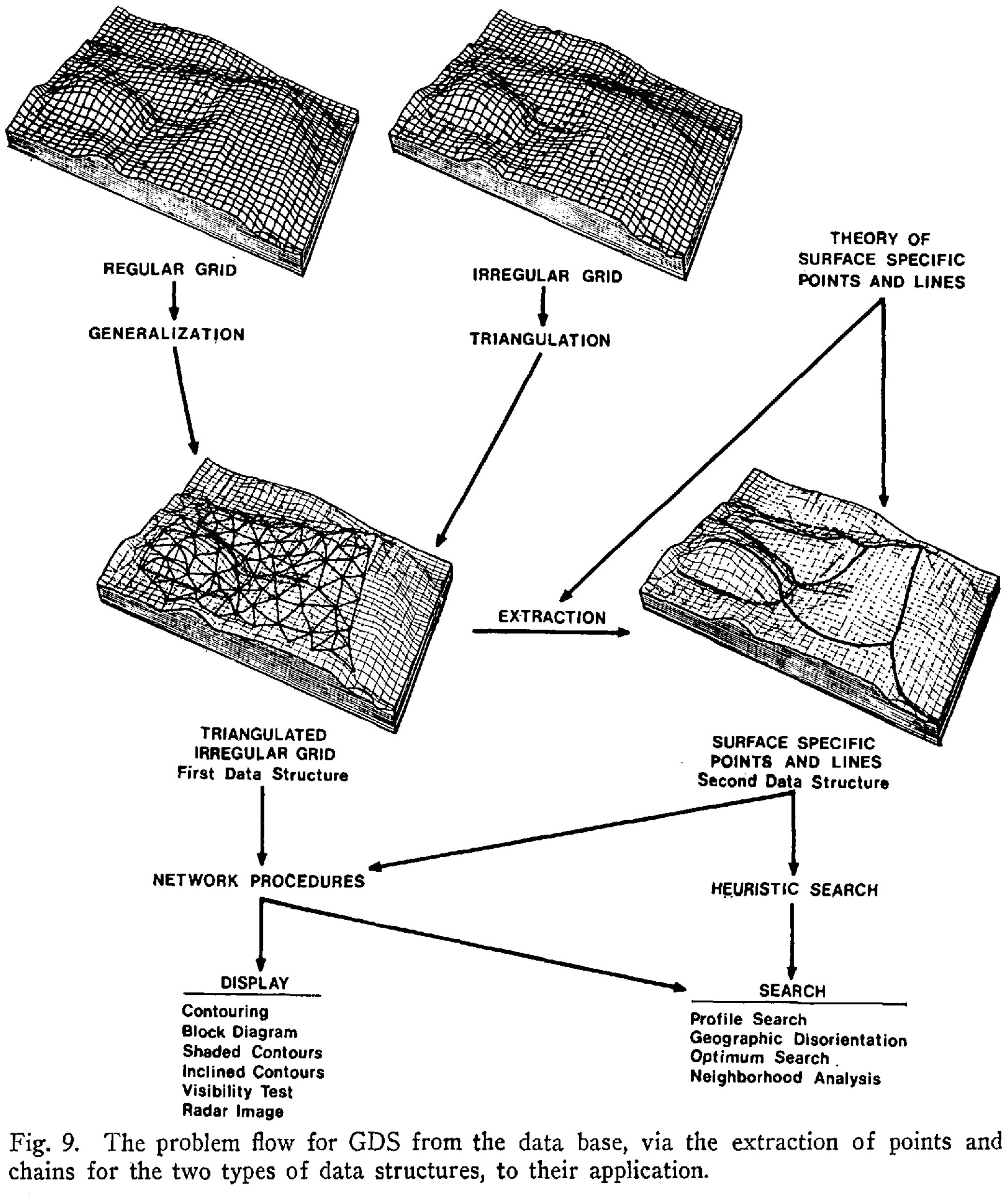


Fig. 9. The problem flow for GDS from the data base, via the extraction of points and chains for the two types of data structures, to their application.

The quintessence of the research so far is the hypothesis that topologically-structured data bases of three-dimensional and planar surfaces can result in reduced efforts in the development and execution of application routines. We have some indication that the hypothesis is correct; the real test will come when the bulk of the application routines is completed.

REFERENCES

Amidon, E. L. and G. S. Aiken (1970), "Algorithmic Selection of the Best Method for Compressing Map Data Strings," *Communications of the Association for Computing Machinery,* Vol. 14, pp. 769–774.

Arms, A. (1970), *MAP/MODEL System: System Description and User's Guide,* Bureau of Governmental Research and Service, University of Oregon, 59 pp.

Boehm, B. M. (1967), "Tabular Representation of Multivariate Functions with Applications to Topographic Modelling," *Proceedings,* 22nd National Conference, Association for Computing Machinery, pp. 403–415.

Brandstaetter, L. (1957), "Exakte Schichtlinien und Topographische Gelaendedarstellung," *Oesterreichische Zeitschrift fuer Vermessungswesen,* Sonderheft 18, Wien, 90 pp.

Cochrane, Douglas (1974), "Cartographic Display Operations on Surface Data Collected in an Irregular Structure," M.A. thesis, Simon Fraser University.

Cooke, D. F. and W. F. Maxfield (1967), "The Development of a Geographic Base File and Its Uses for Mapping," *Urban and Regional Information Systems for Social Programs,* Papers from the 5th Annual Conference of the Urban and Regional Information System Association, pp. 207–218.

Douglas, David H. (1973), "BNDRYNET," Peucker, T. K. (ed.), *The Interactive Map in Urban Research, Final Report After Year One,* University of British Columbia.

Dueppe, R. D. and H. J. Gottschalk (1970), "Automatische Interpolation von Isolinien bei willkuerlich verteilten Stuetzpunkten," *Allgemeine Vermessungs-Nachrichten,* Vol. 129, No. 10 (Oct.), pp. 423–426.

Grist, M. W. (1972), "Digital Ground Models: An Account of Recent Research," *Photogrammetric Record,* Vol. 70, No. 4 (Oct.), pp. 424–441.

Heiskanen, W. A. and H. Moritz (1967), *Physical Geodesy,* W. H. Freeman, San Francisco, Chapter 7.

Holroyd, M. T. and B. K. Bhattacharyya (1970), *Automatic Contouring of Geophysical Data Using Bicubic Spline Interpolation,* Geological Survey of Canada, Paper No. 70–55.

Hsu, M. L., *et al.,* (1975), "Computer Applications in Land Use Mapping and the Minnesota Land Management Information System," Davis, J. C. and M. McCullagh (eds.), *Display and Analysis of Spatial Data,* New York, John Wiley and Sons.

Junkins, J. L., G. M. Miller and J. R. Jancaitis (1973), "A Weighting Function Approach to Modelling of Irregular Surfaces," *Journal of Geophysical Research,* Vol. 78, No. 11 (Apr.), pp. 1794–1803.

Kraus, K. (1973), "A General Digital Terrain Model," translation from an article in Ackermann, F. (1973), *Numerische Photogrammetrie,* Sammlung Wiechmann, Neue Folge, Buchreihe. Bd. 5, Karlsruhe.

Laboratory for Computer Graphics and Spatial Analysis, Harvard University (1974), *POLYVRT Manual,* Cambridge, Mass.

Mark, David M. (1974), "A Comparison of Computer-based Terrain Storage Methods With Respect to Evaluation of Certain Geomorphometric Measures," M.A. thesis, University of British Columbia.

Morse, S. P. (1968), "A Mathematical Model for the Analysis of Contouring Line Data," *Journal of the Association for Computing Machinery,* Vol. 15, No. 2, pp. 205–220.

Nake, F. and T. K. Peucker (1972), *The Interactive Map in Urban Research, Report after Year One,* University of British Columbia.

Nordbeck, S. and B. Rystedt (1970), "Isarithmic Maps and the Continuity of Reference Interval Functions," *Geografiska Annaler,* Vol. 52, Ser. B., pp. 92–123.

Peucker, T. K. (1972), *Computer Cartography,* Association of American Geographers, College Geography Commission, Resource Paper No. 17, Washington, D.C.

Peucker, T. K. (ed.) (1973), *The Interactive Map in Urban Research, Final Report After Year One,"* University of British Columbia.

Peucker, T. K. (1974), *Geographical Data Structures Report After Year One,* Simon Fraser University.

Pfaltz, J. L. (1975), "Surface Networks," *Geographical Analysis* (forthcoming).

Rhynsburger, D. (1973), "Analytic Delineation of Thiessen Polygons," *Geographical Analysis,* Vol. 5, No. 2 (Apr.), pp. 133–144.

Rosenfeld, A. (1969), *Picture Processing by Computer,* New York, Academic Press.

Schmidt, W. (1969), "The Automap System," *Surveying and Mapping,* Vol. 29, No. 1 (Mar.), pp. 101–106.

Shepard, D. (1968), "A Two-Dimensional Interpolation Function for Irregularly Spaced Data," *Proceedings,* 23rd National Conference, Association for Computing Machinery, pp. 517–524.

Sima, J. (1972), "Prinzipien des CS digitalen Gelaendemodells," *Vermessungstechnik,* Vol. 20, No. 2, pp. 48–51.

Switzer, P. (1975), in: "Sampling of Planar Surfaces," Davis, J. C. and M. McCullagh (eds.), *Display and Analysis of Spatial Data,* John Wiley and Sons, New York.

Tobler, W. R. (1969), "Geographical Filters and their Inverses," *Geographical Analysis,* Vol. 1, pp. 234–253.

Warntz, W. (1966), "The Topology of Socio-Economic Terrain and Spatial Flows," *Papers of the Regional Science Association,* Vol. 17, pp. 47–61. ■