Cartographic Data Structures Panel

Allan H. Schmidt, Presiding
Harvard University

Nicholas Chrisman
Harvard University

Thomas K. Peucker
Simon Fraser University

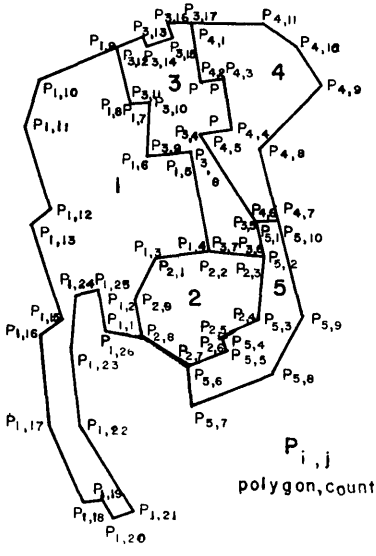
Schmidt: A cartographic structure is the organization of machine-readable cartographic data for computer processing. Too often the importance of the structure is misunderstood or ignored. When developing a data file that will output graphic images, it is necessary but not sufficient to have x-y coordinates recorded on punch cards or magnetic tape.

Within the last 10 yr a variety of approaches have been developed to organize machine-readable coordinates for recording and displaying cartographic data. In most cases data prepared for use with one technique are incompatible with data prepared for a different approach or a different graphical display program. There must be a better way to structure a digital cartographic data base--better in terms of having built-in error detection, scale and detail flexibility, efficient use of computer memory, and adaptability to a variety of computer mapping programs.

When evaluating the characteristics of digital cartographic data bases, it is important to distinguish between external and internal data structures. An external data structure is the organization of data on punch cards, magnetic tape, or other machine-readable media external to the core memory of a computer. An internal data structure is the organization of a given set of data within the core memory. The two data structures are clearly related: the internal structure may include information, such as tables or directories, constructed by the external structure software for use in storing and managing the data, but the internal structure can only contain or draw upon information first provided through the external structure. For example, if the topological attributes of cartographic data are not contained within the external structure, the internal processing cannot include operations which require such information.

One cannot create something from nothing (although the reverse is true); therefore, it is extremely important that the initial external data structure for digitizing data should contain all of the information necessary to satisfy anticipated internal and external data structure requirements. At least two interest groups in automated cartography have significantly different requirements--one for planar data and the other for spatial data such as terrain models. This panel will provide an introduction to both types of requirements.

Chrisman: In spite of the broad possibilities of the field, most cartographic information storage systems solve a limited set of problems and are a poor solution to the more general requirements of geographical analysis. To date, few surveys of methods to numerically represent



SYMAP

LOCATION LIST

A $X_{P_{1,1}}$ $Y_{P_{1,1}}$

$X_{P_{1,2}}$ $Y_{P_{1,2}}$

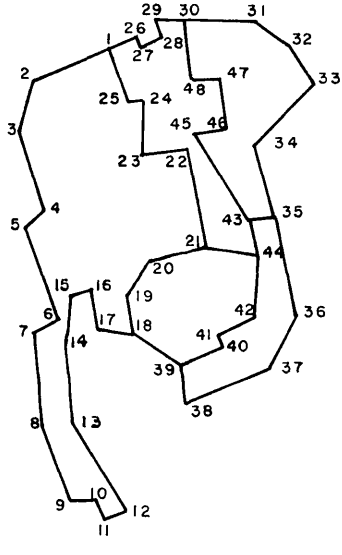
$X_{P_{1,n_1}}$ $Y_{P_{1,n_1}}$

A $X_{P_{2,1}}$ $Y_{P_{2,1}}$

$X_{P_{2,2}}$ $Y_{P_{2,2}}$

$X_{P_{2,n_2}}$ $Y_{P_{2,n_2}}$

Figure 1



CALFORM

POINT DICTIONARY

points

1 X Y
2 X Y
3 X Y
:
48 X Y

polygons

LAOS 1 25 24 23 22 21 44
43 45 46 47 48 30 29
28 27 26 1

Figure 2

space have demonstrated the impact of a specific method on geographical processing. Some surveys have presented many hypothetical possibilities but have lacked evaluation and comparison with a range of problems (Deucker, 1972). Others dive directly into the question of coordinate storage and ignore the basic issue of what to store (Freeman, 1961 and 1973; Peucker, 1973). Methods of coordinate storage, while important, depend on the design of the data structure, which in turn must respond to the present and future needs of cartography and spatial analysis. I will evaluate a range of widely used systems in terms of their ability to handle progressively difficult problems.

• Cartographic spaghetti--In the development of automated cartography, first priority has been the substitution of mechanical drafting for traditional handicraft. Most large cartographic systems have concentrated on the numerical storage of linear features without any expectation of using the cartographic information as part of a larger geographic information system. The resultant mass of digitized features is often as unstructured as a plate of spaghetti. When plotted, identifiable entities and spatial structures may be evident, but the file contains nothing of the sort. For instance, while a map may show the entity called France, there is no guarantee that the cartographic file contains a distinct France or separate treatment of its boundaries.

World Data Banks I and II typify the substantial investments made in this system of storage (Schmidt, 1969). This data structure is simple and compact, and the size of the file is the sum of the space for coordinates. With such simplicity, almost all file correction and verification must be performed by visual inspection and manual intervention.

• Location lists--The earliest and simplest system for numerical representation of geographic entities is the location list. This system describes an entity by specifying coordinates around its perimeter (fig. 1). A continuing series of cartographic efforts from SYMAP and MAP/MODEL through the Urban Atlas project use this straightforward approach (Arms, 1970; Merrill, 1973; Holmes, 1974). In contrast to cartographic spaghetti, each entity is easily identified in the numerical process as well as in the output. Thematic mapping and other analytical applications are based on the ability to identify individual polygons by location lists, but this ability is bought at the expense of the simplicity and compactness of the spaghetti files. Shared lines between zones are recorded and stored twice, causing knotty problems such as sliver areas and overlaps. Correcting of location lists is not automated in most systems but is a major part of MAP/MODEL and MAPEDIT. Besides causing a correction problem, shared lines also nearly double the size of a location list, thereby expanding external storage requirements; however, only the most detailed cartographic files are limited by this constraint.

• Point dictionary files--Point dictionary files were designed to improve on location lists by establishing unique point identification for the whole file (Peucker and Chrisman, 1974). These files contain a list of the coordinate values for the whole map. Polygons and lines are then built up by referencing the dictionary (fig. 2). The result is a more consistent representation minus the sliver-area problems of the location lists. The "cleaner" output led to its use in a number of plotter mapping programs (for example: CALFORM, Laboratory for Computer Graphics, 1969; INTURMAP, Peucker, 1973; GIMS, Waugh, 1973).

The drawback of the point dictionary lies in its storage requirement, but not in the size of the file on external storage. The storage requirement is somewhat less than that of the location list system because the number of point coordinates is replaced by unique points and the old number of point references. The storage problem arises because the entities are not independent. Point references may be random, forcing the whole dictionary to be core-resident. The storage needed for full cartographic accuracy is prohibitive given present technology.

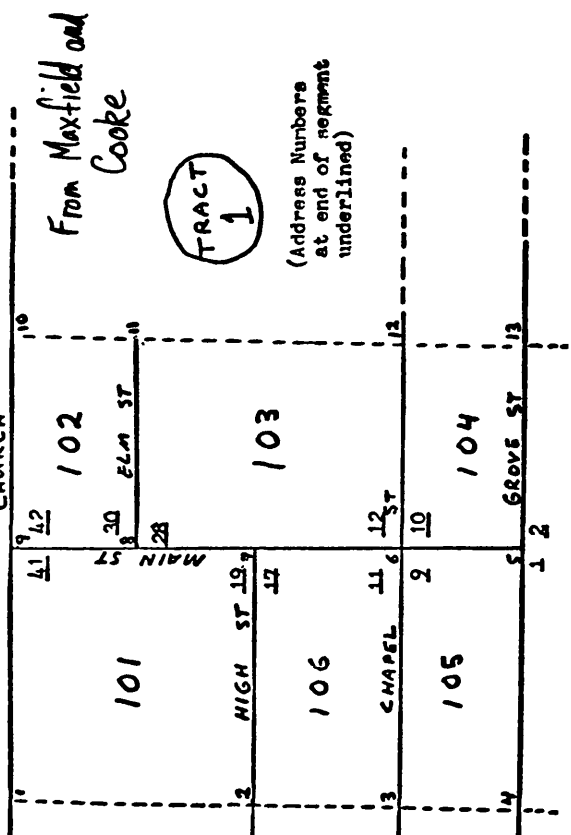
- DIME files--The invention of the Dual Independent Mapping Encoding (DIME) structure by the Census Use Study (Cooke and Maxfield, 1967) signaled a major change in the history of geographic representation. A DIME file does not merely store information about a single targeted entity; features are coded with their topological relationships to other features (fig. 3).

The DIME concept did not spring out of the blue in topological purity. The Census was attempting to organize a mailed self-enumeration in 1970. A system to tie addresses to location descriptors was originally developed as the Address Coding Guide (ACG). It became evident that an ACG was very difficult to verify and impossible to use for graphic display because it contained only streets, not the complete outlines of areas or tracts. By introducing a representation of the connected graph of streets and other features and geographic coordinates, an ACG could be enhanced into a DIME file. The DIME structure allows for topologically based file verification, but the problem of maintenance still remains herculean. A substantial public investment has been channeled into the representation of urban areas, but little cartographic output has resulted.

The mission of the DIME file in the automation of address coding dominates its present form. The unit of storage, the street segment, is small and self-contained. Each segment carries identifiers for a pair of nodes at the ends and a pair of polygon features on either side, plus the address range data. These records are thus fairly bulky. While the segment contains the identifiers of topologically related features, there is no system of directories linking segments together into more complex objects. Due to the fragmentary nature of the file, directories, if created, would be almost as large as the file itself. The file is usually alphabetized by street name to facilitate address matching. Application of the DIME structure to cartographic files has been rare, but the county DIME file is an exception and deserves further investigation.

- Chain files--The chain file, a new structure for representing geographical features, is based on the same topological principles as the DIME system (Laboratory for Computer Graphics, 1974; Peucker and Chrisman, 1974). Whereas the DIME file consists of records that define line segments, the chain file is based on records that define uncrossed boundary lines, each composed of one or many line segments (fig. 4). These boundaries, or chains, connect two end points or nodes and separate two zones. The points between the nodes are cartographically, not topologically, required and thus are subject to line generalization if desired.

The DIME structure, due to its street orientation, ignores the problem of accurate representation of curved boundaries. It is easy to conceive of a DIME application in which 90 percent of the area consists of



DIME Street Segment Records for Main Street

Street	Node(1)	Node(2)	Tract(L)	Block(L)	Tract(R)	Block(R)	Low Add	High Add
Main St	5	6	1	105	1	104	1	10 *
Main St	6	7	1	106	1	103	11	1* *
Main St	7	8	1	101	1	102	19	29 *
Main St	8	9	1	101	1	102	30	42 *

* = Odd left

Figure 3

straight street segments, and the other 10 percent consists of curved features, such as rivers, highways, and suburban streets. If this residual group averaged a modest 10 segments/feature, the total DIME segment file would contain more records from the curved features than from the straight ones. The chain approach groups all the segments of one continuous feature into a single chain. This change makes it possible to approach cartographic files in which 10 segments/feature can be considered small rather than large.

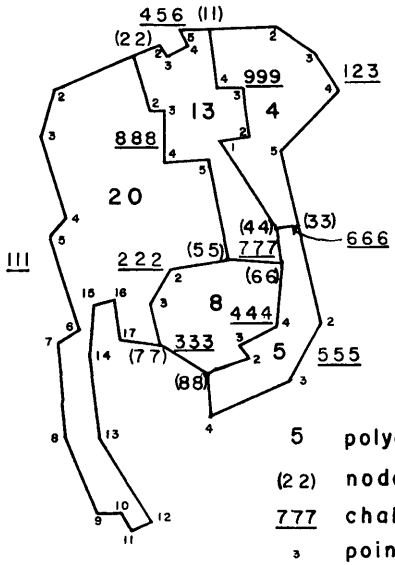
The chain structure can be built from a file as compact as the spaghetti file with the addition of a header for each chain. The header consists of identifiers for the nodes at the two ends and the polygons on the two sides of the chain. Using this information, a set of directories can be constructed to link polygons to their constituent chains or nodes to their incident chains. The space required for these directories is related to the topological complexity of the map, not the cartographic detail. The POLYVART program, a geographic base file converter for which the chain structure was formulated, requires space for all its directories proportional to eight times the number of chains, three times the number of nodes, and twice the number of polygons (fig. 5). Thus to represent the 49 States (plus the District of Columbia) extracted from the County DIME file, the 169 chains and 115 nodes require only 1754 entries of directory space. The original DIME file, perhaps misapplied here, requires 11,372 segment records to perform the same function. Clearly directory building is only practical when boundaries are stored as unified objects.

An extension of the chain structure, due to be implemented over the next year in the GEOGRAF program, will allow for the representation of multiple types of linear and areal features in a single file by expanding the directory system. At the base of the structure the chain and smallest polygon unit will continue to be defined as unbroken linear and areal entities.

PROBLEMS: A discussion of processing problems should demonstrate the relative capabilities of alternative data structures. I will compare the amount of data handled and the storage required, but will avoid chancy comparisons between languages, machines, and installations.

- Line drawing--Production of line drawings is an elementary cartographic requirement and requires little from a data structure, merely the conversion of coordinates into plottable form. Both the spaghetti and chain files insure that each line is processed only once. A DIME file has the same characteristic although random spatial order of the segments reduces plotter efficiency. The location list and point dictionary files are less efficient because shared boundaries are drawn twice.

Frequently a line drawing of a specific part of a file is desired. This part may be specified either by membership in a list of entities or as a coordinate window. All structures except the spaghetti file are capable of inclusion by name or attribute, although the DIME file must be completely scanned for lack of directories. Windowing, however, can be performed with all the structures specified. In the simplest case, the whole file must be scanned to extract the coordinates to include in the window. Any file with objects bigger than a line segment could employ a previously calculated window for each file element. Nevertheless, every object's window must be checked. Further reduction of windowing costs would involve the intersection of a window polygon



POLYVRT
CHAIN FILE (EXTERNAL)

CHAIN	LENGTH	FROM	TO	LEFT	RIGHT
111	18	22	77	20	0
	$X_1\ Y_1\ \ldots\ X_{18}\ Y_{18}$				
999	5	44	11	13	4
	$X_1\ Y_1\ \ldots\ X_5\ Y_5$				
222	4	55	77	8	20
	$X_1\ Y_1\ \ldots\ X_4\ Y_4$				

Figure 4

POLYVRT CHAIN STRUCTURE (INTERNAL)

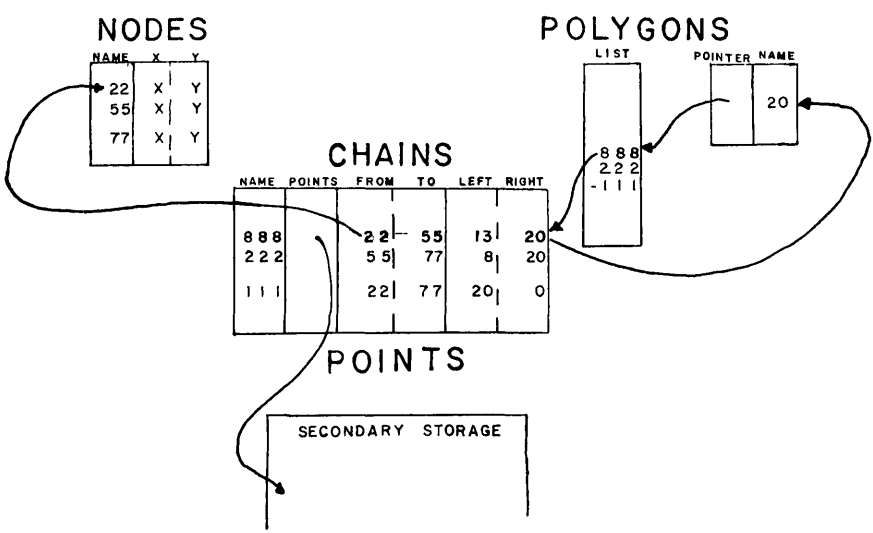


Figure 5

with the rest of the file. A single chain, intersected with a file, could generate markings on all objects within the window at a cost below checking every chain for plotting. Arbitrary windows would then be as easy as rectangular shapes, allowing clipping of files in geodetic form before projection. In practice only the chain file is capable of performing this task at a lower cost than checking every entity's window.

- Retrieving polygon outlines--The problem of assembling the coordinates that define the boundaries of a specific polygon may not occur frequently, but the capability is fundamental in modular systems. Both the location-list and point-dictionary files are designed specifically to serve this need. Since the chain structure includes a polygon directory, the constituent chains can be retrieved, then copied to form the polygon outline. By contrast, the construction of a polygon outline can be costly in the DIME structure. The entire file must be examined to extract segments with the required polygon on either left or right, and then the result must be linked together. By introducing more frequent attempts to link results, the average scan of the file can be reduced, as long as the polygons are single closed entities without islands.

The spaghetti file is unable to retrieve a polygon outline without considerable outside intervention because there is no identification of named polygons and no assurance that any file entity corresponds to a boundary. For this reason the spaghetti structure will not be discussed further.

- Computing areas--The areas of polygons represent a range of geometrical quantities inherent in a cartographic file. Area is computed by summing areas of trapezoids under line segments or of triangles from the Cartesian origin. Thus the general problem is, at the minimum, proportionate to the number of segments in a file. Computing the sums for all the polygons requires duplicate calculations for shared segments. Given the chain structure, the sums for each chain could be added. But if the chains tend to be more complex than straight lines, there would be a significant reduction in processing efficiency; the additional storage of one word per chain, not a great hardship, will improve the efficiency.

Areas can be calculated from DIME files, but neither of the above methods can be transferred. The direct polygon method would require as many passes through the file as there are polygons. The chain approach depends on a directory system and the compaction of chain storage. The best solution would be to read the whole DIME file and generate two records/segment on a scratch file. The scratch records would contain the summation for the segment and either the left or the right polygon identifier; the summation would have opposite signs for the two sides. At the end of the input file, the scratch file will contain double the number of records on the original file. By sorting this file according to the polygon identifier, all sums for a polygon would become contiguous. The sorted output can be read, and polygon areas computed. This process involves not merely the single pass through the source file, but also the writing and rereading of two files of twice the original length (admittedly in shorter records). Furthermore, file sorting is never inexpensive. Computing areas from the DIME file is more expensive than computing from any other data structure.

• Contiguity--Context is the cornerstone of microspatial analysis. Without knowledge of its neighbors each polygon might as well be an independent "spaceship" (Berry, 1973). Despite the growing awareness of contiguity in analytical research, the extraction of contiguity and other analytical data from cartographic base files is not yet general practice because of the complexity of the task. The determination of polygons contiguous to a given polygon is the key to the analytical application of cartographic information.

Only one structure presented here can extract contiguity easily and quickly. Using the polygon-to-chain directory of the chain file, one need only examine the chains forming the central polygon and then collect a list of the polygons to either side. The DIME file also contains contiguity in the left and right codings of each segment, but for lack of directories, a scan of the complete file is required, which is identical to the process of collecting the polygon outline.

Contiguity can be ascertained in a roundabout way from the point dictionary structure if it is properly coded. To find the polygons contiguous to a central one, scan the other polygons for the recurrence of each pair of point references in the central polygon. An optimization can occur once a match is found by attempting to follow the shared boundary (Evey and Mantey, 1974). The number of comparisons will remain proportional to the square of the number of point references, hardly desirable.

Determination of contiguity from the location-list structure resembles the point-dictionary problem, except the coordinate equality test replaces the point reference equality test. Since coordinates can not be expected to match exactly, more costly tolerance-based checks must be used. Unfortunately the use of tolerances can produce undesirable results when different points are closer than the tolerance.

The method described above is only possible if shared lines are coded similarly on either side, a faulty assumption in most cases. In figure 6 the side of polygon C may be coded in either system as extending from point 1 directly to point 3, because point 2 is colinear or nearly so. The MAP/MODEL package, among other features, provides a mechanism to correct these errors in location-list structures (Arms, 1970). The segments are first identified by their center, and a match is attempted. More rigorous matching and correction is performed until every segment's duplicate has been found or broken out of another segment. This process requires the definition of a hull polygon so that the outside edge is also duplicated. In general the computation involved in matching segments greatly increases the process previously described. The economics of contiguity is entirely different for topological and non-topological systems.

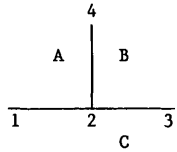


Figure 6

• Polygon overlay--Perhaps the largest analytical use of geographic base files is the comparison of different sets of polygon zones. Most frequently, different variables relate to disparate entities and are thus not easily reconciled. By overlaying and creating an intersection set, an automated process can replace visual inspection as the basic tool of spatial correlation. Furthermore, contiguity statistics will be greatly enhanced when not restricted to information gathered by a single class of polygons. The most pervasive approach to polygon overlay ignores the use of cartographic base files and utilizes uniform grids.

The grid system forces a nasty choice between enormous bulk and resolution inaccuracies. Once a grid is defined, all data must be related to it, and flexibility is limited. The grid system survives not because of its virtues, but because the polygon-based methods of overlay have been expensive.

The overlay of polygons from the location-list structure is performed by MAP/MODEL and many other systems. Every approach reduces to the intersection of each line segment in each polygon of set A with the line segments of the polygons in set B. By using coordinate windows for the polygons in set B, the amount of calculation can be reduced. The time required is proportional to the number of polygons in set A times a subset of the total segments in set B. The size of this subset is determined by the window elimination; the set is smaller for square polygons than for arbitrary shapes. Due to the nature of the shared lines between zones, every boundary intersection must be found twice, once for each side. This solution to polygon intersection makes the best of a limited data structure, but it is expensive when applied to files with many zones or with detailed outlines.

The DIME system is not designed to perform polygon intersections, since it is based on the storage of the smallest urban unit. However, someone will always find a reason to subdivide a unit, no matter how small. The assessor's files of land ownership are an important set of polygons below the block level. Lack of directories makes the economics of overlaying DIME files similar to those of the grid system. The elimination of duplicated lines will be offset by the lack of an entity on which to base window elimination.

Polygon overlay based on the chain structure is remarkably different. The intersection process can be guided by contiguity information to limit calculations. Once a starting point is established, each chain can be treated by using a depth-first search procedure. Depth-first search means that the starting point of each chain will always be known in terms of its location in the other set of polygons and chains. The intersection of each chain can thus be limited to the single polygon it splits. When the chain crosses a polygon boundary, the current polygon is set to the one on the other side. This technique, described in more detail in figure 7, demonstrates the power of topological information in handling complex problems. Only one object in the first set, the starting node, is checked against the whole of the other file. All the coordinate information in the chains is checked only against the polygon currently occupied. With reasonable shortcuts, such as using chain windows in the intersection process, the savings will be great. Additional storage for this algorithm is required for "status flags," for objects processed, and for the stack of the search procedure. Whereas the polygon-based methods involved the multiplication of terms to yield time, this method is not inflated as the files become bigger. The time bound of the algorithm in figure 7 is proportional to the number of chains in the first file and the complexity of polygons in the second. Large files may thus be processed without undue increase in cost.

The problems presented range from line drawings where geographical entities can be distinguished by the human eye to polygon overlay where the automated process must understand the full network of spatial interactions. In the simpler tasks the savings introduced by a complete topological structure are not startling. The first advantage discovered was the ability to retrieve more than one type of entity. As processing developed a greater need for full geographic representation, the gap

An Algorithm for Polygon Intersection

Given: two chain files, set A and set B.

Find by coordinate comparison the object in B overlaying ROOT, an arbitrary node in A. This node is the root of a depth first search which drives the process. Initialize stack for search procedure with ROOT and its associated object in B.

```
while node stack non-empty do begin
   if all chains incident at node ATTOP done then
            pop node stack else
            begin
            CA := clockwise-most undone chain incident at ATTOP;
            POSIT := object in B overlaid by ATTOP
            while segments left in CA do
               begin
               if POSIT is a node then
                    begin
                      decide which chains in B CA falls between;
                      PB := polygon between chains bracketing CA;
                    end; comment node incidence reduced to polygon
                if POSIT is a chain CB then
                    while POSIT=CB do begin
                        if segment of CA departs from CB
                           then POSIT := side of deviation
                           else next segment
                    end;
                if POSIT is a polygon then PB := POSIT
                assemble chains surrounding PB
                mark PB split
                while POSIT = PB do begin
                    if segment still inside PB then next segment
                        else begin; comment found intersection
                            mark intersection; mark CB split;
                            POSIT := object at intersection;
                        end;
                end;
            associate POSIT with GOTTO, terminal node of CA
            mark CA done
            if GOTTO not marked reached
                then begin
                  mark GOTTO reached
                  push GOTTO onto node stack
                end;
end;
```

all polygons in B not marked split can be assigned the A polygon of a neighbor

Figure 7

widened. In the final problem of polygon overlay, the chain-based algorithm registers very significant reductions in processing costs. The increased complexity of the chain file creates a sensitivity to geographic representation. The resulting efficiency can not be ignored.

In cartography the computer should not be used merely to emulate the operation of a draftsman. Cartographic data bases should provide a means for analyzing geographic information. Statistical and cartographic information should be linked to allow thematic and other analyses; the cartographic file should provide a model of the area for the analyses. Relationships between areas in topological structures are vital to both cartographic efficiency and analytic acuity.

Peucker: I would like to trace the discussion of cartographic data structures back to 1967 when Boehm wrote his important but often overlooked article comparing several surface encoding procedures. Surfaces were coded by two similar types of contour storage, the regular grid and a rectangular grid with varying mesh width. The objective of the research was to find optimum systems for data compaction and manipulation. The study showed that the variable grid, or contours, enabled the highest data compression but also resulted in the highest manipulation expenses. The regular grid, on the other hand, was fast to use but voluminous to store. The conclusion--to use one grid for storage and another for manipulation of the surface--is impossible since converting (by interpolating) the surface from its first to its second definition is one of the most time-consuming procedures. The solution must be found elsewhere. Unfortunately, Boehm minimally explored the reasons why opposite approaches gave optimum performance for the two criteria given, and few followed in his steps. This neglect most likely accounts for the relatively slow development of more sophisticated data structures in computer cartography.

Three aspects of surfaces and points on surfaces influence the storage of the surface. The first factor is surface behavior, the average change in height over space. The surface changes from area to area (i.e., it is nonstationary) and with it changes the density of points required to maintain a constant error term for the estimation of a point. Contours adapt to the changing sampling density; regular grids do not.

The second factor is the topology of the surface points. For most manipulations of surface data, a knowledge of the internal connections is necessary. Topology is defined implicitly in the regular grid: the location of a point within the set (matrix) of points simultaneously gives its neighbors. A system of contours, on the other hand, makes it very difficult to find any neighboring points except the two on the same level. All other points require laborious searches through adjacent contours. Other search routines are needed to find a given number of closest points for the interpolation of an irregular grid of points.

The third factor is the relation of sampling points to the surface. One can distinguish between surface-random and surface-specific points. The first does not give any information about the surface beyond their own height. The second allows estimations about the neighborhood of the points and represents certain types of points or lines on the surface breaks. Clearly, encoding the same number of points of each type will result in higher accuracy with surface-specific points.

These three concepts are a good basis for comparing surface information systems, or digital terrain models. The most frequently used data structure is the regular grid. The surface is sampled at regular distances in an orthogonal network. The regular grid is the most efficient data structure for defining the internal topology of the surface points: neighborhood relationships are implicitly defined by the regularity of the grid. However, the structure performs very badly with respect to the other two components of a surface. To maintain a certain error variance the mesh of the grid has be to adjusted to the roughest terrain and thus results in a high redundancy of surface definition in less undulated terrain. Also, it is highly probable that surface-specific points and lines will fall between grid points and therefore the representation of the points of highest information content is very poor.

Data redundancy could be reduced by using a regular grid of varying mesh widths and adjusting the grid density to the roughness of the terrain. Additional computations necessary to adjust to the variation of the density should not be too high nor the programing too difficult. But little work has been done in this area, and only one attempt is described in the literature (Boehm, 1967). This structure would probably have a higher chance of missing surface-specific points and lines than the regular grid.

Systems based on contour encoding adapt readily to the irregularity of the surface. Therefore, the storage density is proportional to the roughness of the terrain. However, this system is very clumsy for the definition of neighborhood relationships and thus for an efficient manipulation of the data. To find neighbors one must search along contour lines above and below the present contour level until the closest points on neighboring contours are found. The process reveals information such as slope and direction of ascent. This system is not very efficient in the representation of surface-specific points and lines, although ridges and rivers are represented quite well in fluvial systems where changes in contour direction are very sudden; in addition neither surface-specific points nor gradual changes in terrain are covered.

When selected to recognize surface-specific points and lines, irregular points adapt quite well to the roughness of terrain. However, a topographic structure has to be explicitly included in the data base. In our research a file has been added to the point file which gives for every point the appropriate labels of all neighboring points.

Aangeenbrug (U.S. Bureau of Census): I don't think that it would be very fruitful at this meeting to continue confusing cartographers. I would like to comment on some of the topics discussed by Chrisman. First, some of the statements about DIME are confusing to me, and I'm somewhat familiar with DIME. DIME is a generic principle that has been described in the literature. The Geographic Base File (GBF) DIME system is an administrative file structure that many metropolitan areas have. Our concern, however, is the dual incidence, or matrix, coding system, which has many properties that apparently I've denied it. Several of the papers concerning these properties will be available tomorrow. One paper, entitled "Arithmecon," describes the multidimensional option of dual-incidence encoding that Corbett is working on. You must be careful not to confuse the operating file system and DIME. DIME is a conceptual model. DIME files themselves are an extension of these models, and there are many types. It's not really relevant to focus one's remarks without more substantiation. I would like to alleviate

this problem by providing literature and a chance for some of you to meet with Corbett and Farnsworth, who have more experience in these areas. There just isn't enough money in urban areas in this nation for ego trips. Too much selling is going on here. I'm much more interested in learning and understanding.

Chrisman: Sir, I'm very sensitive to your feeling that I'm trying to sell something; I am not trying to sell my ego. I have nothing to gain, nothing to lose. I've attempted to establish my concept of the DIME file from all the DIME files that I've seen. The crucial difference between the DIME file and a system of more cartographic ability is the move away from storing separate two-point line segments. Only a very rare cartographic file has many boundaries which are only two-point line segments. I admit that the more complicated topological structure that I'm presenting has evolved from the DIME file. But to get an explicit topology, you can not be continually burdened by a system based on single segment entities. Perhaps I'm making a distinction between DIME in its actual form for present use and DIME as a topological theory for special representation. I am borrowing the topological system for special representation, but I'm sure that the Census Bureau is working on it. What I'm presenting here as the chain structure seems to be the arc structure that USGS is working on, which resembles the system that Raytheon developed.

Aangeenbrug: To clarify, I'm much more concerned with theoretical implications. The working files are, I admit, primarily administrative files, but the DIME file is generically a set of vertices that define boundaries and coboundaries.

Peucker: But that's topology.

Chrisman: To make the DIME file an analytical tool, one must have access not just to the vertices, but to the vertices from the nodes, from the polygons, and from a set of directories of larger chains based on the small chains.

Tsao (Dept. of Natural Resources, Montana): I would like to comment from the user's viewpoint on what these gentlemen have discussed. We tried to develop a system in Montana, called Environmental Resource Geode Information System (EVRGIS), for land-use planning only. This system has four subsystems--data input, data storage, data output and retrieval, and data manipulation. The most important subsystem is data manipulation. You must consider purpose as well as input and output. We are using the microcell rad system for transmitting corridor selections. For example, with a scan resolution of 0.02 in, the cell size is about 50 ft square, and the transmission corridor is 300 ft wide. For transmission the grid system is much better than the polygon system. We tried the polygon system with an overlay, a 12-in-square map, and it took at least 10 min of Central-Processing-Unit time to superimpose one map on another. But the microcell approach takes only 10 to 20 sec.

Chrisman: How many cells do you have for Montana if you use a 50-ft cell?

Tsao: Montana is about 147,000 mi^2.

Chrisman: Yes, but you're storing 50-ft cells.

Tsao: The data are compressed without any problems. Using a study

area of about 60,000 mi^2 mapped at a scale of 1 in to 2 mi, we stored the data on tape using code compression. Each type of data is stored separately so that it is easily retrieved and doesn't get mixed up.

Chrisman: I agree that overlays should be stored separately, but I still think that we have new efficient algorithms for polygon overlay.

Rhind (Univ. of Durham, U.K.): Unfortunately we are often stuck with divisions of our data. We frequently, certainly in the U.K., don't have the choice of using grids or irregular areas. Thus we must have both capabilities. Grid data have just as many disadvantages as advantages. And in overlaying polygons of any kind or shape, you make assumptions about the internal homogeneity of the data distributions.

Chrisman: You're making the same assumptions with the grid system.

Rhind: I agree. But I think that often the errors attendant in the grid system are greater than those in the polygon system. This point cannot be proved; that's why I'm making it now. You can't decide what kind of data structure to use without considering the tasks to be performed. You must consider what kinds of data access you will require.

We might discuss three types of data access. Access methods for much cartographic data, pantograph representations particularly, have been straight sequential methods. With geographic processing we have a predefined aim. For example, (I'll take the grid square case because it is simple) to define population within 5 km of each point in D.C., you go to each point and look around it. The same process is used for contouring. In using the sector typefaces, you look at eight different sectors. Before you start you know that you must look geographically at all the points or at a selected subset of the points. The most complicated kind of accessing involves many more pointers and looking up the data, going to one particular point, and choosing outwards from that point depending on what you find in the data. If you are looking for high densities of population and you do not want to build on the basis of administrative boundaries, then you have an accessing problem in which many pointers are needed to be efficient. Obviously, we can't have an all-singing all-dancing system without dreadful inefficiencies. A great deal of careful planning is necessary to set up these structures.

Dobson (State Univ. of New York): I am probably a strange creature in the audience because I'm a cartographer. I have used pen and ink and have scribed. People who talk about cartographic spaghetti, problems of lack of structure, polygon files with cartographic ability, and cartographic data should realize that many people in the audience are unaware of some of these matters. We are more excited about what the product looks like than about statistical or mathematical manipulations or probabilties. Many people seem to be picking on World Data Bank I, a damn fine tool. I think you are getting confused about objectives and semantics. Consider, for example, "cartographic ability"; I don't know how a polygon has cartographic ability.

(Unidentified speaker): Come to our workshop. We have this structure working for interactive mapping with World Data Bank I. The output was generated from the metropolitan DIME files and the county DIME files. The present structure of World Data Bank I does not support interactive mapping, but we can perform this task because we have added the DIME principle to the cartographic files.

Dobson: I will come to see your output, but the cartographers here are really interested in products. Certainly, there are different types of discussions going on, but some of the theoretical points are not appropriate in this meeting.

Peucker: At least two people at this table are cartographers. Through research they have found that mathematicians and computer scientists have been working in this area maybe longer than cartographers. Cartographic terminology has no more right to be used in this area than mathematical or any other terminology. The term polygon may be used because we are performing mathematical operations on it. As a Canadian I would object to using the term State because I would like to call the area a Province. But, if you suggest that we call it Polygon Virginia, I will accept the term. The terminology doesn't have to be cartographic. At every session we have one guy who says, "Wouldn't you like to use cartographic terms?" Cartographers have changed the terminology, too. Algorithm wasn't algorithm all the time. What is wrong with using other terms?

Dobson: I am not objecting to using other terms but to some types of goals. For instance, you are familiar with my communication with Tobler regarding his paper. He came to the dramatic idea of having a symbol for each different item on a choropleth map. The idea is fine, but it looks like hell. We have 105 county maps and 105 lines. The quorum on stage lacks the balance that exists in the audience.

Chrisman: I don't agree with Peucker. I like 7-class maps; my mother likes 13-class maps. When I bring home fancy output for my mother, I can't bring home a 7-class map.

Dobson: As the Moody Blues say, "There is a question of balance," and balance is what I am looking for.

Peucker: The discussion of Tobler's paper shows a theoretical misunderstanding of the whole issue, and not by Tobler.

Dobson: Tobler suggested that I send my reply to Geographical Analysis.

White (Census Bureau): At the risk of boring the cartographers, I would like to ask Peucker to give some more concrete details about the graph structure that you are coding with surfaces and volumes. I don't understand what you mean by surface-random point.

Peucker: If you put a regular grid on the surface, you get a distribution of the heights, or a height frequency table. This grid gives an average distribution of the heights, but often omits important points, like peaks or passes. Therefore, it is nonrandom with respect to x and y, but random with respect to the surface features. This situation is true for any random x, y distribution of points not directed towards the identification of lines and their special points. We create a data set which approximates a surface with a given error variance and then triangulate the data by assigning pointers to all the neighbors. Usually, we have 6 pointers; but no more than 8. These pointers have label identifications to other surface points, sorted clockwise, so that you can triangulate with any two points. One frequent use of the process is finding triangles and then working with them; another use is crossing an edge and finding the other edge which crosses the traingle again.

Next we extract from that data set a second data set that is restricted to the main ridges and channels of the system. Often we can't link them naturally, and we do so artificially. This structure again makes for a thinner and, therefore, shorter system. The file is created on three levels. First is paging--1,300 points/page; then data sets of 32,000 points, the extent of our numbering, are formed; and finally regions of data are organized.

Moellering (Ohio State Univ.): It is very difficult to say that one data structure is better than any other because the choice of data structure depends on your goals. However, theoretical discussions are not irrelevant. One of the points made yesterday was that automated cartography includes several functions. One is analysis and another is data display. This is what interactive geographical information systems are all about. If you have a CRT system but are interested in a smaller system for research, you can design one to automatically analyze geographical data, sort a cartographic base file, and produce a cartographic display in real-time. But there must be more discussion between the people who are designing geographic information systems and data structures and the people who require cartographic displays and automated cartography.

Schmidt: Your emphasis on goals is a point well taken. I hope no one here got the impression or made the assumption that Chrisman's or Peucker's talks were intended to present the optimum. We are trying to identify a number of alternatives which have been developed as well as some emerging alternatives that we feel have sufficient abilities. We are merely revealing new ideas about how to use cartographic data.

(Unidentified speaker): What is missing is documentation for the optimal use of each method. If someone could formalize this information in one source, it would be very helpful.

Thorpe (Natural Environmental Research Council): We have addressed the same problems discussed yesterday and today regarding hardware and software for the cartographer. The data structure that we developed is a disk-based system. I have a short film concerning the problems of area overlay, automatic area recognition, area mesurement, and point in polygon. We have had this information available for at least a year. What is the big hassle?

Peucker: We also discussed these topics a year ago.

Rockwell (N.J. Dept. of Community Affairs): We not only need to know what is available, but who has it and where it is. I don't know who the cartographers are and who else might be represented at this meeting. A list of the people present and their areas of interest might be helpful.