MAPPING FROM A TOPOLOGICALLY ENCODED DATA BASE: THE U.S. BUREAU OF THE CENSUS EXAMPLE

Frederick R. Broome

Geography Division
United States Bureau of the Census
Washington, D.C. 20233
U.S.A.

ABSTRACT

The use of a topologically encoded data base as the source for map production introduces some new problems along with the benefits of the file structure. The topologically encoded structure (TES) provides the means for the construction of map features that are not easily derived from other data base structures such as the string and polygon structures. The U.S. Census Bureau's Topologically Integrated Geographic Encoding Referencing (TIGER) File is an example of a TES. Because the TIGER File is not primarily a cartographic data base structure, problems occur when using it for automated map production. Many of these problems can be overcome by the introduction of a cartographic shell between the TIGER File and the map producing programs.

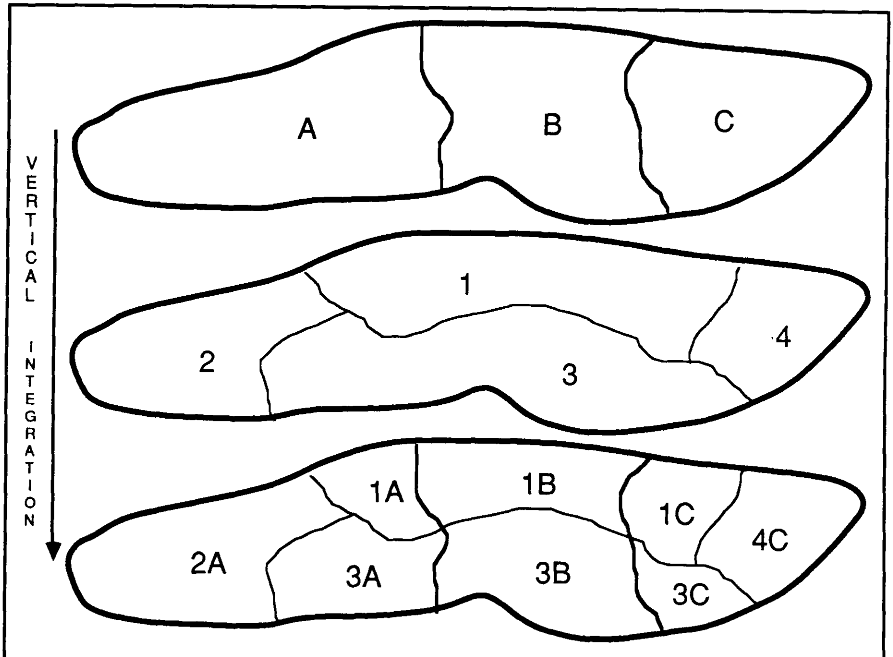
INTRODUCTION

The increased demand for automated geographic systems in recent years has begun to surpass the demand for automated cartographic systems. The expanded use of geographic systems has forced an integration of cartographic needs into the design and construction of geographic systems. This has brought many benefits to the field of cartography while at the same time has exacerbated the problems associated with using an automated system for the production of maps. This has further reduced the cartographic clarity of data structure and content that would result from a pure cartographic system. This paper will discuss the benefits and problems of using a fully topologically encoded data base structure as a cartographic source for the automated production of maps.

TOPOLOGICAL FILE STRUCTURE

A topological data base structure is one that forces a very specific set of relationships upon the basic entities in the data base. At a minimum, a topologically structured data base for geographic information requires the identification and specification of the relationships between the fundamental geographic entities of areas, the lines bounding the areas, the endpoints of the bounding lines, and the lines converging at an endpoint.

A fully topologically encoded data base has two further constraints. First, all information must be encoded as if it has been projected upon the same surface. That is, there are no levels of data in the basic entities of areas, lines, and endpoints. While membership in a level may be an attribute of an entity, the fundamental data base relationships do not recognize entities as existing separately. Thus, an area in a totally encoded structure is the smallest, unique area represented by the intersection of all higher-level areas (Figure 1).



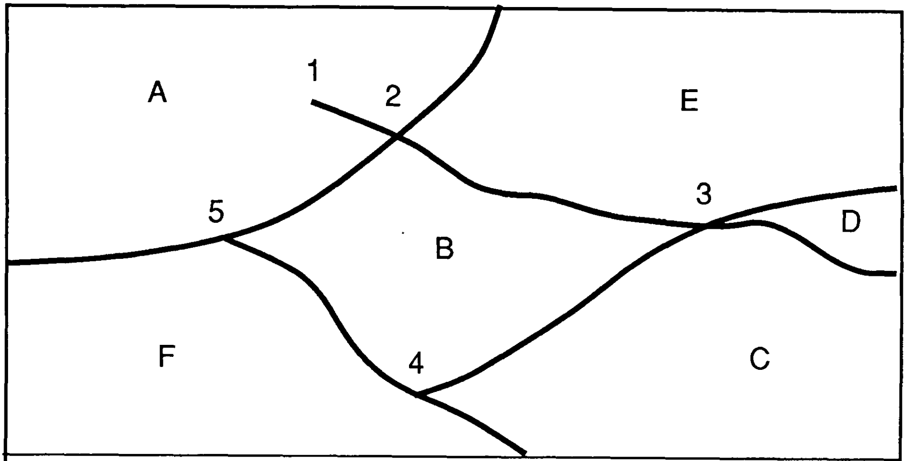
Larger geographic areas are intersected to make 2-cells.

Figure 1

Second, all data base elements, files, codes, descriptors, and so on must be tied to the basic file entities of areas, lines, and endpoints. No data are permitted to be freestanding in the file. All elements must in some way be represented by an attachment to or a grouping of the basic file entities.

The United States Bureau of the Census has implemented a fully topological encoding of its geographic data. The file and its structure are known by the acronym TIGER, which stands for Topologically Integrated Geographic Encoding and Referencing. The TIGER System, which includes the TIGER File and all associated programs and hardware, is designed to support the geographic needs of the Census Bureau through the next decade and beyond. One of the three most important geographic products of the TIGER System is the automated generation of maps.

The terminology used to describe the TIGER File content and structure has precise meaning, as opposed to the more generic entity names of "areas," "lines," and "endpoints." In terms of the TIGER File, a basic area is a 2-cell, a line connecting endpoints is a 1-cell and an endpoint of a 1-cell is a 0-cell (Figure 2). Hereinafter, the terms 2-cell, 1-cell, and 0-cell will be used to describe the basic topological entities of a topologically-encoded structure (TES) as well as the TIGER File. The generic terms of areas, lines, and endpoints will be used in a more general sense.



Alphabetic letters denote 2-cells and numbers denote 0-cells. 1-cells are lines connecting the 0-cells.

Figure 2

The TIGER File structure is founded upon the three basic entities: the 0-, 1-, and 2-cells (Figure 3). The relationships between and among these entities, and in some instances, the relationships to attributes, descriptors, and groupings are implemented through a series of pointers. A detailed description of the TIGER File content relationships is available from the United States Census Bureau. The complete file structure, while is conceptually simple, is somewhat complex to diagram due to the combination of one-to-one, one-to-many, and many-to-one relationships all bound together through a series of pointers.

The reduction of all data in the file to either one of the basic entities or to a relationship to one of them is the strength of a topologically-encoded file structure. This means that all data and information needed and available in the file for mapping can be extracted via the basic entities and their relationships. Some information can be extracted directly because it is given explicitly through the structure or content. Other information is implicit and must be derived through the structure relationships. The difference between explicit and implicit information is a key to the efficiency of any cartographic activity.

Benefits of a TES

There are several benefits to cartographic activities resulting from the

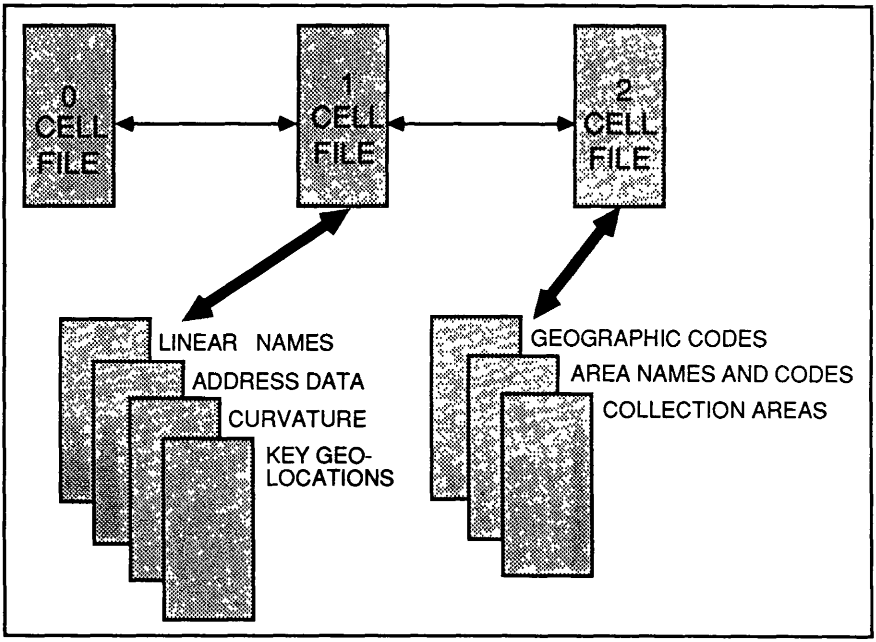
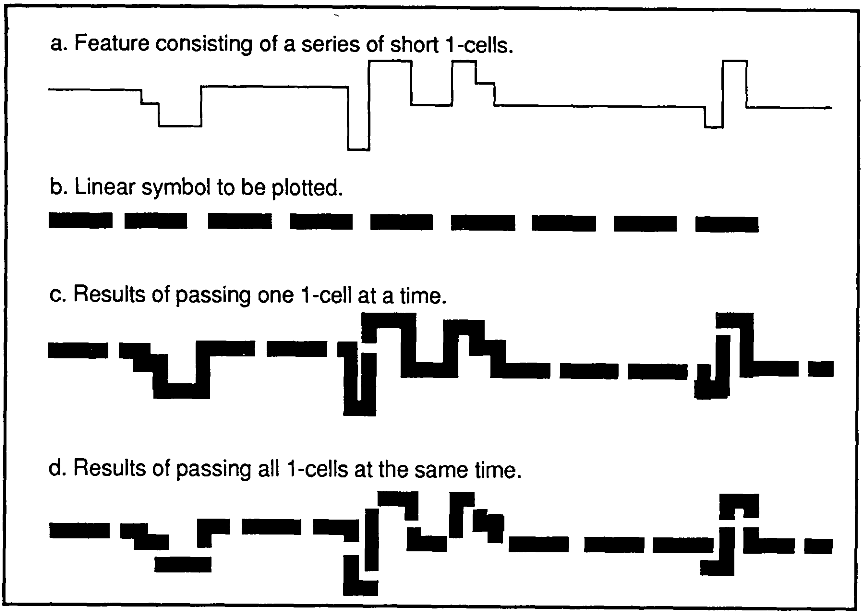


Figure 3

use of a TES; benefits that are not necessarily found in other file structures. A TES contains knowledge of neighbors, both areal and linear. A TES contains knowledge of the intersection of areas and lines. Feature coincidence also is known. Finally, a TES allows almost direct construction of areas and lines that are not explicitly defined in the file.

Knowledge of neighbors permits the aggregration of basic entities into longer lines or larger areas. When symbolizing a linear feature, the images produced through most algorithms yield the best appearance if the whole line is passed at once and not fed piecemeal to the algorithm 1-cell at a time (Figure 4). Note the chopped up pattern when the plot routine is given a single 1-cell at a time, in contrast to the more pleasing pattern when the whole feature is passed at one time.

Likewise, many area-fill algorithms work more efficiently when all contigous areas to be filled with the same pattern are passed at once. (Unless otherwise stated, area fill referenced in this paper means software-generated fill, external to the output hardware.) Since direct knowledge of neighbors is inherent in the TES data structure, all parts of a feature or group of features can be gathered, and--important to some implementations--they can be gathered sequentially along or around a feature. The sequence of parts is encoded at the time the feature is entered into the system. Thus, the sequence does not have to be derived each time the feature is to be symbolized. This represents a major saving in computer processing.



a. Feature consisting of a series of short 1-cells.

b. Linear symbol to be plotted.

c. Results of passing one 1-cell at a time.

d. Results of passing all 1-cells at the same time.

Figure 4

The knowledge in the TES of line feature intersections, and to some degree area intersections, yields other saving in terms of computer processing time. Most cartographic data base structures do not provide this facility directly; particularly those structures that maintain the data on separate levels as if they were independent of other data. Intersection information is useful, for example, when placing text along a line. Knowledge of where all other features intersect eliminates search time to find those points along a line where other lines also may intersect text. Feature intersection determination is one of the most computationally expensive operations. Consequently, the savings are increased vastly when a data base is used frequently to produce maps.

The fact that two features that follow the same path are represented in the TES by only one 1-cell has a major impact upon both file structure and use. The 1-cell is given the code of the most physically apparent characteristic such as "a class 4 road overpassing a river." The other, less apparent characteristics are attached to the 1-cell as flags representing additional coincidence characteristics, such as the 1-cell also is a corporate boundary. Coincidence encoding is a requirement of a TES file because all features are projected upon the same surface. Thus, coincidental features become only one entry in the file. The obvious advantage is that only one 1-cell record is required to store all the information along the 1-cell's path, and only one 1-cell record needs to be searched for this information and the associated relationships. As will be discussed later, this introduces some difficulties in file use for cartography, but the benefits of structure simplicity and data quality more than make up for not having separate coincident records for different features.

A TES allows the almost direct construction of lines and/or areas not explicitly defined in the file structure. By merely querying the file for entities that meet certain criteria, such as all roads within 10 kilometers of a water body greater than 100 hectares in surface area, one can flag these in the file or construct a separate list of, say, "water recreation access routes." In terms of areas, grouping together the 2-cells into some higher, but previously undefined areal unit like school districts, one creates a list for rapid access by school district. This can be carried a step further by using the new grouping of 2-cells to flag the 1-cells bounding them as school district boundaries. The latter capability is particularily useful in finding and symbolizing the boundary of the new units, since each one probably will consist of two or more 2-cells. Thus, the system eliminates the need to test all 1-cells as a school district boundary each time a map is needed. This results in a tremendous saving in computer power for areal and boundary units that are used multiple times.

Problems

It is a law of data bases that no data base structure is the most computationally efficient for all uses. The TES is no exception. The TIGER File implementation is a good example to demonstrate the problems one has in mapping from a TES. The TIGER System was designed primarily to respond to three types of almost mutually exclusive inquiries: geographic coding, geographic structure, and mapping. Geocoding is the assignment of addresses or locations to the geographic areas within which they are located. Geostructuring is the use of the geographic codes to control collection, tabulation, and publication of the data. Many data base structures have been developed to accomplish these tasks separately. The TIGER System plans to do them all.

Each file structure has its cartographic and computer strengths and weaknesses. A polygon file structure has all the boundary coordinates in sequence and ready for use by area-fill algorithms and some boundary-symbolizing algorithims. Thus, a polygon structure is superior for these automated cartographic functions.

A feature-string structure has all the coordinates for a given type of line together. It permits easy symbolization of the string without regard to what the string overlaps or bounds. This also is superior for these automated cartographic functions.

Cartographic structures generally are designed around the type of map to be produced, while a TES, which is oriented to geographic information, is not. Thus, the TES is not as efficient as either the polygon or string structures for many cartographic operations.

The magnitude of the TIGER File prohibits storage and maintenance of all polygon and linear relationships explicitly. While this is not strictly speaking, a problem unique to a TES, it is related in that there is a practical upper limit to explicitly stating within a TES all possible area boundaries. The non-explicit boundaries must have their missing relationships constructed each time they are required for plotting. For example, the state boundaries constitute polygons that affects less than five percent of the records in the TIGER File; the national boundary affect less than one percent. Every record, whether part of the state or international boundary, would have to have pointer space reserved in each entry if the boundaries were to be explicitly linked into a polygon. To maintain sequenced pointers for this would place a significant overhead on the system. The cartographic impact is that there is a problem for area fill or line symbolizing of constructing relationships not stored.

A production problem for mapping also results from file size. The large files must be partitioned into subfiles. This is done geographically in the TIGER File rather than by feature type. This is not unique to a TES, but a TES handles it well. Thus, to map a single feature over a large area requires accessing multiple partitions. A TES has knowledge of its neighbors, thus of neighboring partitions. The problem comes in having to load in partitions which due to system's capacity, probably will not be online. This usually means slower response and increased cost per map when the map crosses partition boundaries.

It is extremely important to remember that a TES is not a graphics structure, i.e., is not merely the digital representation of a map. This means that items like text position, sheet edges, and so forth are not stored in the file and must be recomputed each time they are required. Since there are few, if any, efficient text placement algorithms, this is computationally expensive. In the case of the Census Bureau's use of the TIGER File, interactive editing of the text placement is economically prohibitive due to the volume of maps to be produced and the short time in which to do it (over 400,000 within one year.) Thus we must accept the results similar to that shown in Figure 5.a. while desiring results similar to Figure 5.b.

THE CARTOGRAPHIC SHELL

Some form of cartographically oriented management subsystem is needed to take advantage of the benefits and to overcome the problems of using a TES such as the TIGER File for volume map production. Such a subsystem called a "cartographic shell" is under development at the U.S. Census Bureau. The shell (Figure 6) will act as the interface between the TIGER File and programs for map production, management and digitizing control.

The three primary functions of the cartographic shell are:

1. Manage map production and TIGER File update control files

2. Maintain libraries of routines for access to both the master TIGER File and any cartographic extracts

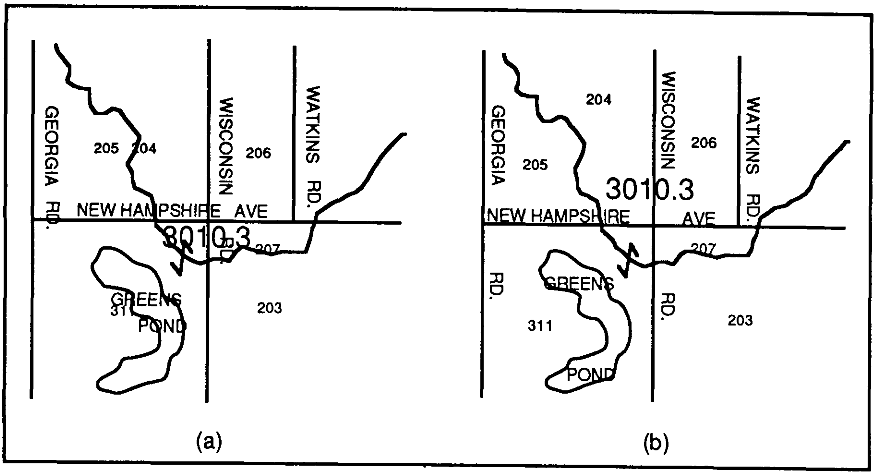


Figure 5

3. Provide an efficient link between the map and the master TIGER File through which coordinate based changes can flow

The value of a cartographic shell is enhanced by the Census Bureau's policy that changes to TIGER File data are to be made only to the master file and not to some intermediate file or product such as a map. This policy overcomes one of the biggest problems facing users of a dynamic data base, that of currentness. When an error is detected, the master file's integrity is not jeopardized because all changes must be applied to the master file and new products generated. Thus, some method of controlling a map oriented application program's access to the master file is necessary.

The production of a map under the control of the cartographic shell may flow as follows:

1 - Application program (map plotting program) checks the TIGER File update control file to see if the master file partition has been flagged indicating that updates have occured.

2 - If yes, all previous cartographic extract files, if any, are re-extracted.

3 - If no, the master file partitions are brought online along with any required cartographic extract files.

4 - The map production control file is checked to see if a map of this type and area has been generated.

5 - If yes, the plot file is automatically identified and a request for plotting is generated.

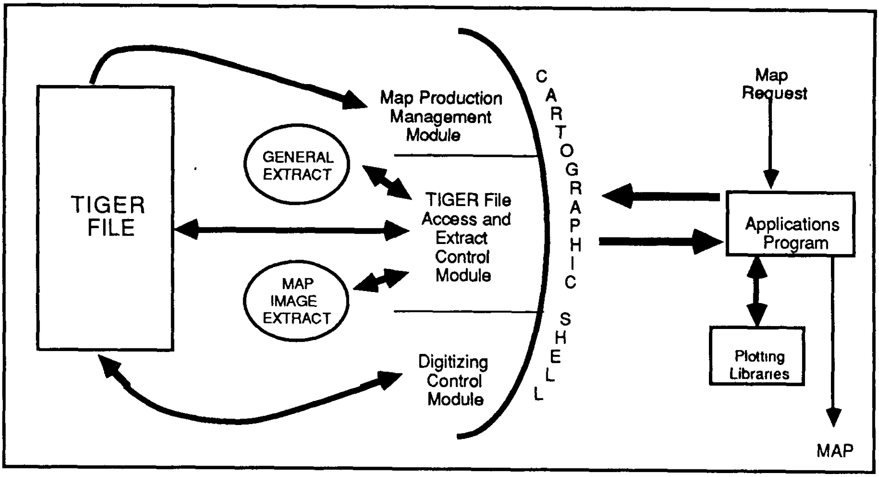


Figure 6

6 - The map plot is produced.

7 - If this is an update of an old map, the map control entry is changed.

8 - If this is a new map, a map control entry is made in the map production control file.

The map production control file contains sufficient information to produce a map. Among the information in the file is the name of the applications program, the date of creation, date of last use, the TIGER File partitions to be accessed, the cartographic extracts to be used, the map identification data and the coverage envelope. The latter is a set of latitude/longitude values for the corner points of the map sheet. The envelope is used for map coverage searches and to compute coordinate transformation parameters for use during digitizing.

Cartographic extract files are of two kinds. First is the general feature extract. When doing a series of maps within a large area, the extraction and linking of certain features into polygons and/or strings is done one time and stored as an extract file. This eliminates the need for applications programs performing that function; thereby simplifying programming effort as well as reducing computer run time.

The second kind of extract is specific to a given map. It is essentially a meta-image of the map where text positioning, spatial filling patterns, programmed cartographic enhancements, and so forth, are stored in plot position coordinates. The purpose is to reduce or eliminate recomputation of map image parameters or text positioning conflict resolution when similar features appear on several different map types within the same area. For example, say the road network with names is to appear on two different maps produced months apart but of the same 7 1/2-minute quadrangle area. One map will contain the roads and statistical area boundaries; the other, roads and political areas. By computing the road image once, and using it twice, a major saving occurs.

The benefits derived from the use of the cartographic shell must be viewed in terms of the size of the map production task. The savings of just five seconds of computer time per map when producing 30,000 place maps translates into a saving of over $62,500. Since the total volume of original maps to be produced for all aspects of the 1990 Decennial Census is almost 1,000,000 maps, even a small savings per unit map becomes a major savings in total cost.

SUMMARY

The storage of geographic data in fully topologically encoded data bases instead of in specially designed cartographic data bases has been both a benefit and problem to mapmaking. The freedom to create with relative ease new features from other features is a major benefit to map production. The relatively complex interrelationships, however, between all the data elements has made programming more difficult. The necessity to link the 1-cells of a feature together in order to symbolize it on a map increases computer time over cartographic structures which would be predefined.

The Census Bureau's intersection of a cartographic shell between the TIGER File and the applications program promises to reduce or remove these and many other problems. Due to the demand for geographic data base systems, many of which are in TES format, cartographic operations will have to be treated as a use of the system instead of the purpose of the system.