

CARTOGRAPHIC DATA CAPTURE USING CAD

Michael E. Hodgson
Department of Geography
University of Colorado
Boulder, CO 80309

Ann L. Barrett
4361 Butler Circle
Boulder, CO 80303

Reese W. Plews
Department of Geology and Geography
Hunter College - CUNY
New York, NY 10021

ABSTRACT

The digitizing of cartographic features is a necessary but laborious task to many research analysts undertaking cartographic or GIS studies. The requirements of a good digitizing module for the capture of such cartographic features are discussed in this paper. To meet these requirements, the advantages and disadvantages of using an existing cartographic digitizing module or a CAD package are discussed. As a CAD package cannot meet all of the requirements, the design of a CAD post-processing program for assembling polygons from chains and for automatically relating attributes to objects is presented.

INTRODUCTION

The manual digitizing of cartographic data (i.e., with a puck and tablet) is an important process in many applications of digital cartography and geographic information systems (GIS). The availability of digital cartographic data from federal, state, or private distributors has provided an enormous supply of existing digital data set for such applications. However, the manual capture of cartographic data will continue to be a necessity for all but the most widely used maps and study regions that may be available in digital form. The capture of such data is hampered due to the availability/cost of good capture software, hardware requirements/cost, and the digitizing time required by the analyst. A variety of capture software packages are available; however, most packages are either expensive, cumbersome to use, have limitations on map complexity, or only create output formats for a limited number of cartographic and GIS packages.

This paper describes the requirements for a good geographic data capture package and the abilities of a computer aided design (CAD) package to meet these requirements. The logic and specific commands for using the AutoCAD package for recording geographic data are described. The design of a post-processing program used to build polygons and relate attributes to cartographic objects from an AutoCAD file is presented.

REQUIREMENTS FOR A GOOD GEOGRAPHIC DATA CAPTURE PACKAGE

Before discussing the requirements for a good cartographic data capture package, the fundamental definitions of mapped features as related to this study must be presented. All geographic features on a map may be recorded as either point, line, or areal objects (assuming that a surface may be recorded as a set of points or isolines with an elevation attribute.) In most digital cartographic data structures, point, line, and areal features may simply be defined by their X and Y coordinates for location and one identification attribute for further describing the object. This attribute may be used to describe the general class that the object represents (e.g., 11 for residential areas) or to describe the unique individual object (e.g., a 5-digit FIPS code for a U.S. county). Further, other attribute values may be "linked" to each object using the unique identification attribute and a relational database.

According to the proposed standards of digital cartographic data, the digital representation of entities on a map are *objects* and may be further defined as a variety of specific 0, 1, or 2-dimensional objects (DCDSTF, 1988). The appropriate objects used in this study are: entity and label points, nodes, strings, chains, rings, and polygons. An *entity point* is a set of coordinates defining the location of a point feature (e.g., tower, building). A *label point* is a set of coordinates used for locating text on a map for feature identification and is used in this study to describe the identification attribute. A topological junction or endpoint (e.g., junction of two or more linear features) that may also specify location is a *node*. A sequence of line segments (without nodes or left and right identifiers) that may intersect itself is a *string* and will be used for recording linear features. The boundaries between areas will be recorded as *chains* - a directed sequence of nonintersecting line segments. The areal features on a map are defined as *simple* or *complex polygons*, depending on the absence or presence of inner rings (i.e., islands or enclaves). Each polygon is further described by one or more *rings*, which may be composed of a sequence of chains that represents a closed boundary around the area defined.

To record these mapped features, a good digitizing package must allow string-node or chain-node digitizing capabilities for polygon capture (formerly referred to as arc-node) and the automatic "relating" of attributes for each object. String-node or chain-node digitizing of polygonal objects is faster and eliminates the sliver lines between adjacent polygons. The relation of an identification attribute for an object (such as a FIPS code for a county) should be automatically performed. The package should also have error detection and display capabilities, be easy to edit, and be easy to learn and use. Ideally, the package supports color displays, is inexpensive, allows unlimited map complexity (e.g., number of polygons, points in a string, etc.), and creates output to all mapping and GIS packages.

ALTERNATIVES FOR DATA CAPTURE

Advantages and Disadvantages of Available Digitizing Modules

Many mapping packages and GIS packages have a cartographic data capture module included as part of the package. The data capture module is designed to work in an integrated environment with the other mapping or GIS modules. The data capture module for a software package will produce an output format compatible with the package which it is a part of. Unfortunately, digitizing modules rarely create output formats for the competing industry products; thus, the development of data interchange software is required or even the procurement, training, and use of more than one data capture modules is necessary. Further, many digitizing modules only operate on a small number of hardware configurations. The lack of available data capture software and compatibility

problems has resulted in a number of "home-grown" systems - most of which may be hardware or output format specific. Additionally, many of the available or "home-grown" capture programs do not offer the chain-node form of digitizing polygons.

The remaining part of this paper will discuss the advantages and disadvantages of using a CAD package for data capture, and the development of a post-processing program for relating attributes with objects and building of polygons from chains. The post-processing program described is also "home-grown" and may suffer from some of the deficiencies of other home-grown digitizing modules. However, the program is designed around a widely accepted and very portable CAD package. Further, one of the intentions of the post-processing program is to create the multitude of output formats demanded by the variety of cartographic and GIS packages.

Advantages of Disadvantages of CAD

CAD packages are currently used in cartography as an automated drafting package; thus, many cartographers are already familiar with at least one CAD package. Because of their present use as a drafting package, many academic departments or agencies already have the software. Also, CAD packages are relatively easy to learn. Most CAD packages have a myriad of display and editing commands that we cartographers wish existed in a data capture module. Many CAD packages support almost every available display card/monitor and digitizing tablet. Some CAD packages even have greater coordinate representation (i.e., number of digitis precision for a coordinate) than some of the most sophisticated and expensive cartographic data capture modules. With a CAD package, a map may be digitized in sections or as a whole; thus, resulting in a 'continuous' digital representation. Finally, many of the better CAD packages allow unlimited drawing complexity.

There are several disadvantages of using a CAD package for map digitizing. CAD packages generally do not allow chain-node digitizing of polygons. An identification attribute is not easily assigned to each point, line, or polygon - a process referred to as relating attributes to objects. CAD packages do not create an output format suitable for most mapping or GIS packages. Finally, some of the cheaper CAD systems do not allow true string or chain digitizing (referred to as a polyline in CAD) and have limitations on the complexity of the digitized drawing. Nonetheless, because of the relative availability, moderate expense, display and editing capabilities, and multitude of hardware configurations supported, a CAD package is an attractive platform on which to design a cartographic digitizing module.

AUTOCAD AS A CARTOGRAPHIC DATA CAPTURE PACKAGE

AutoCAD was chosen for this study as the CAD package to use for digitizing cartographic features because of the myriad of hardware configurations supported, the available capture and editing commands, and the stability of the product. The relative success of the company and product insures its support for years to come. Finally, many departments and agencies that require a cartographic data capture package already have a version of AutoCAD for other purposes.

Digitizing Logic

The digitizing logic of using AutoCAD and the developed post-processing package (named OADIG for Object and Attribute DIGitization) is shown in Figure 1. All objects and attributes on a map are digitized in AutoCAD and an output file created (a .DXF interchange file). The postprocessing package will read the AutoCAD output file and attempt to relate attributes to objects and assemble chains into polygons, with or without islands. If digitizing errors

are encountered, the OADIG program creates an AutoCAD file of error locations/types used as a graphic overlay onto the original digitized map. The geographic data is edited with AutoCAD and a new output file created for input to OADIG. The cycle of editing and building polygons/relating attributes continues until all errors are removed. When all errors are eliminated the OADIG program is used to create a suitable file in the format of many common GIS or cartographic packages.

*Set up layers to be used (e.g., POINT, LINE, AREA).
 Digitize all objects.
 Create .DXF output file.
 With OADIG, attempt to build polygons and link attributes.*

*Until OADIG indicates no errors,
 With AutoCAD,
 Input .DXF file of error layers onto original digitized map.
 Noting the types and locations of errors, edit objects in map.
 Erase objects in error layers.
 Create .DXF output file.
 With OADIG, attempt to build polygons and link attributes.*

With OADIG, select appropriate output format of cartographic objects.

Figure 1. The digitizing logic of using AutoCAD and the developed post-processing package (named OADIG for Object and Attribute DIGitization) presented in pseudocode.

The location of each point, line, and areal feature on a map is recorded using either the POINT or PLINE AutoCAD command. The POINT command is used to record entity points. The PLINE command is used to record strings for linear objects and chains for polygonal objects. The identification attribute of each object is recorded with the TEXT command as a label point. The point, linear, and areal features are recorded in three separate layers - appropriately name POINT, LINE, and AREA layers, respectively (Figure 2). The DXFIN and DXFOUT commands are used to create files for input to OADIG and to read OADIG created files into AutoCAD. For editing, the additional AutoCAD ERASE and PEDIT commands are used.

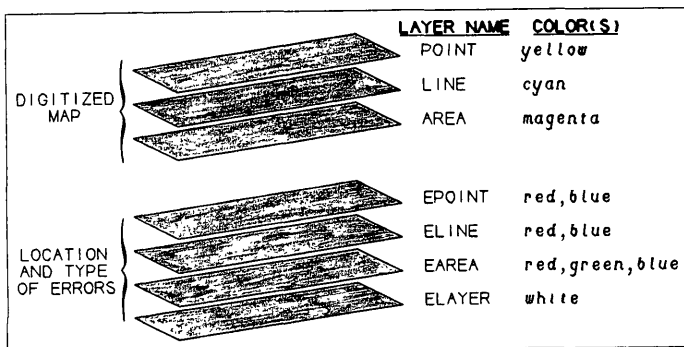


Figure 2. The appropriate layers for digitizing cartographic objects and label points and the error layers created by OADIG containing the location and type of errors encountered.

For an experienced AutoCAD user the commands identified above (with the exception of DXFOUT and DXFIN) should be familiar. A few simple digitizing rules using the described AutoCAD commands must be followed (Table 1).

Table 1
Criteria for Digitizing With AutoCAD and OADIG

Layer Recording:

- Objects and their attributes must be recorded on the appropriate layer: POINT, LINE, AREA.

Location Recording:

- Entity points must be digitized using the POINT command.
- Strings & chains must be digitized using the PLINE command.
- Polygons must be composed of one or more chains that together form a "ring" (i.e., they must not intersect and they must produce closure).
- The endpoints (i.e., nodes) of chains must perfectly join.

Attribute Recording:

- Label points of points, lines, and areas must be digitized using the TEXT command.
 - A label point of a point or line object must be closer to the point or line desired than any other point or line.
 - The label point of a complex area must be inside the outer ring but outside any inner ring.
 - Each inner ring of a complex polygon must also have a label point.
 - Each object must have one and only one label point.
-

Use of Layers

Layers are used in the described digitizing logic for a number of reasons. First, as all label points for the three types of features use the same command (i.e., TEXT), recording the labels on separate layers allows for differentiation (e.g., between the label point for a line and that for a polygon). Similarly, differentiation between strings and chains (both recorded as polylines) is required.

In general, most cartographic data capture packages do not explicitly use the concept of perfectly registered layers of point, line, and area objects. However, digitizing experience of the authors has demonstrated the usefulness of recording the location of certain combinations of geographic features simultaneously visible on the display screen - such as digitizing the location of a river that is also the boundary between two areas. Perfect matching of location for the river (string) and area (chain) is essential.

Finally, layers also allow for the convenient use of colors and separation of errors from original objects. As a convention, the three types of geographic features and the three types of errors are recorded in unique colors. The layers for points, lines, and areas are displayed in yellow, cyan, and magenta. As the error layer is drawn "on top" of the original digitized map last, the objects in error will be illuminated in the appropriate error colors. Red indicates an object without an attribute, blue indicates an attribute without an object, and green indicates a non-closing polygon. The OADIG program creates an error file with point errors displayed in the EPOINT layer, line

errors in the ELINE layer, etc. (Figure 2). Objects digitized in an incorrect layer are placed in the ELAYER layer. An entire layer of errors may be created and turned on (visible) or off (invisible) as necessary for interpreting the location/type of errors and for appropriate editing. When editing the objects, the appropriate error layers may be "turned off" allowing the original objects to be erased or altered. After the editing process is completed, all error layers are erased in AutoCAD and a new output file is created for another use of the OADIG software.

ALGORITHMS FOR RELATING ATTRIBUTES AND BUILDING POLYGONS

The algorithms for relating identification attributes (label points) to the appropriate objects and for the building of polygons from chains are discussed for the benefit of the user. If the user understands the algorithms, the errors are much easier to interpret and edit. This not only results in a better product, but a less frustrated user.

Relation of Attributes to Objects

The correct attribute (described by a label point) for each object is identified using a distance or polygon inclusion criteria. The correct attribute for a point is the nearest label point. Determination of distance from an entity point to a label point uses the simple Euclidean distance in two-dimensions. The correct attribute of a line is the nearest label point to any line segment in the string. A label point may be identified as within a given distance to a string using a path search around the line segments making up the line. A thorough discussion and implementation of the path search algorithm in the FORTRAN language may be found in Baxter (1976). A critical distance is used as a constraint to aid in relating label points and points or lines (Figures 3 and 4). This critical distance is necessary as either a label point or an object may be incorrectly omitted; thus, creating two different types of errors. The attribute for a polygon is the label point that is located inside the polygon. A complex polygon must have label points for the outer ring and for each inner ring (Figure 5). The point-in-polygon test for determining whether a label point is inside, outside, or on the boundary of a polygon uses the algorithm by Edwards and Coleman (1976).

Building of Chains into Polygons

The goal of the chain-to-polygon procedure is to create a set of polygons (either simple or complex) that are defined as a sequence of X-Y coordinate pairs representing line segments enclosing the area. If the polygon is complex (i.e., has inner rings) then the outer ring will appear first in the list, followed by the coordinate pair sequence representing each inner ring. This definition of a polygon is well suited to the requirements of most cartographic and GIS packages although may not be suitable to those GIS packages requiring DIME format chains. While the presented algorithm does not explicitly retain the left and right identifications of each chain, minor bookkeeping could be added to create a DIME format output.

Several algorithms exist for the building of string or chains into polygons (Cromley, 1984; Burrough, 1986). Cromley (1984) presents a method for digitizing conformant zones (i.e. polygons without islands) that requires the user to enter left and right areal codes as well as node identifiers. This additional user input is time-consuming although important for the data structure created. Cowen, et al. (1986) digitized polygon boundaries as chains and then used the GIRAS Arc-to-Polygon procedure to assemble the chains. The GIRAS interface worked well except for the mainframe requirements and

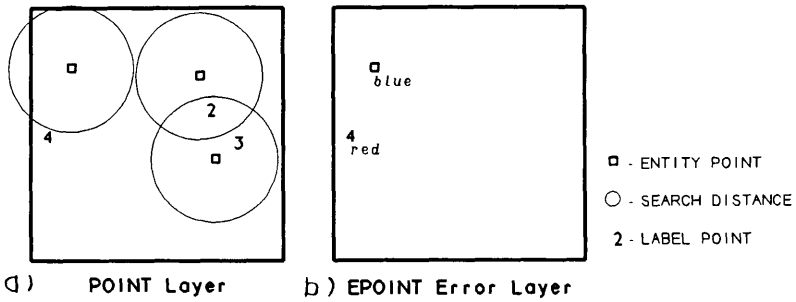


Figure 3. Determination of attributes to be associated with cartographic **point** objects. The original digital map is shown in a) and the resultant error layer produced by OADIG is shown in b). One entity point did not have a related label point and one label point (i.e., number 4) did not have a related entity point.

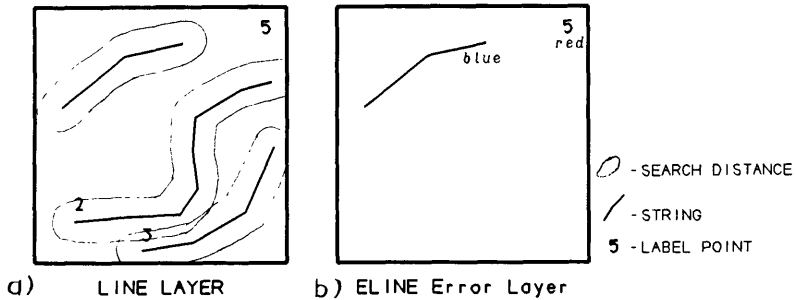


Figure 4. Determination of attributes to be associated with cartographic **linear** objects. The original digital map is shown in a) and the resultant error layer produced by ODIG is shown in b). The label point for a string was missing and the string for label point 5 was not recorded.

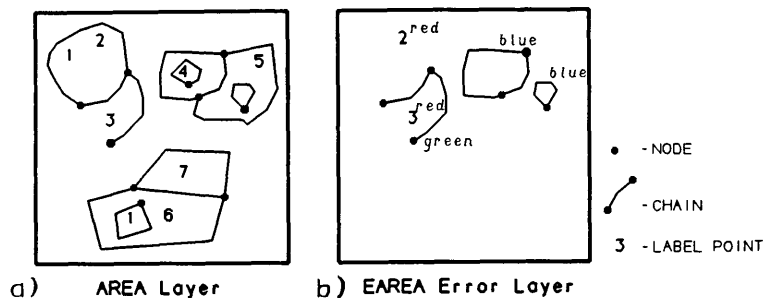


Figure 5. Determination of attributes to be associated with cartographic **areal** objects. The original digital map is shown in a) and the resultant error layer produced by ODIG is shown in b). The three types of errors that may occur when digitizing areal features are illustrated.

lack of interactive graphic editing. A somewhat abbreviated discussion of building polygons from chains and identifying the islands within each polygon is presented by Burrough (1986).

The algorithm developed and presented in this study began from a discussion with Fegeas (1985) on the GIRAS Arc-to-Polygon algorithm procedure for polygon building, although it is not identical. The presented algorithm assumes that the digitized linear boundaries are indeed chains - they do not intersect and begin and end at nodes. Thus, the user must insure non-intersection of a chain with itself or other chains and that the endpoints are nodes. Visual inspection easily solves the intersection constraint and the OSNAP function in AutoCAD may be used to force the chain to 'snap' to the node (i.e., critical point) of another chain. (If the chain does not begin and end at the endpoint of other chains, an error will be graphically identified with the OADIG program.)

The algorithm begins by retrieving one chain from the available set of chains (Figure 6a). This chain becomes the first chain in the current ring (Figure 6b). All other chains are examined to identify the chains that also joins at the ending node of the current ring. The chain with the smallest counterclockwise angle is identified and added to the current ring, reversing its direction if necessary (Figure 6b). The process of identifying the next chain ends when the ending node of the current ring is the same as the beginning node (Figure 6c). Each chain is used twice, once for building clockwise rings and once for counterclockwise rings (as in Figure 6d). After being used twice, the chain is discarded from the available set of chains (Figure 6f).

Only clockwise rings may be a true outer ring of a simple or complex polygon. Counterclockwise rings are only needed for describing the inner rings of a complex polygon. The clockwise constraint for a valid outer ring eliminates the counterclockwise rings built around adjacent areas (Figure 6e). This algorithm for building simple and complex polygons from rings is presented in pseudocode (Figure 7).

SUMMARY

The advantages and disadvantages of using a CAD package for digitizing cartographic data have been discussed in this paper. The logic of using the AutoCAD package for digitizing point, line, and area features as entity points, strings, and chains with the related attributes as label points was presented. A post-processing package for assembling chains into rings and then polygons and the relation of label points to objects was developed in the study and outlined.

ACKNOWLEDGEMENTS

The use of a CAD package for cartographic data capture in chain-node form was the original idea of David Cowen. The work presented here was based on exploratory research originally conducted by David Cowen, Lynn Santure, and the lead author of this paper at the Social and Behavioral Sciences Laboratory, University of South Carolina. The authors would also like to express their gratitude to Marty Garcia-Cotter of AutoDesk for his support in the implementation of the software and to Eva Nesmith of the Computer Sciences Department, University of Colorado for her suggestions.

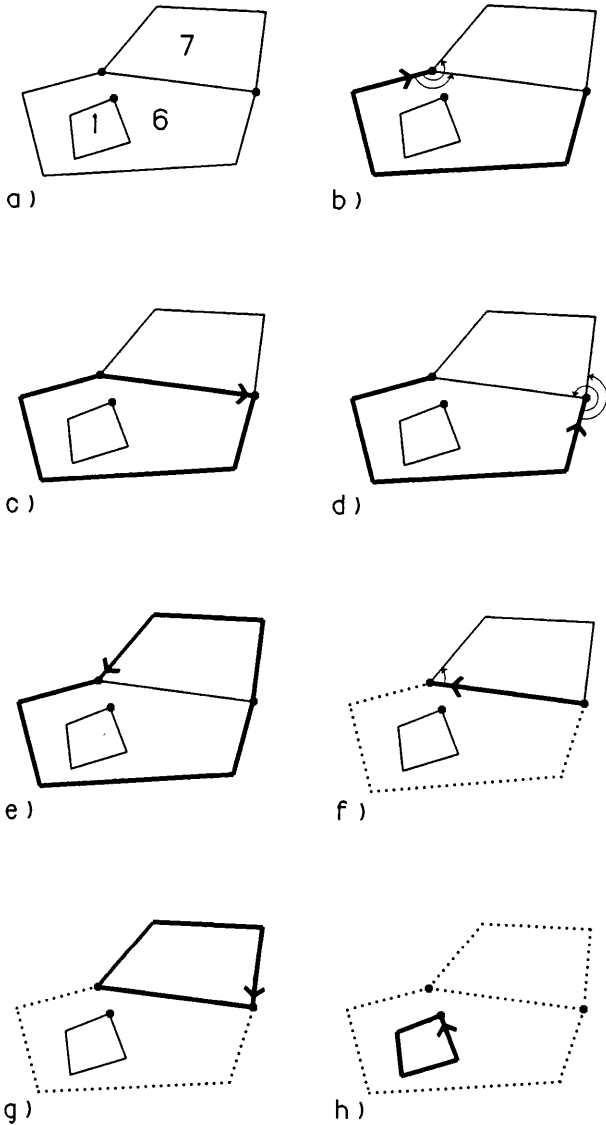


Figure 6. The sequence of steps for building a complete set of rings from chains. The original digitized chains are shown in a). The first chain encountered is used to begin building a ring in b). In c), the chain with the smallest counterclockwise angle is added to the current ring to complete a closed ring. Another chain is used to begin building the next ring (d) and angles are measured to identify the next chain. The second ring is completed in e). In f), the chain used twice has been eliminated and a new ring is begun. The third ring is completed in g) and the final ring is begun and completed in h).

For each ring,
 If ring is clockwise,
 /* This clockwise ring is referred to as the outer ring of a simple
 or complex polygon */
 Polygon = outer ring.

 /* Identify a candidate set of counterclockwise rings that are
 inside outer ring */
 Candidate Set = nil.
 For each ring,
 If ring is counterclockwise & inside outer ring,
 Add to candidate set.

 /* Add each appropriate inner ring in candidate set to polygon */
 For each ring in candidate set,
 If ring is not inside another ring of candidate set,
 Add ring to polygon.

Output completed polygon.

Figure 7. Algorithm in pseudocode for the creation of simple and complex polygons from rings identified previously.

REFERENCES

- Baxter, R.S., 1976. Computer and Statistical Techniques for Planners. (Methuen and Co., Ltd: London), 336 p.
- Burrough, P.A., 1986. Principles of Geographical Information Systems for Land Resources Assessment. (Clarendon Press: Oxford), 193 p.
- Cowen, D.J., M.E. Hodgson, L. Santure, and T. White, 1986. "Adding Topological Structure to PC-Based CAD Databases," Proceedings, Geographic Information Systems Workshop, Atlanta, GA, pp. 198-205.
- Cromley, R.G., 1984. "An Efficient Digitizing System for Encoding Conformant Zone Maps on a Vector Mode Device," Proceedings, International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 181-188.
- Edwards, P.G., and P.R. Coleman, 1976. IUCALC - A FORTRAN Subroutine for Calculating Polygon-Line Intersections, and Polygon-Polygon Intersections, Unions, and Relative Differences. (Oak Ridge National Laboratory: Oak Ridge, TN), ORNL/CSD/TM-12, 30p.
- Digital Cartographic Data Standards Task Force (DCDSTF), 1988. "The Proposed Standard for Digital Cartographic Data," The American Cartographer. 15(1): 1-142.
- Fegeas, R., 1985. Informal discussion of the GIRAS arc-to-polygon building algorithm, March 12.