DATA EDITING

MR. DEAN EDSON: I identified the data editing as a topic unto itself because of the extensive work that is going on in this field and the impact it ultimately will have on the usefulness of digital cartographic data.

To lead the discussion for this important subject, Harvard Holmes has traveled from the Lawrence Berkeley Laboratory, just on the other side of the Bay Bridge, and is the leader of the computergraphics group at this particular laboratory. Harvard's group has been involved in thematic mapping efforts for a number of federal agencies over the last five years, and is one of the largest general purpose computer centers in the federal government. They have specialized in the use of very large data bases such as census and so forth, which are on-line at the computer center at Cal Berkeley. Without further ado, then, I now introduce Harvard Holmes who will lead the discussion concerning data editing.

MR. HARVARD HOLMES: I will start with a few words of what data editing is all about. It includes the original data capture, with perhaps the exception of some mass digitization efforts. It includes the correction of mistakes and perhaps more importantly it includes alterations and additions to the original data base. I would like to just describe the areas of work that are going on at Lawrence Berkeley Laboratory and then summarize what we have learned from these projects. The largest project to date has been the Urban Atlas project in cooperation with the Department of Labor and the Census Bureau. This project required digitizing about 35,000 polygons. That is where we got our experience.

The editing that we did used a refresh terminal. In our case, it was connected to a large general purpose time sharing system. That leads to problems, problems in the area of response time, primarily. I guess the two things that we did to get good response time were, first, we worked at night. That always helps. Secondly, we made some deals with the computer operators. Anyway, what did we learn from all of this? We were involved in the Census Bureau project with mass digitization, and we had a very early version of a line following digitizer. We learned a very painful lesson, which was "get it right the first time." It is far easier to lavish a great deal of care and attention on a high quality base map and digitizing operation than it is to go back and fix it up later. The second thing that we think we learned, and I know at least one panel member wil disagree with me, is that BATCH editing is a drag. We feel that interactive editing is the only hope of correcting the

errors and keeping track of what is right and what is wrong. Apropos of that, we discovered that interactive editing can generate very high interaction rates. Over a two-hour period we have measured interaction rates as high as one operator input every six seconds. And on every operator input, the computer system must respond with some response. On every third or fourth input, there is a significant graphic output to go along with that. We think that only a mini-computer can keep up with the need for this kind of responsiveness. Finally, we discovered that subsequent uses of the base file map, especially aggregations, disaggregations, modifications and so forth, demand a fairly sophisticated file structure like the DIME structure or the chain structure, which you will be hearing about. Finally, we discovered that we should use the same file structure throughout. In an early part of the project we were converting a file structure to use an existing graphic editor. That killed us. That was a very expensive, very frustrating period. We discovered that every time we had one minor bug, we were going to spend five minutes of computer time converting the file to the edit format, fixing this one thing, five minutes changing it back, and then retry your program. That is just not going to work. So if you possibly can, you should keep the same file structure throughout.

I would like to introduce our first speaker, Marv White, who is from the Census Bureau. He is in the Statistical Research Division. He has been concerned for a number of years with ARITHMICON, which is a project to exploit the topological aspects of cartographic data bases for editing and verification.

A Geometrical Model for Error Detection and Correction

Marvin S. White, Jr.

INTRODUCTION

Practitioners of automated cartography are well aware of the inevitable intrusion of errors into their maps and of the tenacity that these errors exhibit. Errors seem to have a survival instinct. To combat these intruders and to provide a sound foundation for automated cartography, we turn to the mathematics of maps.

The maps we have been concerned with all represent the surface of the earth and so the appropriate mathematics is the geometry of 2-dimensional surfaces. By studying the geometrical character of 2-dimensional surfaces, we can understand map phenomena from a mathematical point of view. This understanding provides the basis for encoding and decoding maps for automated cartography and subsequently for detecting and correcting errors in the encoding.

The model described below is a topological model which is the basis for the well known DIME map encoding method. The DIME edits are also well known but their mathematical foundations and fundamental nature are not widely understood. These edits are not ad hoc tests, rather they are questions about the fundamental properties of the encoded map.

Mathematical Character of a Map

A map may be regarded as an assembly of elements of dimension 0, 1 and 2. This is a combinatorial view of maps, which is illustrated in Figure 1. The elements are points, called 0-cells, line segments, called 1-cells, and areas, called 2-cells.

The elements of a map are called n-cells after Poincare, who invented the terms. A 0-cell is merely a point; a 1-cell is a line segment stretched and formed to the desired shape but not crossing itself; and a 2-cell is a disk stretched and squeezed to the necessary shape but neither torn nor folded onto itself. Figure 2 illustrates 0-, 1- and 2-cells.

Dual Independent Map Encoding (DIME)

An automated map consists of numerical representations of the 0-, 1-, and 2-cells and their interrelations. The fundamental relations are the incidence relations, viz., which cells touch which other cells. In DIME we code the incidence relations for each 1-cell, i.e., the pair of 0-cells that bound the line and the pair of 2-cells that the 1-cell separates, as shown in Figure 3.

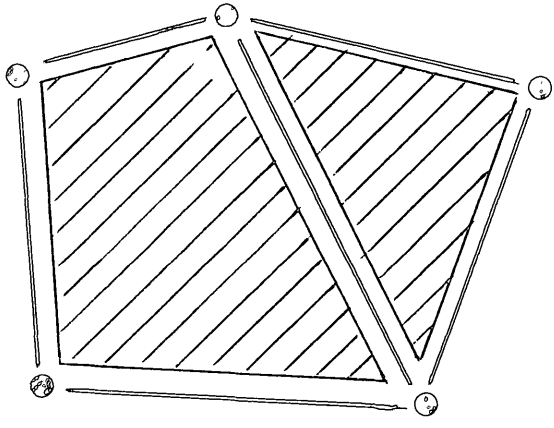
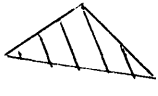


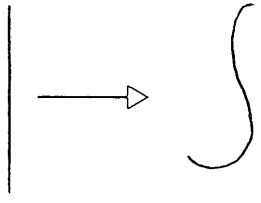
FIGURE 1. A MAP MAY BE REGARDED COMBINATORIALLY AS AN ASSEMBLY OF

POINTS 

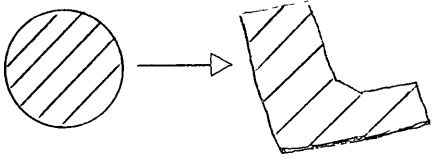
LINES

AREAS 

A 0-CELL IS A POINT



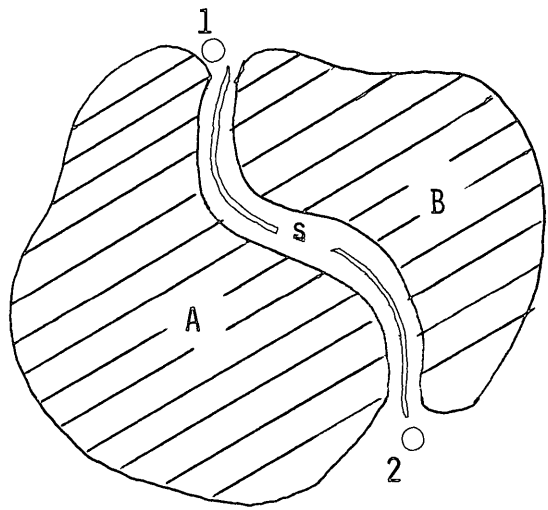
A 1-CELL IS A LINE SEGMENT STRETCHED AND FORMED



A 2-CELL IS A DISK STRETCHED AND FORMED

FIGURE 2.

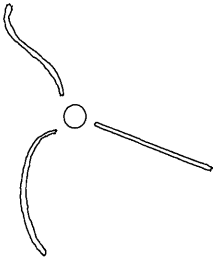
A DIME file contains one record for each 1-cell in the map, which contains the incidence relations for the 1-cell and from that information alone, all topological relations can be computed. For example, the set of 1-cells incident to a particular 0-cell can be assembled by searching the file for all references to that 0-cell (this is called the coboundary of the 0-cell). Similarly, the set of 1-cells bounding a 2-cell can be constructed by searching the file. The neighborhood of a 0-cell is constructed in stages, as shown in Figure 4.



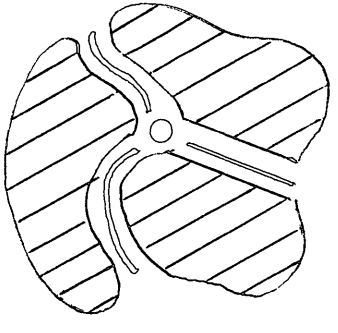
	From	To	Left	Right
s:	2	1	A	B

FIGURE 3. IN DUAL INDEPENDENT MAP ENCODING (DIME) THE INCIDENCE RELATIONS BETWEEN A 1-CELL AND ITS ASSOCIATED 0-CELLS AND 2-CELLS ARE CODED.

A 0-CELL



THE INCIDENT 1-CELLS ARE ADDED



THE 2-CELLS INCIDENT TO THE 1-CELLS ARE ADDED

FIGURE 4. CONSTRUCTING THE OPEN NEIGHBORHOOD OF A 0-CELL. THE 0-CELL, 1-CELLS AND 2-CELLS TAKEN TOGETHER FORM A LARGER 2-CELL.

Error Detection and the Mathematical Model

The mathematical model of a map provides a simple but telling question to be asked of files allegedly representing maps: can this file possibly represent a smooth 2-dimensional surface? If not, the allegation is false, because maps are drawn on spheres or planes, which are smooth 2-dimensional surfaces. Corbett ("Topological Principals in Cartogtaphy," 1976) has shown that this question can be answered by a series of easy questions about parts of the file.

The questions are:

1. Is every 1-cell incident with exactly two 0-cells or for a loop incident twice with a single 0-cell?
2. Is every 1-cell incident with exactly two 2-cells or for interior segments, incident twice a single 2-cell?
3. Is each 2-cell bounded?
4. Does each 0-cell have a neighborhood equivalent to a disk?

If the answer is 'yes' in every case, i.e., for every 1-cell for questions 1 and 2 and for every 2-cell for question 3 and for every 0-cell for question 4, then the file can be interpreted without contradiction to be a smooth 2-dimensional surface. Otherwise the file must represent some torn, folded or higher dimensional or higher genus surface, if it is to be interpreted as a geometrical object at all. Figure 5 illustrates affirmative and negative answers to each of the questions.

For a DIME file, questions 1 and 2 are automatically answered affirmatively. Every DIME record represents a 1-cell and gives the two bounding 0-cells and the two cobounding 2-cells. Questions 3 and 4 are answered via the DIME block and vertex edits. To determine whether a particular 2-cell is bounded, all the incident 1-cells must be assembled and chained together on their bounding 0-cells. If they form a single closed chain, the 2-cell is bounded and the condition for a smooth 2-dimensional surface is satisfied. This bounding test is illustrated in Figure 6 (this figure and several of the following figures are taken from the file correction of the Washington, D.C. GBF/DIME file using ARITHMICON, a

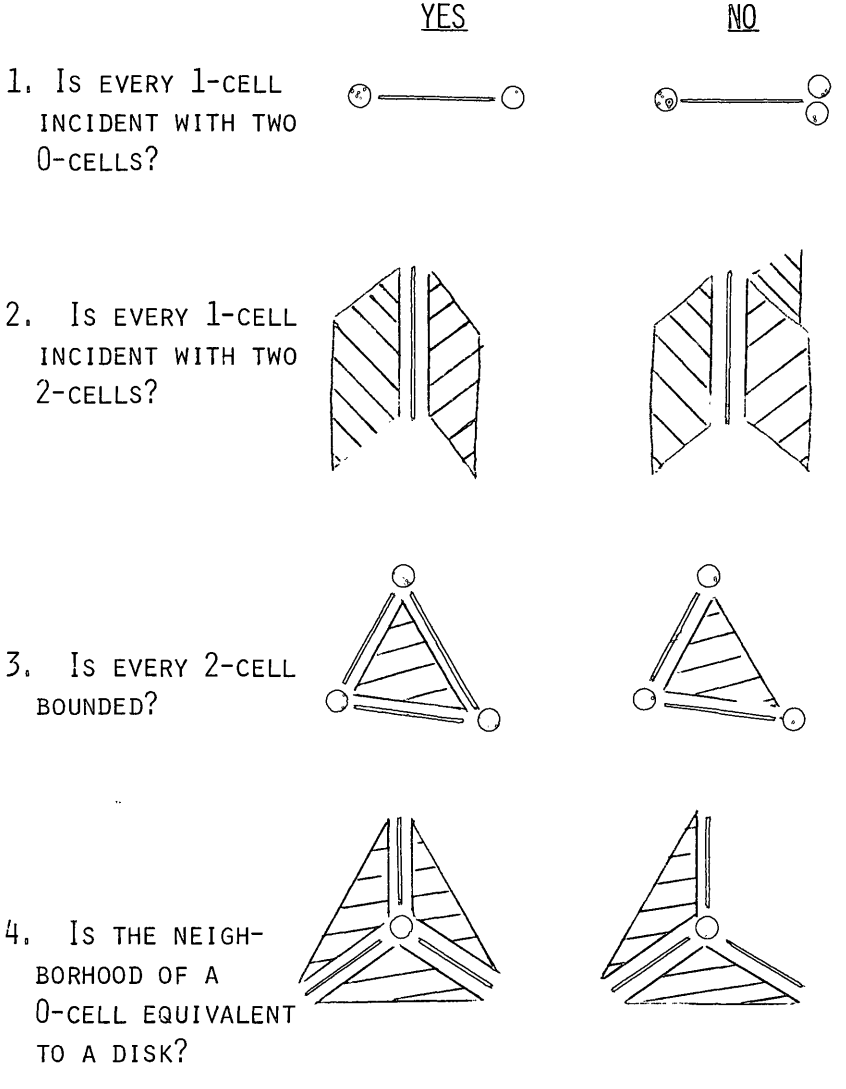


Figure 5. For a smooth 2-dimensional surface, the answer to questions 1 - 4 is always 'yes'. If any answer is 'no' the surface is not smooth.

research system developed at the Census Bureau).

Question 4 is the dual of question 3. It is answered in exactly the same way but with 0-cells and 2-cells interchanged. All of the 1-cells incident to a particular 0-cell must be assembled and chained together on their cobounding 2-cells. If they form a single closed chain, then the neighborhood of the 0-cell is equivalent to a disk. The vertex edit is illustrated in Figure 7.

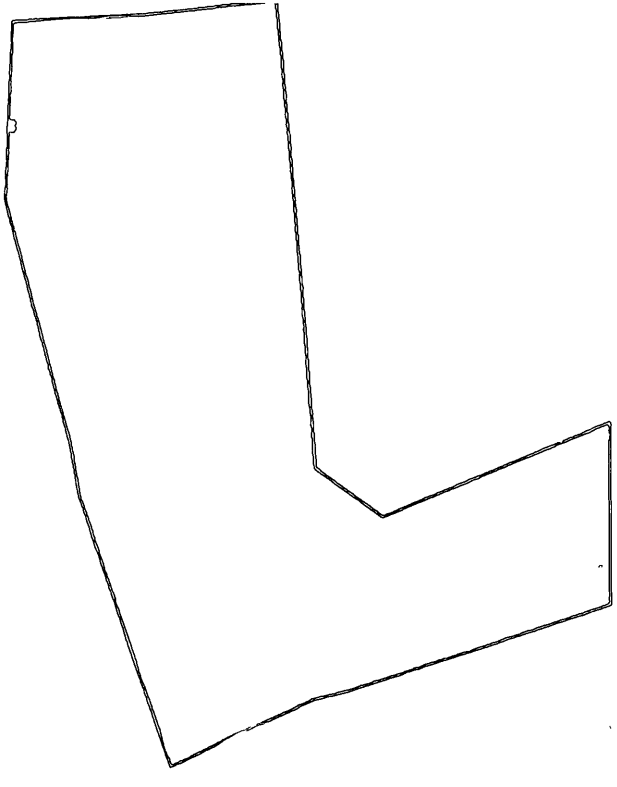


FIGURE 6. THE BOUNDARY OF A 2-CELL IS A CLOSED CHAIN OF 1-CELLS. THIS 2-CELL IS THE NEIGHBORHOOD OF A PARTICULAR 0-CELL.

Duality

The symmetry between 0-cells and 2-cells mentioned above is worth further study. The symmetry between questions 3 and 4 is clearer in a reformulation of the questions: 3' Is each 2-cell bounded by a chain of 1-cells; and 4' Is each 0-cell bounded by a chain of dual 1-cells (a dual 1-cell intersects with the primal 1-cell but connects the two 2-cells that the primal 1-cell separates --- the dashed line in Figure 7 is composed of dual 1-cells).

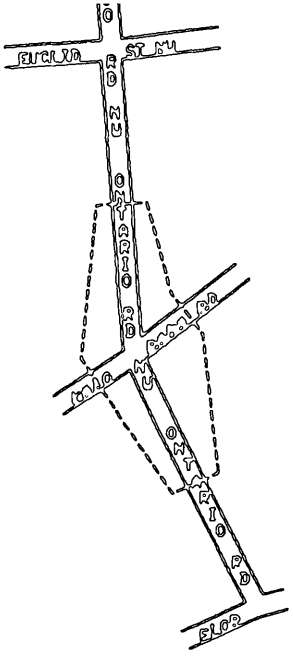


FIGURE 7. THE 1-CELLS INCIDENT TO A 0-CELL MUST CHAIN IN A LOOP, AS INDICATED BY THE DASHED LINE. THIS IS EQUIVALENT TO THE NEIGHBORHOOD OF THE 0-CELL BEING A 2-CELL.

Questions 1 and 2 exhibit the same symmetry --- the questions themselves are interchanged by merely interchanging 0-cells and 2-cells.

This symmetry is called duality and appears in graph theory as well as in topology. The 0-cells and 1-cells of our map form a graph, which is a set of points and a set of edges, in which each edge is terminated by points in the set. The circuits in a graph may be regarded as boundaries of regions and the dual graph is formed by interchanging the roles of the regions and points. It is the dual graph that is the subject of the four color theorem, which states that every map may be colored with no more than four colors so that no two adjacent regions have the same color.

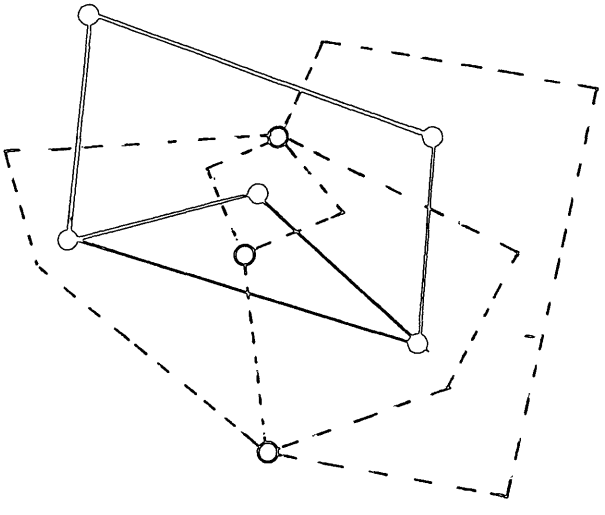


FIGURE 8. A GRAPH (○═══○) AND ITS DUAL (○- - -○):
EACH 0-CELL IN THE GRAPH IS A 2-CELL IN THE DUAL;
EACH 1-CELL IN THE GRAPH HAS A CORRESPONDING 1-CELL IN THE DUAL;
EACH 2-CELL IN THE GRAPH IS A 0-CELL IN THE DUAL.

Figure 8 illustrates a graph and its dual. Each region in the primal graph is represented in the dual by a vertex and each vertex in the primal is contained in a unique region in the dual. This symmetry appears in other places in both graph theory and topology, such as the question of planarity, but a discussion of those topics would be too lengthy for this paper.

Planarity and Orientability

Once we have determined that a file represents a 2-dimensional smooth surface, we may ask further whether it is orientable and if so whether it is a plane (or equivalently a sphere). An orientable surface is one on which left and right may be assigned consistently over the entire surface. It is surprising that there are surfaces on which left and right make no sense, the non-orientable surfaces. However, there are such surfaces and map files frequently represent them, albeit inadvertently. The simplest example is the Moebius strip shown in Figure 9.

When we code a DIME file, we distinguish between left and right --- we code the oriented incidence relations. To answer the question about orientability, we merely ask whether the assignment of left and right is

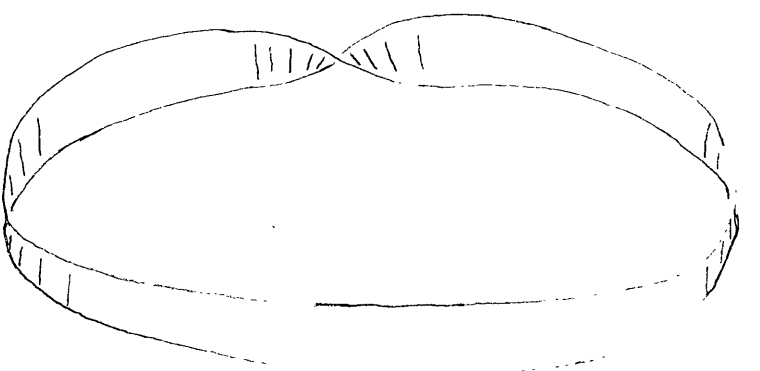


FIGURE 9. A MOEBIUS STRIP IS A SMOOTH 2-DIMENSIONAL SURFACE BUT IT IS NOT ORIENTABLE.

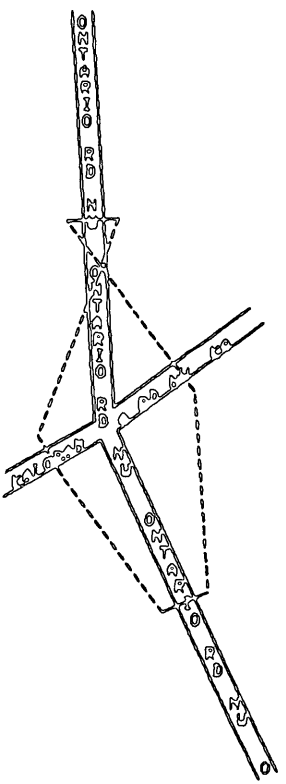


FIGURE 10. THE VERTEX EDIT DETECTS AN ORIENTATION ERROR.

consistent throughout the map, which is easily incorporated into the DIME block and vertex edits. Rather than asking whether there is a chain of 1-cells, we ask whether there is a chain of 1-cells that maintains the encoded orientation. For the block edit this consists in keeping the block being edited on the left while chaining and for the vertex edit keeping the vertex in the "from" position. Figure 10 illustrates the vertex edit in the case of an orientation error.

The question of planarity is answered by a straightforward computation of what is called the Euler characteristic. This tells us the genus of the surface, i.e. how many handles, like handles on a coffee cup, the surface has. For zero handles we have a sphere, which is the very thing we hope for. A handle may be inadvertently created in a DIME file by mislabelling a vertex so that it is identified with another distant vertex. This is equivalent to grasping the map at the mislabelled vertex, pulling and stretching that portion of the map and attaching it to the map at the distant point. Figure 11 illustrates how a mislabelled vertex may be interpreted as a handle on the map.

The Department of Redundancy Department

It has been suggested that the DIME edits are merely redundancy tests and thus we might, by abandoning the tests, actually encode less information but still encode a map. However, these tests are consistency tests that apply to any coding scheme, not only DIME. No matter how we code a map, we can ask whether the code can possibly represent a smooth 2-dimensional surface and further whether it is planar. To illustrate, two different schemes are described below with a discussion of how the consistency tests are applied.

A very popular scheme is the polygon encoding method in which the ordered list of vertices on the boundary of the polygon is coded. This amounts to encoding the incidence relations for 1-cells and 0-cells in pairs along with one of the incidence relations for 1-cells with 2-cells -- we have a DIME segment without the left 2-cell. Each DIME segment is indicated by a pair of vertices in succession. Figure 12 illustrates this encoding of a map. To recover the missing incidence relation, i.e. which 2-cell is on the other side, we can match 1-cells (vertex pairs) occurring in the file

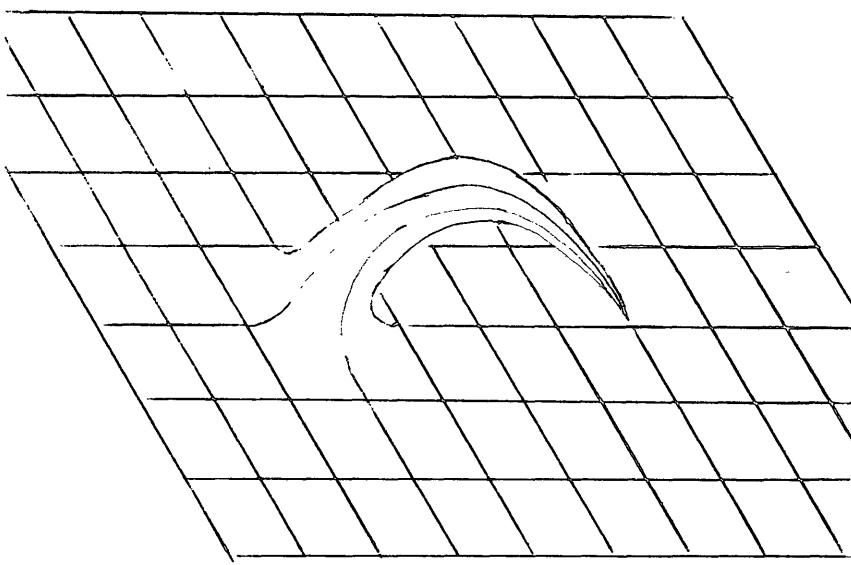
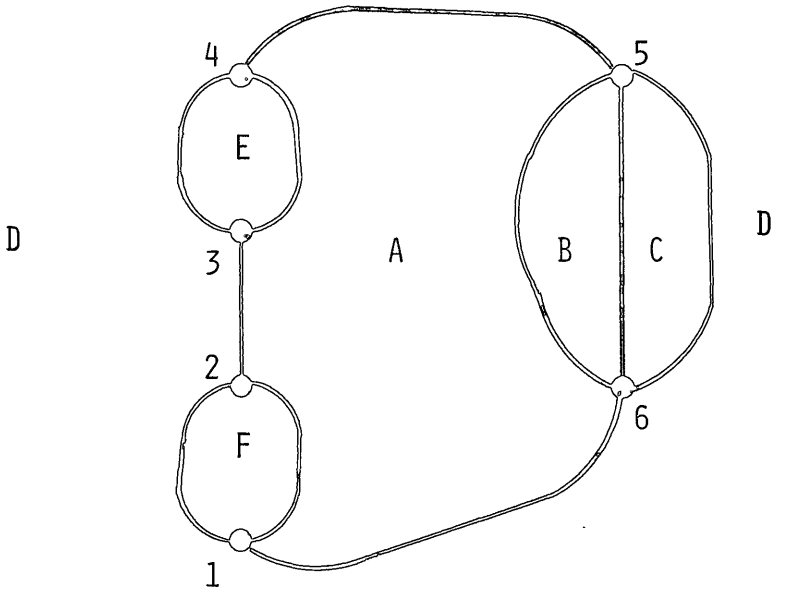


FIGURE 11. A MIS-IDENTIFIED 0-CELL MAY CREATE A HANDLE IN THE MAP. THIS CONDITION IS DETECTED BY COMPUTING THE EULER CHARACTERISTIC.



A: 1 2 3 4 5 6 1
B: 5 6 5
C: 5 6 5
D: 1 6 5 4 3 2 1
E: 1 2 1

POLYGON ENCODING

1: A F D A
2: A D F A
3: A E D A
4: A D E A
5: A B C D A
6: A D C B A

VERTEX ENCODING

FIGURE 12

with the negatively oriented version, which should also be in the file, ultimately constructing a DIME file. We pay for not encoding all the incidence relations in doing the match.

The same questions can be asked of a file constructed by the polygon method: Does it represent a smooth 2-dimensional surface? Is it orientable? What is its genus? The first question again becomes the four: 1. Is every 1-cell incident with exactly two 0-cells? (Yes, the coding forces it). 2. Is every 1-cell incident with exactly two 2-cells? (This question is answered in the match). 3. Is each 2-cell bounded? (Yes, the coding forces it) 4. Does each vertex have a neighborhood equivalent with a disk? (This is answered with the vertex edit).

Note that two of the questions (1 and 3) are automatically answered affirmatively and that two (2 and 4) must be answered by computation, as for DIME encoding. Whatever savings might have been achieved in encoding did not affect the nature of the edit, only the details. We did not sacrifice the edit; satisfying the edit means exactly the same thing, viz., the file represents a smooth 2-dimensional surface.

Even if the coding had been perfect, the matching of 1-cells for the map in Figure 12 would have been ambiguous. It is impossible to determine from the polygon encoding whether the 2-cell B is between A and C or between C and D. This ambiguity will arise whenever a pair of 0-cells are connected by several 1-cells, i.e. when they occur in several ordered lists. This ambiguity arises only because the necessary information was not encoded.

Another interesting encoding method is the vertex method, which is just the dual of the polygon method. The dual graph around each vertex is coded, rather than the bounding graph around each 2-cell. So for every 0-cell, the list of regions incident to the point are named in the order that they would be seen in a counter-clockwise sweep. Each pair of regions implies a 1-cell separating them and the 1-cells must be matched to determine 0-cell adjacencies. Figure 12 also shows this encoding for the example map. Now ambiguities arise when 2-cells are adjacent several times, as for regions A and D. Of course, questions 2 and 4 are

automatically answered by the form of the encoding and questions 1 and 3 must be answered by computation. But the consistency test remains the same.

The vertex encoding allows one to implicitly label 0-cells rather than explicitly labelling them on the source map. The numerals identifying the 0-cells in Figure 12 were unnecessary; they could have been automatically generated and never have appeared on the map. This is a great advantage, since node numbering is an expensive manual process. Polygon encoding allows 2-cells to be named implicitly, but this is not much of an advantage, since we generally wish to maintain common names for regions. In any case, one must weigh the cost of resolving ambiguities and matching 1-cells against the advantages in these encoding methods to decide on the least expensive approach.

Controlling Error Correction and Updating

The DIME edits discover the existence of errors and even localize them very well. The geometrical theory also helps in controlling the correction of errors and the introduction of new information into the file. The remarkable tenacity of errors is in part due to lack of control over the correction process. It is a common practice to stop entering corrections when some small percentage of the file seems to be in error, because the marginal cost of correcting the errors is high and avoiding the intrusion of new errors is very difficult.

File correction and editing in general can be controlled in an interactive system so that the cost of making the last correction is no worse than for the first correction and so that the introduction of new errors is immediately discovered and can be reversed. This is the case in the ARITHMICON system. The question also arises whether file correction is a finite process --- does it ever end? This is a serious matter and should give one pause before undertaking the correction of a map file.

Whether the correction process terminates depends on whether corrections force changes in parts of the file that have already passed the test. In testing orientation on a Moebius strip, one would eventually discover an inconsistency and correct it. But this would lead to further inconsistencies finally forcing changes in portions of the file already tested. The

conclusion is that the surface is not orientable and the so-called correction process would never terminate. Fortunately, few maps are coded on Moebius strips or other non-orientable surfaces but even so, we can detect such a condition and know that 'correction' is impossible.

Correcting for a smooth surface does however finally cease, provided the corrector is not perverse. Controlling the correction process so that it does cease may be accomplished in more than one way. In ARITHMICON, the system we have developed at the Census Bureau, we maintain a check list of 0-cells to be edited and remove them from the list individually as they pass the vertex edit. Whenever a change is made to a 1-cell (all changes in ARITHMICON are made to 1-cells), its bounding 0-cells and all the adjacent 0-cells are pushed back into the check list, so that they are retested. So the check list shrinks and grows as the edit proceeds but the overall effect is that it shrinks. The size of the list gives us a good indication of the remaining work. The file is declared consistent only when the check list is empty.

The check list does finally empty and the corrections cease, provided the person making the corrections always makes changes so that the file corresponds to the source map after the change. Our confidence that the process terminates comes from our confidence that the source map is a smooth 2-dimensional surface.

Summary

A geometrical model provides the foundation for understanding maps and the automation of maps. The topological tests that determine whether a file could possibly represent an orientable smooth 2-dimensional surface are the DIME edits. This topological test is appropriate even for files not encoded directly as DIME files and gives us the same assurances of consistency. Finally, the geometrical model is the basis for controlling the correction and update processes so that they may ultimately terminate.

MR. HOLMES: Our next speaker, Robin Fegeas, is with the U.S. Geological Survey in Reston, Virginia. He will describe his work with automatic digitizing and how to incorporate that data into a chain file structure. Robin?

MR. ROBIN FEGEAS: Thank you, Harvard. To clear up a small point right off the bat. We not only have experience with automatic digitizing, but manual tables as well. The system I wish to describe can be called an operational system. We have been using it for four years to generate data and convert data from graphic form to a data base.

Could I have the first slide. This shows the status map of the work that the U.S. Geological Survey is doing in land use mapping. We are trying to map the entire country by 1982. As I said, we started three, four years ago. This slide was made two years ago and shows status at that time. The red was what was completed then. The blue, what was in production. Today, the blue is pretty much completed except for around here. The West Coast, north of Los Angeles is all that is not completed. By the end of this year we hope to complete approximately 23 states and much of the coastal areas.

This shows the classification scheme used (table 1). I will not go into it too much except to say that it is based on using high-altitude aerial photographics and some Landsat imagery as well as medium-altitude photography to compile the maps at the regional scale of 1:250,000. For those of you interested in the classification scheme, there is a Geological Survey Professional Paper, 964, which may be purchased for 75 cents.

This is an example of one of our land use sheets. This is from the West Palm Beach 1:250,000 quadrangle showing an area around Fort Lauderdale, Florida (see figure 1). This shows about four percent of a total 1:250,000 quad.

Since we are talking about in excess of 400 of these quads, and this is only less than one-twenty-fifth of that, you can see how much data we are talking about. This is a blow-up of the center portion of that last slide. This covers an area roughly equivalent to a 7½-minute quadrangle.

This indicates that data volumes by overlay. I forgot to mention that--along with the land use, we are compiling political boundaries--in other words, county boundaries, census tracks and minor civil divisions, hydrological units, and federal land ownership. As you see, the land use is the predominant data type. Once again, this slide was made two years ago, and you can increase the volume per map by about 50 percent.

1 URBAN OR BUILT-UP LAND	
11	Residential
12	Commercial and Services
13	Industrial
14	Transportation, Communications and Utilities
15	Industrial and Commercial Complexes
16	Mixed Urban or Built-up Land
17	Other Urban or Built-up Land

2 AGRICULTURAL LAND	
21	Cropland and Pasture
22	Orchards, Groves, Vineyards, Nurseries, and Ornamental Horticultural Areas
23	Confined Feeding Operations
24	Other Agricultural Land

3 RANGELAND	
31	Herbaceous Rangeland
32	Shrub and Brush Rangeland
33	Mixed Rangeland

4 FOREST LAND	
41	Deciduous Forest Land
42	Evergreen Forest Land
43	Mixed Forest Land

5 WATER	
51	Streams and Canals
52	Lakes
53	Reservoirs
54	Bays and Estuaries

6 WETLAND	
61	Forested Wetland
62	Nonforested Wetland

7 BARREN LAND	
71	Dry Salt Flats
72	Beaches
73	Sandy Areas Other than Beaches
74	Bare Exposed Rock
75	Strip Mines, Quarries, and Gravel Pits
76	Transitional Areas
77	Mixed Barren Land

8 TUNDRA	
81	Shrub and Brush Tundra
82	Herbaceous Tundra
83	Bare Ground Tundra
84	Wet Tundra
85	Mixed Tundra

9 PERENNIAL SNOW OR ICE	
91	Perennial Snowfields
92	Glaciers

For definitions of Level I and Level II categories see U.S. Geological Survey Professional Paper 964, *A Land Use and Land Cover Classification System for Use With Remote Sensor Data*, 1976, by Anderson, J R., E E. Hardy, J. T. Roach; and R. E. Witmer. Minimum mapping units are: 4 hectares (10 acres) for Level II categories 11-17, 23-24, 51-54, 75, and urban occurrences of 76; and 16 hectares (40 acres) for all other Level II categories.

Table 1.--U.S. Geological Survey Land Use and Land Cover Classification System

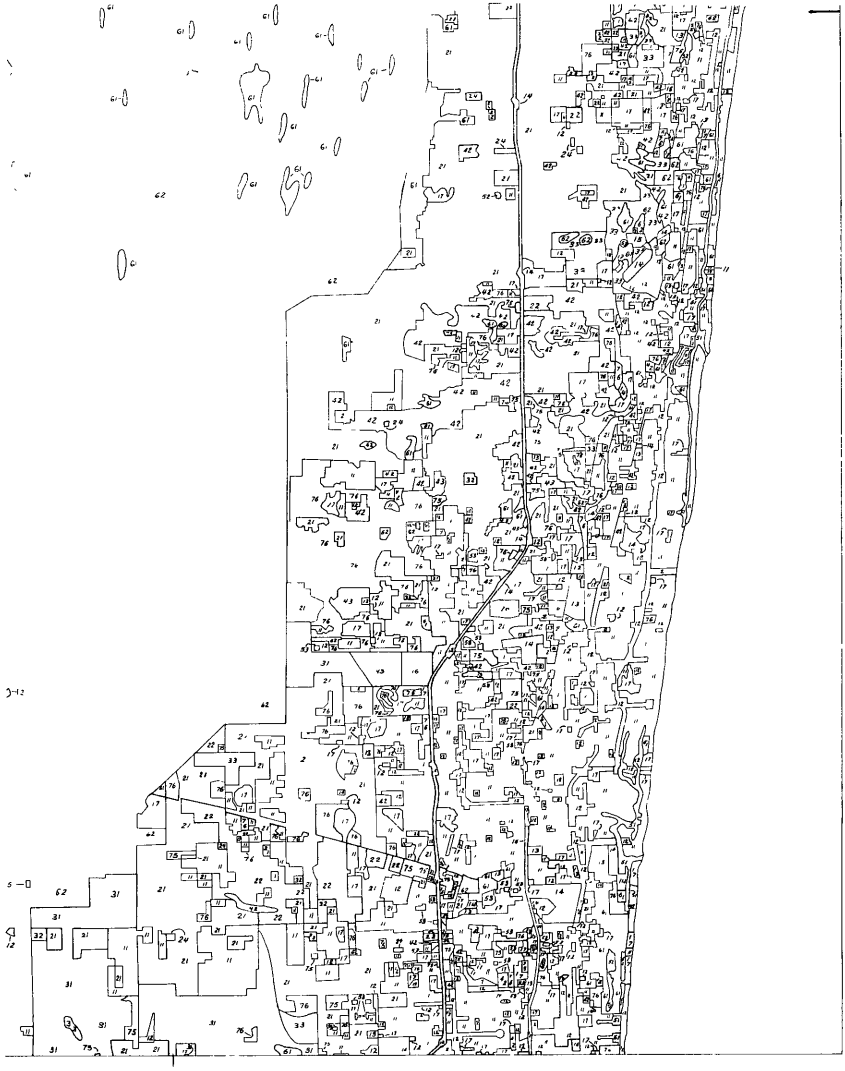


Figure 1.--A Portion of the West Palm Beach 1:250,000 Land use and Land Cover Map

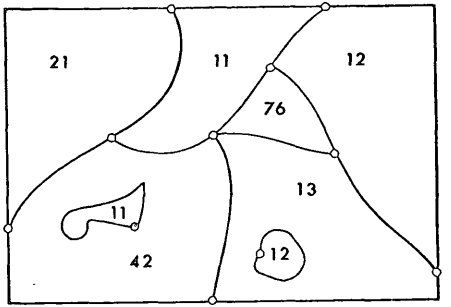
To follow up on some of Marv's discussions, we use a similar structure to DIME, one based on the polygon boundaries, which we call arcs (see figure 2). An arc begins a node, must begin a node, must end in a node, and never pass through a node. A special case is an island, simple island completely enclosed by a larger island. In this case an arbitrary point is chosen to be the beginning and ending point. The final data that we generate from the graphic maps consists of both arc records and polygon records.

This is a depiction of a fixed portion of an arc record (see figure 3). It gives the unique identification number and a pointer to the number of coordinates which make up that arc, that is, the variable portion of the record, plus the indication as to polygon left, polygon right; attribute left, attribute right; a window; length; beginning node and the ending node. For each polygon we also end up with a unique sequence number, a pointer to its variable length portion, which gives the numbers of the arcs which make up the boundary of the polygons (see figure 4). There is a code that points to its descriptive information, the attribute code. Then there is area information, the window, perimeter length, whether it is an island or not, number of islands within it.

This is what our final output is. This input procedure is only part of a larger information scheme, information system that we are working on. We call it GIRAS, geographic information retrieval and analysis system. There is another professional paper just out, which I have some copies of up here. You can get a copy of it after the session, which describes GIRAS see figure 5).

Today I will only be talking about the first four boxes, from source material to the simple data base. In detail, the input procedure consists of the steps you see here (see figure 6). First, the source material is digitized. Then the data is converted to a standard format, and that is what the "read" data box is meant to say. The data is then compacted, and then coordinate transformations are done. For large data sets, data is split up and segmented into manageable spatial sections. There is an automatic edit and error detection performed, and then a manual edit of the arc data. Once the arc data are clean, they are combined with polygon attribute data to form the final files. If necessary, the polygon files, polygon information may also be edited.

As I indicated at the beginning, in the initial digitization stage we have used many separate hardware devices. The methodology we chose has allowed us this flexibility. We have used very simple manual tables, blind tables such as the Bendix Datagrid and Wang table, and also tables which have an interactive system associated with them so you can do some editing while you are digitizing, onto



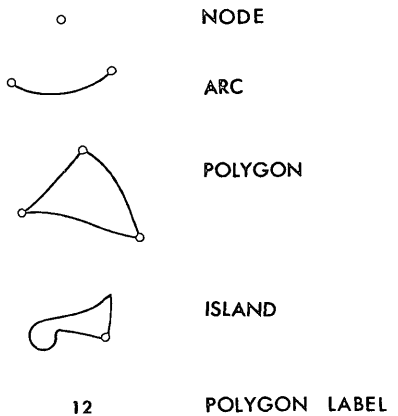


Figure 2.--Elements of a Polygon Map

ARC RECORD

AID	PLC	PL	PR	PAL	PAR	XMNA	YMNA	XMXA	YMXA	ALEN	SN	FN

Name	Description
AID	Arc number.
PLC	Position of last arc coordinate in COORD file.
PL	Polygon number of polygon to left of arc.
PR	Polygon number of polygon to right of arc.
PAL	Attribute of polygon to left of arc.
PAR	Attribute of polygon to right of arc.
XMNA, YMNA	Minimum x,y coordinates in arc.
XMXA, YMXA	Maximum x,y coordinates in arc.
ALEN	Arc length in coordinate units.
SN	Node number at beginning of arc.
FN	Node number at end of arc.

Figure 3

POLYGON RECORD

PID	PLA	CX	CY	ATT	AREA	XMNP	YMNP	XMXP	YMXP	PERL	NIW	NIP

Name	Description
PID	Polygon number.
PLA	Position of last arc number of polygon in FAP file.
CX,CY	Coordinates x,y of an interior point.
ATT	Polygon attribute.
AREA	Area of polygon.
XMNP, YMNP	Minimum x,y coordinates of polygon.
XMXP, YMXP	Maximum x,y coordinates of polygon.
PERL	Perimeter length of polygon.
NIW	Number of islands contained within polygon.
NIP	Number of the polygon containing this polygon, if it is an island.

Figure 4

INFORMATION SYSTEM FLOW

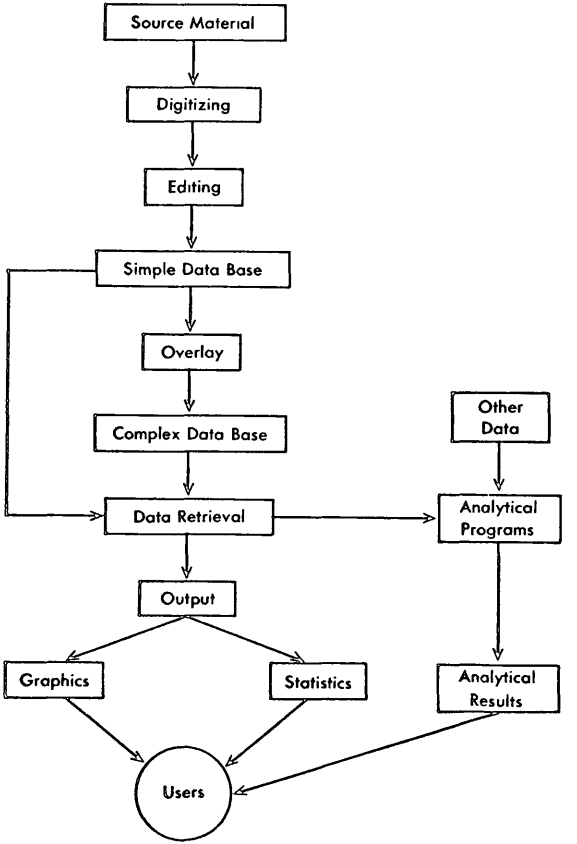


Figure 5

automatic line following, done by I/O Metrics Corporation, now IOM - Towill in California. Broomall Industries is now raster scanning our data and then converting it to vector before sending it to us. Of course, with the many sources that we have, the first step has to be the conversion to standard format. Once the data is delivered to us in Reston, Virginia, all the steps from the conversion to the standard format through final file formation are done on an IBM 370. All the modules used were developed in-house, coded in FORTRAN. They operate in anywhere from 72K bytes to 360K bytes of core memory.

After the data have been converted to standard format they are compacted, the arc data are compacted merely by eliminating points unnecessary to define the lines within a given spatial tolerance. I will not go into the algorithm used. We feel it is a very efficient one. It has allowed us to compact data that has been initially defined by about 200 points per inch down to 30 points per inch, and still retain an accuracy of five mils. That is about an 85 percent reduction. I will not mention too much about the conversion and splitting. That is a very minor operation.

The first step in assuring a clean data set, at least clean arc data, is an automatic edit and error detection routine which is basically an arc end-point matching routine. I am not sure I mentioned it, but the flexibility that we have attained in allowing different hardware to digitize our data is because of the limitations or the limited amount of information we require from the digitizers. We do not require the digitizer operator or device to code the arcs in any way. All they are is just a bunch of spaghetti at this point. A separate file of polygon information is digitized, consisting merely of a polygon label with an arbitrary point within the polygon. At this stage all we are dealing with is unlabeled arc data.

I will not read the slide (see table 2). I think you should be able to read it. It is very elementary editing. What error resolution the program cannot do, it lists errors for the manual editors to then take and perform corrections.

Now, to help us in our work we have acquired a stand-alone mini-computer system developed at the University of Saskatchewan. Many of you have heard of it. It is Cart/8. It is now called Intermap, developed under the direction of Ray Boyle. This shows the digitizing table (see figure 7). Anytime we have to add data into the files, we must, of course, go to the digitizing table.

The rest of the system is shown here, consisting of a small PDP 8 mini-computer, a couple of Tektronix display tubes. The system

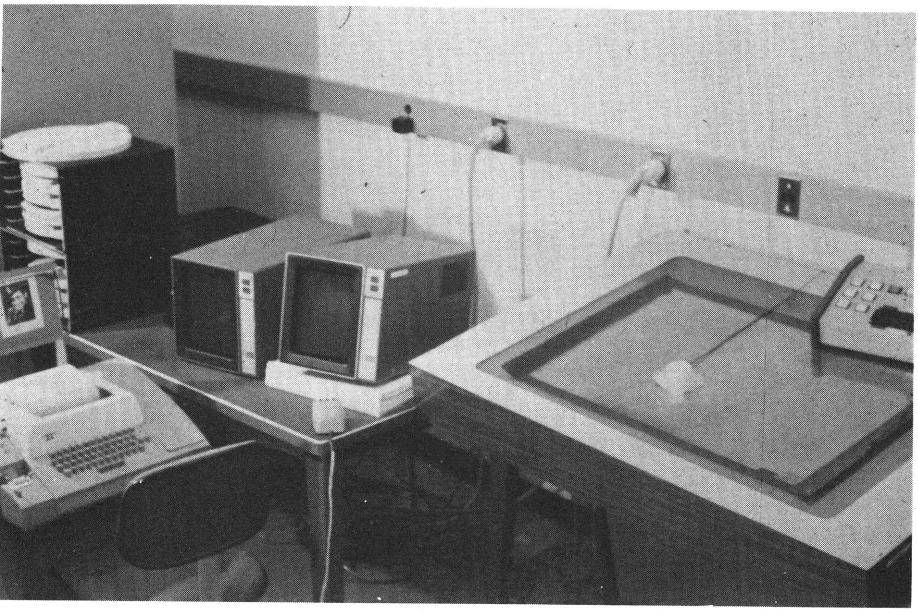
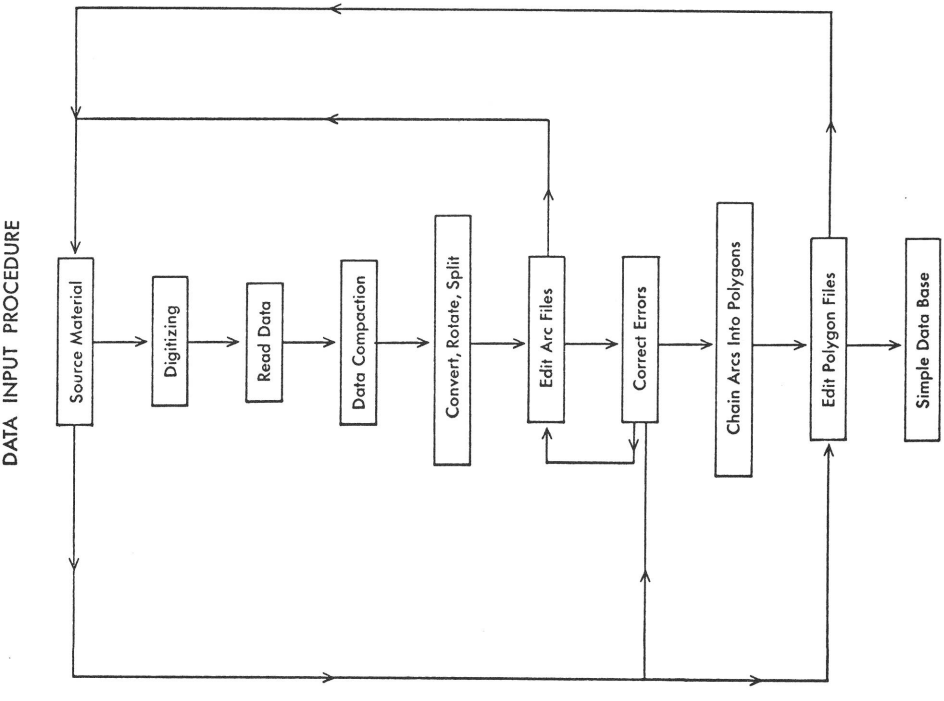


Figure 7


DATA INPUT PROCEDURE
Source Material
Digitizing
Read Data
Data Compaction
Convert, Rotate, Split
Edit Arc Files
Correct Errors
Chain Arcs Into Polygons
Edit Polygon Files
Simple Data Base

AUTOMATIC EDITING AND ERROR DETECTION

An arc end-point matching routine forms the basis of first step toward ensuring a 'clean' or logically correct data files. As part of the routine the following editing is performed:

1. Deletion of one point arcs.
2. Deletion of arcs of length shorter than an allowed tolerance;
3. Deletion of duplicate arcs (duplicate arcs are defined as those whose end-points match and whose bounded area to length ratio is less than an allowed tolerance);
4. Adjustment of arc end-points to meet exactly at nodes; and
5. Deletion of arc points within positional accuracy tolerance of nodes.

Since not all errors can be automatically corrected, the node matching routine detects and lists the following errors:

1. An end-point of an arc matches no other end-point;
2. Only two arcs meet at a node; and
3. One and only one arc end-point matches the node of a one-arc island (this is actually a special case of the second error called "apple in a window").

Table 2.

can also perform a wide range of interactive editing. However, we do not use the system for that because of the volume of data we must process. The system is a one-user station only, and most of the time it is being used for digitizing. We do some in-house digitizing as well as error addition digitizing. Just to give you an example of what can be seen on the screen, this is the land use of a portion of Louisiana shown on the screen (see figure 8), just the arc data again.

This is a blow-up of around Ferriday, Louisiana (see figure 9). This is a portion of a land use sheet from Florida (see figure 10). There are polygon labels also displayed here. You cannot read them very well, but they are there. As I say, we do not use the system much. We are forced to use what we have, and that is an IBM 370, which has limited us to BATCH editing. Therefore, practically all of the editing we do is in a BATCH environment.

We give the error listings and listings of data plus the plots of the data. That is the simple Calcomp drum plotter we use to plot the data. This is a sample plot. We give these to our manual editors to then determine what kinds of corrections are to be made to the data. The errors introduced in data may come from two sources, the digitizing or the original compilation of maps. Digitizing errors introduced, of course, depend greatly on the device used. There are more errors introduced in manual editing than in automatic scanning. Since most of the data that we have processed so far--and I believe I am safe in saying this--has been digitized by I/O Metrics, using their automatic line following scanner, digitizing errors have been reduced considerably. We are left with compilation errors which, becuase of the large amount of data and complex data, are considerable, and we spend a lot of time doing our edit because of compilation errors. I will echo a previous statement made that the more editing you can do before you perform your initial digitization, the better off you are. Right now, once the data is in digital form, it is expensive and time consuming to edit, especially in a BATCH environment. Regardless of whether you edit in a BATCH environment or interactive environment, these are the kinds of options available to our editors. I will not read them (see table 3).

Sometimes we are forced to perform a massive update of our data because of a map being sent out for digitizing before a field check has been made. This slide represents a large amount of updating that was necessary because of field checking. The red represents additions to be made. The blue, deletions. In this case we are forced to go to interactive editing. This (figure 11) shows on the interactive screen the additions that were digitized on the Intermap system for the last area shown, the previous slide.

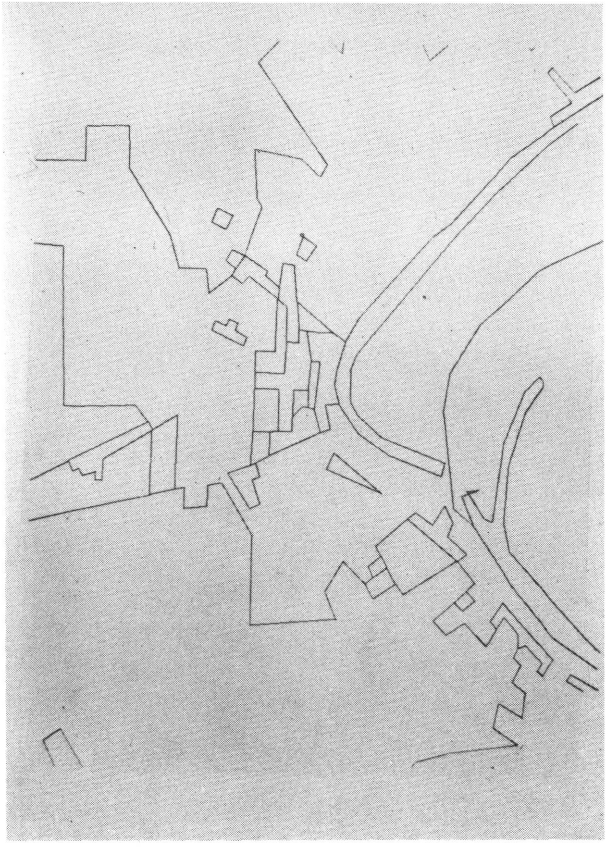


Figure 9

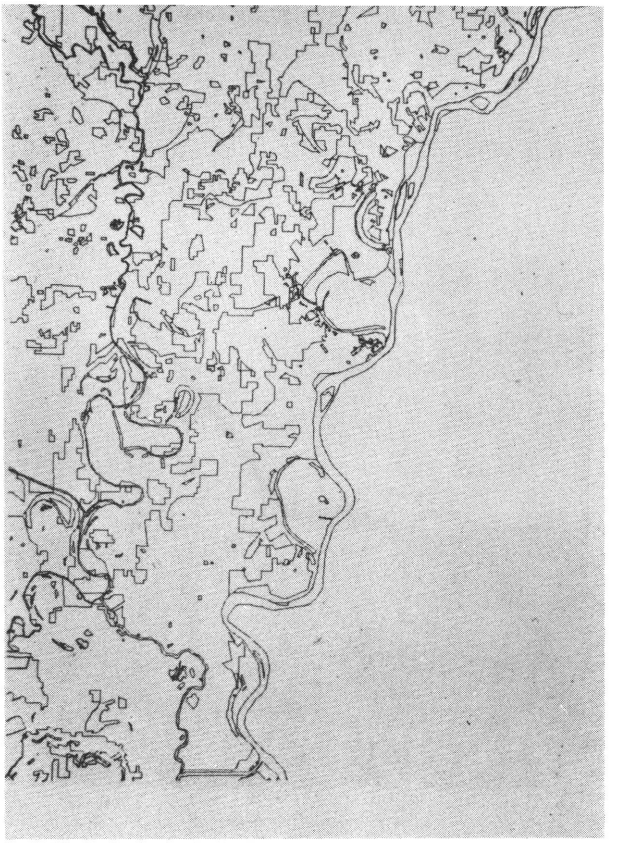


Figure 8

The following commands are available to correct the arc data:

1. Join two arcs into one;
2. Divide one arc into two;
3. Delete an arc;
4. Add an arc;
5. Delete a segment of an arc;
6. Add a segment to an arc; and
7. Translate, rotate and/or stretch/compress an arc.

The arc data are again checked for node errors and the editing process continues until no futher errors are found.

Table 3

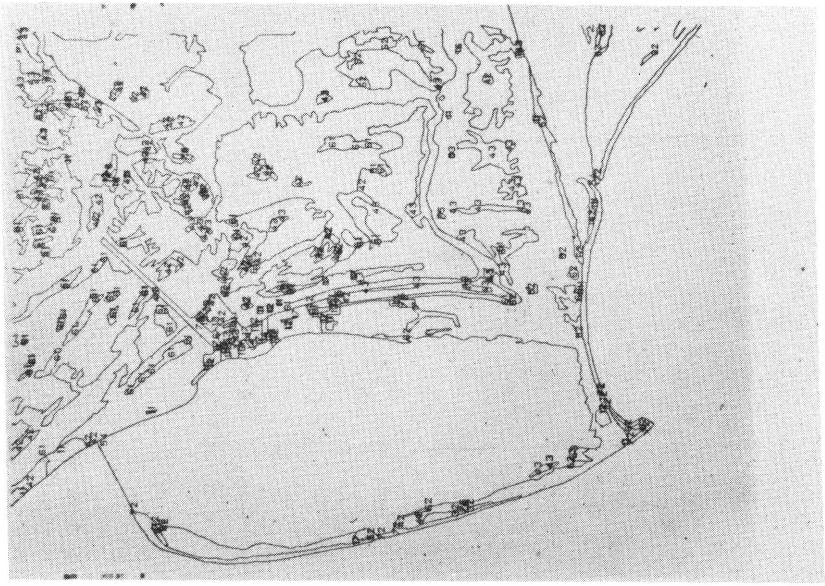


Figure 10

Figure 11

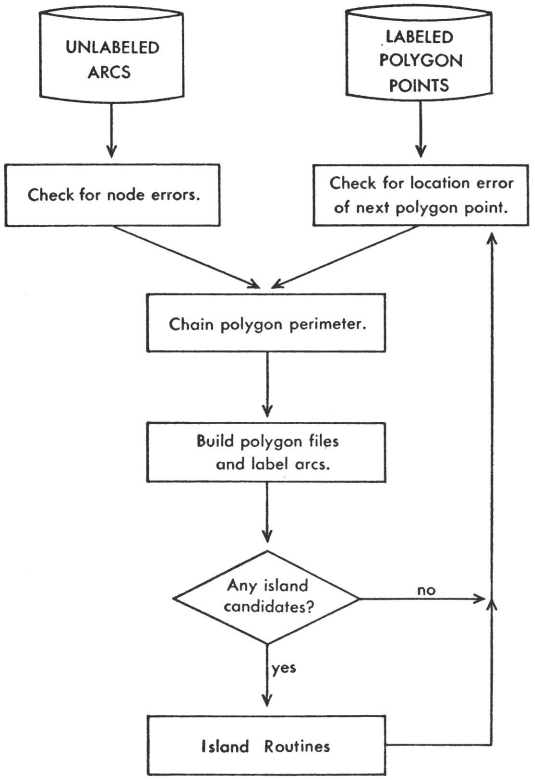


Figure 12

Once the arc data have been cleaned, they can then be merged with the polygon label data to form the final clean files (figure 12). This basically consists of taking each polygon label point and chaining a polygon around it, then labeling the arcs as to right and left. By the way, the arcs have already been labeled as to beginning node and ending node at this point. However, a check is made again to make sure this is correct.

Special routines must handle islands (see figure 13). If any islands are found within a given polygon, this information is also added to that polygon's file, and the arcs are also coded as to left and right. In this method we also can check for identification conflict within one polygon if it is identified by more than one label. Finally, the arc-to-polygon step forms the last topological edit check (see figure 14). Once all the polygons have been chained for those labeled polygon points input, then a check is made to make sure that all arcs are labeled as to right and left, and if that arc is labeled by the same label, same polygon label right and left, the polygon does not necessarily have to be the same, but the labels which are not necessarily unique may be the same. That is an error. Areas are totaled.

As a topological check, this process assures topologically error-free data, but not necessarily attribute error-free data. We must also use procedures to make sure the polygon labels are correct. We can do this by summarizing and seeing if the area summaries look right. We can also plot out selective uses. Here is an area around Little Rock showing urban and water.

You can also then compare the data that we just edited with data that is already in the file. This is a plot of the entire State of Kansas, consisting of 12 separate sheets. So we check the edge information. I think I will end there. I will show a sample of a graphic map that can be produced from the data in publication form.

ISLAND ROUTINES WITHIN ATP

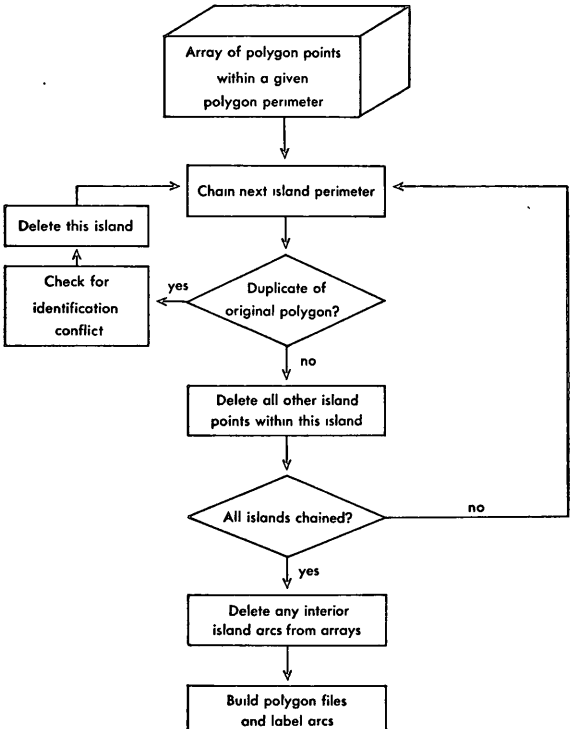


Figure 13

FINAL ATP ERROR CHECKING

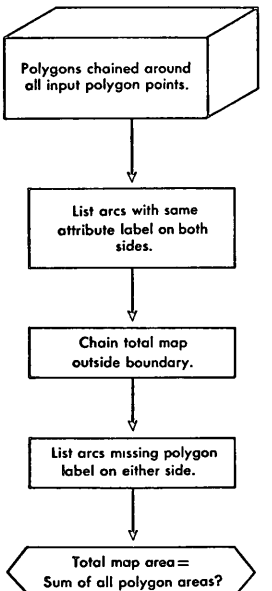


Figure 14

MR. HOLMES: Our next speaker is Marv Grimes. He is Project Manager for the County Mapping Project for the Alameda County Assessor's Office. I think he has developed some unique approaches to urbanized mapping projects. Marve?

MR. MARV GRIMES: Thank you, Harvard. Good afternoon. It is a pleasure to be here. I would like to give you a little history of our automated mapping project, and then get into more of the data characteristics of the system.

In March of 1973, Mr. Hutchinson, the Assessor of the County of Alameda, attended a symposium and was made aware of the possibilities of a computerized mapping system and its application to the mapping of the County of Alameda.

The concept and supporting data for the automated mapping program were presented to Mr. Loren Enoch, County of Alameda Administrator. The Board of Supervisors gave support, and funding was provided subject to annual progress reviews. The actual beginning of the automated mapping project began with my hiring in April of 1975.

I should like to take a minute of your time to present some of the problems in developing this mapping system. May I have the first slide, please.

How does the property on the assessor's map relate to the property as it actually exists on the ground?

How to develop a system to capture for the computer, data from the 1890's to the present for over 300,000 parcels?

How to create an accurate mapping system to ensure the complete inventory of all the land in the County of Alameda?

Finally, how to develop a system that has the flexibility and accuracy to satisfy requirements of today and in the future?

The project was divided into two sections:

Section I - The establishment of monuments to control map data.

The accuracy of our mapping system is based on knowing exactly where the property is located on the face of the earth. The problem in the development of a horizontal control network was to produce accurate state plane coordinates for monuments within the budgeted time and cost parameters.

In order to achieve this objective, an extensive analysis was conducted of technologies to provide adequate ground control monumentation. These methods included satellites, photogrammetry, field survey teams, and inertial guidance systems.

As a result of our research, we selected the auto-surveyor system. This is a picture of the system. This system incorporates inertial measurement techniques similar to those employed in guided missiles, satellites, airplanes and submarines.

The auto-surveyor system's outstanding features include speed, mobility and precision, greatly reducing the time, manpower and expense required in more conventional surveying methods.

The auto-surveyor is completely self-contained:

- no signals are transmitted or received.
- no angles or distances need to be measured.
- neither weather nor temperature retards its operation.
- surveys can be accomplished day or night.

The precision operation is executed by

- the use of gyros that sense the earth's rotation;
- accelerometers that measure the acceleration to as small as 1 part in 10 million;
- a computer that instantly indicates the distance travelled.

To maximize the use of this highly sophisticated system, routes were prepared prior to its arrival. Here is an example of a route.

We prepared the routes to begin and end with a first or second order United States Coast and Geodetic Survey monument. Within each route was a United States Coast and Geodetic Survey monument. This monument allowed us to check the accuracy of the system while it was in operation. Cadastral control monuments were marked by the auto-surveyor, and each of our routes was run in a forward and reverse direction.

Here is a picture of the vehicle shown on one of its routes. There was a total of 1,045 monuments marked by the auto-surveyor. Our average accuracy for all the routes was one foot error in 60,000 feet. The time to accomplish this survey was one calendar month.

Section II - The gathering of map information and the production of finished maps.

This is pretty hard to see, but on the next slide, it will be better.

In Step 1, the map information from the assessor's map, filed maps and recorded documents is input to a micro-processor by clerks. This machine is made up of a screen, a tape cartridge holder, keyboard and memory. The screen allows the clerk to see a picture of the map as it is being input into the system. The clerks do not need prior mapping experience. They are only required to have average typing skills.

Using this methodology, we are able to train clerks in a week and a half, and they become proficient in their input. As you can see in this slide, the input is tutorial. The clerk is asked to enter the book number, page number, block, street address and parcel number. Then metes and bounds information is input into the system.

This is what the input clerks actually see as the map data is input into the system. Now, as you can see, as they put in the metes and bounds information, the actual parcel is being drawn on the screen, so visually the input clerks can actually see what they are doing. This information is collected on a tape cassette.

Once the clerk has input all of the map information, we go to the next step.

At this step, the mapping supervisor views the information on the map. We do this on a block-by-block basis by parcel outline only. The mapping supervisor can visually confirm that the map data have been input correctly by the clerk.

The clerks are able to pick up errors rapidly while they are inputting the map data, and this becomes quite a challenge to them.

When the information has been visually checked, the information is transferred to a mini-computer. Mapping engineers check the map pages and make the necessary adjustments and corrections. Once the mapping supervisor has certified that the information is complete, the computer generates the information to produce the map by a plotter.

This is a picture of our communicator. It is the one on the left. We labeled it "DH" or the Don Hutchinson unit, because it has to communicate. We take the information off the tape cassette and transfer it to a magnetic tape. This is all done off-

line. We are able to capture a number of tape cassettes on one magnetic tape.

From that point, the information is put into our computer. There you see a picture of our computer. We use 80 megabyte disks, and we have three of them on this particular computer. We use an interactive terminal, the 4014 by Tektronix. This is a picture of the data from the mini-computer on the display CRT as actually received from the clerks.

This picture does not look too good as you can see letters upside down, so we want to clean up the map. We may want to delete these two; we may want to move this down here some place; we may want to take and flip some of these over; we may want to put our addresses on these two points right here. Almost instantaneously, as you can see, we do that on the next slide.

Now, we may want to make a picture of this map. We take that picture right off the tube.

This is a view of the completed map. The map is ready to be plotted. This is a picture of our plotter in operation. We are able to give the scaling, the paper size, the position that we want the map on the paper. This is a completed map.

This is just to demonstrate that we can put in a series of scale maps on the same plot output. It is a very simple thing to do by just changing the scaling size.

In summary, I should like to point out a few of the highlights of the system:

- metric to English conversion is instantaneous.
- we have perimeter and area information.
- we can blank, delete, change, move data, and obtain plane coordinates instantaneously for any part of the map.
- translation and rotation of map data.
- the whole map is constructed by metes and bounds information.
- we do not use any digitizer in our system.
- we use basic geometry and mechanical drafting and geometric analysis techniques.

This is a very fast, thumbnail sketch of what we have in our system. We have over 175 different things we can do to a map.

This concludes our presentation of the mapping system for Alameda County. Thank you. (Applause).

MR. HOLMES: Next is Tom Patterson from the Southeastern Wisconsin Regional Planning Commission. Tom is going to describe the minicomputer system which they use. Tom?

MR. TOM PATTERSON: The Southeastern Wisconsin Region is a seven-county planning unit. Its 2,700-square-mile area comprises about five percent of Wisconsin's total land area, but its resident population of 1.8 million people is about 40 percent of the state's population. It also contains about 40 percent of the state's employment and about 40 percent of the state's equalized valuation of property. So, in a very real sense, it is a major portion of the state. The Planning Commission has been in existence since 1961, and is a long-range physical facilities planning agency operating in the areas of land use, transportation, housing, air quality, water quality, waste water collection and treatment systems, and public water distribution systems.

In addition to the specialized engineering and planning data necessary for the individual planning programs, the Commission also acquires and analyzes large quantities of demographic and economic information about the Region. The end result of almost two decades of detailed planning work has been the emergence--not too surprisingly--of a large data management problem. This problem has been attacked in a number of ways, including microfilming, acquisition of larger and more powerful mainframe computers, use of interactive remote entry terminals for program and data set editing and updating, and the use of a digitizing table for conversion of information contained on maps and aerial photographs directly into machine-readable data for analytical and modeling uses. The digitizing hardware itself was acquired and installed during the fall of 1976. The initial purpose for acquisition of this hardware was to convert, directly into machine-readable form, land use information coded directly onto prints of low-flight aerial photography taken in the spring of 1975.

Analysis of the situation prior to system acquisition indicated that there might be a slight cost advantage in the use of a digitizer, even including the hardware and software acquisition costs, as opposed to the manual measurement of land use parcels by polar planometer or dot screen, entering the data by hand on coding sheets, and keypunching. This slight cost advantage was not sufficient in and of itself, however, to tip the balance in favor of acquisition of a digitizing system. There were four additional considerations, however, that made acquisition of such a digitizer quite attractive. These were: ease of update; computation of polygon intersections; the scale independence of the collected data; and the ability to measure areas as part of the data collection procedure. The primary advantages to acquisition of

digitizing capability were perceived to be long-term rather than short-term.

As finally configured, the system has a data input and editing station. This consists of a free cursor digitizing table and cathode ray tube display device. The processing station consists of a mini-computer, a disk storage unit, and a tape drive. Communication with the mainframe computer is by tape, and the digitizing system has software to generate plotting tapes for the Commission's off-line drum plotter. An interactive configuration was obtained in the expectation that data could be collected and edited as a single operation. Experience has shown, however, that this was not a good assumption. The reason, however, cannot be attributed to either hardware or software shortcomings, but turned out to be inadequate quality control operations on the interpreted manuscripts or, in other words, a management shortcoming.

At least in our experience, "data editing" is at least of equal magnitude a management function as it is a hardware or software function. Over a period of time, and utilizing experience gained on two short data collections projects, a data editing system has evolved. It begins before a manuscript is ever mounted on the digitizing table. Hand-coded aerial photograph prints are received in batches of about a dozen at a time. They are reviewed by the lead digitizer operator for logical consistency and completeness. Any questions are referred back to the person who has prepared the print and may result in a revision to the coded manuscript. After passing this initial review, a record form is attached to the document, and it is assigned to an operator for digitizing. All operations performed on the document and its image are logged on the form.

After the document has been digitized, the operator will carefully review its image on the CRT and make any necessary changes before placing the image in storage. After digitization is complete, the lead operator will recall the image from storage, review the image against the document, make any additional revisions as necessary, and will then generate a check plot of the image and a summary report of the coded information. These items are returned to the originating division where the plot is checked against the coded document for a third time. Any desired changes are annotated on the plot, and the material is returned to the digitizing personnel. The image is recalled from storage and revised, plotted, and returned again with the document for another review. This sequence is repeated until the image is approved. When the image is finally approved, it is placed on tape for long-term storage. A final summary report of the coded information is also generated on tape for use on the mainframe computer. These summary tapes represented

the total end product under our previously existing manual inventory system.

Through this new system, in addition to obtaining the summary report of land use categories and areas contained within categories, we now have an image of that information in machine readable form. There are several advantages that have accrued to us because of that. One is that we can replot those images back at requested scales. This has proven to be quite valuable. The data, once collected, are scale independent. Secondly, ease of update as the result of future inventories seems assured. The present inventory of land use that is being coded is the fourth such inventory and a fifth is scheduled to begin in 1980 at the same time as the federal census.

The digitizing operation has given us some additional analytical capabilities. I mentioned before the possibility of computing polygon intersections. We have attempted this on some initial data sets and it has been successful. It does not work well yet, but we can do it, and we can do it on a small mini system as opposed to a mainframe computer. We are quite excited about the information system possibilities that this and some other "problem areas" that we are working on will give us. But there are also some disadvantages, or, if you want to look at them that way, opportunities, for the acquisition of this type of equipment. For a small agency such as ours, staffing poses a severe problem, particularly in that is required a minimum of one or two people who are very well grounded in systems programming, geographic information systems, and state-of-the-art hardware technology.

It can also cause environmental problems or problems in the operating environment which, again, for small agencies is something that may not be considered in the original acquisition. The hardware itself throws off heat like a small furnace, which has to be dissipated, and can result in expensive air conditioning requirements. We have also found that the operating environment can be quite noisy, particularly in small areas, and may require sound deadening and/or protective earplugs for operators.

In summary, then, while we have gained some initial advantages from switching to this type of an operation, we feel that the long-term payoffs, if they finally come about, will be far more substantial than originally anticipated and in the end will probably be the only justification for small agencies to ever embrace technologies of this type. Thank you.

MR. HOLMES: We can open the floor for questions.

MR. TONY VAN CURREN: Tony Van Curren, San Bernardino County. I think this panel has pointed up a very important controversy among small governmental agencies that perhaps the rest of the audience should be explicitly aware of, and that is the relative requirements for accuracy in building a virtual map. The Alameda County project is obviously one in which cadastral locations are considered to be of prime importance. I presume they are paying a price in dollars for that accuracy. The Wisconsin project is one in which the acquisition of data and its manipulability seems to be uppermost in the minds of the users. This divergence of user philosophy is something that the purveyors should keep in mind. I have one question I would like to ask Mr. Grimes, and that is: Have you any estimate what your cost per parcel is going to be by the time you develop your entire data base: And, a somewhat separate question, if you could elaborate somewhat on the uses you hope to put this data base to.

MR. GRIMES: We do not have it down to the price per parcel on putting the information in. We have a bogey figure we are shooting for, which is $1,600,000 for the whole thing, including the equipment. As far as other applications are concerned, it has all the engineering standards. Right now we are going from one inch equals one foot up to one inch equals 2,000 feet, but these are variable. We can go out to double precision of 64 bits. It just depends on how much people want to utilize them.