Lawrence Berkeley National Laboratory
Recent Work

Title
THE MAPEDIT SYSTEM FOR AUTOMATIC MAP DIGITIZATION

Permalink
https://escholarship.org/uc/item/2qk3z63c

Authors
Holmes, H.H.
Austin, D.M.
Benson, W.H.

Publication Date
1974-08-06

U J U U 4 I U / 8 J 8

THE MAPEDIT SYSTEM FOR
AUTOMATIC MAP DIGITIZATION

H. H. Holmes, D. M. Austin and W. H. Benson

August 6, 1974

SEP 12 19

DISCLAIMER

The MAPEDIT System for Automatic Map Digitization

H.H. Holmes, D.M. Austin and W.H. Benson

Lawrence Berkeley Laboratory, University of California

Berkeley, California 94720

ABSTRACT

A system for the automatic digitization of polygon boundaries is described. Digitized map files are created from a driver tape containing identification codes and approximate centroids of polygonal boundaries (e.g., census tracts), and a film image of the map. The digitizer scans on the film plane in an automatic line-following mode, producing the first stage of the map file for the editing system. The MAPEDIT system, which can be used either interactively or in batch mode, reads maps in several standard formats and provides for combining and selecting maps by census (or other) geocodes or by longitude and latitude. This system provides several stages of data compression, analysis, and verification, including algorithms for detecting straight lines, finding corners, fitting insets of maps together and matching boundaries common to a pair of polygons. Auxiliary programs (1) provide a very high resolution (1 part in 25,000) C.R.T. plot of the map, (2) allow a detailed examination and editing of the map and (3) supply missing geocodes using auxiliary tapes such as the Medlist tapes.

The MAPEDIT System for Automatic Map Digitization

H.H. Holmes, D.M. Austin and W.H. Benson

Lawrence Berkeley Laboratory, University of California

Berkeley, California 94720

INTRODUCTION

A computer-based mapping system for the display of statistical data by geographical and political boundary is being developed by the Math and Computing Group of the Lawrence Berkeley Laboratory with the cooperation of the Manpower Administration of the Department of Labor, the Geography Division of the Bureau of the Census and the Department of Housing and Urban Development. The primary purpose of this project is to provide a system for producing high-quality, low-cost maps for use by Federal agencies in graphically representing statistical data.

The project consists of three major components: a system for digitizing base maps; a system for editing and coding of the digitized maps into error-free map files; and a system for correlating statistical data with geographic coordinates for the production of print quality graphic output.

This paper describes the digitizing hardware and the editing system; a companion paper [1] describes the graphic output. The goal of this system is to start with a hand drawn map and to produce a machine readable representation of it. It is not difficult to simply reproduce the original map, using this system. The difficulties arise when one attempts to make a data base instead of a simple reproduction. One must then worry about accuracies and errors in the original maps and how maps are joined together. We must adjust and correct

the maps to eliminate as many of these errors as possible. The steps which
we have found necessary and useful are detailed in this paper.

These machine readable maps have often been done by hand in the past,
[2,3] and recent work has been done based on raster scanning devices [2,3,4].
In our work we chose to use a novel device which has a hardware technique for
following lines, using a point of light which orbits in a very small circle.
As this work was carried out by a subcontractor (i/o Metrics Corporation),
this paper deals largely with the subsequent data processing steps, steps
which we feel would be of use in a raster scanning approach as well.

This project involves digitizing the census tracts for all the Standard
Metropolitan Statistical Areas (SMSAs). This comprises about 30,000 polygons,
with each polygon including up to 2,000 points.

Digitizing Hardware.

The heart of the digitizing hardware is a rapidly spinning prism, which
displaces the light beam into a small circular orbit. This beam, which is
produced by a small gas laser, is then deflected by mirrors to the desired
spot on the film. A photomultiplier tube measures the amount of light
transmitted. A pair of interferometers measure the x and y position of
the mirrors, and a shaft encoder measures the angle of the light beam in its
orbit. The mirrors can be read to an accuracy of 1 μm in the film plane,
and the angular position of the beam is measured to one part in 4,000.
A PDP-15 is in control of the entire apparatus. The line following operation
proceeds as follows: (1) the computer commands the x and y servos to
position the center of the orbit to a particular part of the film, (2) the
photomultiplier tube is enabled and the computer is interrupted when a line has

been seen, (3) the computer reads the angles of the edges of the line, computes the coordinates of the center of the line and a new x,y position. By monitoring the successive widths of a line as it moves along, the computer detects when it is nearing a corner and avoids getting lost inside of very wide lines. When more than one line crossing is detected, the computer always chooses the one on the right, thus performing a clockwise scan of each polygon. A more extensive description of this hardware is available [6].

The advantages of this hardware approach over raster scanning techniques is that the bit map processing is done by hardware rather than software (if we consider the film image as a hardware bit map) and is thus cheaper. We expect that the use of more specialized scanning devices will ultimately favor the use of raster scanning techniques. In our case, we were also motivated to avoid, for a time, the additional program development necessary to use raster scanning devices. The difficulty with this approach is that the line following algorithms are constrained by the hardware. More sophisticated algorithms and techniques which use a scan other than a circle cannot be used. A problem relevant to our application is that of completeness, i.e., how do we ensure that every line has been digitized exactly once (or twice in our case). We solved this problem together with the problem of identification by providing what we call a "driver" tape. This tape contains the identification to be associated with a given polygon and the coordinates of a point inside the polygon. These points were taken from the Medlist files which contain centroids for all the census tracts. The procedure then is to place the light beam inside the polygon and to scan sideways until a line is encountered. The border is then traversed clockwise until it closes upon itself. Each record from the digitizer, then, contains an identifying label and the points associated

with this polygon. The records retain this structure throughout processing,
since it is a convenient form for the mapping programs, the ultimate destination
of the map. This scheme seems much superior to one in which the digitizer
supplies an unlabeled collection of lines, requiring software to associate
each label with the proper set of line segments. Of course this scheme is
not foolproof and an operator is required to confirm that each point is inside
the proper polygon. The position of the beam is compared with the named tract
on a paper reference copy of the map. If it is not correct, then the beam
is repositioned. This new position then can be used to update the Medlist
file. Often, the Medlist points are known to be in error and then the
reference map is suitably marked in advance.

Software

The software for processing these digitized records currently includes
four programs. The first, MAPEDIT, is responsible for converting between
the formats required for other programs, for noise removal and smoothing,
for inset correction, and for boundary matching. The second program, BIGPIC, [5],
provides a very high resolution plot of the map. The third, PICASSO [5],
allows one to edit the map at any magnification up to 60X and the fourth,
FILLDES, supplies missing geocodes from auxiliary tapes.

The coordinate system chosen for internal use is west longitude and
north latitude in degrees. The choice of latitude and longitude is practically
unavoidable considering the areas which the map is to cover; the choice of
degrees over radians is simply good human factors, since it allows easy
comparison with other maps, and computers have no difficulties converting degrees

to radians for the trigonometric functions. For intermediate results an equirectangular projection provides good shapes with a minimum of computation. Final publications, of course, use a conic projection, unless the area is small, in which case, the equirectangular projection is just as good. (The equirectangular projection is obtained by shrinking the horizontal scale of the map according to the cosine of the middle latitude so that equal horizontal and vertical distances on the ground are shown as equal on the map. This compensates for the fact that the meridians become closer together as they approach the poles). A mapping system which operated near the poles might be compelled to choose a different coordinate system or to use a more complicated projection for intermediate viewing.

The file structure which we chose was designed to be simple to understand, yet efficient. Since both the input and the mapping programs processes the data as an identifier and a set of points, we decided to retain that format for internal manipulations. Each polygon thus appears as (1) an array length and an array of geocodes; (2) four words for the x and y coordinate limits; (3) an array length and an array of alphanumeric labels; and (4) the number of points followed by the points themselves. The file is designed to be read by a single FORTRAN read statement. Each of these polygons is a record, and a map is composed of as many records as necessary. All of the coordinate information is stored in the hardware floating point representation. Programs which access the file randomly prepare an index to the file by reading it sequentially once.

Reading the digitizer tapes involves several operations: (1) decoding the tape format supplied by the digitizer, (2) removing noise and squaring corners, and (3) transforming digitizer coordinates to longitude and latitude. The

first step is fairly simple: unpack the various appropriate bit strings from a buffer and perform numeric or character conversion as appropriate.

The procedure for removing noise and straightening up lines is based on a least squares fit to the points of a line. Starting with 2 points the best fit straight line is constructed and the next point is checked to see if it is within epsilon of the straight line; if so, the best fit to the three points is constructed. If not, the average of the next two to four points is taken and checked for closeness; this will include the point if it is a random deviation not supported by the next points, but will exclude the point from the line if a general trend in a new direction is indicated. The least squares fitting algorithm is formulated to compute the fit to $n+1$ points given the fit to n points and the $n+1$ st point. In this way, the computation required to continually update the least squares fit is very small. When a segment is complete, the end points are moved perpendicular to the fitted line, to lie on the line, and the intermediate points are discarded. At the same time, the input routine searches for rounded corners and attempts to make them sharp. This is necessary because the digitizing hardware is confused by the rounded corners and wide intersections which appear on the film image. The software recognizes a corner when it finds two long line segments connected by one or two short segments. These two long segments must have an included angle of 45 to 135 degrees. Upon finding a corner, the software extends the long segments to an intersection and replaces the previous endpoints by the intersection. The short segments are deleted. The corner recognition logic and the line straightening algorithm both use an error criterion which is usually set at execution time to correspond to the digitizer step size. The

usual reduction in the number of points is from 4 to 1 up to 10 to 1.

The transformation from digitizer coordinates to longitude, latitude is performed with the aid of three fiducial points whose longitude, latitude and digitizer coordinates are known. From this information, a matrix transformation is computed which is then applied to all the points in each polygon.

We also plan to include corrections for map projections and for lens effects involved in photographing and scanning the maps. The effects of the map projection and the lens are smaller than the inaccuracies in the maps themselves, at least for the maps currently in use (the Census Bureau's Metropolitan Map series).

Inset correction is the process necessary to match pieces of the map together. Although the transformation to longitude, latitude has positioned the various pieces of the map to the same scale and orientation, it is still necessary to make minor adjustments in the fit of the pieces. There are many causes for these inaccuracies including inaccuracies in the fiducials, cartographic generalizations, and variations in the projections used to make the original maps. For the small areas (individual cities or urban areas) which we have been working with, we can generally disregard the type of projection being used on the input map, provided we know the vertical and horizontal scale factors. These factors are supplied by the fiducial points. In the few cases where the projection cannot be ignored, the effect is to introduce a slight taper in the map from north to south. We compensate for this effect by using a transformation which distorts the map slightly. The general procedure for correcting the position of an inset is to select one or more points on the inset and give their old and new (desired) positions.

From one to four of these points can be used. The effect varies with the number of points. If one point is specified the inset is translated from the old position to the new position. If two points are specified, the inset is translated and the horizontal and vertical scale are changed so that the positions specified at both points are satisfied. If three points are given they specify an arbitrary linear transformation on the inset points. If four points are specified, then a distortion of the inset results which satisfies the positions at all four points. The distortion which we use corresponds to the transformation:

$$x' = r_{11}x + r_{12}y + r_{13}xy + r_{14}$$

$$y' = r_{21}x + r_{22}y + r_{23}xy + r_{24}$$

where x,y are the old coordinates x',y' are the new coordinates, and the r's are the coefficients used in the transformation (given here with subscripts corresponding to the matrix notation used in the program). With 4 pairs of old and new coordinates given, 8 equations result which are solved by matrix inversion to get the 8 coefficients (r_{ij}).

Figures 1A and 1B illustrate the four point transformation.

The border matching portion of the program is used to improve the esthetic quality of the map and to find the adjacencies within the map. The first step in this procedure is to read the entire map and prepare an index to the file. For each polygon, the index includes the x,y limits of the polygon and the disk address where it may be accessed. The first polygon (the "subject" polygon) is then read in for matching. The first point of this polygon is selected and the index is scanned to find those polygons ("matching" polygons) whose coordinate limits overlap the selected point. Each of these polygons is read in from the disk and all of their points are checked to see if any are close to the selected point. If this test
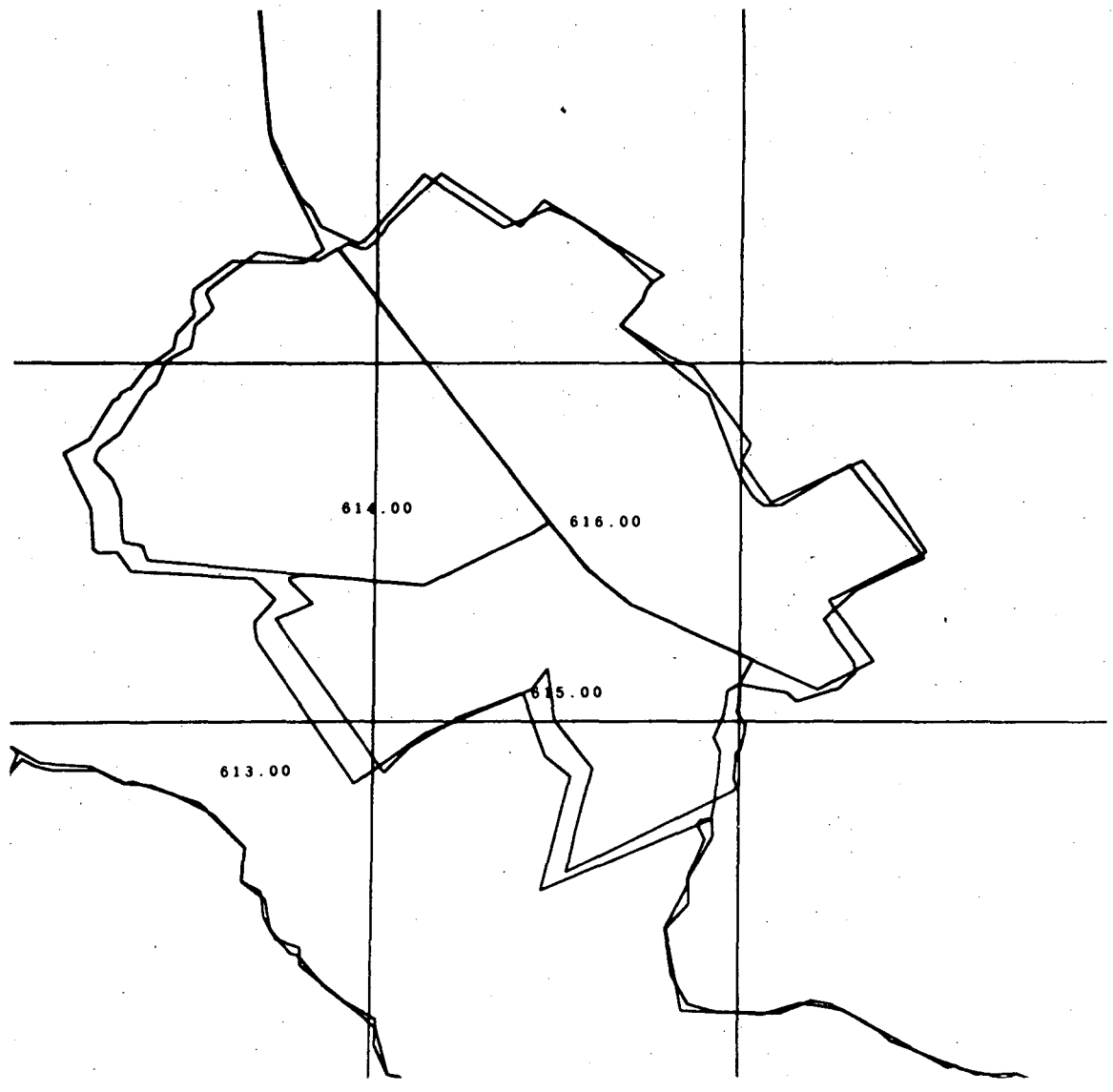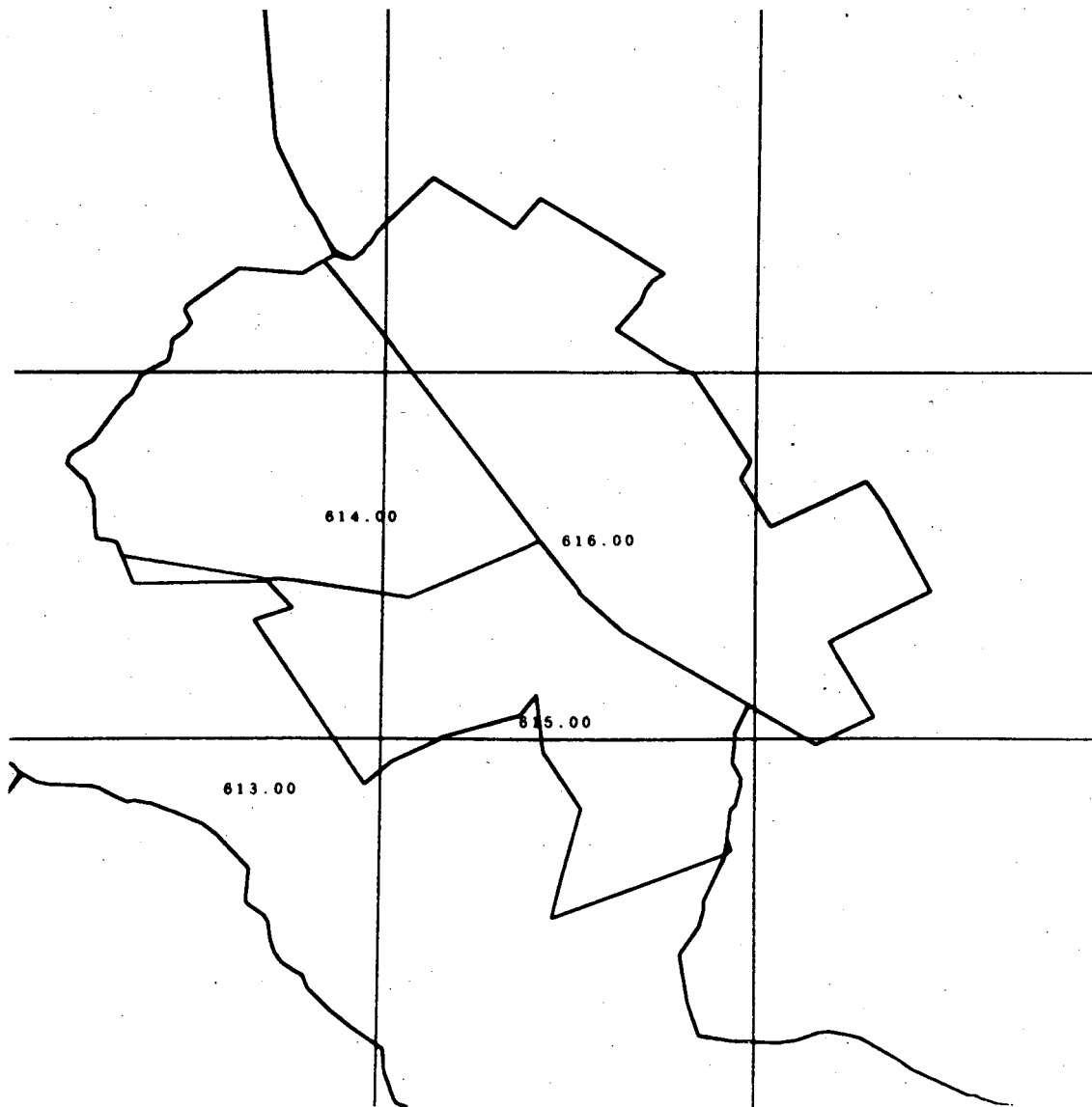
XBL 746-967

Fig. 1A. The original map, showing an inset and the surrounding area.

614.00

616.00

5.00

613.00
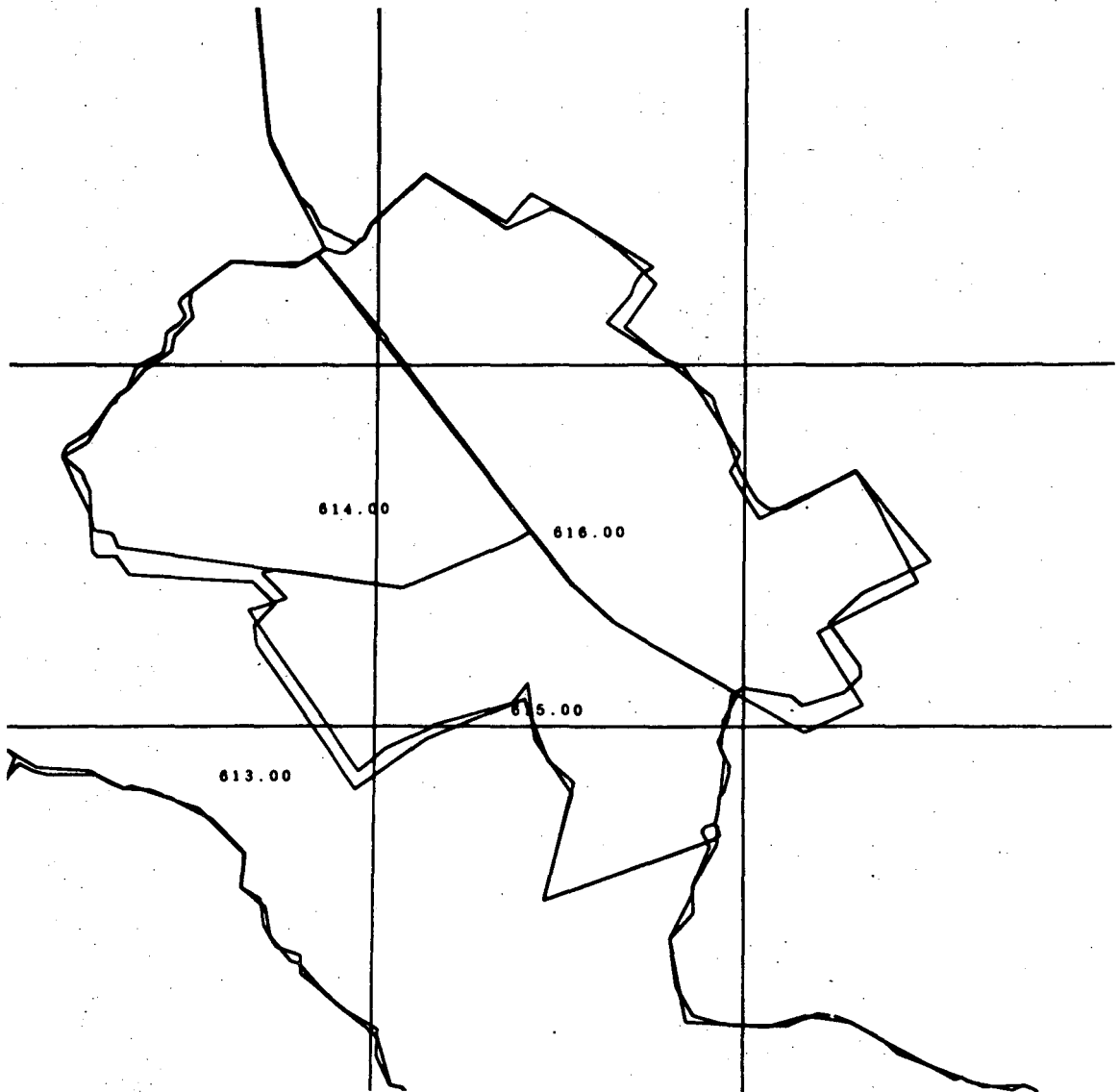
XBL 746-968

Fig. 1B. The inset has been adjusted for a somewhat better fit.

XBL 746-969

Fig. 1C. The border matching algorithm has been applied to the inset.
One line segment failed to match.

fails, then the selected point is compared to the line segments implicit in the points of the matching polygons. Whenever a point or line segment is acceptibly close (within the "miss" distance) then that polygon and the matching point are remembered, and the rest of the search is conducted with the miss distance reduced to the value found. As with the line straightening algorithm, this miss distance can be adjusted as part of the input data. When the search is complete, the best matching polygon is again read in (if not already in main memory) and control passes to the matching section. Beginning with the selected point of the subject polygon and the closest point of the matching polygon, the matching section gets the next point of each polygon (forward for the subject polygon and backward for the matching polygon, since both are drawn clockwise) and decides what kind of match to make: a point to point match, or a point to line match, and which point to which point or line. An interpolating point is then computed, using a weight which is explained below. Pointers to the selected and/or matching points are then advanced. The routine thus works its way around the common border of the two polygons, with the interpolated points put in a separate array. The matching routine will eventually fail, due to the divergence of the common boundary. The interpolated points are saved for use at the end of the next step. The next point of the subject polygon is then selected and the search for a matching polygon and point is repeated. When a matching polygon is found, the matching section first searches backward toward the previously saved interpolation points. Again, in the backward direction, the boundaries will eventually diverge, terminating the matching routine. The two arrays of interpolation points (one forwards and one backwards) are examined for overlap, and if so, the point of closest matching is selected for the transition from

one set of interpolation points to the next. The original points of the subject polygon are then replaced by the interpolated points. When the subject polygon is complete, it is written to another disk file and the index is updated so that subsequent references are to the new version.

Several aspects of the border matching are worth some comment. First, the border matching is performed twice, once with the "left" polygon as subject, and once with the "right" polygon as subject polygon. The algorithm is relatively cheap, so we have not worried about this problem. When a polygon is changed, the new version is used subsequently, which reduces the necessity for an absolutely symmetric matching procedure. Second, the miss distance is keyed to the scale of the map as well as the input parameters. The scale of each polygon is derived from the transformation used to convert from digitizer coordinates to longitude and latitude. The miss distance must be correctly adjusted to achieve proper results. If the miss distance is too large, then non-adjacent boundaries (e.g., rivers) may be collected together; if it is too small, then adjacent boundaries may not be recognized as adjacent. Third, the interpolation weight is also tied to the relative scales of adjacent polygons. If the scales are relatively equal, then the weight is set to 0.5, while if one polygon has a much larger scale than the other, then all the weight will be given to those points. The result in this case is to include the cartographic detail of the large scale boundary in the smaller scale boundary. Fourth, the production version of MAPEDIT uses extended core in place of disk storage to reduce access time. An example of the operation of the border matching routine is given in Figures 1B and 1C.

On-Line Editor

The on-line editing facilities allow a variety of operations to be performed on the map using a CRT display, a lightpen and a teletype. All of the commands are initiated by pointing to a menu with the light pen. The teletype is used for the entry of text only. The basic editing commands allow the drawing of a line or a series of line segments, the erasure of a line, adding a point to a line and positioning it, and moving a point and its connected lines. These commands may be applied simultaneously to several borders, in case the borders of two or more polygons coincide. Figure 2 illustrates the command to move points. The display manipulation commands allow the selection of magnification up to 60X of any area of the map. Having selected a magnification and an area, one may then step the display in any of the four directions by half a screen width at a time. This is ideal for stepping around a boundary at very high magnification. The operator is able to select any polygon or have the next one in sequence selected automatically. In addition, all the polygons around the selected one may be displayed for simultaneous editing. The file management routines keep track of each polygon on disk or in main memory and pages them in and out of main memory as necessary. In addition, these routines provide a conversion from internal to external format for flexible input and output.

The hardware used for editing is a CDC 6600 mainframe and a buffered memory display driver (VISTA 250) connected to a channel. Less expensive hardware configurations have been described in the literature [3,7,8].

High Resolution Plotting Package

The high resolution plotting package is based on our Computer Output
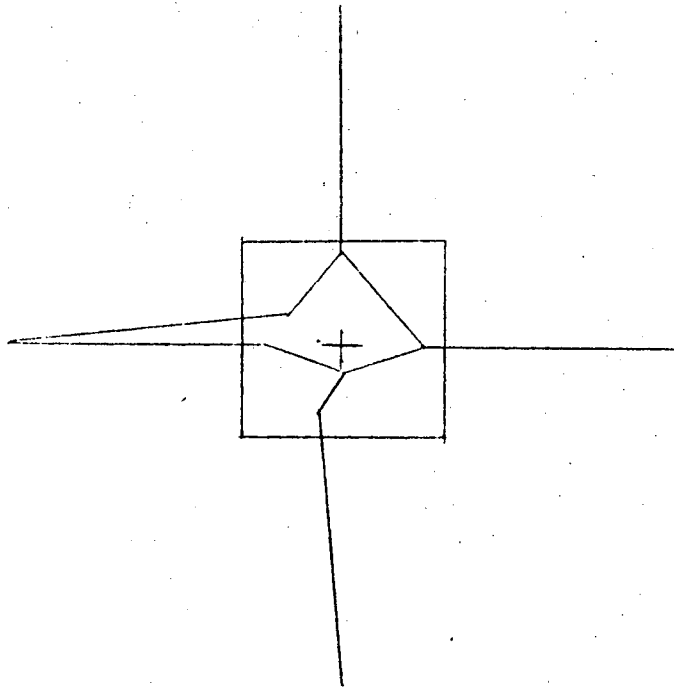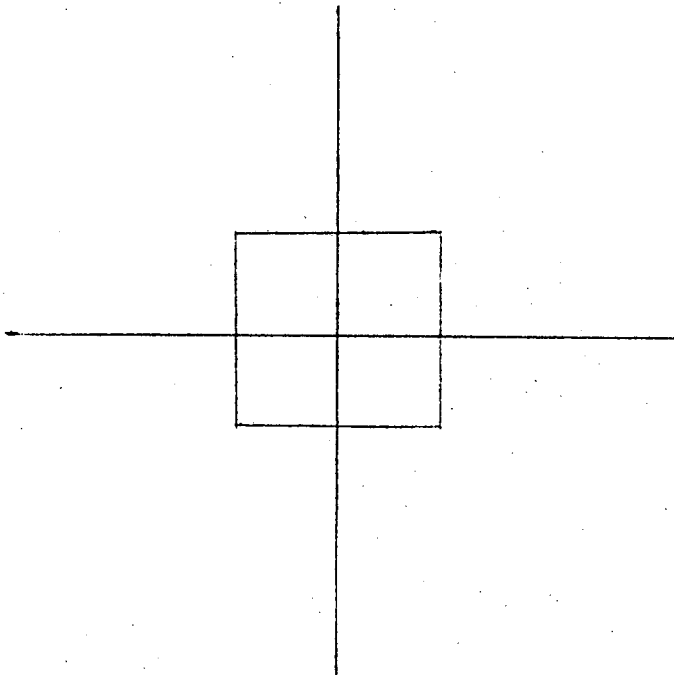
Figure 2A

Figure 2B

Fig. 2(A) A typical corner, shown during the on-line editing phase. The cross shows the last position read by the program and the square outlines the area affected by the current command; (B) The points within the square have been moved to the cross.

to Microfilm (COM) unit which produces 70 images on a 4" by 6" microfiche. Our unit has been adjusted so that these images overlap very slightly. Each image has a resolution of 1 part in 4000 and there are seven rows and 10 columns of images. The plotting package reads the map and prepares 70 lists of polygons, one for each image. Each list contains the polygons which appear partially or wholly in the corresponding image. Each image is then plotted in sequence, retrieving the relevant polygons from mass storage.

Supplementary Geocodes

The addition of supplementary geocodes is performed in a straightforward manner. A program reads the relevant geocode records into main storage from a Medlist tape. Each polygon is read and matched to one of the geocodes in memory. The additional geocodes are extracted from the remainder of the Medlist record. In addition geocodes which appear in one list but not in the other are printed as diagnostics.

Performance

This system has been oriented toward the editing functions rather than the digitizing operations. As the digitizing was done by a subcontractor on an experimental basis, it is difficult to measure the true cost of digitizing, but we would estimate a cost of from 3 to 10 cents per final point. The editing operation is performed at a cost of 80 cents per tract, with an average of 30 points per tract (our computer is cheap, however). Tracts were edited at the rate of 30 to 40 per hour of operator time, with an average of 3 or 4 points being adjusted per tract. This is comparable with the rate achieved by a similar project [9] at the IBM Research Laboratory in San Jose, California.

They report a rate of over 75 segments per hour, using a DIME (Dual Independent Map Encoding) file structure. Since most of our operator time is spent in verification, we expect that new data could be entered at a rate of up to 300 points per hour.

Conclusions

We have found the job to be harder than we anticipated. We were fairly well prepared for misnamed and missing polygons and we were also fairly well prepared for refining the alignment of map pieces. We were poorly prepared, however, for the need of an extensive border matching algorithm in the software. We surprised by the notion of computer generated enlargements of portions of the map. In retrospect, it is natural that given a computer and a map, the user will enlarge the city until his block fills the screen, thus displaying a multitude of minor imperfections, but we were surprised by the resulting need to keep as many details as possible, even those which were smaller than some errors. So far, our algorithms for this are shakey; they occasionally respond with dramatic change to minor variation in parameters. Nevertheless, we feel that border matching algorithms, such as ours, will ultimately allow small scale maps to benefit from the detail available in large scale maps.

We feel that our algorithms for line straightening and map fitting have worked out fairly well. If we were to do the job over again, however, we would require more fiducial points and perform consistency checks among them. A fruitful area for future research is the automatic computation of inset position, possibly from center of mass or similar calculations.

ACKNOWLEDGMENTS

REFERENCES

1. Wood, Peter M., and Austin, Donald M.,

    "CARTE: A Thematic Mapping Program", Computers and Graphics, July, 1974

2. An Integrated System of Digitization of Cartographic Data, [Interim

    Report, July 1972], Graphic System Design and Application Group,

    Electrical Engineering Department, University of Saskatchewan.

3. Tomlinson, R.F., ed.,

    Geographical Data Handling, International Geographical Union

    Commission on Geographical Data Sensing and Processing, 1972,

    Chapters 7, 9, 15, 16.

4. Program Description of Application System for Cartographic

    Scanner/Plotter, Final Report, U.S. Army Engineer Topographic

    Laboratories, Fort Belvoir, Virginia. July, 1972.

5. Holmes, Harvard H., and Austin, Donald M.,

    "Picasso: A General Graphics Modeling Program", ACM SIGPLAN:

    Symposium on Two-Dimensional Man-Machine Communication,

    Los Alamos, New Mexico, Vol. 7, No. 10, October, 1972.

6. Schweitzer, Richard H., "Mapping Urban America with Automated

    Cartography", Proceedings of the American Congress on Surveying

    and Mapping, Fall Convention, Lake Buena Vista, Florida,

    October 2 - 5, 1973, pp. 265-283.

7. Shepherd, Wesley H., "Editing of Digital Data in the Automated Cartography System at U.S. Army Engineer Topographic Laboratories", Proceedings of the American Congress on Surveying and Mapping, Fall Convention, Lake Buena Vista, Florida, October 2 - 5, 1973, pp. 370 - 402.

8. Van Dam, Andries, and Stabler, George M., "Intelligent Satellites for Interactive Graphics", National Computer Conference, Vol. 42, 1973, pp. 229 - 238.

9. Giddings, Gary M., and Carlson, Eric D., "An Interactive System for Creating, Editing and Displaying a Geographic Base File", Research Report, October 15, 1973, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10598