RASTER DATA ACQUISITION AND PROCESSING

Richard F. Theis
U.S. Geological Survey
526 National Center
Reston, Virginia 22092

BIOGRAPHICAL SKETCH

Richard F. Theis received a Bachelor of Science degree in geography from the University of Maryland. He is presently employed by the U.S. Geological Survey, National Mapping Division. His responsibilities include research and development of digital cartographic systems in support of the National Mapping Program.

ABSTRACT

The National Mapping Division of the U.S. Geological Survey has the responsibility for building and maintaining a Digital Cartographic Data Base (DCDB) to support the National Mapping Program. A major task which must be accomplished to fulfill this mission is the conversion of data contained in over 40,000 published 1:24,000-scale topographic maps to digital form for entry into the DCDB. The raster data acquisition and processing system being developed for this task is comprised of three components: (1) raster scanning and editing on a Scitex Response 250 System; (2) raster-to-vector conversion on a DEC minicomputer; and (3) vector editing and attribute tagging on an Intergraph Interactive Graphics Design System. The system, still in the early stages of development, has established the feasibility of raster scanning and processing as a viable method of data capture.

INTRODUCTION

The National Mapping Division (NMD) of the U.S. Geological Survey (USGS) is responsible for fulfilling the objectives of the National Mapping Program. One primary objective of the program is to collect and disseminate selected topographic information in digital form. The vast majority of this digital topographic information will come from more than 40,000 published 1:24,000-scale quadrangle sheets that are archived on stable base color separation films.

The current procedure used to collect data from archival materials is by manually digitizing features and attaching attributes to the digitized features; editing the line and attribute data; and finally, storing the data in the Division's Digital Cartographic Data Base (DCDB). The

Any use of trade names and trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the U.S. Geological Survey.

DCDB is the digital counterpart of the film archive. Manual digitization works sufficiently well for the collection of most map features with the notable exception of contours. Because manual digitizing is a slow, labor-intensive procedure, it is not feasible to be used as the primary method of capturing the contour data from over 40,000 quadrangles in a timely, cost-efficient manner. An automatic means of digitizing data was called for. To fill this need the NMD acquired a Scitex Response 250 System with raster data scanning, editing, and plotting capability.

The Scitex system components include a scanner, a color design console, and a laser plotter, each with a HP 21MX E host processor. The scanner and design console are used for data capture. The scanner is a drum-type raster scanner capable of digitizing sheets up to 36 inches by 36 inches in size at resolutions between 4 and 47 points per millimeter (.25 and .02 millimeters) at a speed of 130 revolutions per minute. The design console, used to interactively edit the raster data, is composed of a color CRT, a digitizing tablet with pen cursor, and a function keyboard. The design console also carries an extensive set of system batch-edit commands.

While the Scitex has solved the problem of time and cost-efficient contour data collection, it has added some additional problems--the foremost being the need to convert the Scitex raster-formatted data to the vector format required by current DCDB processing software. To overcome this problem, NMD obtained a raster-to-vector (R/V) processing program from Dr. A. R. Boyle of the University of Saskatchewan, Canada. This program underwent considerable modification although the basic algorithm was unchanged. In addition, pre-and post-processing routines were written to form a complete R/V processing system. This software is currently operational on PDP-11/70 and PDP-11/60 minicomputers.

The last phase of the R/V processing software builds an Intergraph design file. The Intergraph is a PDP-11/70 based interactive graphics design system. On this system the contour data, in vector format, undergo attribute tagging and final editing in preparation for input to the DCDB processing system.

Three phases of raster data acquisition and processing are discussed in the following text: raster scanning and editing, R/V processing, and vector editing and attribute tagging (see figure 1). All phases are in operation at the National Mapping Division but are still under development. The Unionville, New York, New Jersey, quadrangle contour feature set is used for reference where explanation of a procedure is enhanced by using an example.

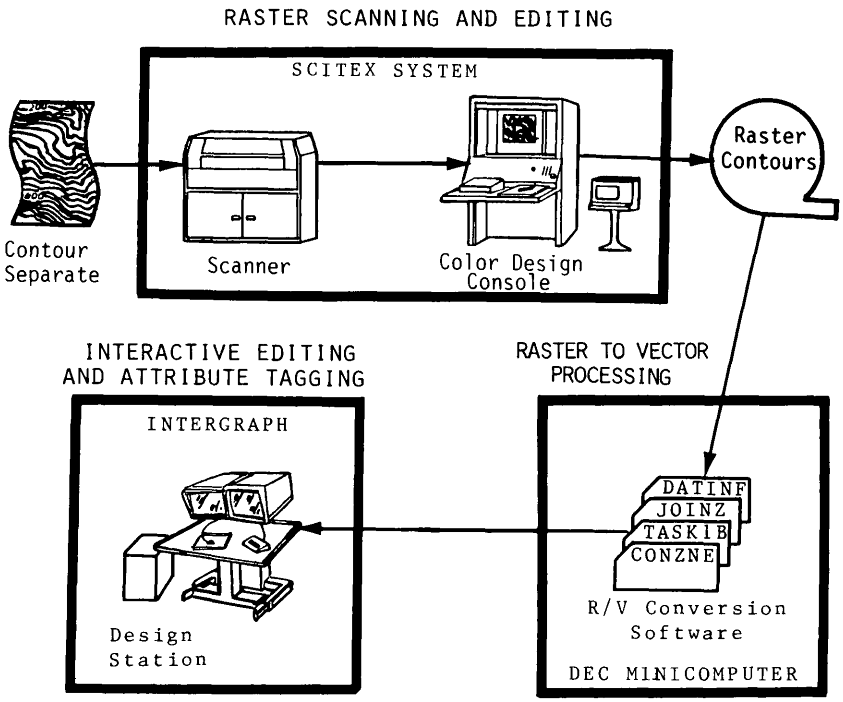

Figure 1. Raster Data Acquisition and Processing

RASTER SCANNING AND EDITING

Scanning

The scanning operation, aside from initial operator setup, is fully automatic. The scan setup procedure begins by securing the film positive source over a white background to the scanning drum. The operator then executes the SCAN command. From this point the SCAN program controls all operator activity. SCAN directs the operator to perform a series of scan initialization procedures, including color calibration and identification of the source area to be scanned. SCAN also prompts the operator to enter scanning specifications such as film thickness and scanning resolution. Upon completion of these initialization procedures, scanning begins. As the scanner drum rotates, the electro-optical scanning head digitizes a column of pixels (picture elements), commonly referred to as a "scan line". The scan head interprets the qualities of light reflected from each pixel and then assigns the pixel the closest calibrated color code. Although the scanner can recognize up to 12 colors; when scanning film positives only two colors are calibrated during scan initialization --black for line work and white for background. The color code assigned to each pixel is stored in digital form on a magnetic disk. After each revolution of the drum, the scan head steps across the drum to digitize the next scan line. When the head has moved across the entire predefined area of the source film, scanning stops and the feature data are then ready for display and edit on the Color Design Console.

The Unionville contour sheet, which measures 17 inches wide by 23 inches high, was scanned at a resolution of 20 points per millimeter (.002 inches). It took approximately 1 hour and 20 minutes to scan. In digital form the sheet measured 9,515 pixels (scan lines) wide by 12,359 pixels high (including 1 inch overedge). The size of each pixel was .002 inches (.05 mm) square.

Editing

The ultimate goal of the editing procedure is to produce a clean contour data set in which only the contour lines and registration ticks reside. All contour lines must be separate, continuous, and one pixel in width . The first task is for the operator to separate distinct features in the file by assigning unique colors to them. Using the design console and the PAINT command, the operator interactively steps through the file and places a dot of color on elevation labels, index contours, registration ticks, and any spots of dirt or noise picked up by scanning. The PAINT command will automatically change the color of a feature to that of the dot placed upon it. Elevation labels and spots of dirt are changed to the background color, thereby deleting them from the file. Index contours and corner ticks are assigned the same color. This procedure results in a three-color file which includes the background color, the index contour and registration tick color, and the intermediate contour color. This feature separation procedure on the Unionville contour set was completed in 4 hours.

Before editing can begin on individual contour lines, the lines must be thinned to a one-pixel width centerline. This operation is performed automatically on all elements in the file by the MIDLINE command. MIDLINE retains the original width definition of the lines and produces in the same file one-pixel width centerlines in a contrasting color. This allows the operator to edit the contour centerlines utilizing the original line width representation as a guide. MIDLINE processed the Unionville data in 3 hours.

After MIDLINE, the FINDEP (find end point) command is initiated. FINDEP is used to locate breaks and spurs in contour centerlines, miscolored contours, and unwanted line symbolization such as depression ticks and dashed lines. The command automatically searches the file for end points of lines. When one is found, it is displayed on the console CRT and control is returned to the operator. The operator can then interactively correct any problem with the displayed line. Breaks in contours are closed, spurs and depression ticks are deleted, and miscolored contours are corrected. When editing of the located line has been completed, the operator then returns control to FINDEP which proceeds to search for the next end point. FINDEP continues executing until all end points in the file have been visited. This process was completed in 4 hours on the Unionville contours.

The next step in the editing procedure involves separating those contours which coalesce. The NODE command is used for this operation. It functions in the same manner as the FINDEP command except that it locates points where lines intersect. As each intersection is found, the operator interactively separates the lines. During this operation lines are never shifted more than one pixel width from their original location. The NODE operation on the Unionville contours was completed in 2 hours.

During the interactive editing procedures on the centerline contours, the operator draws lines in the file either to close gaps or realign contours. The lines input in the drawing process are not of uniform one-pixel width; thick portions of lines are produced where the operator paused while drawing or where new lines meet existing lines. A second automatic thinning operation (MIDLINE) is performed on the data set to remove these thick portions. This second thinning operation on the Unionville contour set took 1 hour.

At this stage the contours are separate, continuous, one-pixel width lines which are ready for R/V processing. The contour file is written to magnetic tape which is then transported to the PDP computer for processing.

RASTER-TO-VECTOR PROCESSING

Scitex raster data designated for entry into the DCDB must first be vectorized so that individual features can be identified and assigned appropriate attribute codes (e.g. elevation). The R/V processing software used to perform this function consists of four programs: CONZNE, TASK1B, JOINZ, and DATINF. CONZNE converts run-length encoded data of the Scitex tape to bit-serial form, and then segments the bit-serial data into 512 by 512 pixel (bit) blocks called zones. TASK1B vectorizes the data set one zone at a time. JOINZ joins together lines which cross zone boundaries to produce a continuous data set. DATINF filters the vector data set and builds an Intergraph design file. In short, the R/V processing software reads a Scitex raster data tape, converts the data to vector form, and produces an Intergraph design file in which the data are ready for interactive editing and tagging. With the exception of TASK1B, all software is coded in FORTRAN. Because TASK1B performs bit processing, it is coded in Assembler language. The functions of each R/V processing software module are described in greater detail in the following text.

CONZNE

The role CONZNE serves in the R/V software is to prepare data which has been scanned on the Scitex for TASK1B processing. This preparation is accomplished through the performance of two tasks. The first task is that of converting the Scitex run-length encoded data to bit serial form. Run-length encoded data are data in which each scan line is defined by a varying number of 16-bit, run-length words. Each run-length word defines a number of consecutive pixels of like color in a scan line. Run-length words carry two values: one specifies the color code; and the other denotes the number of consecutive pixels which exhibit that color. Run-length encoding is an efficient and compact form in which to store multicolor raster data sets. Bit-serial format, on the other hand, represents scan lines as a fixed-length series of bits. Each bit represents a pixel in the raster data set. Only two colors can be stored in this format: an active color represented by set bits (1); and a background color represented by clear bits (0). Although bit-serial format represents only one color and requires significantly more storage than the run-length format, it is much more conducive to the bit processing performed in TASK1B.

The second task performed by CONZNE is that of segmenting the bit-serial raster data into 512 by 512 bit blocks called zones. Zoning serves to break down the large data set into manageable units. A zone is the maximum unit of data which the vectorization routine, TASK1B, will process at one time.

One 512 by 512 bit zone requires 16,384 words of memory to store. This figure amounts to one half of the maximum task image size supported on the PDP 11/70 and 11/60 minicomputers. The Unionville contour set was segmented into 475 zones--19 zones wide by 25 zones high. At a scan resolution of 20 points per millimeter (.05 mm), each zone covered approximately one inch square on the map. Even though the zone is a small segment of the entire data set, it is in itself a large volume of data. A column of zones, made up of 512 scan lines, is called a band. Zones are stored on disk in files according to the band in which they reside.

Because the Unionville data set was comprised of three colors representing index contours, intermediate contours, and background, the data set was processed through the R/V software twice--once to process the index contours, and again to process the intermediate contours. CONZNE processed the Unionville index contours in 1 hour 15 minutes and the intermediate contours in 1 hour 19 minutes.

TASK1B

TASK1B is the key program in the R/V processing software. It is this routine which actually performs the R/V conversion. Conversion is performed on one zone at a time. TASK1B utilizes three important components in the R/V process: a decision template; a node table; and a next position table. The decision template, which is defined in the program code, can be described as a 3- by 3-square matrix of which the eight peripheral squares have assigned to them the values of 1, 2, 4, 8, 16, 32, 64, and 128. The central square has a value of zero. The sum of any combination of template values will produce a unique number between 0 and 255. These numbers directly correlate with locations containing decision information in the node and next position tables. The node table is a node/intermediate point determination table which takes the form of a 256-bit array in the program. Each bit in the array holds a predetermined decision as to whether a bit (pixel) in the raster data set is a node or an intermediate point. A set bit (1) designates a node; a clear bit (0) designates an intermediate point. The next position table is also an array within the program and is 256 16-bit words in size. Each word in the array holds a predetermined address offset value which points to an address in the zone data to which the decision template is to be placed next.

Utilizing the decision template, TASK1B can examine a pixel (bit) of the zone array in relationship to its immediate eight possible neighbor pixels and subsequently make two decisions. The first decision, determined by the node table, is whether the pixel under examination (pixel of interest) is a node or an intermediate point. The second decision, made by the next position table, is the address of the next pixel in the zone array to be examined. Vectorization of a zone proceeds in the following manner. The decision template is moved sequentially through the zone array until a set bit (pixel) is encountered, indicating the presence of a line. This set bit becomes the pixel of interest, and the decision template is positioned directly over it . The pixel of interest always falls within the zero square of the decision template. Once placed, the eight peripheral squares of the template are examined. Values of template squares containing set bits are added together. The sum of these values becomes the pointer value which points to locations , containing decisions, in the node and next position tables. The node table is referenced first. If the node table determines the pixel of interest is a node, an end point of a line has been found. Whenever a node is encountered, a vector-line record is either opened or closed, depending upon whether the node is the first point of a line or the last point. A line record is opened if one does not already exist when a node is found. In this case, the zone coordinates of the node (current pixel of interest) are computed and written to the new line record. The line record is closed upon encountering a second node immediately after the coordinates of the node have been written to it. Whenever an intermediate point is identified, its zone coordinates are simply appended to the current line record.

After the pixel of interest has been identified as node or intermediate and its coordinates added to the line record, the next position table is referenced. The decision template is moved to the location of the adjacent set bit indicated by the table. The bit at this new location becomes the next pixel of interest, and the evaluation procedure begins again. The template continues following the line in this fashion, recording its coordinates to the line record as it goes, until the end node is encountered. All lines in all zones are vectorized in this manner. Completed line records are stored by zone in files on a disk according to the band in which they reside.

TASK1B processed the Unionville index contours in 14 minutes and the intermediate contours in 21 minutes.

JOINZ

After TASK1B processing, the vectorized data set remains segmented into zones. A single line several inches in length on the original scan manuscript would exhibit the following characteristics in the data set produced by TASK1B: the line will cross several zones, resulting in a line divided into several segments (records) with each segment being stored in a different local zone coordinate system; adjacent segments of the line may flow in opposite directions; and common endpoints of adjacent segments will not be the same but will fall at least one pixel apart.

What needs to be done to this line, as well as all other lines in the data set, is to join all of its component segments together to produce a single, continuous line which is stored in one coordinate system. In other words, the zoned data set must be unsegmented and restored to the point where it exhibits in digital form the same characteristics of its prescanned graphic counterpart. This operation is performed by the JOINZ program.

JOINZ utilizes endpoint matching to join together segments of a line. Before matching is begun, all line segment coordinates in the entire data set are converted from local zone (0 to 511) to absolute sheet coordinates (0 to 32,767). When this conversion is completed, JOINZ initiates a sequential search through the zones for the next unprocessed line segment. As line segments are processed, they are flagged as such so that they will be skipped by the sequential search. Upon initiation of the sequential search, no line segments have been processed and it stops at the first line segment of zone one. When an unprocessed line segment is found, a joined-line record is opened and the coordinates of the segment are written to it. The last coordinate written to the joined-line record is retained as the search coordinate. The search coordinate is then examined to determine the candidate zones in which a possible match endpoint might be found. The number of candidate search zones can range from one to four depending upon the position of the search coordinate within the zone and the position of the current zone within the data set. Endpoint coordinates of all line segments within the candidate zones are compared to the search coordinate. If an endpoint falls within four pixels of the search coordinate, then a match is found. The matched line segment coordinates are then appended to those already in the joined-line record. If the matched coordinate happens to be the last coordinate of the matched line segment, then the coordinates of the matched segment are written to the joined-line record in reverse order. The last coordinate written to the joined-line record becomes the new search coordinate and the match search process begins again. JOINZ continues following a line through the zones, joining the segments as it goes, until no match coordinate for the search coordinate is found. When this occurs, the joined-line record is written to a joined-line disk file and the line record is closed. The sequential search is then reinitiated, and the line joining cycle begins again. All lines in the file are joined in this manner.

JOINZ joined the 475 zones of the Unionville index contour data set in 9 minutes. The number of index contour lines input to the program was 1,831 and the number output was 330. The intermediate contours were processed in 54 minutes, and the 6,910 intermediate contour lines input were reduced to 1,198 upon output.

DATINF

The final step in the R/V conversion process is to build an Intergraph design file with the line elements contained in the joined-line file created by JOINZ. In order to speed the interactive editing and tagging procedure on the Intergraph, it is also necessary at this stage to filter out coordinates or vertices of each line which are not essential to retention of line definition. The smaller the design file in terms of number of vertices, the faster interactive editing proceeds. DATINF processes the data set a single line at a time. It reads a line record from the joined-line file, filters the vertices, and performs a simple format conversion on the line data before writing it to the design file. DATINF processed the Unionville index contours in 3 minutes and the intermediate contours in 7 minutes. The number of vertices defining the 1,528 index and intermediate contour lines was filtered down to 69,670. The entire contour data set occupied 1,470 blocks (256 words per block) of the Unionville design file.

VECTOR EDITING AND ATTRIBUTE TAGGING

Vector editing and attribute tagging are performed interactively on the Intergraph System in the design file generated by DATINF. Editing and tagging operations are made utilizing an Intergraph Design Station and its associated edit functions. The design station is composed of: a digitizing table, menu tablet, free-floating cursor, alphanumeric keyboard, and twin CRTs. Editing functions are initiated via cursor through the command menu. For repetitive operations such as attribute tagging, user commands have been written to automate the procedures. An Intergraph user command is simply a user-defined procedure in which a series of individual system functions are combined and executed under a single command.

Discussion here will be limited to the editing and tagging of the Unionville contour data set to be processed for entry into the Digital Elevation Model (DEM) DCDB. Because the Unionville contour set underwent extensive editing on the Scitex system in raster form, no line editing was required at this stage for DEM production; the only requirement was that each contour be tagged with an elevation.

Contour tagging was done with the aid of a user command. This tagging user command is comprised of three system functions--place text, add to graphic group, and change line symbology. In addition to these functions, the user command is responsible for storing and updating elevation and contour interval values after their initial entry by the operator. Upon initiation, the user command prompts the operator to enter an elevation value and a contour interval value. After entry, the user has the option of incrementing or decrementing the elevation by the contour interval amount with each successive touch of a cursor button. When the elevation has been set to a desired value, the operator then identifies, with the cursor, the particular contour in the design file to be tagged. From this point the functions of the user command are executed automatically. The contour is located and a text string of the elevation is placed in the design file at the point of identification. The text string element and the contour line element are then graphically grouped, thereby linking the two elements together. The user command next changes the symbology of the contour line from a solid line to a short dash so that the tagged contours can be distinguished from the untagged contours. Finally, the elevation value within the user command is automatically incremented by the interval value. At this point the user command is ready for the operator to either identify another contour or to change the current elevation value. The operator, utilizing the source litho or contour film positive as reference, tags all design file contours in this fashion. After tagging is completed and the elevation tags have been verified through visual inspection, the data are ready for extraction and entry into a DEM processing system.

Tagging and verification of the Unionville contours required approximately 32 man-hours.

CONCLUSION

Development of the raster data acquisition and processing system has brought the National Mapping Division a step closer to solving the problem of converting more than 40,000 published sheets to digital form in a timely, cost efficient manner. The system, though operational, is still in the early stages of development. It has, at this point, served to establish the feasibility of raster scanning and processing as a viable method of data capture. The scanning of high-volume, contour map separates has proven faster and more accurate than manual digitization.