AN APPROACH TO MICROCOMPUTER-BASED CARTOGRAPHIC MODELING

Andrew P. Mitchell
Harvard University Graduate School of Design
Cambridge, MA 02138

BIOGRAPHICAL SKETCH

Andrew P. Mitchell received his B.A. degree in Geography-Ecosystems from the University of California, Los Angeles, and his Master in Landscape Architecture degree from the Harvard University Graduate School of Design. He is currently a senior analyst at Criterion, Inc., San Diego, CA, where he is involved in development of cartographic software for planning applications. Andrew is a member of the ASP, APA, and the Southern California Computer Aided Mapping Association.

ABSTRACT

The use of computer-based cartographic modeling techniques in environmental planning has increased dramatically in recent years. However, many users of this technology must rely solely on microcomputers for applications involving large cartographic data bases. Existing microcomputer-based cartographic modeling systems are inadequate for this task due to limitations in size, speed and/or specific operations. An approach has been developed which utilizes a compressed raster data structure and local processing of data to overcome these limitations. This approach has been implemented through the development of a software program which contains rudimentary data input/output and transformation operations. Testing of this program confirms that use of a compressed-raster data structure significantly reduces data storage requirements, while the local processing structure results in acceptable processing time. While this approach does have limitations, it presents a viable method for modeling large data sets using microcomputer technology. However, its greatest potential may be as one component of a system which utilizes vector, raster and compressed-raster structures.

INTRODUCTION

Environmental planners, landscape architects, foresters and others involved in the planning and management of the physical environment are confronted with increasingly complex demands and large amounts of data on which to base decisions. Cartographic modeling is one method available to these managers to utilize the increasing amounts of data in the decision-making process. Tomlin refers to cartographic modeling defined as "the act of formally synthesizing geographic information as part of a decision-making process" (Tomlin, 1983). In practice, this includes methods such as overlay mapping and proximity analysis.

While this technology has become well established in the past decade and continues to advance, many users still do not have access to mainframe or minicomputer-based systems. Many of these users are involved in cartographic modeling with data bases that cover large land areas. This is especially true of developing countries which are involved in the development of resource management plans for large areas but may not have the infrastructure or resources to support large systems.

However, recent advances in microcomputer technology have created the possibility of access to this technology for those users who do not have access to the larger systems (Youngmann, 1981). Microcomputers also present the opportunity to make this technology available in the field where it can be utilized on a daily basis by resource managers, thus reducing reliance on systems operated through a distant regional or national office.

Within the past several years, work has begun on a number of micro-based cartographic modeling systems. Despite recent advances in both hardware and software, these existing systems have shortcomings which prevent their full utilization for cartographic modeling involving large data bases. These problems result mainly from the limited storage capacity of microcomputers, and from increased processing time. Most grid based systems are limited to a maximum map size of 100 by 100 cells, or 10,000 cells total. While vector based systems can accomodate larger data bases, the processing time these systems require limits their use in interactive cartographic modeling.

AN APPROACH TO MICROCOMPUTER-BASED CARTOGRAPHIC MODELING

In order to overcome these limitations, research was undertaken to develop a micro-based system which will allow cartographic modeling of a data base virtually unlimited in size while maintaining acceptable processing speed. The approach developed utilizes a compressed data structure in conjunction with local processing.

Compressed Data Structure

A compressed data structure based on a grid rather than polygon structure was chosen as the basis for the system in order to maintain the cartographic modeling capabilities associated with grid format.

Background Compressed raster format has been used frequently in digital image processing applications, but has been used less often in regard to cartographic data, and rarely in cartographic modeling applications. This structure, the general term for which is run-length encoding, has been discussed in numerous texts on digital image processing. (Gonzalez, 1977; Pratt, 1978; Hall, 1979) As applied to digital image processing, the method involves scanning along a row of an image and comparing gray values of adjacent picture elements (cells). Adjacent cells for which the gray value does not significantly differ are grouped together and termed a 'run'. As described by Gonzalez, "the sequence of image elements along a scan line (row) x1, x2...xn is mapped into a sequence of pairs (gi, li) (gk, lk) where g denotes the gray value and l denotes the run length of the ith run", that is, the number of image elements in the run (Gonzalez, 1977). Pratt distinguishes between run-length encoding and what he terms "run-end" encoding. Rather than specifying the number of image elements in a run, run-end encoding specifies the location of the end of the run in relation to the beginning of the row (Pratt, 1978).

Miller discusses the application of compressed raster structure to the encoding and storage of cartographic data and documents several alternative methods of compressed raster encoding (Miller, 1980). One such

VALUE	LENGTH	ROW
0	3	1
1	2	1
1	2	2
0	2	2
1	1	2
1	3	3
2	2	3
2	1	4
1	3	4
2	1	4
2	3	5
1	1	5
2	1	5

RUN LENGTH

VALUE	COLUMN	ROW
0	3	1
1	5	1
1	2	2
0	4	2
1	5	2
1	3	3
2	5	3
2	1	4
1	4	4
2	5	4
2	3	5
1	4	5
2	5	5

RUN END

VALUE	COLUMN
0	3
1	5
-1	
1	2
0	4
1	5
-1	
1	3
2	5
-1	
2	1
1	4
2	5
-1	
2	3
1	4
2	5

ENDING COLUMN

VALUE	POINT
0	3
1	7
0	9
1	13
2	16
1	19
2	23
1	24
2	25

VALUE-POINT

	1	2	3	4	5
1	0	0	0	1	1
2	1	1	0	0	1
3	1	1	1	2	2
4	2	1	1	1	2
5	2	2	2	1	2

figure 1. Comparison of four compressed-raster data structures. Note that run length and run end notation require three integers per run as compared to two per run for ending-column and value point notation. Note also that value-point notation requires the least number of integers to store the map, but also requires the largest integers.

system records the position of the first element of the run and the last element of the run, both identified by column number, as well as the value of the attribute of the run and the number of the row. An alternative system records the row, beginning column, and attribute value of the run.

A third system, termed "ending-column notation", is similar to the run-end encoding described by Pratt, but utilizes a "-1" in the data field to indicate the end of a row. The position of the run is specified by the number of the column in the x-y matrix in which the run ends. This approach has several distinct advantages over systems which use both beginning and ending column since only two values are stored instead of three. In addition, data searches are facilitated since the location of the required data can easily be determined from the column and row values, which are known. Data search becomes much more difficult using run-length encoding since data must be located by summing the run length values to determine the beginning and ending points of a run (Miller, 1980). Figure 1 compares several compressed raster data structures.

Value-Point Notation A variation of the ending column notation was developed as part of this research. Termed "value-point notation", this approach uses two integers to describe each run: the value of the attribute associated with the run, and the ending point of the run, which is the position of the point in the x-y matrix (figure 1). This method has the advantage of further reducing the total number of runs, and thus the storage required for the map, since a run may encompass several rows. Using this approach, a map of a constant value would be stored as one run. Using ending column notation, the same map, of x rows, would be composed of x runs. For comparison, the same map stored in grid format would be composed of x * y points, each represented by an integer value. Thus an overlay stored in value-point notation can be thought of as a continuous string of runs of varying length and value, with the endpoint of the final run equal to the number of cells in the overlay. While data search is more readily accommodated using this method, operations involving display of data are made somewhat complex. In addition, the largest value stored using value-point notation is the value associated with the last point of the map, which is equal to the total number of points (cells) in the map. This may be a limitation for some microprocessors. In contrast, the largest value stored is equal to the number of columns in the map for ending column notation, and equal to the length of the longest run for run-length notation.

Discussion of compressed-raster structure There are disadvantages associated with compressed raster encoding, such as the complexity of algorithms for cartographic modeling required by compressed raster, and the increased difficulty in data searching. The predictable nature of traditional raster, in which each data element is explicitly located, facilitates both of these tasks. However, when working within the limitations of microcomputer technology the greater storage and processing requirements of traditional raster offset the advantages to be gained. One major advantage of the compressed raster approach is that fine resolution mapping, approaching that associated with traditional cartographic methods, can be achieved. When using a traditional raster format, doubling the resolution of a data set will quadruple the number of cells. The result has been that users decrease resolution in order to decrease the number of data elements. This problem is reduced with a compressed-raster data structure since doubling resolution only doubles the number of runs (figure 2).

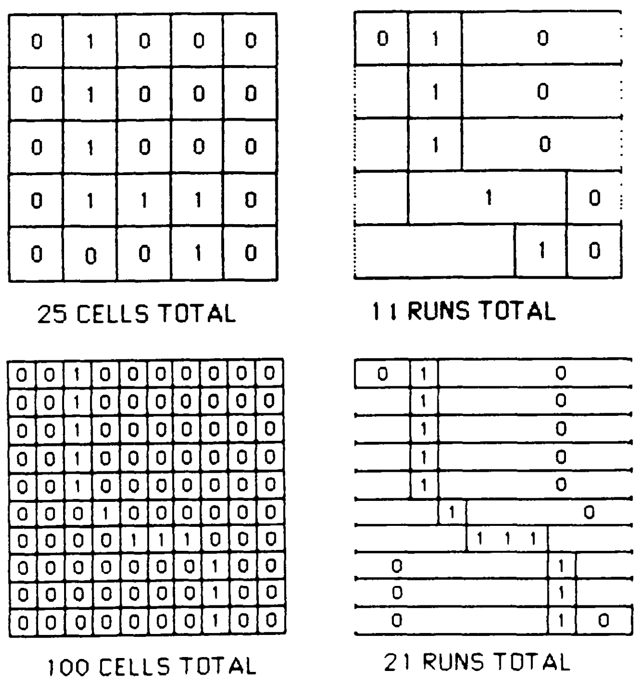


figure 2. Comparison of the amount of additional data generated for traditional grid vs. compressed raster when resolution is increased.

The savings in storage to be gained from use of compressed raster will vary based on the configuration of the overlay being encoded. Hall has noted, in regard to digital image processing, that "The efficiency of run length encoding depends on the number of gray level transitions or edges and could therefore be expected to be most efficient for images with a small number of edges and gray values"

(Hall, 1979). When applied to cartographic data, this means that those maps with few attributes distributed over large areas will require fewer runs than those maps which have many attributes dispersed widely and in small clumps. The extreme example is surficial data, such as elevation, in which each cell can be expected to have a value unlike its neighbors, thus resulting in many runs. The "worst case" example would involve a map in which each cell requires its own run--since a compressed raster structure requires two values per run (value of the attribute and run length or ending column or point), such a map would require twice the space of the same map stored in traditional raster format. However, since for many modeling purposes the surficial map would be reclassified to several ranges of value (slopes of 0-8%, 9-15%, 16-25% and >25% for example), the result would be a map which could efficiently be stored in compressed format.

Local Processing of Data

In order to allow processing of large data sets using microcomputer technology, it is necessary to take into account the limited main memory capacity of existing microcomputers. While microprocessor technology continues to develop, thus increasing the memory capacity of microcomputers, many machines in use are limited to as little as 16k bytes of main memory. One approach to accommodating this limitation is to implement local processing of data. This approach entails retrieving a portion of the data set from secondary storage, processing it, and returning it to secondary storage. This process is repeated such that the data set is processed sequentially from beginning to end, one portion at a time.

The advantage of this approach is that only a small amount of data is contained in main memory at a given time, thus main memory capacity can be quite small. By comparison, many traditional raster systems require that entire maps be stored in main memory at one time, and often two are required for data transformation operations.

The major disadvantage of local processing is that the number of input/output operations is greatly increased and thus processing time is increased. The result is a trade off between space and time: the more main memory available, the larger the portion of the data set that can be processed at a time, and the fewer the input/output operations. Theoretically, the size of the portion to be processed would be specified by the user to be as large as possible with a given microcomputer. To process data sets stored in traditional raster format using local processing would probably result in prohibitive processing time and cost. However, by local processing of compressed raster data, the amount of data to be processed is already greatly decreased, thus the number of input/output operations will be decreased. It is this combination of compressed raster data structure and local processing which allows the capability for processing of large data sets using microcomputers.

IMPLEMENTATION

A software program using the described structure has been written to test the viability of the above approach as a basis for cartographic modeling of large data bases using microcomputer technology.

Structure of the System

The current program, tentatively entitled MAP2, is written in FORTRAN 77 high level programming language and utilizes the value-point notation data structure, described previously, as well as local processing. Runs are represented as two-dimensional arrays and are labeled "VP" (for value/point, the order in which indicator integers are listed in the array). To take advantage of local processing, arrays are packaged in records of a specified length. In the current version, each record contains one array of 256 runs. Since each run is represented by two integers of 4 bytes each, each record requires 8 * 256 bytes or 2k bytes.

Although structured to run on a microcomputer, the program was developed on an IBM 370 mainframe computer to take advantage of the CMS operating system utilized by the IBM. CMS files in binary format are created by the program, stored in CMS, and are called as input by the program as required. The program uses a prompt format for user interface: users are asked to enter input for specific operations (for example, the number of the map file to be processed, the number of the map file to be created, and data to be used in the processing, such as values to be assigned when reclassifying a map).

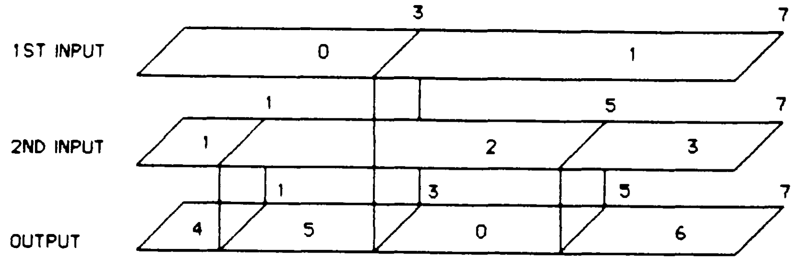
Operations

A number of data input, output and transformation operations have been developed for this system. Data input operations include the ability to download existing raster format maps to compressed raster format, encode maps directly into compressed raster format and create a map of a constant value. Data output operations include the ability to display a map using alphanumerics to represent the values associated with the map, and to convert an existing compressed raster format map to traditional raster format.

Data transformation operations include the ability to alter the values associated with the attributes of a map, assign a unique user-specified value to each combination of values encountered when two maps are compared (figure 3), and to create a map with values obtained by summing the values on two or more existing maps.

Demonstration of MAP2

To test the efficiency of the compressed raster structure and local processing approach as implemented in the MAP2 program a comparison between MAP2 and a traditional raster based system was undertaken.



OLD1	OLD2	NEW
0	1	4
0	2	5
1	3	6

CONVERSION TABLE

	1ST INPUT VALUE	1ST INPUT POINT	2ND INPUT VALUE	2ND INPUT POINT	OUTPUT VALUE	OUTPUT POINT
1ST SEARCH	0	3	1	1	4	1
2ND SEARCH	0	3	2	5	5	3
3RD SEARCH	1	7	2	5	0	5
4TH SEARCH	1	7	3	7	6	7

figure 3. The Cross operation. A new run is created each time the endpoint of an existing run is encountered on either input array. Note that on each search the array incremented is the one which contained the shorter run on the previous search.

Hardware and software utilized The comparison utilized an IBM 370 mainframe computer and compared storage and processing time of MAP2 with the Map Analysis Package (MAP) (Tomlin, 1983), a typical, in terms of data structure, traditional raster based system. MAP assigns a 2-byte integer to each cell in a map and requires, for most operations that two entire maps be in main memory for processing. MAP2 defines each run in terms of two 4-byte integers and processes maps in records of 256 runs each. Thus, this comparison is of the efficiency of both the compressed raster structure and the local processing approach combined.

Data bases The two programs were compared using two separate data bases. The first, Petersham, consists of a 36 square mile area in central Massachusetts. The data base is 180 rows by 180 columns for a total of 32,400 cells. The data base consists of approximately ten data maps, three of which, Waterbodies, Soils and Slope, were chosen for use in the comparison. The second data base, Alaska, covers a 54 square mile area of the Tongass National Forest in Southeastern Alaska. The data maps are 250 rows by 90 columns for a total of 22,500 cells. The data base consists of six maps, four of which, Waterbodies, Wildlife, Visual quality, and Land Management Zones, were used in the comparison. All maps were originally encoded in traditional grid format and were downloaded to compressed raster using the Gridin operation of the MAP2 program.

Comparisons When comparisons were made of storage space required, substantial savings were achieved for almost all maps when stored in compressed raster format (Table 1). The exception, as would be expected, was the Petersham slope map which required more space in compressed raster than in traditional grid format. Maps with large continuous areas of a single value, such as the Petersham soils map or the Alaska land management zones map, required the least space. It should be noted that since both of these data bases consist of less than 32,768 cells, the compressed raster arrays could actually have been specified using two 2-byte integers, thus cutting required storage space to half of that shown by the figures in the table for MAP2.

Petersham Data Base	MAP2	MAP
Waterbodies	39kb	64kb
Soils	32kb	64kb
Slope	190kb	64kb
Alaska Data Base		
Waterbodies	10kb	45kb
Wildlife	10kb	45kb
Visual Quality	14kb	45kb
Land Management Zones	12kb	45kb

Table 1. Comparison of storage requirements of several maps utilizing MAP2 and MAP.

An objective comparison of processing time for specific operations, was difficult to achieve since many extraneous factors were involved in running the programs (for example, the time required to type in commands). However, comparisons of processing time seem to indicate no substantial increase in processing for the compressed raster data, this despite the fact that the specified MAP2 record length was 2k bytes. Presumably, increasing the record length would result in more efficient processing for the compressed raster data.

SUMMARY

It is apparent that use of a compressed raster data structure presents a viable approach for most cartographic modeling operations. In addition, when combined with the use of local processing, this structure provides the basis for a microcomputer based system. However, the data structure does make certain operations more complex in development and execution, specifically those which involve sampling of points surrounding a cell. In addition, the efficiency of this structure does not apply to surficial data.

Possible applications of this approach, apart from as a basis for a microcomputer-based system, include high resolution cartographic modeling and image processing.

However, its most valuable application may be as one component of an integrated system which utilizes polygon, raster, and compressed raster data structures as appropriate for the type of data utilized and modeling operations performed.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Dana Tomlin for his collaboration on some of the concepts presented in this paper, and Dr. Carl Steinitz and Mr. Denis White for their valuable suggestions.

REFERENCES

Gonzales, R.C. and P. Wintz. 1977. Digital Image Processing. Addison-Wesley, Reading, Mass.

Hall, E.L. 1979. Computer Image Processing and Recognition. Academic Press, New York.

Miller, S.W. 1980. A compact raster format for handling spatial data. Technical Papers of the ACSM Fall Technical Meeting, Niagara Falls, NY.

Pratt, W.K. 1978. Digital Image Processing. Wiley, New York.

Tomlin, C.D. 1983. Digital cartographic modeling techniques in environmental planning. PhD. Dissertation, Yale University, New Haven.

Youngmann, C.E. 1981. Portable geographic information systems for on-site environmental data collection. Harvard Library of Computer Graphics/1981 Mapping Collection, vol. 15, Computer mapping of natural resources and the environment. Harvard University. Graduate School of Design.