2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
4 Sep 90

3. REPORT TYPE AND DATES COVERED
Scientific Paper - 23-27 Jul 90

4. TITLE AND SUBTITLE
Scan-Line Methods in Spatial Data Systems

5. FUNDING NUMBERS

. AUTHOR(S)
Michael McDonnell

'. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Topographic Laboratories
ATTN: CEETL-LO
Fort Belvoir, VA 22060-5546

8. PERFORMING ORGANIZATION REPORT NUMBER
R-152

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release;
distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper explains the advantages of scan-line methods in raster or grid-cell spatial data management systems with an emphasis on display and data manipulation techniques. The concept of scan-line methods is examined. Scanline methods use the structure of a raster display to facilitate computation. The technique is especially promising for parallel computers. I show how scan-line methods can expedite data management and display in grid-cell GIS applications by presenting concrete examples.

14. SUBJECT TERMS
Grid-Cell, Spatial Data Management, Data Manipulation, Scan-Line

15. NUMBER OF PAGES
10

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C -	Contract	**PR** -	Project
G -	Grant	**TA** -	Task
PE -	Program Element	**WU** -	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD -	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE -	See authorities.
NASA -	See Handbook NHB 2200.2.
NTIS -	Leave blank.

Block 12b. Distribution Code.

DOD -	Leave blank.
DOE -	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA -	Leave blank.
NTIS -	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report) An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

SCAN-LINE METHODS IN SPATIAL DATA SYSTEMS

Michael McDonnell
U. S. Army Engineer Topographic Laboratories
Bldg. 2592, Fort Belvoir, Virginia 22060-5546
email: mike@etl.army.mil

Abstract

This paper explains the advantages of scan-line methods in raster or grid-cell spatial data management systems with an emphasis on display and data manipulation techniques. The concept of scan-line processing is explained and the effect of basing a system design on scan-line methods is examined. Scan-line methods use the structure of a raster display to facilitate computation. The technique is especially promising for parallel computers. I show how scan-line methods can expedite data management and display in grid-cell GIS applications by presenting concrete examples.

Introduction

This paper examines a method of manipulating data in grid-cell spatial data management systems such as Geographic Information Systems (GIS) and digital cartographic systems. While scan-line methods have advantages when used only in traditional GIS operations such as manipulation of region polygons (binary overlays), much of this paper discusses digital imagery, digitized map backgrounds, and other types of "pictorial" or multibit data. This is because the integration of these kinds of data with GIS will become more prevalent and because scan-line techniques are especially useful for pictorial data. The incorporation of remote sensing data into GIS is a major current issue. Jensen[1] notes that "The full potential of remote sensing and GIS can best be achieved if the technologies are integrated." Remote sensing data consists mostly of digitized photography, so it is important to be able to handle photographic data in a GIS.

Modern digital cartography entails the handling of ever-larger data sets. And it may take many of these data sets to cover a region of interest. This amount of data cannot be handled well on present computer systems, and extrapolation of computer capabilities shows that handling large amounts of data will be difficult for many years to come.

> It is therefore essential that attention be focused on both the data structures that are used to store geographic information and the efficiency of the procedures that are used in relation to these data structures to store, analyze, and retrieve data.[2]

[1] J.R. Jensen, **Introductory Digital Image Processing; A Remote Sensing Perspective**, Chap. 10, "The interface of Remote Sensing and Geographic Information Systems", Prentice-Hall (1986).

[2] T.R. Smith, "Requirements and principles for the implementation and construction of large-scale geographic information systems", Int. Jour. GIS, 1(1) 13-31 (1987).

Even though handling data sets as large as those described above is beyond the capabilities of most current computer systems, we want be able to handle these large amounts of data as well as our computers will let us. We also have to design our software as cleanly as possible so that we are not forced to redesign it when better hardware becomes available. To quote Niels Bohr: "It is very difficult to predict, especially the future," but we need to try and foresee our future systems because we must plan them now.

This paper is an attempt to predict a part of the future of GIS and to design a basis for future software, especially with regard to data management, analysis, and display. The key techniques I will examine are called scan-line algorithms. I will first explain scan-line algorithms in general and then will proceed to outline aspects of design of a generic GIS based on some specific scan-line algorithms. Finally I will analyze the effects of scan-line algorithms on GIS design.

The techniques outlined in this paper promise to be portable to many computer architectures. Scan-line programs run equally well on conventional and parallel computers and give an excellent software strategy for the design of systems that can easily be moved to future computers while allowing their development on current machines. It is my experience that scan-line techniques also run faster than other methods on current computers and therefore provide an immediate payoff.

First, let's look at what scan-line algorithms are and why they are important.

Scan-line Algorithms And Computer Graphics

All data on a computer is stored as numbers. It is our interpretation of these numbers that gives the data meaning. When I refer to a "point" I mean a set of numbers interpreted as a position in some space. Modern computer graphics is based on two kinds of graphical entities, vectors and rasters. A vector is stored as two points which define the ends of a line segment. Drawing these lines on a display screen is the fundamental action in vector graphics. Vectors have a single value or color throughout their length. A vector may, for example, be red or blue but may not change from red to blue along its length. A raster is a set of horizontal lines which together fill a rectangular region on a display. Unlike vectors, the value assigned to each point of a raster can differ from the value at adjacent points.

In a grid-cell GIS the notion of a grid cell is very close to that of a raster in that a grid is a regular square decomposition of a space, usually some terrain. Values assigned to a grid cell do not necessarily relate to a display variable such as a color but rather act as indices defining attributes of the region, such as soil types. There is another difference between the notion of a "grid cell" and a "raster". A grid-cell data structure has no preferred direction but is just an array of numbers corresponding to the regular division of the area represented. A raster has an ordering of data in horizontal strips. Raster data is stored in "scan lines". A scan line is a single horizontal line of pixels on a display such as a CRT. Scan lines have a privileged status over other [illegible] of lines on a display because standard television display hardware generates [illegible] raster scan. Raster graphics has advantages over vector graphics when it comes [illegible] continuous-tone imaging and for certain calculations such as Boolean combinations of regions. There is a well-developed set of techniques to convert vector data to raster data.[3] The processes used in converting vector data to raster data may themselves be scan-line methods and therefore have the advantages to be discussed in this paper. When I am talking about display I will refer to rasters or scan lines, and when I am talking about storage I will refer to grid cells.

Many current developments in graphics are concerned with raster scans. An important characteristic of a raster is that data is read in as a group of horizontal lines or scan lines. Usually, a raster scan is thought of as a one-dimensional, time-sequenced writing of pixels on a CRT screen, but there are benefits from thinking of the raster as a unique data structure consisting of a set of parallel and *independent* lines of data. The emphasis on the independence of scan lines from one another is what gives scan-line techniques their power. Scan-line algorithms are graphics computations that treat a raster in this way.

This assumption of independent scan lines is true for many operations of interest in GIS calculations. Some recent results, which I will present, show that scan-line independence can be made to be true for other operations by using appropriate geometric transformations. This is currently an active area of research.

Over the last decade or so, several researchers in computer graphics have developed algorithms for display and analysis which use the scan-line structure of the display to good advantage. An example is Lane and Carpenter's 3D surface rendering technique.[4] As these methods have become further developed, other aspects of scan-line algorithms have become evident. In particular, scan-line algorithms offer advantages in data storage and handling, and in ease of running efficiently on parallel computers. We'll examine each of these advantages.

A System Design

To make things more concrete, let's consider the design of a GIS based on scan-line techniques and including the ability to use pictorial and photographic data. Elements of a GIS include a user interface and a means of accessing non-graphic data, but I want to ignore these aspects of the system so we can focus our attention on those graphical presentation and analysis operations which show the unique features of scan-line methods. Other aspects of GIS design, such as storage of attribute data, are not much impacted by the method of graphical data representation anyway. This independence of sections of a GIS allows separate design of the various parts of the system with well-defined and unchanging interfaces between the parts; the essence of good software engineering.

[3] D.F. Rogers, Procedural Elements for Computer Graphics, Chapter 2 "Raster Scan Graphics," McGraw-Hill (1985).

[4] J. Lane and L. Carpenter, "A Generalized Scan Line Algorithm for the Computer Display of Parametrically Defined Surfaces" Comp. Graphics & Image Proc., vol. 11, 290-297 (1979).

What, then, are the problems in handling graphical data? These problems are outlined by Smith[5] and are:

- Encoding for storage and transmission
- Storage and retrieval
- Manipulation and analysis
- Display

Pictorial or photographic data has its own set of problems, some of which overlap with those stated above. These are given by Green:[6]

- Cataloging and accessing
- Radiometric Modification
- Spatial transformation
- Spatial frequency transformation (filtering)

Here is a table that shows which of these operations can be performed using scan-line methods:

TECHNIQUE	SCAN-LINE METHOD?
Encoding for storage and transmission	yes
Cataloging and accessing	no
Storage and retrieval	yes
Manipulation and analysis	yes
Display	yes
Radiometric Modification	no
Spatial transformation	yes
Spatial frequency transformation (filtering)	yes

Cataloging and accessing are database operations not concerned with the details of an individual picture, so they are not suitable for scan-line methods. Radiometric modification treats the image as a whole through histogram or pixel modification. These operations are also not suitable for scan-line methods, although a histogram may be derived efficiently using scan-line operations. Spatial transformations include rotation, scaling, map projection transformations, and perspective displays. These types of operations are suitable for scan-line techniques.

The spatial transformations of rotation and scaling can be done using Fant's resampling method,[7] which impresses me as very original work. Perspective displays can be calculated using Robertson's[8] method, which can produce marvelous images with comparatively little computer time. (This technique introduced me to the power of scan-line methods.) Spatial frequency transformation is done by

[5] op. cit.

[6] W.B. Green, **Digital Image Processing, A Systems Approach**, Van Nostrand (1983).

[7] K.M. Fant, "A Nonaliasing Real-Time Spatial Transform Technique," IEEE Computer Graphics & Applications, 71-80 (Jan 1986).

[8] P.K. Robertson, "Fast Perspective Views of Images Using One-Dimensional Operations," IEEE Computer Graphics & Applications (Feb 1987).

convolution, or smoothing of data, and can be done using scan-line methods if the convolution kernel is separable. A separable convolution kernel is not a special case, but is rather the usual case for rectangular and gaussian smoothing functions, the ones most used for processing images and other pictorial data.[9]

A special case of data modification, which is of great interest for GIS, is vector-to-raster conversion and its opposite, raster-to-vector conversion. Good scan-line methods exist to do vector-to-raster conversion, but this is not true for raster-to-vector conversion. The referenced book by Rogers contains some details of the vector-to-raster conversion process. Vector-to-raster conversion is much less expensive and simpler than raster-to-vector conversion. If data must be converted, then it is best to do vector-to-raster conversion if possible.

Raster-to-vector conversion must balance several competing sources of error to minimize a total error function.[10] This is a type of operation known as linear programming[11] which is known to be computationally expensive. Since the error function is global over the polygon of interest, scan-line methods cannot be used.

The difficulty of raster-to-vector conversion seems to mandate a system design that uses vector formats for data storage, wherever possible, and raster formats for data display and manipulation, such as Boolean combination of regions. This design provides a good match with available data for those of us in the U.S. Department of Defense, since the Defense Mapping Agency (DMA) releases digital terrain feature data in a vector format; as do most other U.S. federal agencies.

Now I would like to examine a few scan-line algorithms in detail to show some of the implementation issues.

Data Compression

Storage and transmission times can be reduced by using compression techniques. In general there is a tradeoff between degree of compression and amount of computer time needed to compress and uncompress the data. Excessive compression also causes loss of image quality. Of course, if you are out of disk space, or have to transmit image data over a slow channel, the higher cost and lower image quality associated with greater compression may be tolerable.

Compression algorithms can be one-directional or bidirectional. For a one-directional algorithm, the data are operated on only along scan lines (right-to-left or left-to-right). A bidirectional technique requires that the data also be operated on in the vertical direction. If the algorithms are one-directional then scan-line methods fit them very well, although efficient transposition algorithms, which I will discuss, can make it reasonable to use bidirectional algorithms too. For scan-line methods we also have to distinguish between one-*directional* compression and one-*dimensional*

[9] J. Canny, "A Computational Approach to Edge Detection," IEEE Pattern Analysis and Machine Intelligence, PAMI-8(6), 679-698 (1986).

[10] Y. Kurozumi, W.A. Davis, "Polygonal Approximation by the Minimax Method", Computer Graphics and Image Processing, 19 248-264 (1982).

[11] R. Sedgewick, **Algorithms**, Chap. 43, Addison-Wesley (1988).

compression. In one-dimensional compression, all pixels have to be visited in a given order, which is typically the same as raster scan order. One-dimensional algorithms, such as the Lempel-Ziv-Welch technique,[12] are unsuited to parallel computers because of the sequential character of the program, which builds an encoding table as it goes through the data. Luckily, there are good one-*directional* compression algorithms, such as run-length coding[13] in which each scan line can be independently compressed. These are the algorithms to use in a parallel scan-line system.

Data compression is usually only used for long-term storage of image or other grid-cell data or for transmission of the data. The data is "uncompressed" before use. An exception to this is color quantization which is essential to display data on a system with a limited color resolution.

File Storage Format For Scan-Line Access

Speed and convenience of accessing displayable data are related to compression but can be considered separately. This is where scan-line methods may have their biggest payoff. To display some data you must:

1. Find the data for the region you want to display
2. Move it off disk into computer memory
3. Modify it, if needed
4. Display it

These steps are all difficult problems when dealing with as much data as we have to handle in a GIS. If data is stored in a form suitable for scan-line operations, it makes manipulation much easier. Curiously, the best storage format for scan-line processing is not itself organized as scan lines but rather as square blocks.[14]

The storage format is a blocked format in which a large display is stored as a plane of "tiles" of data, each tile being some power of 2 cells on a side. A size of 256 cells is reasonable because of the way computers are organized, i.e. it is consistent with usual disk block sizes[15].

[12] T.A. Welch, "A Technique for High-Performance Data Compression," IEEE Computer, 8-19 (June 1984).

[13] R.C. Gonzalez, P. Wintz, Digital Image Processing, Second Edition, Chap. 6, Addison-Wesley (1987).

[14] L.H. Quam, "A Storage Representation for Efficient Access to Large Multidimensional Arrays," Proceedings DARPA Image Understanding Workshop, 104-111 (1980).

[15] This is also true of the 128 X 128 storage format called ARC Digitized Raster Graphics (ADRG) and used by the U.S. Defense Mapping Agency for its digitized map sheet products.

A tiled data format has several advantages:

(1) For roaming over a plane of image data, the speed with which data has to be read from the disk is independent of the direction of roaming.

(2) Amount of data in computer memory (RAM) is minimized easily. Only those tiles which have a portion visible on the screen need to be in RAM.

(3) Transposition of data may be done incrementally. Transposition of arrays is an important step in many scan-line image processing methods such as convolution. Using raster-based techniques,[16] large two-dimensional data sets may be transposed by first internally transposing individual tiles of cells and then accessing the tiles in transposed order. Note that it is not necessary to move tiles around on the disk, it is only necessary that the tiles be *accessed* in transposed order.

Perspective Display

One of the reasons we have to be concerned with transposition of data, and the principal cause of my interest in scan-line techniques, is the recent invention by Robertson, referenced above, of an algorithm for viewing digitized photographs or maps as true perspective displays using only scan-line methods. Robertson's method requires that the data be transposed to create perspective displays for data sets too large to fit in RAM at one time. In fact, Robertson's method requires no more than one scan line at a time to be held in RAM. The exciting result is that even small personal computers can create true perspective views of data sets far too large to fit in their memory.

Scan-line generation of perspective views is very fast compared to other techniques. This is crucial to their usefulness since it is our experience from field test demonstrations that a slow algorithm will simply not be used because it is too tedious. I have programmed Robertson's algorithm on a Sun 3/260 (about a 3 MIP machine) and can generate a 512 X 512 perspective in 30 seconds; much faster than the algorithms we are now using in field tests. When you consider that this is a scan-line algorithm, and therefore easily parallelizable, very rapid views of terrain can be calculated on future computers without special-purpose graphics hardware. And without much reprogramming.

Line-of-sight and masked-area (hidden region) plots use the same calculations as perspective displays, so these capabilities are also provided by Robertson's algorithm.

Map Projections And Rectification

A major problem with using data from different sources is that they will in general not be in the same coordinate system. For example, we may have a digitized oblique aerial photograph and want to register it with data in geographic coordinates such as is provided by DMA. Fant's algorithm, which I mentioned earlier, can be

[16] D. Fraser, R.A. Schowengerdt and I. Briggs, "Rectification of Multichannel Images in Mass Storage Using Image Transposition," Comp. Vision, Graphics & Image Proc., 29 23-36 (1985).

used to do simple rotation and linear scaling of data, but what can we do if we want to do a more complicated transformation? Even complicated warping (a "mapping" in mathematical terminology) can be done using scan-line techniques. This is shown schematically in Figure 1 where a square block of data is mapped into a complex shape through two successive scan-line transformations. The dashed lines are drawn to show how the interior of a region is reshaped. The mapping is first done in the horizontal direction from A to B and then done in the vertical direction from B to C. The step from B to C can also be done along horizontal lines by first transposing B, just as I mentioned when discussing perspective displays. The dotted lines in Figure 1 show that all movements of data occur along parallel straight lines; that is along scan lines.

This mapping works by first finding the equations for the borders of a region after transformation. These borders are the curved solid lines in part C of figure 1. These borders may be expressed as cubic spline curves. Working backwards to part B of Figure 1; projections of the borders in part C must be found to get the solid curved lines of part B. These borders may also be expressed as cubic spline curves. Mortenson[17] shows a mathematical formalism to use. I am still trying to work out a good method for performing a general mapping using scan-line techniques. In particular, it is not obvious how a user should specify these kinds of transformations. It is conceptually simpler to make purely mathematical transformations such as cartographic projections which do not require extensive interaction with the user.

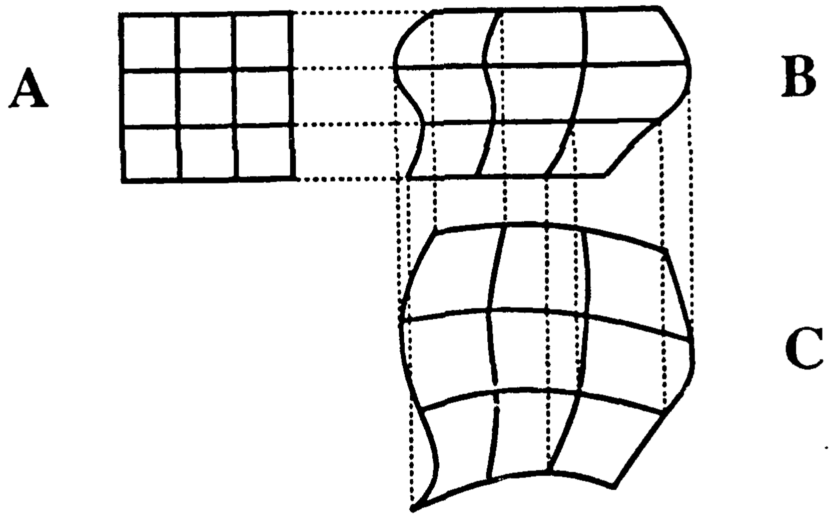

Figure 1: Mapping a square block of pictorial data into a complex shape. The data is first transformed along horizontal scan lines from A to B and then is transformed along vertical scan lines from B to C. See the text for a discussion.

[17] Michael E. Mortenson, Geometric Modeling, John Wiley and Sons (1985), ISBN 0-471-88279-8.

Use Of Parallel Computers

Another exciting aspect of scan-line methods is that adapting them to parallel computation is quite easy. You just assign a processor to a given scan line (or to a set of scan lines). This applies to scan-line algorithms of all sorts such as the techniques used for data compression and vector-to-raster conversion. Compare the ease of moving scan-line algorithms to a parallel computer with almost any other problem or technique requiring computation. It has been said that training programmers to "think parallel" will be one of the biggest problems we will have to face as the age of parallel computers comes upon us. Scan-line algorithms are a happy exception to this. A scan-line program does not change conceptually when it is rehosted on a parallel computer and little reprogramming must be done. A more subtle point is that, since scan lines are processed independently, there is minimum communication needed between processors and near-linear speedup of computation with number of processors can be expected. So it is not only easy to perform scan-line algorithms on parallel computers, it is efficient as well.

Other Operations

I have indicated some of the unique features of scan-line processing of cartographic data. Many other operations of interest in spatial data handling can be performed using scan-line techniques, especially if transposition is allowed. These include:

- polygon (region) filling
- region thinning and growing
- boolean combination of regions
- perimeter and area calculations
- centroids
- intervisibility analysis
- shape classification

For a somewhat dated, but still basically valid analysis of raster algorithms useful for GIS, see Peuquet.[18] Remember, not all raster-graphics algorithms are scan-line algorithms, but many raster operations which used to be considered incapable of being reduced to scan-line methods have been cast into a scan-line model by an appropriate transformation. Research on these transformations is continually providing us with new scan-line techniques. Most of the work I have mentioned in this paper is recent, and I expect to see more new scan-line algorithms in the next few years.

[18] D.J. Peuquet, "Raster Processing: An Alternative Approach to Automated Cartographic Data Handling," The American Cartographer, 6(2), 129-139 (1979).

Review And Conclusions

There are many other considerations to be handled in designing a real GIS, but let's examine how the decision to use scan-line techniques has affected our conceptual system design. The salient features of the system are:

(1) Displays, and some calculations, are based on methods using independent scan lines, minimizing the data held in computer memory. Therefore there is no inherent system limit on the size of data sets that may be handled.

(2) Moving the software to a parallel-processing environment should be easy. This preserves the software investment while allowing efficient use of new (and faster) computers.

(3) Raster data is stored as square tiles of cells with dimensions being some integer power of 2.

(4) Data for long-term storage is compressed using run-length coding or is stored as vectors.

(5) Efficient algorithms for vector-to-raster conversion are available, as are algorithms for viewing and manipulating data; the system runs fast.

Notice how the early high-level decision to use scan-line methods has determined every aspect of that part of the system concerned with display and some parts of the GIS that are concerned solely with data manipulation. Closer consideration of other, non-graphical, aspects of the system such as handling attribute data would certainly show some dependence of these parts on the decision to use scan-line techniques in the display parts of the system and may even reveal some disadvantages of scan-line methods. An example of an operation which is more difficult in a raster system than in a vector system is the assignment of multiple attributes to a region; more difficult but not at all impossible.

Any investigation such as this must consider hardware advances. It is difficult to design graphics software because of the many special hardware architectures being designed, or even already marketed, which require particular programming techniques to get the most speed out of these architectures. The rise of this special-purpose graphics hardware is simultaneously promising and threatening. It is promising in that we have the possibility of graphics speed unimaginable a few years ago at an affordable price. It is threatening because the major cost of any computer system these days is in the software, and special-purpose hardware requires special-purpose software which may not be transportable to other machine types without expending lots of time and effort. My personal bias is to protect the software investment, even at the cost of somewhat reduced speed, but the techniques I have discussed in this paper have a high degree of both speed and portability.

I am therefore confident that scan-line techniques form a fruitful basis for GIS design. Scan-line methods are efficient on present computers and will be even more efficient on future computers. These techniques are also portable and allow rehosting of programs on parallel computers without extensive reprogramming or rethinking of algorithms, thereby preserving the software investment.