

# Raster and Vector Processing for Scanned Linework

David D. Greenlee

EROS Data Center, U. S. Geological Survey, Sioux Falls, SD 57198

**ABSTRACT:** Recent advances in scanning technology have made it possible for linework to be scanned on relatively inexpensive systems and output as raster images. These images can be edited, automatically thinned to connected lines, and converted to vector representations using conventional image processing techniques. An investigation of raster editing techniques, including thinning, filling, and node detecting, was performed by using specialized software. The techniques were based on encoding the state of a three-by-three neighborhood surrounding each pixel into a single byte. This byte has a range of 0 to 255, and stores all possible surround conditions for a pixel. Implementing various methods for thinning, filling, and node detection was greatly simplified by making use of this surrounding topology. Also, it was possible to develop and test decision tables for raster editing tasks, and to assure symmetry and consistency in their application. A prototypical method for converting the edited raster linework into vectors was also developed. Once vector representations of the lines were formed, they were formatted as a Digital Line Graph, and further refined by deletion of nonessential vertices and by smoothing with a curve-fitting technique.

## INTRODUCTION

THE CONVERSION of rasterized linework to vector chains or arcs has provided researchers with a challenging set of problems for several years. Peuquet (1981) gives a technical overview of the methods that have been developed for raster-to-vector conversion. Based on a large body of available techniques, this paper describes the use and refinement of several methods for processing linework in raster and vector form. Automated aspects of the raster-to-vector process are described, but interactive editing steps are beyond the scope of this study.

This paper also attempts to bridge a gap between theoretical work on raster-to-vector processing and the many turn-key production systems that are in use but are often not well understood. During this project, techniques were developed that created gaps, spikes, spurs, snow, node slippage, and various other unsavory artifacts of automated processing. Through this experience has come a better understanding of the automated techniques that work best for a given application and some insight into the optimal mix of automated versus manual editing techniques.

## BACKGROUND

Techniques have recently been developed by Jenson (1985) for extracting hydrologic basins and for delineating drainage networks from raster-formatted digital elevation models (DEM). In applying these techniques to project work, it was determined that a conversion of drainage lines to vector format would best allow for efficient overlay with other mapped data, for measuring lengths and areas, and for ordering of streams. Techniques were available for converting classed hydrologic basins into polygonal form (Nichols, 1981), but not for converting rasterized drainage lines into vector form.

A method was needed to convert rasterized linework, as extracted from the DEM, into vector chains or arcs that could be input to a vector-based geographic information system (GIS). It was observed that this problem is analogous to the problem of converting scanned map linework into vectors. It was hoped that future scanning of map data could be facilitated by developing a generic solution to this problem and by better understanding the processing and conversion steps involved.

## ENCODING OF TOPOLOGICAL SURROUNDS

Rasterized linework can be visualized as a bilevel or binary image having pixels represented by ones if a line is present, and represented by zeros for the space between lines. Many techniques are available for processing and refining lines in this

form (Rosenfeld and Kak, 1982; Pavlidis, 1982). Most techniques utilize a set of rules for operating on each pixel based on its state and that of its eight contiguous neighbors. Because neighborhood operations tend to be time consuming and computationally expensive to perform, special attention was given to this problem.

The encoding method described here was inspired by Golay (1969), who described 64 surround conditions for a hexagonal data structure. Because we usually work with image data in a raster format, our technique encodes for each pixel a single byte of eight bits that completely describes the state of all adjacent neighbors. For raster-processing operations to be performed (for example, filling, thinning, or node detection), a set of decision rules is developed and placed in a table with one entry for each possible surround condition. The raster image is then processed in the following fashion:

- (a) encode the surround condition for each pixel, and store it
- (b) for each pixel,
  - (1) find the decision rule referenced by the surround condition
  - (2) make changes to the pixel where indicated by the rule
  - (3) if changed, update the surround values for neighbor pixels
- (c) if any changes were made on this pass, go to b)

By encoding the surround values on the first pass and then updating them only when changes have been made, unnecessary encoding can be avoided in subsequent passes. This can greatly improve the speed of processing, because many of the processing functions are applied iteratively until no changes are made in an entire pass.

Encoding can be manually performed by adding together the values shown on the following diagram for the neighbor pixels that have a line present:

64	128	1
32	X	2
16	8	4

Digitally, encoding can be performed by setting the corresponding binary bits of an eight-bit byte. In either case, the resulting value occupies the range of values 0 to 255, or 256 possible states. An example of a Fortran subroutine to perform this encoding is shown in Figure 1.

The complete set of 256 possible surround conditions is shown in Figure 2. In this figure, surround conditions have been grouped to allow decisions to be developed easily and consistently. The 256 surround conditions were combined into pattern groups by



```

SUBROUTINE ENCOD (NBR8,IVAL)
C
C   IVAL = output value (byte)
C   NBR8 = input array of eight   MASK = bit masks corresponding
C   neighbors ordered as:         to NBR8 as follows:
C       1 2 3                     64 128 1
C       4 5                       32 2
C       6 7 8                     16 8 4
C
C   DIMENSION NBR8(8),MASK(8)
C   LOGICAL*1 IVAL
C   DATA MASK /64,128,1,32,2,16,8,4/
C
C   IVAL = 0
C   DO 100 K=1,8
C       IF (NBR8(K).EQ.0) GO TO 100
C       IVAL = IVAL.OR.MASK(K)
100  CONTINUE
C   RETURN
C   END

```

FIG. 1. A FORTRAN subroutine for performing the digital encoding.

sorting on a set of attributes. Within each of the count groups (labelled 1 to 8 and organized vertically) are subgroups that may contain as many as eight patterns. These are referred to as pattern groups and may consist of a pattern; rotations of 90°, 180°, and 270°; and its mirror image and rotations. Because many of the patterns are symmetrical, they generate non-unique or duplicate patterns when rotated. As a result, some subgroups will contain only four, two, or as few as one unique pattern. Grouping allows the 256 possible surround conditions to be aggregated into a more manageable 49 pattern groups. None of the operations described in this paper requires subdivision below the pattern group level.

Each of the pattern groups can be represented by a set of attributes that allows decision tables to be efficiently developed. Figure 3 is a table of pattern groups and associated attributes. One basic attribute is the number of neighbors that have a line present. For thinning operations to be performed, it is useful to know whether the pattern is classed as simple (Rosenfeld and Kak, 1982) or non-multiple (Pavlidis, 1980). For this condition to be true, a pixel must have exactly one pixel or a connected set of pixels adjacent to it. Landy and Cohen (1985) use similar information, which they refer to as the number of black-to-white transitions present. In addition, they determine the number of lines that result when the center pixel is deleted. This is referred to as the number of gaps, because it measures whether a gap in connectivity would be created by thinning. These characteristics are constant for each pattern group, and were used to define and order the groups shown in Figure 2.

### RASTER OPERATIONS

Several raster operations were performed on two example data sets. These examples demonstrate the various raster and vector operations that are involved in the conversion of raster-linework to vector arcs.

Sample data set 1 is a digital elevation model (shown as a shaded-relief display in Figure 4a). Figure 4b shows rasterized linework that has been extracted from the DEM as described by Jensen (1985). The linework is portrayed as a binary image with lines as ones and background as zeros.

Sample data set 2 is a small portion of a topographic map, scanned using commercially available raster scanning equipment. Figure 5a represents the original image of 256 possible levels, with the lines being the lower values and the image background as the higher values. In Figure 5b, a convolution filtering process has been performed to enhance the local contrast of the lines against the background. This is commonly referred to as edge enhancement or high-pass filtering, and is



FIG. 2. The 256 possible surround conditions for a pixel and its eight adjacent neighbors.

described by Moik (1980). The data were then converted to a binary valued image by selecting a threshold value that best separated lines from background (Figure 5c).

### FILLING OPERATIONS

The purpose of a fill operation is to eliminate small voids in a line that add unnecessary complexity. The term fill is also used in computer graphics applications, where it is synonymous with flooding or painting of the interiors of polygons. While somewhat similar, the latter is not the process we require in automated editing of lines. In the process of scanning the linework, small breaks or voids are sometimes created within the lines that are artifacts of the scanning process or are due to imperfections in the original lines. In some cases, we wish to eliminate valid inclusionary voids that are too small to be considered significant. In addition to filtering out small voids, filling also tends to generalize and fill in acute angular features



PAT GRP	NUMBER OF 8 NGHRS	PATS IN GRP	NUMBER OF TRANS	NUMBER OF GAPS	FILL RULE	THIN RULE STEP1	THIN LANDY STEP1	THIN RULE STEP2	THIN LANDY STEP2	NODE MARK RULE	NODE MARK SAKAI
1	0	1	0	0				Y		T	
2	1	4	1	1				Y		T	T
3	1	4	1	1				Y		T	T
4	2	8	1	1				Y		T	T
5	2	8	2	2							
6	2	4	2	1				Y	Y		
7	2	4	2	2							
8	2	2	2	2							
9	2	2	2	2							
10	3	8	2	1				Y	Y		C
11	3	8	2	2							C
12	3	8	2	2							C
13	3	8	2	2							C
14	3	4	1	1		Y	Y	Y	Y	T	
15	3	4	1	1		Y	Y	Y	Y		M
16	3	4	3	1	Y			Y		C	M
17	3	4	3	2						C	M
18	3	4	3	3						C	M
19	3	4	3	3						C	M
20	4	8	1	1		Y	Y	Y	Y		C
21	4	8	2	1		Y	Y	Y	Y		C
22	4	8	2	2							C
23	4	8	3	1	Y			Y	Y	C	M
24	4	8	3	2						C	M
25	4	8	3	3						C	M
26	4	4	2	1				Y	Y		C
27	4	4	2	2						C	C
28	4	4	2	2						C	C
29	4	4	2	2						C	C
30	4	4	2	2						C	C
31	4	1	4	1	Y				Y	C	C
32	4	1	4	4						C	C
33	5	8	2	1	Y			Y	Y	C	C
34	5	8	2	1	Y			Y	Y	C	C
35	5	8	2	2						C	C
36	5	8	2	2						C	C
37	5	8	3	1	Y			Y	Y	C	M
38	5	8	3	3						C	M
39	5	4	1	1	Y	Y	Y	Y	Y		
40	5	4	1	1		Y	Y	Y	Y		
41	6	8	1	1	Y			Y	Y	C	C
42	6	8	2	1	Y			Y	Y	C	C
43	6	4	2	1	Y				Y	C	C
44	6	4	2	2						C	C
45	6	2	2	1	Y				Y	C	C
46	6	2	2	2						C	C
47	7	4	1	1	Y						
48	7	4	1	1	Y						
49	8	1	1	1	Y						

FIG. 3. Pattern groups of topologically similar surround conditions and their corresponding decision rules.

not be created by deletion of the center pixel (that is, where 8-way connectivity is maintained), and where the pattern is simple (that is, where only one black-white transition is found as the neighbors are tested in circular order). A special case is posed by the "T" connection (pattern group 16), that will become a "Y" connection unless given consideration. Also, single or isolated pixels (pattern group 1) may be deleted during thinning, provided they are considered erroneous or insignificant. Endpoints (pattern groups 2 to 4) may or may not be preserved because some may be valid lines and some may be spurs that are created as artifacts of thinning of thick lines (more than four pixels wide). Spurs may also be deleted after vector conversion, when length may be calculated and used as an editing parameter. Figure 4d shows drainage lines after filling and two-step thinning have been performed. Figure 5d shows scanned linework after filling and two-step thinning. In both examples, the thinning operation has reduced the original binary image to skeletal connected lines, and has eliminated many unnecessary pixels.

## NODE MARKING

Once lines have been filled and thinned, node marking may be performed. Two rule tables for node marking are shown in Figure 3.) Nodes that are endpoints or line terminations (shown as Ts in Figure 3) can be detected as 1-count pixels, or pixels with a single occupied neighbor. Connecting nodes (shown as Cs in Figure 3), or nodes where lines join, can be detected as patterns that have three or more transitions, indicating three lines emanating from the node. A special case is the "+" (pattern group 31), that is a connector, even though the gap value is one (that is, no gap would be created if the center pixel were removed). Multiple connecting nodes (shown as Ms in Figure 3), or node clusters, make it difficult to properly locate the intersection of lines. Unless reduced to a single point, multiple connectors may also create a web of short interconnected arcs during vector generation. Rosenfeld and Kak (1982) suggest that one solution is to perform a second pass through the image, testing for adjacent connectors, and calculating a single point (a centroid) that can be used as the connector.

## CREATION OF DIGITAL LINE GRAPH (DLG)

With nodes marked, the process of converting linework into vector arcs is relatively simple. We begin on an endpoint or connector node and simply follow the lines that are guaranteed to have only one path to follow until another node is found. To avoid following a line twice, the pixels are marked as their coordinates are added to the line. This approach works well when the entire image can be resident in computer main memory, or in an image display memory, as in the case of the system used for this study.

The format for the resulting lines, chains, or arcs may be expressed in relative coordinates (for example, below, above, and left), or maybe converted to absolute measure (for example, Cartesian coordinates). the USGS Digital Line Graph (DLG) stores absolute coordinates, usually in map units expressed as inches, or in ground units in meters. For this project, the DLG format was chosen because several available systems read and write data in this form.

## VECTOR OPERATIONS

### POINT REDUCTION

Linework that has been converted to vectors is made up of short line segments that have a length of one pixel unit for four-way adjacent neighbors, or of 1.414 units for diagonal neighbors. Straight lines can be reduced by eliminating intermediate points on the line. In addition, lines may be further reduced by using a generalization method such as described by Douglas and Peucker (1973). For the examples used in this study, point

on the exterior edges of the linework. This may sometimes be appropriate, especially because subsequent thinning is performed from the line edge inward and is greatly affected by a rough line edge. Figure 4c shows how filling can be used to eliminate small holes or void inclusions in drainage lines that have been extracted from a DEM. The rules for a filling operation are shown in Figure 3, and are labelled Y for surround conditions that indicate that a center cell should be set from zero (no line) to one (line).

### THINNING OPERATIONS

Several alternatives exist for thinning rasterized linework to create a single connected line (often referred to as a line skeleton). Most are iterative, and several use a two-step processing approach. One method performs the bulk of the thinning in a first step with a second step that eliminates all but the single connected lines (Landy and Cohen, 1985). The decision tables for thinning in the first step must not allow thinning to less than two pixels wide. This is to ensure that gaps are not formed by thinning on both sides of a line during the same pass. A similar approach, described by Rosenfeld and Kak (1982), also requires two steps. Their method alternately thins first the top and left sides of the linework and then the bottom and right sides, in order to avoid the creation of gaps.

Two thinning rules are shown in Figure 3, each containing two steps. The first is the method used in processing the example data sets, and is more liberal in its thinning than the second method, labelled Landy. Both are included to show that there is no single set of correct rules for all data sets. In general, thinning can be performed on all patterns where gaps would



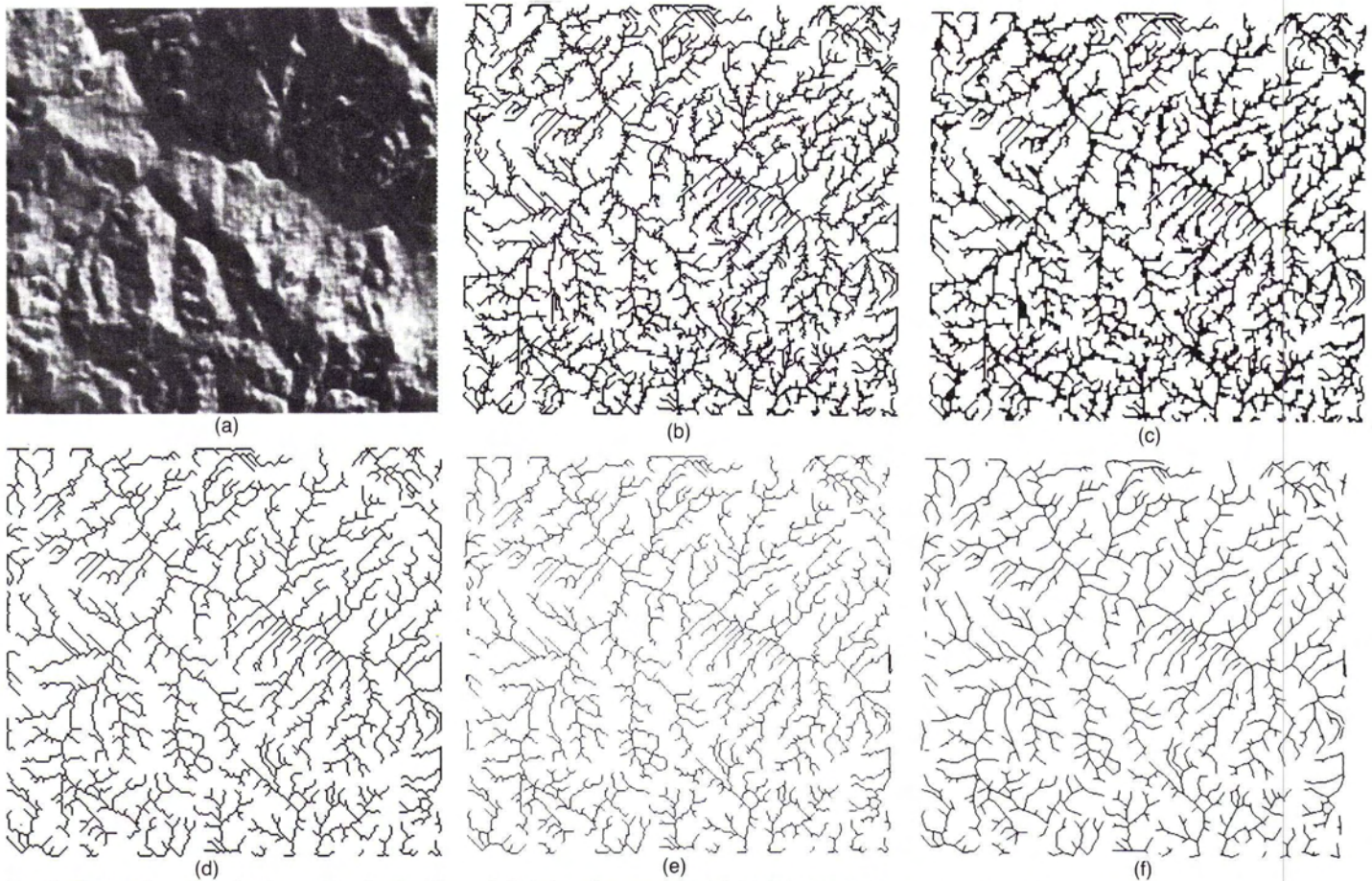


FIG. 4. Processing steps for linework extracted from digital elevation models. (a) Digital elevation model displayed as a shaded relief image. (b) Raw linework extracted from elevation model. (c) Linework after filling operation. (d) Linework after two-step thinning process. (e) Linework after conversion to vector format. (f) Vector linework after bicubic spline smoothing.

reductions of 50 to 80 percent were possible with little perceivable difference in the quality of the lines.

#### LINE SMOOTHING

Point reduction does not reduce the appearance of sharp jagged arcs that show remnants of their raster derivation. In fact, point reduction, when used exclusively, will remove blunt angles and leave sharper angles in the lines. Sharp angles that exist at points of inflection can be treated by fitting a curve through the set of points. This will introduce new points but will also yield arcs with a more acceptable appearance. One such technique is cubic spline interpolation (described by Monmonier (1982)). In Figure 4f, spline interpolation has been performed on the drainage lines with the result showing a smoother appearance. In Figure 5f, spline interpolation has been performed and, in addition, short arcs have been deleted. In this case, the short arcs that were removed were largely made up of erroneous spurs (generated in the thinning process) and bridges between lines.

#### SUMMARY AND CONCLUSIONS

An examination of the results of these processing techniques on the sample data sets indicates that many errors remain in the linework. This points out the necessity for a good interactive editing facility. Most noticeable are several breaks in the lines and erroneous connections between contours. Both sample data sets required considerable interactive editing after conversion to vector form. This will be true for most lines of the type we create from the DEMs or for lines scanned from maps which have not been specially prepared.

This study was conducted to convert rasterized lines to vector arcs in a format appropriate for geographic information system

processing. This was accomplished by assembling several image-processing and vector-editing techniques into a flexible but incomplete prototype system. The organizing and documenting of surround conditions may have a much broader application than that required for this project, and it is felt that the use of pattern groups can provide an efficient and consistent framework for many neighborhood operations that are performed in the raster domain. The experiences gained in this project have also helped to objectively characterize the steps required in raster-to-vector processing. An understanding of these processes will become even more valuable as scanning systems become more affordable and turn-key systems are better tailored to meet specific requirements.

#### REFERENCES

- Douglas, D.H., and T.K. Peucker, 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, v.10, pp. 112-122.
- Golay, M.J.E., 1969. Hexagonal parallel pattern transformations. *IEEE Transactions on Computers*, v.C-18, no.8.
- Jenson, S.K., 1985. Automated derivation of hydrologic basin characteristics from digital elevation model data. *Proceedings of Auto-Carto VII*, 1985, Washington, D.C.
- Landy, M.S., and Y. Cohen, 1985. Vectorgraph coding: Efficient coding of line drawings. *Computer Graphics and Image Processing*, v.30, pp. 331-344.
- Moik, J.G., 1980. *Digital processing of remotely sensed images*. NASA SP-431, pp. 130-140.
- Monmonier, M.S., 1982. *Computer assisted cartography*. Prentice Hall, pp. 108-110.
- Nichols, D.A., 1981. Conversion of raster coded images to polygonal



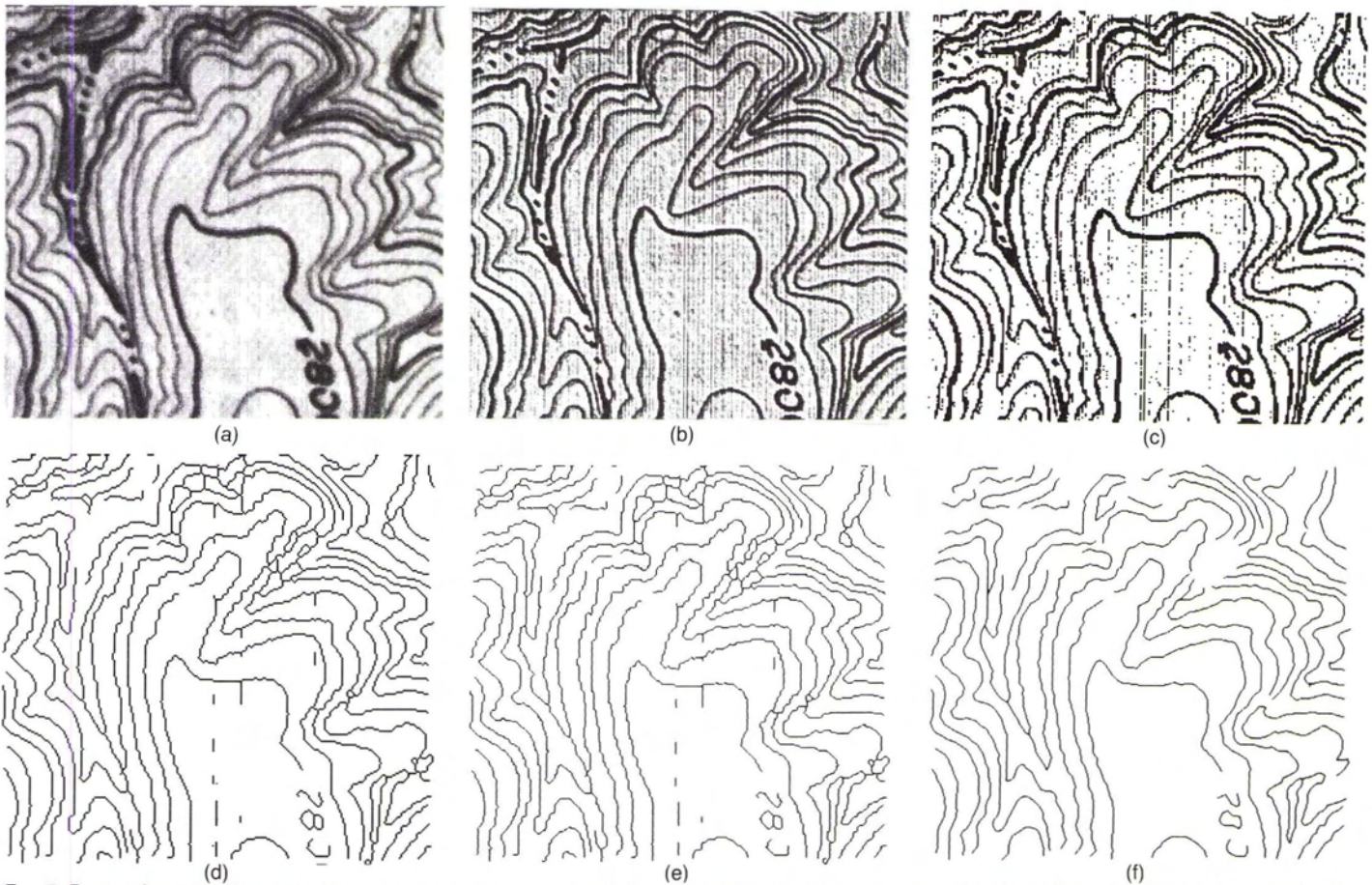


FIG. 5. Processing steps for scanned topographic linework. (a) Portion of scan-digitized topographic map. (b) After edge enhancement by convolution filter. (c) Linework after thresholding to a bilevel (binary) image. (d) Linework after filling and two-step thinning. (e) Linework after conversion to vector format. (f) Linework after splining and deletion of short arcs.

data structurers. *Proceedings of Pecora VII Symposium*, 1981, Sioux Falls, South Dakota.

Pavlidis, T., 1980. A thinning algorithm for discrete binary images. *Computer Graphics and Image Processing*, Vol. 18, pp. 142-157.

—, 1982. *Algorithms for Graphics and Image Processing*. Computer Science Press, Chapters 7-9, pp. 128-214.

Peuquet, D.J., 1981. An examination of techniques for reformatting digital cartographic data, part 1: the raster-to-vector process. *Cartographica* 18, vol. 1, pp. 34-48.

Rosenfeld, A., and A.C. Kak, 1982. *Digital Picture Processing* [Second Ed.], Academic Press, v. 2, pp. 232-240.

## BOOK REVIEW

*Computational Models in Surveying and Photogrammetry*, by B. D. F. Methley. The Blackie Publishing Group, Wester Cleddens Road, Bishopbriggs, Glasgow G64 2NZ, U.K., 1986. 346 pages, 137 figures, hard cover. Price \$70 (£40.00, U.K.).

ACCORDING TO THE PREFACE, *Computational Models in Surveying and Photogrammetry* is designed for the college or university student considering the topic for the first time or the practitioner who needs help with software design. Before reaching conclusions with respect to the validity of this statement, consider the content of this volume.

The book is subdivided into three major parts which are focused on (1) surveying (Chapters 2, 3, 4); (2) photogrammetry (Chapters 5, 6, 7); and (3) statistical concepts and application of adjustments to problems in surveying and photogrammetry (Chapters 8, 9, 10). The basic principles of surveying and photogrammetry are covered, very briefly, in the first two portions, with the major emphasis here being given to rigorous development of the mathematical models associated with standard

problems in these areas, and in each case followed by examples of traditional approximate adjustments particularly in surveying and to a lesser degree in photogrammetry. The third portion contains rigorous development of statistical concepts and least-squares estimation applied to problems in surveying and photogrammetry. Appendices A, B, and C contain detailed derivations for linearization using Taylor's theorem; coefficients for the linearized collinearity equations; and useful, general purpose mathematical formulae.

Chapter 2, titled "Mathematical Models in Surveying," provides brief reviews of the basic concepts involved in coordinate systems, rectangular coordinates, angles, directions, bearings, and height differences. Equations (the mathematical models) are derived for computation of coordinates, triangulation inter-