Watershed Delineation with Triangle-Based Terrain Models

Article *in* Journal of Hydraulic Engineering · October 1990

DOI: 10.1061/(ASCE)0733-9429(1990)116:10(1232)

Norman L Jones

Brigham Young University - Provo Main Campus

engineering." *Comput.-Aided Design*, 19(4), 171–187.
Silfer, A. T., Kinn, G. J., and Hassett, J. M. (1987). "A geographic information system utilizing the triangulated irregular network as a basis for hydrologic modeling." *AutoCarto 8, Proc. 8th Int. Symp. on Comput.-Assisted Cartography*, N. R. Chrisman, ed., American Society for Photogrammetry and Remote Sensing/American Congress on Surveying and Mapping, 129–136.
Watson, D. F. (1981). "Computing the n-dimensional Delauney tesselation with application to Voronoi polytopes." *The Comput. J.*, 8(2), 167–172.
Watson, D. F., and Philip, G. M. (1984a). "Systematic triangulations." *Comput. Vision, Graphics, and Image Processing*, 26(2), 217–223.
Watson, D. F., and Philip, G. M. (1984b). "Triangle based interpolation." *Math. Geology*, 16(8), 779–795.

Watershed Delineation with Triangle-Based Terrain Models

By Norman L. Jones,[1] Stephen G. Wright,[2] and David R. Maidment,[3] Members, ASCE

Abstract: An algorithm is presented for tracing the path of steepest descent from a given starting point on a terrain model defined by a triangulated irregular network. This algorithm is then extended to solve several problems. The flow patterns for a site are generated by tracing flow paths from a large number of starting points. The approximate stream network or channel network is found by tracing the channels upstream from pits or exit points. Once the stream network is found, the source areas or contributing areas for each of the sections of the stream are delineated. The source areas are then used to delineate the watersheds of selected nodes in the channel network. The format of a possible data structure for the channel network is presented along with pseudocode examples for traversing the channel network and the source areas. Execution times on a desktop computer are presented along with suggestions for optimal use of the algorithms.

Introduction

The use of computers for design has created new opportunities to employ relatively sophisticated, accurate models to represent terrain surfaces in electronic form. These models are used to generate contour maps and design drawings, to compute earthwork quantities, and to support simulation and design programs. Computer-based terrain models can also be used to automate the delineation of drainage patterns, stream networks, source areas, and watersheds.

Representions of land surfaces typically involve one of three schemes: (1) A set of points, randomly scattered over the surface connected to form a triangulated irregular network (TIN), that results from a land survey of a building site; (2) a rectangular grid of points, as contained in data from the digital elevation model (*Digital* 1987); or contour strings, as presented in topographic maps. The fundamental goal of this paper—namely, the automated determination of drainage paths and source areas—requires different algorithms for each of the three representations.

Jensen and Domingue (1988) have presented algorithms for determining drainage direction on rectangularly gridded surfaces. These algorithms are hinged around the concept of a grid point being surrounded by eight neighboring points, four in the coordinate directions and four on the diagonals. Drainage from any given point is considered to occur downhill toward the lowest of the eight neighboring points. Source or watershed areas are determined for any grid square by counting the upstream grid squares whose drainage subsequently flows through the specified square.

[1]Grad. Res. Asst., Dept. of Civ. Engrg., Univ. of Texas at Austin, Austin, TX 78712.
[2]Ashley H. Priddy Centennial Prof., Dept. of Civ. Engrg., Univ. of Texas at Austin, Austin, TX.
[3]Assoc. Prof., Dept. of Civ. Engrg., Univ. of Texas at Austin, Austin, TX.
Note. Discussion open until March 1, 1991. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on October 27, 1989. This paper is part of the ***Journal of Hydraulic Engineering***, Vol. 116, No. 10, October, 1990. ©ASCE, ISSN 0733-9429/90/0010-1232/$1.00 + $.15 per page. Paper No. 25132.

Moore et al. (1988) have analyzed terrain represented by digitized contour maps. Their algorithms are based on modeling the terrain as an approximately orthogonal network, comprising connected quadrilateral elements whose ends coincide with adjacent contour lines and whose sides are lines of steepest descent between the contours. Drainage flows down the surface from one quadrilateral to its neighbor below, bounded by the continuation of the lines of steepest descent. Moore et al. (1988) also discuss the quantification of subsurface drainage rates and other physical variables based on their surface representation.

While the foregoing algorithms are applicable for planning purposes over fairly large areas, most engineering design for site plans or other forms of construction is preceded by a site survey from which a set of scattered (x,y,z) coordinate points is obtained that capture the essential features of the terrain. These points are joined by a triangulated irregular network from which a contour map is generated if needed. It is more effective, however, to work directly from the TIN model rather than contours for determining drainage directions, stream networks, and watershed source areas, as shown in this paper.

Several previous investigators have studied drainage on a TIN model. Silfer et al. (1987) developed a kinematic wave model for flow hydrographs based on partitioning the flow across the triangles. Grayman et al. (1975, 1979, 1982) described a similar concept. These authors do not address the issue of how to define the watershed source area from any point on the stream network.

Triangle-Based Terrain Models

General terrain surfaces may be represented by a series of discrete points like those shown in Fig. 1(*a*). Each point is characterized by its x, y coordinates in plan and an elevation, z. The points could represent either measurements taken directly by various land-surveying techniques, including aerial photogrammetry, or they could be points that were digitized from an existing contour map. It is assumed that each pair of x, y coordinates is unique, i.e., there are no duplicate points or points with the same x, y value but different z values. It is possible to connect the x, y projections of the points with edges as shown in Fig. 1(*b*), thus forming a triangulated irregular network.

Triangle-based terrain models have certain advantages over grid-based models. Triangle-based models provide a more accurate representation of the original survey data. Grid-based techniques often involve interpolating from the original data points to points on a rectangular grid. Much of the detail and information present in the original data set is lost in the interpolation process. Moreover, the boundaries of a triangulation can conform to any shape desired, and because the triangles can represent planes or facets, the network of triangles can be thought of as a continuous piecewise linear interpolant of the scattered data points. The triangulation can be quickly contoured by linearly interpolating across the triangles. If smoother contours are desired, polynomial surface patches can be formulated on each triangle with slope continuity across triangle boundaries (Akima 1978; Franke and Nielson 1980; Lawson 1977; McCullagh 1981; Watson and Philip 1984b).

It can be readily imagined that there are many ways that x, y projections

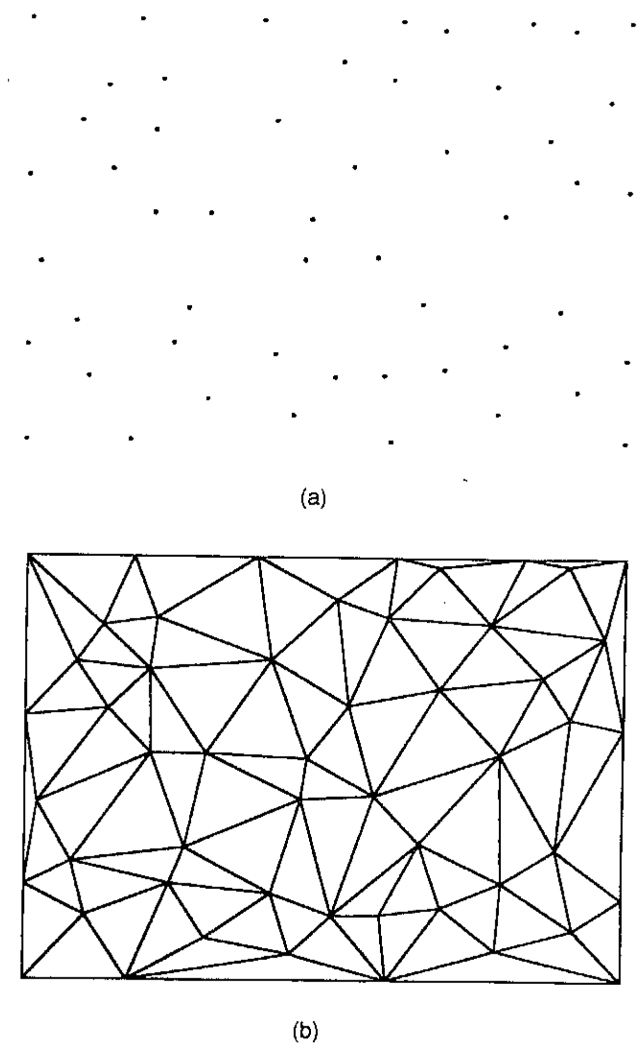

FIG. 1. Set of Scattered Data Points: (*a*) Plan View; and (*b*) Corresponding Delauney Triangulation

of the set of data points in Fig. 1(*a*) could have been connected to form a triangulation. Various schemes have been devised in order to get some degree of repeatability or uniqueness in the triangulation and to automate the triangulation process. One of the best schemes for this purpose is the Delauney triangulation scheme (Lee and Schacter 1980; Watson 1981; Watson and Philip 1984a). The Delauney triangulation is based on the criterion that when three points are joined to make a triangle, the circumcircle of that triangle (the circle drawn through its vertices) shall not encompass any of the other data points (Fig. 2). The effect of this criterion is to maximize the

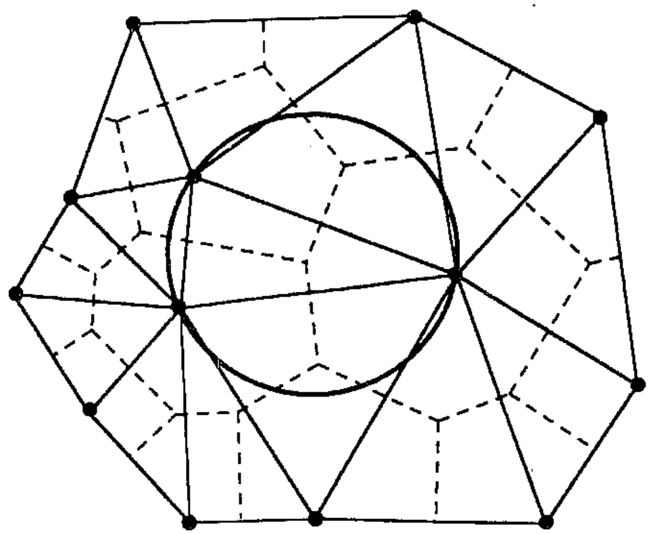

FIG. 2. Graphic Representation of Delauney Criterion for Triangulation and Underlying Thiessen Polygon Network

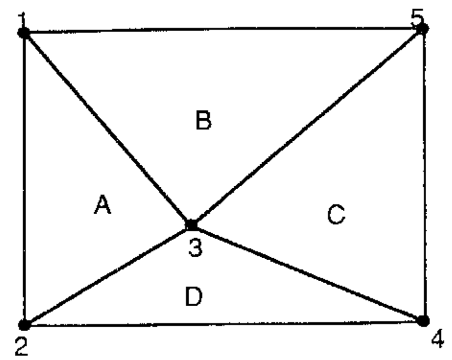

FIG. 3. Sample Triangle Network

minimum interior angle of the triangles so formed, thereby avoiding long thin triangles and connecting points that are the nearest neighbors in the data set. The Delauney triangulation is the dual of the Thiessen polygon network, which can be computed from the same set of points (Fowler and Little 1979). The centroids of the circumcircles correspond to the points of intersection of the perpendicular bisectors of the triangle edges (i.e., the corners of the Thiessen polygons).

Several schemes exist for representing the triangle network in the computer. A scheme that is convenient for the algorithms discussed in this paper is illustrated in Fig. 3 and Tables 1 and 2. The triangulation is represented by a list of vertices and a list of triangles. (Note: The lists could be imple-

TABLE 1. Vertex List for Sample Triangle Network Shown in Fig. 3

Vertex (1)	x (2)	y (3)	z (4)
1	0.0	12.0	2.0
2	0.0	0.0	3.0
3	5.0	4.0	4.0
4	12.0	0.0	4.0
5	12.0	12.0	3.0

TABLE 2. Triangle List for Sample Triangle Network Shown in Fig. 3

Triangle	Vertices			Adjacent Triangles		
(1)	V1 (2)	V2 (3)	V3 (4)	A1 (5)	A2 (6)	A3 (7)
A	1	2	3	nil	D	B
B	1	3	5	A	C	nil
C	3	4	5	D	nil	B
D	2	4	3	nil	C	A

mented as linked lists, arrays, or some other data structure. Some of the issues related to triangle data structures are to be discussed in a forthcoming paper by the two senior writers. Throughout this paper, the term *pointer* will refer to a data item that points to an individual item for access. In the case

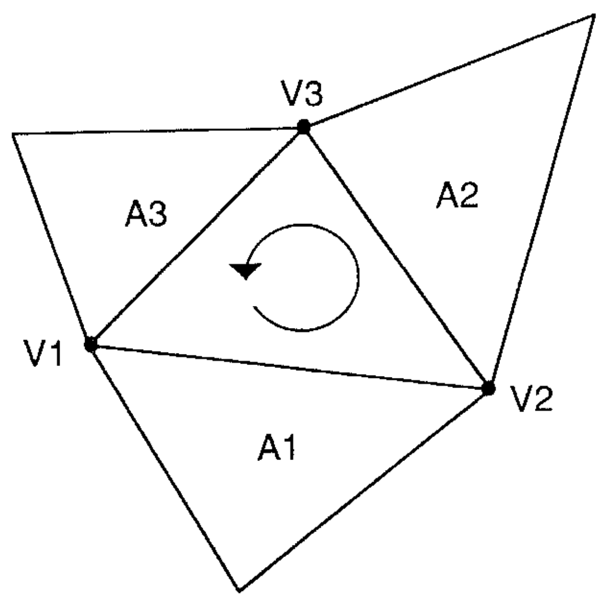

FIG. 4. Relationship of Vertex and Adjacent Triangle Numbering Schemes

of arrays, the pointer is simply an index of the array. In the case of linked lists implemented in Pascal or C, the pointer is an address of the location in memory of the data item being targeted.) The vertex list contains the x, y, z values of the vertices, and the triangle list contains the data defining the triangles. Each triangle is defined by a list of pointers to its three vertices and a list of pointers to the three triangles that surround the triangle. The list of triangle pointers is ordered in the same direction as the list of vertex pointers (counterclockwise). The vertex index is related to the adjacent triangle index as shown in Fig. 4. Thus, adjacent triangle i is the triangle opposite the edge defined by vertex i and the next vertex from i, based on a counterclockwise traversal of the vertices. Computationally the next vertex from vertex i is expressed as vertex (i mod 3) + 1.

Paths of Steepest Descent

The piecewise linear nature of the triangle-based model makes it an efficient base for the computation of paths of steepest descent. As water begins its flow across a surface, the initial direction of flow will depend on the gradient of the surface and the surface roughness. If the roughness is not significantly different in different directions of possible flow, the initial path of flow will be in the direction of steepest descent. As the flow proceeds across the surface, subsequent directions of flow will depend on both the gradient at any point and the momentum. As long as the momentum is negligible, the direction of flow will be governed primarily by the direction of steepest descent, which is the case in most natural flow over watersheds. Accordingly, paths of steepest descent can represent a good approximation of flow paths over a surface and will precisely indicate the directions in which flow will be initiated over a surface with homogeneous roughness.

Paths of steepest descent are constructed from any arbitrary point on a triangulated terrain surface by following the path of maximum gradient (Fig. 5). Location of the path of steepest descent is initiated by starting at a selected point in a triangle (point a) and computing the path of maximum gradient across the triangle from the starting point to the edge of the initial triangle (path segment ab). For planar triangles computation of the direction of the path of steepest descent across the triangle is quite simple. The equation of the plane defined by the three vertices of the triangle is as follows:

$$Ax + By + Cz + D = 0 \quad (1)$$

where A, B, C, and D are computed from the coordinates of the three vertices (x_1, y_1, z_1), (x_2, y_2, z_2), and (x_3, y_3, z_3):

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \quad (2a)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \quad (2b)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \quad (2c)$$

$$D = -Ax_1 - By_1 - Cz_1 \quad (2d)$$

Eq. 1 can also be written as

$$z = f(x,y) = -\left(\frac{A}{C}x + \frac{B}{C}y + \frac{D}{C}\right) \quad (3)$$

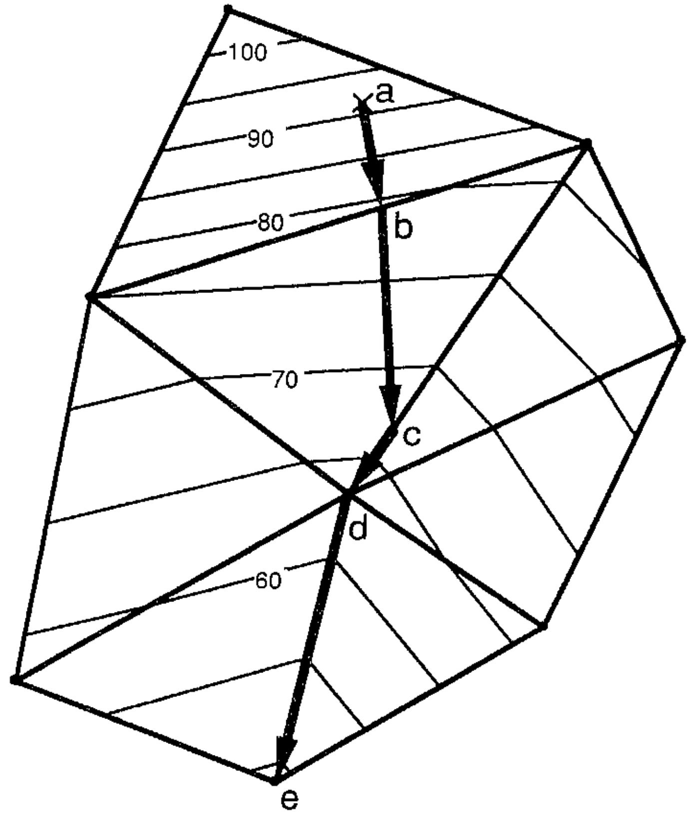

FIG. 5. Sample Path of Steepest Descent

which is the form of the plane equation used to compute the elevation at any point on the triangle. The gradient of a scalar function ∇f points in the direction of greatest increase in the function or in the direction of steepest ascent. The gradient of the function f of Eq. 3 is computed as follows:

$$\nabla f = \frac{\partial f}{\partial x}\hat{\imath} + \frac{\partial f}{\partial y}\hat{\jmath} \quad (4a)$$

$$\frac{\partial f}{\partial x} = -\frac{A}{C} \quad (4b)$$

$$\frac{\partial f}{\partial y} = -\frac{B}{C} \quad (4c)$$

The direction of steepest descent is opposite to the direction of steepest ascent or $-\nabla f$. Thus, the direction of steepest descent is

$$-\nabla f = \frac{A}{C}\hat{\imath} + \frac{B}{C}\hat{\jmath} \quad (5)$$

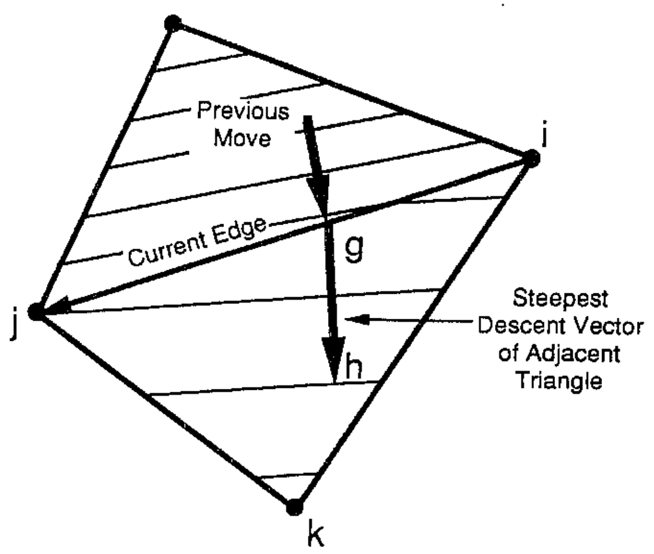

FIG. 6. Data Used to Compute Next Move from Edge

Eq. 5 describes the direction of the path of steepest descent, expressed as the projection onto the xy plane. The path of steepest descent, the gradient vector, and the normal to the plane all lie in the same vertical plane. The endpoint of the first linear segment of the path of steepest descent (point b) is found by projecting the line defined by the direction of steepest descent and the point where flow was initiated, then locating which edge it intersects. This completes the calculation of the first segment, ab, of the path of steepest descent shown in Fig. 5.

Once the point on the edge of the triangle is located, the gradient across the adjoining triangle is tested to determine if the path of steepest descent should continue across the adjacent triangle (overland flow) or along an edge (channel flow). Since the data structure for the triangles contains the adjacency information, no searching needs to be done through the data to find the adjacent triangle. This becomes an important factor with very large triangulations. If the adjacent triangle slopes downward from the edge, the flow is overland. For overland flow, this process is repeated for the adjacent triangle to locate an opposite edge where the path of steepest descent will exit. If the adjacent triangle slopes toward the edge, flow is considered to be along the edge; in that case, the path of steepest descent then continues along the edge toward the vertex with the lowest elevation. The method illustrated in Fig. 6 can be used to determine whether the adjacent triangle slopes toward the current edge or away from the current edge. The cross product of the steepest descent vector of the adjacent triangle, **gh**, and the vector defined by the current edge based on a counterclockwise traversal of the adjacent triangle, **ij**, is computed. If the result of the cross product **gh** $\times$ **ij** is positive (out of the page), the adjacent triangle slopes towards the current edge. If the result of the cross product is negative, the adjacent triangle slopes away from the current edge. For the example shown, the adjacent triangle slopes

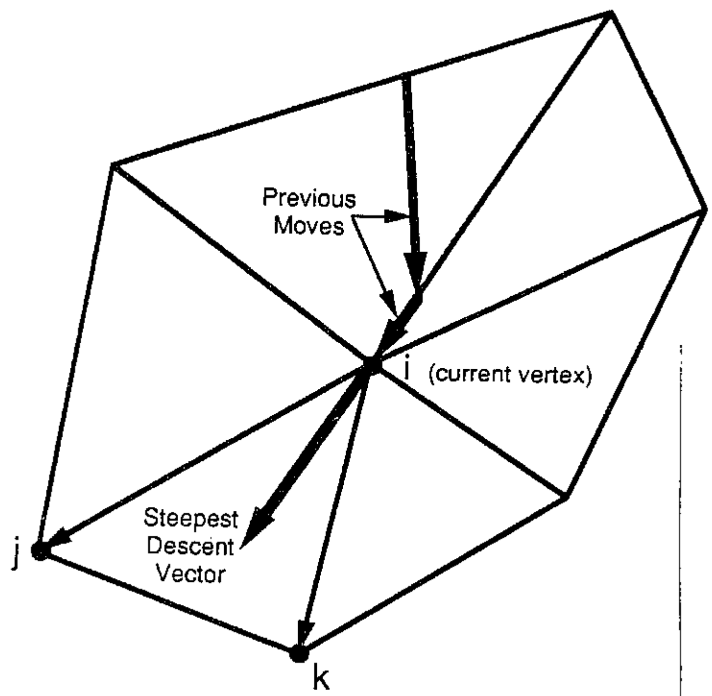

FIG. 7. Data Used to Compute Next Move from Vertex

away from the current edge and the path of steepest descent is tracked across the adjacent triangle using the procedures described earlier (path segment bc in Fig. 5). Upon reaching the next edge, however, the adjacent triangle slopes towards the edge and the path of steepest descent continues along the edge to the vertex with the lower elevation (path segment cd).

At the vertex corresponding to point d in Fig. 5, the gradients of the surrounding triangles are tested to determine if the path of steepest descent should continue along an edge or across the face of an adjacent triangle. Both the edges and the triangles adjacent to the vertex are traversed counterclockwise and tested one at a time. To test an edge, the steepest descent vectors of the triangles on either side of the edge are tested using a cross product to determine if both triangles slope towards the edge. If both triangles slope towards the edge (i.e., if the edge lies in a channel) and if the edge slopes away from the vertex, the path of steepest descent continues along that edge. To test a triangle, the steepest descent vector for the triangle based at the current vertex is tested to determine if it lies between the two adjacent edges as shown in Fig. 7. If the cross product of the steepest descent vector and the vector **ij** is negative, and the cross product of the steepest descent vector and the vector **ik** is positive, then the steepest descent vector lies between the two edges, and the path of steepest descent continues across the face of the triangle being tested. If either cross-product test fails, then the path of steepest descent will not continue across the face of the triangle.

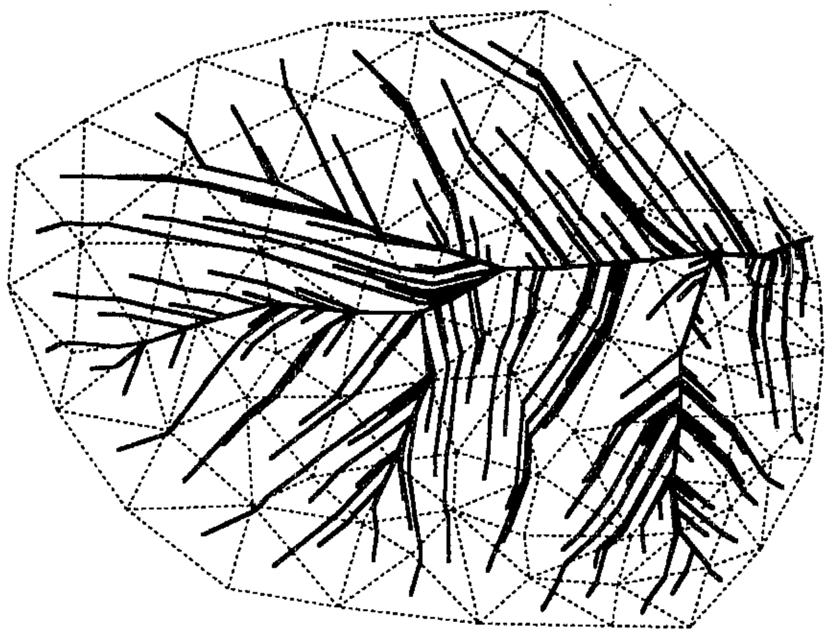

FIG. 8. Lines of Steepest Descent Initiated at Centroid of Each Triangle

If all of the adjacent edges slope towards the vertex, the vertex lies in a pit, and the path of steepest descent is terminated. For the example shown in Fig. 5, the path of steepest descent continues along an edge (path segment de). Tracing the path of steepest descent continues in this manner until either the outer boundary of the triangulated terrain region is reached or a pit is encountered.

Paths of steepest descent are useful for analyzing a terrain model. By selecting a point on the surface and having the computer plot the path of steepest descent from that point, the user can obtain insight into the potential drainage patterns of a particular area of the terrain. By automatically initiating paths of steepest descent at the centroid of all of the triangles on the terrain surface, the pattern of flow becomes evident and areas where streams will develop are suggested by a convergence of overland flow paths into flow paths along edges (channels). A typical set of paths of steepest descent for a triangulated terrain surface is shown in Fig. 8.

Channel-Flow Lines (Stream Network)

Paths of steepest descent coinciding with edges of triangles and representing channel flow suggest a stream network (Fig. 8). However, the lines of channel flow are dependent to some extent on the arbitrary points where the paths of steepest descent were initiated, and they may not include all channel-flow paths. One way to compute the complete channel network would be to test each of the edges in the model and mark those that lie in channels. This would produce a set of edges representing the network. The set would

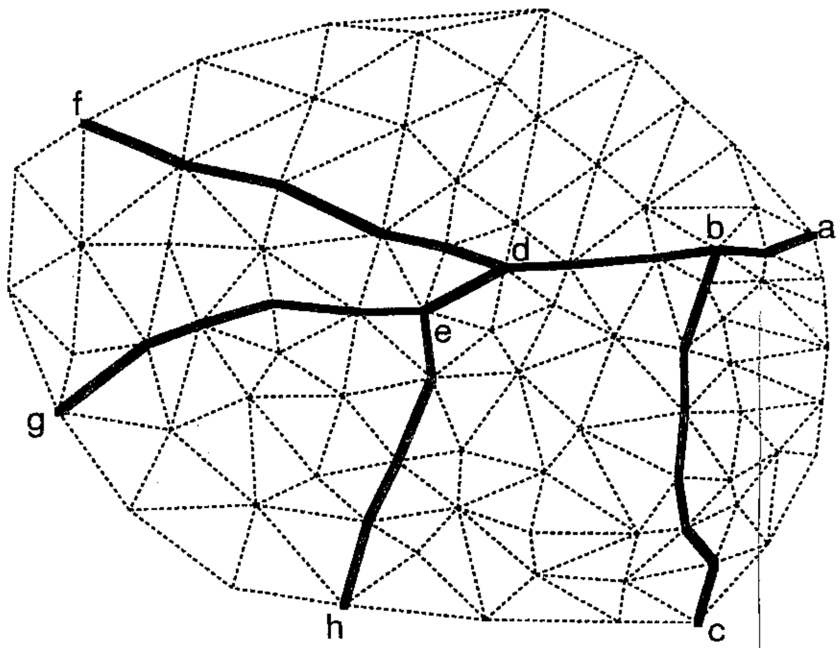

FIG. 9. Channel Network of Terrain Model Shown in Fig. 8

be more useful if the edges were structured into an ordered network with the upstream and downstream connections between the edges explicitly defined. The set could be sorted, but sorting can become time-consuming with a large number of edges. The complete channel network corresponding to the approximate stream network can be determined more directly by employing the set of algorithms described earlier for computing the path of steepest descent in a slightly different manner.

Lines of channel flow are generated by tracing the channels in reverse from their terminal points to their uppermost reaches. The process is initiated by traversing the vertices on the outer boundary of the terrain model. At each vertex on the boundary, the edges terminating at the vertex are checked to determine if any edge lies in a channel, i.e., if the two adjoining triangles slope toward the edge. If an edge lies in a channel and the channel slopes upward (ascends) from the boundary, the edge is traced upward to the next vertex. At the next vertex, the surrounding edges are again checked to locate edges that form channels and slope upward from the vertex. Channel-flow lines are then traced along those edges that constitute channels up to the next vertex and the process is repeated. At some vertices the channel lines will split along more than one edge, and they will need to be traced along several edges. The tracing of a given channel is terminated when either a peak or the outer boundary of the terrain model is encountered. This process is repeated for all channels located on the outer terrain boundary. Additional stream channels can also be located at vertices that make up pits. Such vertices are then used to trace additional channels upstream. The set of channels

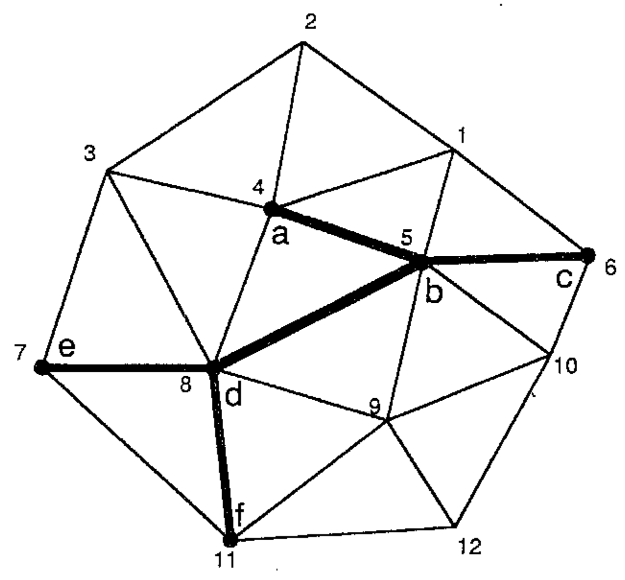

FIG. 10. Sample Channel Network

determined for the terrain of Fig. 8 is shown in Fig. 9.

The logical representation of a data structure convenient for representing the channel network in the computer is shown in Fig. 10 and Table 3. The channel network is represented by a set of channel nodes, each with a pointer to the corresponding vertex in the terrain model, a pointer to the downstream node, and a list of pointers to upstream channel nodes. A separate pointer is maintained to the terminal node of the network (the node with no downstream pointers). For a given terrain model, there may be several channel networks, each with its own terminal node. If a programming language that supports recursion (such as Pascal or C) is used, very simple recursive algorithms can be written to traverse the channel network upstream from its terminal node. For example, to draw the network, the following algorithm

TABLE 3. List of Stream Nodes and Pointers Used to Represent Sample Channel Network Shown in Fig. 10

Node (1)	Vertex (2)	Downstream (3)	Upstream (4)
a	4	b	—
b	5	c	d, a
c	6	—	b
d	8	b	e, f
e	7	d	—
f	11	d	—

(written in pseudocode) could be used:

```
procedure Draw_Network (Node)
  For each UpstreamNode from Node do
    Move_to(Node) {Move the current pen position to Node} . . . . . . . . . . . (6)
    Draw_to(UpstreamNode)
    Draw_Network(UpstreamNode)
```

The initial procedure call would be made using the terminal node of the network. The algorithm proceeds by drawing one of the edges upstream from the current node, then drawing one of the edges upstream from the upstream node, etc., until a node with no upstream nodes is reached. The recursive process then backs down the stream and draws the branches corresponding to other upstream nodes. For example, the network shown in Fig. 10 would be drawn in the following order: cb, bd, de, df, and ba.

Delineation of Source Areas

It is often necessary to delineate contributing source areas in order to indicate watersheds or compute the quantities of flow that might occur in a stream system or channel network such as the one shown in Fig. 9. Source areas represent the area adjacent to a section of the stream from which overland flow would come directly to that section of the stream. Each stream section has two contributing source areas—one on each side of the stream.

A stream section can be defined in several ways. Through the remainder of this paper it will be assumed that each portion of the channel network between branching nodes or between a branching node and a terminal node or an uppermost node in the network is considered to be a stream section. The nodes defining the stream section will be referred to as delimiting nodes since they delimit stream sections. A stream section defined in this way could include several sequential triangle edges or linear channel segments. For example, the delimiting nodes for the channel network of Fig. 9 are a, b, c, d, e, f, g, and h, and the stream sections are ab, bc, bd, de, eh, eg, and df. In practice, extra intermediate delimiting nodes could be added and deleted at selected points on the channel network between the default delimiting nodes if desired. Determination of potential source areas can be accomplished using the following three-step process.

Step 1: Calculating Channel Network

The first step in the delineation of source areas consists of establishing the channel network. A channel network like the one shown in Fig. 9 is established using the procedures described in the previous section.

Step 2: Delineating Drainage Boundaries and Subdivision of Triangles

To simplify subsequent steps, it is necessary to be able to represent a source area with a set of triangles. However, some triangles straddle the boundaries between source areas and thus lie in more than one source area. In other words, a given triangle may contribute to the flow in more than one stream section, depending on where the flow was initiated in the triangle. The purpose of the second step is to identify such triangles and subdivide

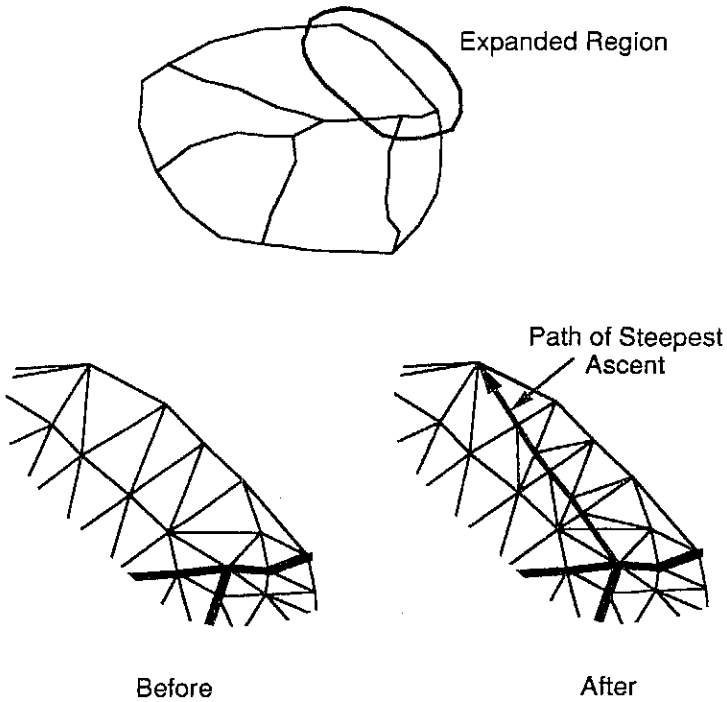

FIG. 11. Terrain Model Showing Triangulation before and after Subdivision Process of Step 2

them so that each triangle contributes directly to the flow in only one stream section.

The triangle subdivision is accomplished by beginning at each delimiting node and tracing paths of steepest ascent. The path of steepest ascent from a delimiting node corresponds to the boundary between two source areas. At least two paths, one on each side of the channel, are traced from the node; three paths are traced if the node coincides with a point where the channel splits into two branches. Tracing paths of steepest ascent involves the same algorithm employed to trace the paths of steepest descent, except that the paths of maximum upward gradient rather than maximum downward gradient are traced, i.e., Eq. 4 is used rather than Eq. 5 to find the direction of the path. As the path of steepest ascent is traced uphill, the triangles intersected by the path are subdivided so that the new triangles lie entirely within one source area or the other (Fig. 11). A subdivided triangle typically consists of one triangular area and one quadrilateral area. The quadrilateral area is subdivided into two triangles, so that the entire terrain surface consists only of triangles. This is necessary to maintain compatibility with the algorithms and data structures. Tracing the path of steepest ascent is continued until either a peak or the boundary of the terrain model is reached. If a ridge is encountered before a peak or a boundary, the triangle edges along the ridge are traced upstream. If the ridge ends before a peak or bound-

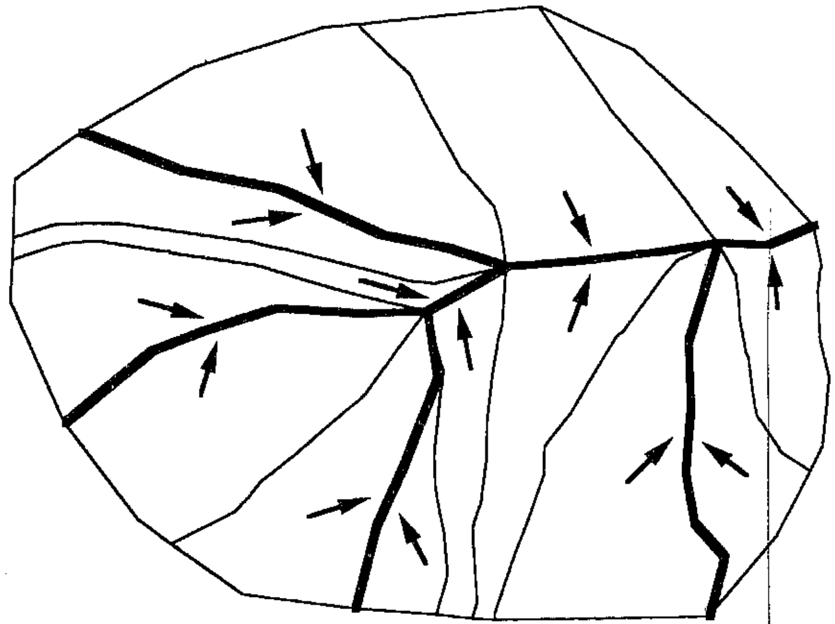

FIG. 12. Completed Source Areas

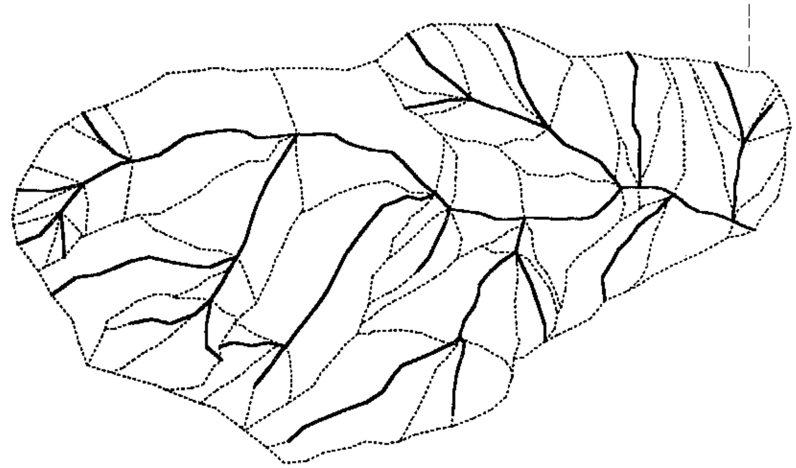

FIG. 13. Channel Network and Corresponding Source Areas for More Detailed Terrain Model

ary is encountered, the path of steepest ascent is once again traced across triangles, splitting the triangles as explained earlier.

Step 3: Grouping Triangles into Source Areas

Once the triangles are subdivided as necessary in step 2, the triangles are grouped according to source area. To group the triangles, paths of steepest descent are initiated at the centroid of each triangle and followed to the point where they first reach a channel segment. (Although paths of steepest descent are initiated at the centroid of each triangle, the path from any point on the triangle will arrive at the same channel segment because of the subdivision process described earlier.) The triangle in which the path of steepest descent is initiated is then added to a list of triangles that "drain" to a certain side of the channel segment encountered. The list of triangles represents the source area for that segment. The source areas computed in this manner for the channel network of Fig. 9 are shown in Fig. 12. The source areas for a more complicated terrain model are shown in Fig. 13.

WATERSHEDS

Once source areas are calculated, they can be used to delineate watersheds. The watershed corresponding to a node in the channel network is the area of the terrain model that could potentially contribute to flow through the node. The watershed for a delimiting node is simply the sum of the source areas upstream from the node. Once the source areas have been computed, simple algorithms can be written to find the watershed corresponding to a given delimiting node by recursively traversing the source areas upstream from the node. For example, to color or fill the watershed above a delimiting node in the channel network, the following algorithm could be used:

```
procedure Color_Watershed(DelimitingNode)
  For each DelimitingNode upstream from DelimitingNode do
    For each source area associated with
        (DelimitingNode, UpstreamDelimitingNode) do    ........... (7)
      For each triangle in source area do
        Color_Fill_Polygon(Triangle)
    Color_Watershed(UpstreamDelimitingNode)
```

The algorithm draws the source areas for each of the edges directly above the node as well as the rest of the source areas above the node by making recursive calls on each of the upstream delimiting nodes. The initial procedure call would be made with the delimiting node whose watershed is desired.

PREPROCESSING

The writers' experience with the preceding algorithms has shown that there are certain special cases where care must be taken to preprocess the data and fix approximation errors in the data or ambiguities will arise and the procedures will run into difficulty. Four situations that need to be identified and corrected before executing the algorithm are as follows:

Diverging Streams

It is uncommon for an actual stream to split in two as it flows downstream. However, the approximations of a triangulated terrain model may cause channels to split due to local irregular features and inaccuracies used to create the model. The algorithms outlined earlier for computing paths of steepest descent and channel networks will find diverging streams, but the data structure described for representing the channel network assumes that channels do not diverge. One solution for this problem is to edit the model when diverging channels are found so that the channels do not diverge. This solution is applicable when it is obvious that the divergence is simply due to irregularities in the terrain model. Typically, only minor changes need to be made to the model to correct this problem. A more rigorous solution is to use a data structure that allows a stream node to have more than one downstream node.

Flat Triangles and Channels

Another special case that must be considered is flat triangles or flat channel edges. The path of steepest descent at such points is undefined. Perfectly flat surfaces or streams are not likely to exist in nature but may exist in the terrain model. Flat triangles occur frequently when the data points used to generate the triangulation have been digitized from a contour map and there are many sets of points (corresponding to contour intervals) with the same elevation. When flat triangles or flat edges are found they can be edited appropriately, or extra feature lines or break lines can be added and the points can be retriangulated. A break line is a piecewise line defined by vertices in the triangulation. No triangle edge is allowed to intersect the break line, i.e., the triangle edges are guaranteed to coincide with the segments of the break line (Lee and Schacter 1980; Petrie and Kennie 1987). Break lines are typically defined on all obvious stream channels and ridges. Proper use of break lines will minimize the occurrence of flat triangles.

Channel Flow to Edge Flow

Another ambiguity arises when flow changes from channel flow to overland flow as it passes through a vertex. The algorithm for computing the path of steepest descent can handle this situation without any problem. However, difficulties arise when defining a channel network because parts of the network become disjointed. Once again, it is not expected that this situation would occur frequently in nature, and it is often due to local errors in the model that can be rectified with minor changes in data-point elevations. When preprocessing the data, it helps to traverse the edges of the network and draw all of the edges that lie in a channel. These edges can be examined for disjoint sections of obvious channel networks. Once again, this problem is minimized with proper use of break lines.

Pits

Pits are to be expected in a natural piece of terrain. However, occasionally false pits occur in the terrain model due to the triangulation process or errors in data-point elevations. As mentioned, in terms of the terrain model, a pit is a vertex whose surrounding edges all slope upward from the vertex. Pits can be found and edited if necessary during the preprocessing stage.

TABLE 4. Execution Times for Source-Area Delineation

Terrain model (1)	Step 1: creation of channel network (sec) (2)	Step 2: subdivision of triangles (sec) (3)	Step 3: grouping into source areas (sec) (4)	Total execution time (sec) (5)
Model 1	0.53	1.37	5.52	7.42
Model 2	10.15	11.27	62.93	84.35

EXECUTION TIMES

The algorithms described in this paper were coded in Pascal and executed on an Apple Macintosh II computer. The following figures show execution times (in seconds) of the steps in the algorithm for finding the source areas of the two sample terrain models contained in this paper. Both of the models were constructed from points digitized off U.S. Geological Survey (USGS) contour maps. The first terrain model is the model depicted in Figs. 8–12. This model had 86 vertices and 147 triangles prior to execution of the algorithm. The second model is the one shown in Fig. 13. This model had 708 vertices and 1,332 triangles prior to execution of the algorithm. The execution times of the various steps already described are shown in Table 4. Upon completion of step 2's subdivision process, the first model contained 198 vertices and 361 triangles, and the second model contained 1,098 vertices and 2,088 triangles. Once the source areas have been calculated, delineation of the watersheds of selected delimiting nodes is extremely fast since it involves simply traversing the data structure as described. The watersheds of any of the delimiting nodes of the two models tested could be computed and drawn in 1–2 sec or less.

CONCLUSIONS

Triangle-based terrain models offer opportunities for generating graphic representations of potential water-flow patterns over terrain with relatively little computational effort. Paths of steepest descent representing lines of potential water flow can be computed and drawn. In addition, channel networks may be automatically generated from the triangle-based terrain model as an approximation of potential stream networks. Finally, source areas representing the potential areas of contributing flow into channel networks are readily delineated once the channel network has been determined. The source areas can be used to delineate watersheds for selected points on the channel network, or they can be output to a file with the channel network and used as input for other analysis programs. Very efficient computational algorithms and data structures have been developed and employed in the work presented in this paper to compute the paths of steepest descent, channel networks, source areas, and watersheds. These algorithms make it possible to perform all of the computations described in this paper within a small amount of time on a personal computer.

Because efficient computational algorithms can be developed for a triangulated surface model, the models are attractive for use in interactive design software. Points on a channel network may be selected interactively and contributing source areas may be delineated. Similarly, points of origin, such as the potential source of hazardous-liquids spillage can be selected and the potential path of drainage over the terrain surface identified. In the design of site grading, the geometry of the site can be altered and potential impacts on drainage over the area of the site can be seen. Almost all of these applications can be done interactively with almost immediate response even on a personal computer. Such immediate response from a model during design makes it possible for the designer to develop insight into the impacts of decisions on site drainage and create an optimum design for the site layout.

The algorithms described in this paper are also useful for generating the geometry required for finite element modeling of surface water flow. The triangulated terrain surface may be used to produce a finite element grid of triangles for flow computations. Also, the boundaries of the source areas described in this paper represent good approximations of zero-flow boundaries. Accordingly, such boundaries may be used to "substructure" a large area into smaller regions for numerical modeling of the overland flow.

ACKNOWLEDGMENTS

The research presented in this paper was supported by Grant MSM-8717452 from the National Science Foundation.

APPENDIX. REFERENCES

Akima, H. (1978). "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points." *ACM Trans. on Math. Software*, 4(2), 148–159.

Digital elevation models, data users guide 5. (1987). Nat. Cartographic Information Ctr., U.S. Geological Survey, Reston, Va.

Fowler, R. J., and Little, J. J. (1979). "Automatic extraction of irregular network digital terrain models." *Comput. Graphics (ACM)*, 13(2), 199–207.

Franke, R., and Nielson, G. (1980). "Smooth interpolation of large sets of scattered data." *Int. J. Numer. Meth. in Engrg.*, 15(11), 1691–1704.

Grayman, W. M., et al. (1975). "Land-based modeling system for water quality management studies." *J. Hydr. Div.*, ASCE, 110(5), 567–580.

Grayman, W. M., et al. (1979). "Joint state/regional environmental planning using the PEMSO/ADAPT geographic information system." *AutoCarto IV, Proc. Int. Symp. on Cartography and Computing: Applications in Health and Envir.*, R. T. Aangeenbrug, ed., American Congress on Surveying and Mapping/American Society of Photogrammetry, 546–553.

Grayman, W. M., Males, R. M., and Harris, J. J. (1982). "Use of integrated spatial data and modeling capabilities for urban runoff analyses." *Int. Symp. on Urban Hydrol., Hydr. and Sediment Control*, Univ. of Kentucky, Jul. 27–29.

Jenson, S. K., and Domingue, J. O. (1988). "Extracting topographic structure from digital elevation data for geographic system analysis." *Photogrammetric Engrg. and Remote Sensing*, 54(11), 1593–1600.

Lawson, C. L. (1977). "Software for C^1 surface interpolation." *Mathematical software III*, J. R. Rice, ed., Academic Press, New York, N.Y., 161–194.

Lee, D. T., and Schacter, B. J. (1980). "Two algorithms for constructing a Delauney triangulation." *Int. J. Comput. and Information Sci.*, 9(3), 219–242.

McCullagh, M. J. (1981). "Creation of smooth contours over irregularly distributed data using local surface patches." *Geographical Anal.*, 13(1), 51–63.

Moore, I. D., O'Loughlin, E. M., and Burch, G. J. (1988). "A contour-based topographical model for hydrological and ecological applications." *Earth Surface Processes and Landforms*, 13(4), 305–320.

Petrie, G., and Kennie, T. J. M. (1987). "Terrain modeling in surveying and civil