DNESYS--An Expert System for Automatic Extraction of Drainage Networks from Digital Elevation

Jianzhong Qian
Roger Ehrich
J.B. Campbell

TR 88-23

DNESYS—an EXPERT SYSTEM for AUTOMATIC EXTRACTION of DRAINAGE NETWORKS from DIGITAL ELEVATION DATA

by

Jianzhong Qian
Roger W. Ehrich
James B. Campbell

Departments of Electrical Engineering, Computer Science, and Geography
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

August 6, 1988

ABSTRACT

The determination of drainage networks and drainage basins is one of the more tedious yet important uses of topographic maps, and geographic information systems are now used extensively as a manual aid to facilitate that task. However, the wide availability of digital elevation maps has stimulated attempts to automate the process even further.

In this paper, the problems that arose in earlier programs to map drainage systems are analyzed in detail. An expert system called the Drainage Network Extraction System (DNESYS) is described which uses both local operators and global reasoning to extract drainage networks and ridge lines. A stream representation called a parameterized directed graph (PDG) is constructed to model a drainage system. The construction of a model begins with an initial pixel labeling procedure. Then a network tracing and property measurement procedure converts the 2-D low-level labeling information into a symbolic database for high-level processing. By applying Dempster-Shafer evidence theory, evidence collection and uncertain reasoning are performed against the DNESYS knowledge-base that contains the drainage system model and the organized expert knowledge. By discarding erroneous information and supplying missing information, DNESYS produces a complete PDG which can be converted into the final drainage system.

1 INTRODUCTION

Digital elevation maps are one of the most valuable earth resource data sources because topographic information is basic to so many aspects of earth science and because so many other variables, including slope, aspect, drainage, and solar insolation, can be derived from the elevation matrix. In recent years the increasing popularity and utility of geographic information systems (GISs) and the availability of accurate and detailed elevation data for selected regions within the United States has focused new interest upon the use of digital elevation data.

Two types of digital elevation models (DEMs) are commonly available. One is 7.5-minute DEM data which are produced by the U. S. Geological Survey (USGS). They correspond in coverage to standard 1:24,000-scale 7.5- × 7.5-minute quadrangles with a spatial resolution of 30m × 30m. The other is 1:250,000-scale DEM data which are produced by the Defense Mapping Agency (DMA). They correspond in coverage to 1° × 1° blocks (one half of standard USGS 1:250,000-scale 1° × 2° quadrangles). The spatial resolution for this type of DEM data is approximately 90m in the north-south axis and a variable dimension (approximately 90m at the equator to 60m at 50° latitude) in the east-west axis due to convergence of the meridians. The vertical accuracy of these two types of DEMs depends on the accuracy of the source data from which the DEMs were derived [1].

The problem of delineating drainage basins may be approached by first solving the problem of determining the drainage network. Then the drainage network and the ridge lines may be used to determine the corresponding basins. The usual method for deriving the drainage network is to trace or digitize streams as recorded on topographic maps. This is not always practical, especially if detailed data are required for a large region, due to the problems in handling and matching data from large numbers of large-scale maps. In addition, conventional cartographic representations of the drainage network are not always satisfactory in the context of a GIS. Drainage information derived from the stream network on topographic maps may be incomplete due to arbitrary definitions of streams, omission of the smallest branches of the drainage network, inconsistencies in interpretation by cartographers, and effects of cartographic generalization.

As a result there may be good reasons to abandon traditional cartographic representations of drainage networks in favor of interpretations of digital elevation data. Once stream channels have been identified within the digital data, many of their most important characteristics, including size, channel length, gradient, and sinuosity, can be calculated. The major problem is to derive the drainage network from the topographic information accurately and efficiently. In principle, the task is straightforward — water drains from high elevation to low along channels that are connected to one another according to well-defined rules. Exceptions are so rare or so easily recognized that they do not influence the usefulness of a successful technique.

The problem is not quite so easy in practice. The sampling of topography by the digital elevation data generates ambiguities, including artificial pits and ridges. Additional errors may be artifacts of the digitization process used to derive the digital elevations from the source maps or photogrammetric models. Due to noise and quantization errors, true valley points may be obscured completely. Streams that flow through narrow gorges may appear to cross ridgelines. In smooth terrain or when a stream passes through a body of water, local features lose their coherence and cannot be traced easily. A practical method of identifying stream channels must be capable of recognizing and resolving such problems.

Most previous efforts to derive drainage information from digital elevation data have used a moving window to derive information concerning the configuration of the land surface within a local neighborhood. For example, the notable techniques proposed by O'Callaghan and Mark [2] and Jenson [3] are based on low level pixel labeling by using a local operator and subsequent pixel grouping according to local criterea. Using a local operator, O'Callaghan and Mark first label the drainage direction for each pixel. Then they carry out an iterative computation of *drainage accumulation values*. In each iteration, each pixel is relabeled with a weighted sum of the drainage accumulation values of neighboring pixels that drain into that pixel. Then the drainage channels are labeled according to the accumulation values which are greater than the user-specified threshold. Jenson uses a 3×3 pixel window to label so-called *drainage pixels* by searching for local minima between two of its non-adjacent neighbors. Minima are located in any of 12 configurations within the 3×3 window. Then, based on local criteria, a drainage basin linkage is established from the resulting drainage label map given user-specified distance and elevation thresholds.

Although such approaches produce a satisfactory approximation of portions of a drainage system, we have found that they do not yield a reasonable representation of drainage systems considered as a whole. The deficiencies in these techniques are the following:

- The extracted drainage networks are broken into disconnected segments because the gaps between them may be quite large. It would exceed the ability of these algorithms to establish linkages between spatially separated segments.

- Some stream segments may erroneously be merged with others, even when separated by a ridge line or a saddle point. Local techniques may not be able to reject these kinds of mislabelings.

- A single extracted segment may have two opposing elevation slopes within the segment. This means that the segment has two conflicting flow directions along its length.

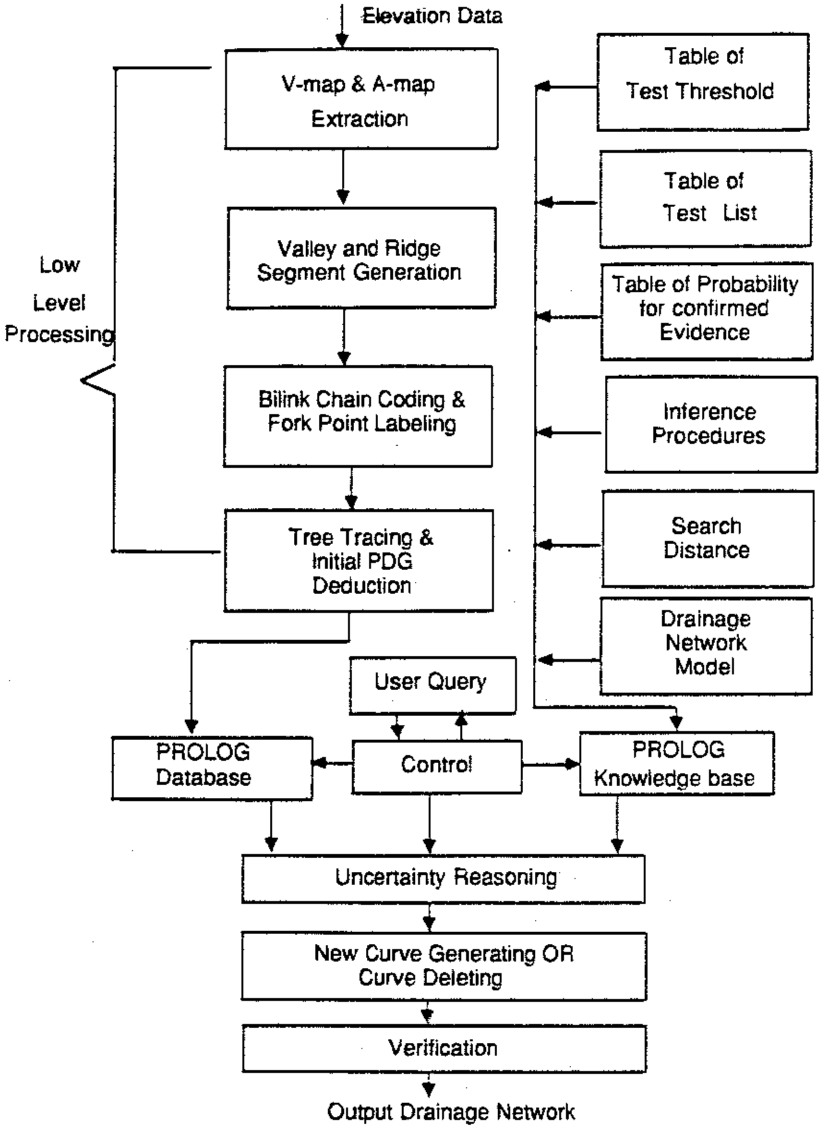


Figure 1. *The DNESYS system diagram.*

- Stream segments may be so thick that the exact stream locations cannot be determined.

In this paper, an expert system called the Drainage Network Extraction System (DNESYS) (Figure 1) is presented. DNESYS uses both local operators and global reasoning to extract the drainage networks and ridge lines from the DEMs by merging lower level structures into globally consistent networks. Unlike other techniques, the DNESYS system organizes the initial labels directly into more global abstractions called *segments* without a pixel-by-pixel grouping process. The system views these as the drainage primitives which are more reliable than individual pixels for the grouping and reasoning process later on. An attribute vector containing all factors that participate in the reasoning process is assigned to each segment. After that, a hypothesis generator proposes links between pairs of spatially related segments. For each pair of segments, N_p predicates are evaluated to produce an N_p-ary evidence vector for asserting a new segment or rejecting it. Finally, all segments with monotonically decreasing elevations are connected to their *stream tree*, which is a hierarchical data structure used to represent a portion of the stream network. The roots of all the stream trees correspond with segments that intersect the edge of the DEM and carry water out of the map area. Since there are many segments and many trees, all segments are uniquely labeled by their segment numbers and tree numbers. DNESYS also extracts topographic ridge lines as part of the reasoning process.

The DNESYS system was developed at the Spatial Data Analysis Laboratory at Virginia Polytechnic Institute and State University. The system consists of three separate parts: low-level primitive labeling and attribute vector measurement, high-level spatial reasoning, and an interface between these two parts. This interface includes a set of special procedures to construct the PROLOG database and to convert the PROLOG database back into a 2-dimensional image. These low-level procedures are written in RATFOR and are implemented as GIPSY (the General Image Processing System [4]) commands. The high-level expert is written in PROLOG to carry out the uncertainty reasoning. The interface is designed to construct a PROLOG data base automatically with an intermediate representation of all useful spatial information about drainage primitives, which the high-level expert can access efficiently to determine relevant events.

In the remainder of this paper, DNESYS is analyzed and described in four parts. The model of drainage networks used in the DNESYS system will be given in Section 2. Section 3 describes the low-level processing which extracts the drainage primitives for the high level reasoning. Then, at the high level stage, the evidence collection method and the uncertainty reasoning algorithm are analyzed in Section 4. Section 5 presents the data structure and control. Finally, the experimental results produced by running the DNESYS on DEMs are shown in Section 6.

2 THE MODEL OF DRAINAGE NETWORKS

The principal topographical features on which drainage networks are based are called *valley segments*. A valley segment is a connected chain of pixels which are local minima in the direction normal to the segment. Hydrographically, a drainage network consists of all valley segments on which runoff is sufficiently concentrated and through which the flow is downhill. These are called *stream segments*. Valley segments and stream segments differ in that valley segments are measured features whereas stream segments are components of an actual drainage network. Due to soil porosity, geologic structure, slope, and climatic conditions, many relatively large and deep valleys contain no definite stream channel. These are still included in the network determined by DNESYS. A single unbranched stream segment may have one open end which may be considered to be its flow source. However, its other end must connect to other channels which drain the water that it collects.

In DNESYS, the model of a drainage system is a *parameterized directed graph* (PDG) deduced from the DEM's data. The graph is, in fact, a set of loop-free stream trees, and the parameters are the attributes associated with each stream segment. The nodes of the graph are either source nodes to which no other stream segments connect, sink nodes which are on the edge of the DEM, or fork nodes where stream segments merge together. Water flow direction is assumed to be from the source nodes toward the sink nodes. The total flow into a node must equal the flow out from the node (flow conservation). IN degree and OUT degree are used to describe the connectivity of a node. Source nodes have IN degree zero and OUT degree one. For fork and sink nodes, the IN degree is equal to the number of directly connected branches which have negative end-elevation slopes, and the OUT degree is equal to the number of directly connected branches which have positive end-elevation slopes. Any node with zero OUT degree is a sink node. The branches of the the PDG are the stream segments, and the reasoning process will determine which valley segments are stream segments and which are not. The parameters associated with each stream segment include length, drainage direction, and the elevation slope of the segment.

Stream segments have the following properties:

1. The width is only one picture element.

2. The length is arbitrary.

3. The elevation values along it decrease monotonically in the flow direction.

4. There is no fork point on the segment between its endpoints.

5. The elevation value of any point on the segment is a local minimum point of the stream segment cross section.

Any branch which does not satisfy above five conditions is called an *inconsistent branch* in the PDG. A graph with any inconsistent branch is an *incomplete PDG*. For example, a valley segment between two adjacent nodes but with conflicting flow directions is an inconsistent branch in that graph, since it does not satisfy Condition 3. It must be split into several consistent branches by inserting nodes at suitable points on the segment so that all five conditions are satisfied. Since the inserted nodes will have either zero OUT degree or zero IN degree, linkages to other branches must be located. A graph with a sink node occurring anywhere except on the map edge is also called an incomplete PDG since sink nodes don't make sense except at the physical edges of the DEM where valleys are discontinuous.

When the reasoning process begins, each branch consists of a source node and a sink node, and sinks and sources are joined as the merging process progresses. However, in some cases global evidence will cause sinks and sources to be connected even though that is not supported by local evidence. The the direction of a branch of the PDG is the flow direction, i.e. the the direction of decreasing elevation along that segment.

Branching angle is another important issue in reasoning about the structure of a drainage network. The angular relationships among segments incident at a node are determined by drainage slopes near the node; the angles between inward flowing streams are usually less than 90 degrees [5,6]. In the literature there are two principal theoretical models for determining branching angle from slope—the Hortonian model [7] and the Minimum Power model [8]. From the theoretical point of view, Roy found that both angular geometry models are based upon the optimality principle, which implies that the drainage system must "perform its task" with maximal efficiency and minimum cost [9]. Because both models yield similar theoretical angles and because the Hortonian model is simpler to implement, DNESYS uses the Hortonian model as part of its knowledge base.

Under the assumption that overland flow on the valley slopes follows the line of steepest gradient, Horton's model hypothesizes that

$$\cos\theta = \frac{S_c}{S_g}, \tag{2.1}$$

where θ is the angle between the line of overland flow and the receiving stream, S_c is the stream gradient, and S_g is the ground slope. For the case that a receiving stream is joined by a single tributary stream, Horton modified his model to

$$\cos\theta = \frac{S_0}{S_1}, \tag{2.2}$$

where θ is the junction angle, S_0 is the gradient of receiving stream, and S_1 is the gradient of tributary stream as in Figure 2. Usually, $S_1 > S_0$. However, when two streams merge

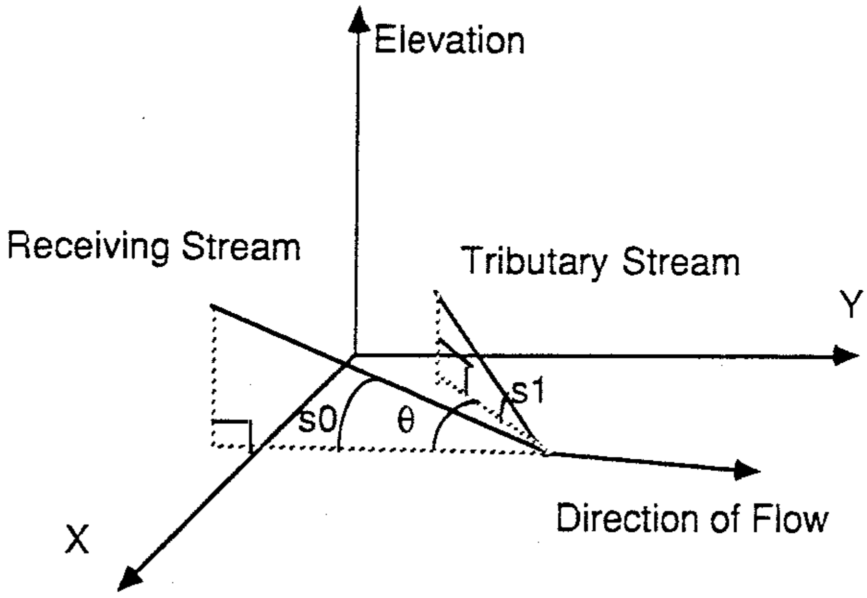


Figure 2. *The stream branching model (Hortonian model).*

with nearly equal gradient, i.e. $S_1 \approx S_0$, this model will erroneously predict a zero junction angle [10]. Nevertheless, this model clearly states that the junction angle θ must vary within a range of $0 < \theta < 90$ degrees. This is true for most practical cases.

In DNESYS, an incomplete parameterized graph with sink nodes, source nodes, and possibly inconsistent branches is obtained from low-level processing. The graph consists of many disjoint trees, and it is the function of the high-level reasoning process to make mergers of trees and to split branches in order to remove inconsistencies due to conflicting flow directions.

3 LOW LEVEL PROCESSING

3.1 Initial Drainage Point Marking

This section begins with a description of the process for determining the initial labelings of candidate pixels for valley segments as well as for ridges by using an extension of Jenson's procedure. These pixels are either *V-shaped* or *A-shaped* depending on the shape of their local elevation profiles. The size of the local neighborhood operator used to label these pixels depends upon the resolution of the DEMs. Generally, the finer the resolution, the larger the operator. However, if the resolution is extremely coarse, the minimum size 3×3 operator must be used to enable the operator to extract meaningful topographic features. These masks are logical operators instead of approximation operators to avoid the risk of making any unforeseen local interpretations.

There are two differences from Jenson's method for defining the operator because DNESYS's grouping and linking algorithms are based on different information. One difference is that instead of resampling the DEM's data according to a coarser resolution grid when the DEM resolution is very fine, the operator itself is extended to a size greater than 3×3. The other is that the DNESYS operator considers only 4 symmetric cross sections instead of using an additional 8 asymmetric cross sections. That is, the operator detects valley points and pits (or ridge points and peaks) when they are centered on the operator and run in diagonal, vertical, or horizontal directions. This improves efficiency and reduces the risk of ambiguous labeling.

Let the size of the general operator be $3N \times 3N$, where $N = 1, 2, 3, \ldots$ as shown in Figure 3. Let $E(m,n)$ be the pixel elevation value at the position $(m,n), 1 \leq m,n \leq 3N$, within the operator. Define a new 3×3 operator Z consisting of 9 $N \times N$ *operative units* of the original operator. Then the operative unit $Z(i,j)$ is defined by

$$Z(i,j) = \frac{1}{N^2} \sum_{m=s}^{T_1} \sum_{n=t}^{T_2} E(m,n) \qquad (3.1)$$

where $s = (i-1)N + 1$, $T_1 = iN$, $t = (j-1)N + 1$, $T_2 = jN$, $i = 1,2,3$, and $j = 1,2,3$.

Thus, a central unit $Z(2,2)$ is a V-shaped unit if and only if any one of the following conditions is satisfied:

$$\begin{gathered} Z(1,1) > Z(2,2) \text{ and } Z(2,2) < Z(3,3) \\ Z(1,2) > Z(2,2) \text{ and } Z(2,2) < Z(3,2) \\ Z(1,3) > Z(2,2) \text{ and } Z(2,2) < Z(3,1) \\ Z(2,1) > Z(2,2) \text{ and } Z(2,2) < Z(2,3) \end{gathered}$$

$Z(2,2)$ is an A-shaped unit if and only if any one of the following conditions is satisfied:

$$
\begin{gathered}
Z(1,1) < Z(2,2) \text{ and } Z(2,2) > Z(3,3) \\
Z(1,2) < Z(2,2) \text{ and } Z(2,2) > Z(3,2) \\
Z(1,3) < Z(2,2) \text{ and } Z(2,2) > Z(3,1) \\
Z(2,1) < Z(2,2) \text{ and } Z(2,2) > Z(2,3)
\end{gathered}
$$

All $N \times N$ pixels within the unit $Z(2,2)$ are labeled with the same label as that of unit $Z(2,2)$. Thus, the local configuration of topography is based upon averages of the pixels within each operative unit. By using the Z-operator, separate V-maps and A-maps are produced. The V-map by itself does not contain sufficient information for identifying valley segments. Since adjacent minima might not be minima in the same direction, some V-shaped pixels must be initially labeled as ambiguous until the valley orientation can be determined. Worse, a particular pixel can be labeled as V-shaped in one cross section, and simultaneously A-shaped in the normal direction. This is a saddle point which further confuses the labeling of valleys and ridges because the valley slopes downward from both sides of the saddle point.

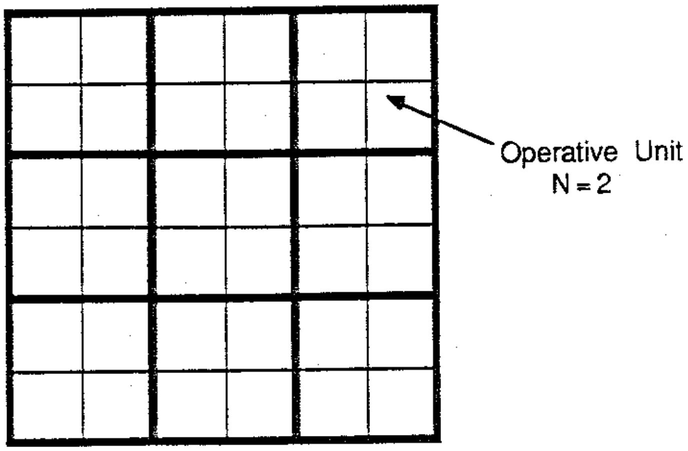


Figure 3. 3×3 *operator with* 2×2 *operative unit.*

After low level pixel labeling, instead of using a pixel by pixel grouping procedure, an A-map masked thinning method is applied to get better valley segments from the V-map. All V-shaped pixels on the V-map which overlap with A-shaped pixels on the A-map are identified as saddle points and are removed from the V-map. Disregarding any drainage property, the thinning algorithm based on that of Arcelli and Sanniti [11] is used. It treats the V-map as a binary picture and thins the V-label figure down to a one-pixel-thick line drawing without destroying the original shape information. As a result, the thinned line corresponds approximately to the bottom line of the valley, and ambiguous pixels are removed. At the completion of this step, all valley segments in the resulting map will satisfy the following three conditions:

1. All segments have a width of one pixel.

2. All segments mark V-shaped topographic features.

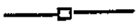
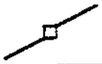
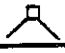
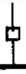
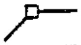
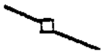
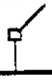
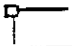
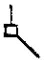
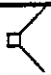
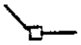
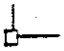
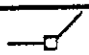
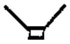
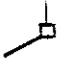
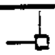
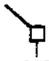
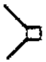
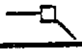
3. No segments cross ridge lines.

These segments are much more useful as primitives for grouping than primitives based on V-shaped pixels alone. However, at this point these may still have pits, conflicting flow directions, and other ambiguities. Resolution of these problems will be left to the high level expert. The output of this stage consists of three images: two symbolic images store the valley segments and ridge segments, and one numeric image is used to store elevation values along these segments. All other pixels in these images are set to zero.

3.2 Network Tracing and Property Measuring

TABLE 1

The link patterns

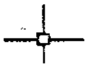
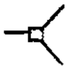
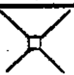
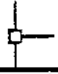
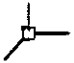
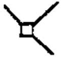
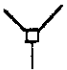
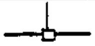
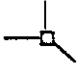
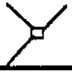
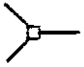
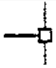
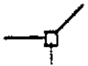
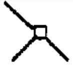
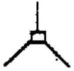
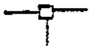
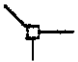
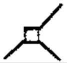
type code	pattern	type code	pattern
1		11	
2		12	
3		13	
4		14	
5		15	
6		16	
7		17	
8		18	
9		19	
10		20	

A special tracing algorithm is used to measure the set of spatial properties of each valley segment. Each valley segment has two endpoints; an endpoint is called open if it is not shared by other segments, and it is called a fork point if it is. Fork points are points where valley segments merge to form trees. For implementation efficiency, a doubly linked chain of pixels is generated for each segment according to twenty predefined linking patterns as in Table 1.

Meanwhile, all fork points in trees are detected (see Table 2), and all sources and sinks of tree branches are marked. These three tasks are done in one pass. A bidirectional linked chain is a sequence of pixels along a valley segment in which each pixel between its endpoints has pointers to its two neighbors. Bidirectional linked chains facilitate the network tracing process since the tracing sequence for the pixels in different segments may be different. The bidirectional linked chains, the fork points, and the open endpoints provide a clear guide for tracing and measuring all spatial information for segments simultaneously.

TABLE 2

The fork patterns

type code	pattern	type code	pattern
1		10	
2		11	
3		12	
4		13	
5		14	
6		15	
7		16	
8		17	
9		18	

The network tracing process illustrated in Figure 4 is the following:

1. Scan the image from top to bottom, line by line.

2. If an open endpoint is encountered, the scanning process is interrupted, and a tracing and measuring process is initiated to follow the bidirectional link codes.

3. Record the segment length, positions, elevations, and tangent directions for all segment pixels and compute the elevation slopes, mean curvatures, end directions, and mean elevations for both ends of each segment. Assign the segment label and tree

label for the traced segment. This process terminates when a fork point or an open endpoint is reached. All pixels which have been traced are set to zero.

4. If a fork point is found, push the fork position onto a fork stack. Search the next untraced branch leaving the fork point in clockwise order.

5. If an open endpoint is encountered, then check the fork stack. If it is not empty, pop the stack and start tracing a new segment from the popped position by repeating Step 3 to Step 5. If the fork stack is empty, then the whole tree in the network has been completely traced. Continue scanning with Step 1.

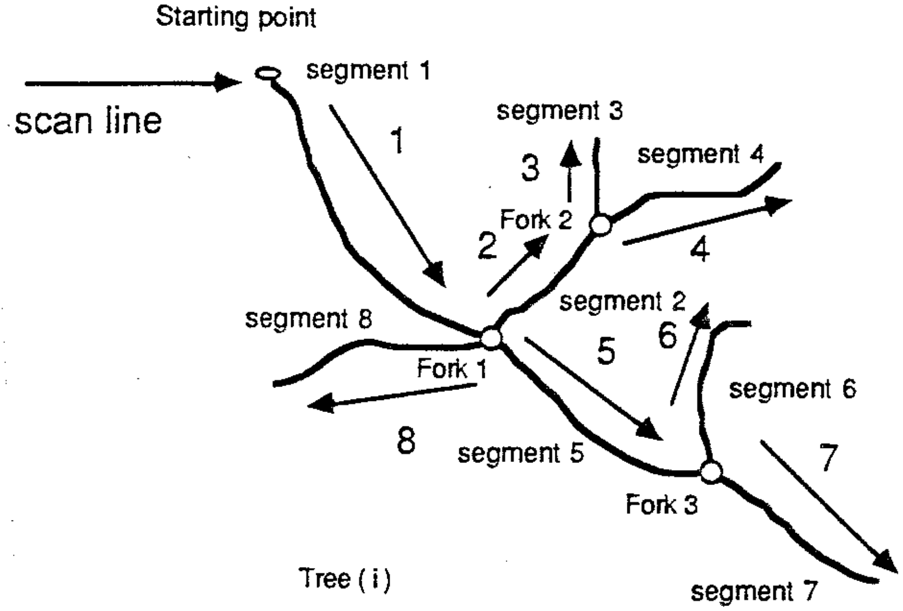


Figure 4. *The tracing order.*

After the whole image has been scanned, the output file contains all the spatial information for each segment. All segments have their own unique labels.

4 EVIDENCE COLLECTION AND UNCERTAINTY REASONING

Reasoning about a drainage system at a high level requires that DNESYS have the ability to handle uncertainty. Uncertainty is caused both by imprecise data and by imprecision in the rules used for extracting the drainage networks. In the reasoning process, the evidence from the observed data is often incomplete or conflicting; some rules are based simply on intuitive knowledge. Therefore, there are no simple measures or rules for deciding how a pair of broken segments should be connected or how an inconsistent segment should be split. To reason with uncertainty and to resolve possible conflicts in the data produced by the low-level measurements, DNESYS applies evidential reasoning to these problems.

Evidential reasoning is a collection of inference techniques for reasoning with uncertain information. Among others, Dempster-Shafer evidence theory [12], Shortliffe's certainty factor [13,14], Zadeh's fuzzy set theory [15,16], and Bayesian updating methods [17] are well-known. An evidential reasoning system is used to construct a global representation using partial representations, each of which demonstrates local consistency. Thus, it can simultaneously form and evaluate alternative hypotheses. An interpretation of a portion of a drainage network depends not only on each individual piece of evidence but also on interactions among various pieces of evidence. Then the problem becomes one of multicriteria optimization. To solve it requires consideration of all the evidence from each partial representation and utilization of higher level knowledge to resolve conflicts.

4.1 Evidence Collection

A set of functions are designed to extract specific partial evidence in support of all hypotheses that are generated. The output of the feature extractors are inputs to a set of tests which are organized as a procedural part of the knowledge base of the expert system. These tests are made at higher-level processing stages. The test results are used as inputs to the process of reasoning with uncertainty, which is discussed later. Let $\tau_1, \tau_2, \ldots, \tau_8$ be the set of tests. The following are the tests made on the low-level spatial features in the PROLOG database.

Distance Test τ_1: This test provides support for the connection of two segments based upon proximity of their endpoints. Let $P_i(x_i, y_i)$ and $P_j(x_j, y_j)$ be endpoints of two arbitrary segments; then the geometric distance between P_i and P_j is given by the usual Euclidean distance

$$D = \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}. \tag{4.1}$$

Given a threshold, T_1, then

$$\tau_1 = \begin{cases} true, & \text{if } D \leq T_1; \\ false, & \text{otherwise.} \end{cases}$$

Curvature Similarity Test τ_2**:** Since most stream segments are locally smoothly connected, this test is for confirming this property. Let CV_1 and CV_2 be the mean curvatures of the endpoints of two candidate segments and let T_2 be a threshold. Then CV_i is defined as

$$CV_i = \frac{1}{n} \sum_{j=1}^{n} \frac{\theta_{j1} - \theta_{j2}}{\Delta S_j} \tag{4.2}$$

where $(\theta_{j1} - \theta_{j2})$ is the local measure of change of the orientation of the tangent vector and ΔS_j is the change in arc length corresponding to above angle changes. Thus

$$\tau_2 = \begin{cases} true, & \text{if } |CV_1 - CV_2| \leq T_2; \\ false, & \text{otherwise.} \end{cases}$$

Segment Length Test τ_3**:** The system assigns higher credibility to longer segments. All isolated valley points are ignored. Let L_i be the lengths of two distinct segments and let T_3 be the test threshold. Then

$$\tau_3 = \begin{cases} true, & \text{if } L_i \geq T_3 \text{ for } i = 1, 2; \\ false, & \text{otherwise.} \end{cases}$$

Elevation Slope Test τ_4**:** This test is to confirm that if the stream flow directions of two segments are consistent, then the signs of the elevation slopes at the nearest endpoints of these two segments should be opposite according to the reference direction. The positive reference direction is defined as the direction from the center of the segment toward the endpoint. The elevation slope of the endpoint is calculated from the end section of valley segment. Then let ES_1 and ES_2 be the elevation slopes at the ends of the valley segment.

$$\tau_4 = \begin{cases} true, & \text{if } (ES_1 \geq 0 \text{ and } ES_2 \leq 0) \text{ or } (ES_1 \leq 0 \text{ and } ES_2 \geq 0); \\ false, & \text{otherwise.} \end{cases}$$

Elevation Similarity Test τ_5**:** This test is to confirm that the mean elevation difference between the endpoints of two mergable segments does not exceed a threshold value, given the search distance. Let ME_1 and ME_2 be the mean elevations in a local neighborhood near the ends of the segments, let T_5 be the threshold for this test, and let β_1 and β_2 be the predicates

$$\beta_1 = ES_1 < 0 \text{ and } ES_2 > 0 \text{ and } 0 < (ME_1 - ME_2) \leq T_5$$
$$\beta_2 = ES_1 > 0 \text{ and } ES_2 < 0 \text{ and } 0 < (ME_2 - ME_1) \leq T_5.$$

Then

$$\tau_5 = \begin{cases} \mathit{true}, & \text{if } \beta_1 = \mathit{true} \text{ or } \beta_2 = \mathit{true}; \\ \mathit{false}, & \text{otherwise.} \end{cases}$$

Forward Elevation Test τ_6: This test is based on the knowledge that the elevation slope does not usually change rapidly in a local neighborhood. Therefore if a segment is extrapolated toward another endpoint based on its slope, there should not be a large disparity between the actual elevation and extrapolated elevation at that endpoint. Let EE_1 and EE_2 be the elevations at the ends of two valley segments, let ES_1 and ES_2 be the corresponding elevation slopes, let D be the Euclidean distance between these two ends, and let T_6 be the test threshold.

$$\tau_6 = \begin{cases} \mathit{true}, & \text{if } |EE_1(1 + ES_1 D) - EE_2| \leq T_6 \text{ or } |EE_2(1 + ES_2 D) - EE_1| \leq T_6; \\ \mathit{false}, & \text{otherwise.} \end{cases}$$

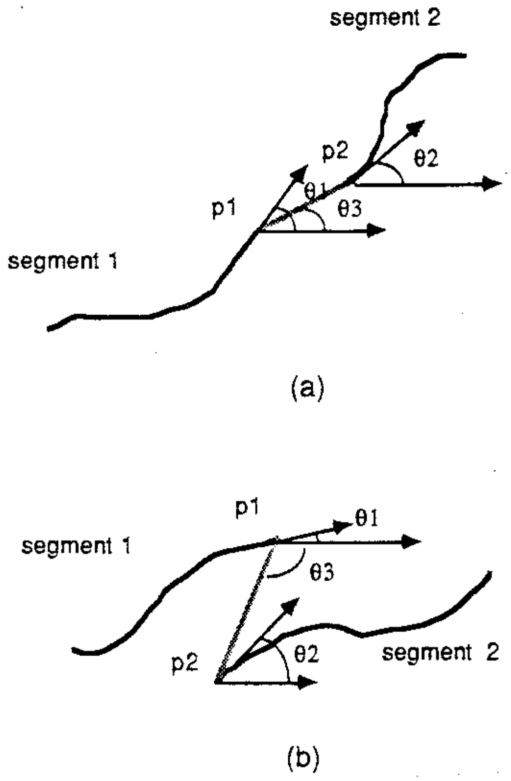
Orientation Test τ_7: Since local curvatures of valley segments tend to be low, one would expect that a segment connecting the endpoints of two disjoint segments would not introduce sharp local curvatures. Therefore τ_7 compares the orientations at segment endpoints and the orientation of a segment between them. The smaller the orientation differences, the higher the probability that the endpoints should be merged. The following three angle differences completely specify the geometric orientation and transition properties of three joined segments.

Let P_1 and P_2 be the endpoints of two distinct segments. Let θ_3 be the direction of P_1P_2, θ_1 be the angle of the tangent vector to the flow direction at P_1, and let θ_2 be the angle of the tangent vector to the flow direction at P_2 as in Figure 5. Then the orientation test checks the orientation of the line P_1P_2 with respect to these orientations. Specifically,

$$\tau_7 = \begin{cases} \mathit{true}, & \text{if } |\theta_3 - \theta_2| \leq T_{71} \text{ and } |\theta_3 - \theta_1| \leq T_{72} \text{ and } |\theta_1 - \theta_2| \leq T_{73}; \\ \mathit{false}, & \text{otherwise.} \end{cases}$$

where T_{71}, T_{72}, and T_{73} are the thresholds for this test.

Ridge Intersection Test τ_8: This test is for checking for a ridge crossing in the gap between the endpoints of two distinct segments. Let R be any ridge line. Let $P_i(x_i, y_i)$ and $P_j(x_j, y_j)$ be the endpoints of two distinct segments. Then the straight line P_iP_j is defined by the slope



(a)

(b)

Figure 5. *The angles in orientation test.*

$$y = \frac{\Delta y}{\Delta x}(x - x_j) + y_j$$

where $x_i \leq x \leq x_j$, $y_i \leq y \leq y_j$, $\Delta y = y_j - y_i$, and $\Delta x = x_j - x_i$.

$$\tau_8 = \begin{cases} \mathit{true}, & \text{if } P_iP_j \text{ does not intersect R;} \\ \mathit{false}, & \text{otherwise.} \end{cases}$$

4.2 The Inference Process

DNESYS maps the evidence that is extracted from the observed data into a Boolean logic set by using the functions $\tau_1, \tau_2, ..., \tau_8$. In general, each τ_i is a function of a data vector Z_i and a parameter vector T_i, that is, $\tau_i = f_i(Z_i, T_i)$. In the case of $1 \leq i \leq 7$, the parameters are all thresholds which are determined in one of two ways — from training data or by experts. An individual hypothesis is supported according to the success of its corresponding tests. Each τ_i is merely a contribution to the overall support, and the influence of τ_i on hypothesis h is determined either by a conditional probability $p(h|\tau_i)$ or by experts subjectively. DNESYS interprets this probability as evidential strength or as a degree of confirmation. An inference network that propagates evidence strength can then be constructed.

Let h be a hypothesis, let $\tau_1, ..., \tau_8$ be the evidence for that hypothesis, and let $p(h|\tau_1), ..., p(h|\tau_8)$ be the evidence strengths. Then the inference network is illustrated in Figure 6. Note that several of the τ_i may logically conflict, but other τ_i may still verify the hypothesis. Thus, the rule format should *not* have the simple form

```
CONDITION 1 AND CONDITION 2 AND ...CONDITION N ACTIONS
```

Instead, DNESYS chooses a belief function which is able to synthesize all partial representations of evidence and to achieve the multicriteria optimization. DNESYS accumulates partial supports from different evidence iteratively throughout the inference network. The reasoning system in DNESYS can be directed by the Dempster-Shafer evidence theory, which is an extension of the more common Bayesian probability analysis [12].

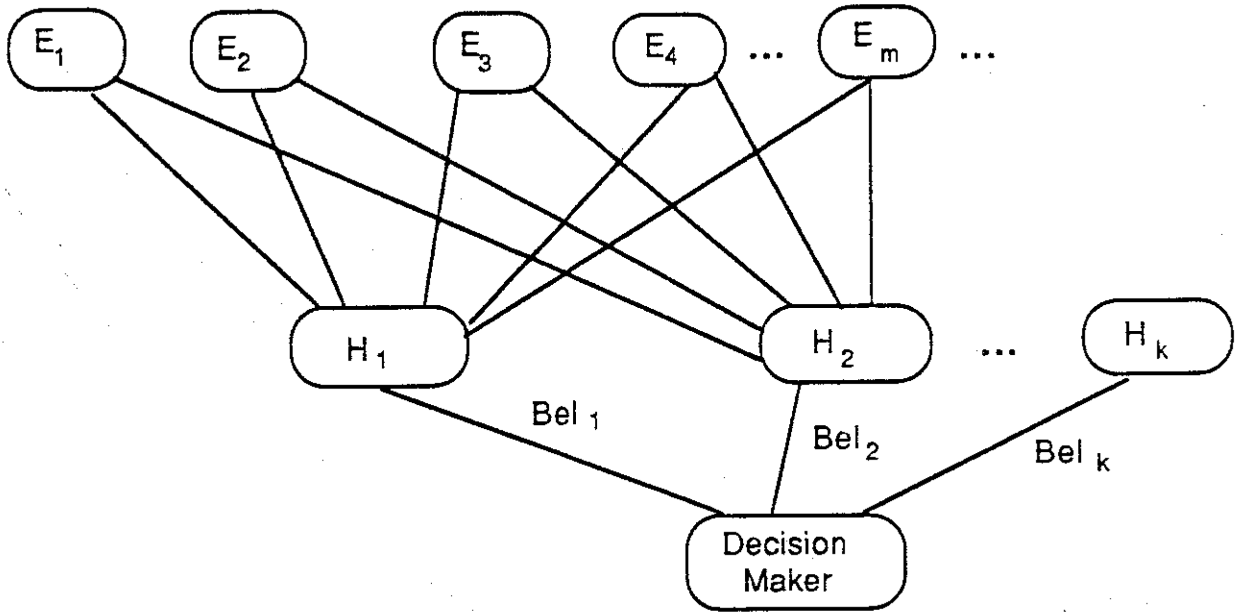


Figure 6. *The inference network.*

A *frame of discernment* Θ in the D-S model is defined as the set of all different possibilites under consideration. It is so named to emphasize the epistemic nature of the set of possibilities Θ that is determined and meaningful, dependent on our knowledge. In DNESYS a frame of discernment Θ, for producing a complete PDG, consists of the following exhaustive and mutually exclusive subsets:

1. A source node is connected to a sink node.

2. A source node remains open (ie, not connected to a sink node).

3. A source node is backward merged to a sink point which splits an inconsistent branch (with a *V-shaped* elevation profile) into two consistent branches.

4. A sink node is forward merged to a source point which splits an inconsistent branch (with a *A-shaped* elevation profile) into two consistent branches.

5. A sink node is located at the boundary of the map and remains open.

Each subset of Θ corresponds to a hypothesis. Frame of discernment Θ delimits a sample space which contains all possible linking hypotheses, only one of which is true at any one time. The evidence provided by each test τ_i is represented as a basic probability assignment (*bpa*) over the hypotheses discerned by a frame Θ. The *bpa*, in fact, is a generalization of a probability mass distribution (or m-function) which represents the impact of each distinct piece of evidence on the subsets of a frame Θ. It is denoted by m and must satisfy the following conditions:

(1) $0 \leq m(A) \leq 1$ *for all* $A \subseteq \Theta$

(2) $m(\phi) = 0$

(3) $\sum_{A \subseteq \theta} m(A) = 1$

The initial bpa's in DNESYS are provided by the $p(h|\tau_i)$ determined by each test described above.

Thus, for an event A of the frame of discernment Θ, we can calculate the belief function, the plausibility function (defined below), uncertainty, and the combination of belief functions. These quantitatively characterize the hypothesis A.

The *belief function*, denoted *Bel*, is the sum of the beliefs committed exactly to every subset of A by m. That is

$$Bel(A) = \sum_{B \subseteq A} m(B),$$

where A is any other subset of Θ and $Bel(A)$ measures the degree of belief in the hypothesis A.

The *plausibility function*, denoted *Pl*, is the maximum extent to which one may believe event A. That is

$$Pl(A) = 1 - Bel(A^c) = \sum_{B \cap A \neq \phi} m(B),$$

where $Bel(A^c)$ is the support for negation of A or the doubt on A. Note that $1 - Bel(A^c)$ is not the same as $Bel(A)$ since evidence partially in favor of a event A does not necessarily mean that the evidence is partially in favor of its negation. In the Dempster-Shafer model, $Bel(A) \leq 1 - Bel(A^c)$. This avoids the Bayesian restriction that requires $P(A) = 1 - P(A^c)$.

One of the advantages of Dempster-Shafer model is that it allows the representation of uncertainty as well as the representation of ignorance. The amount of uncertainty with respect to a hypothesis A given the evidence can be represented as the interval of the $Bel(A)$ and $Pl(A)$, ie, $[Bel(A), 1 - Bel(A^c)]$. The degree of ignorance for a hypothesis A is the difference $Pl(A) - Bel(A)$.

Dempster's rule provides a way to combine the support for a hypothesis A based on multiple bodies of evidence. Let Bel_1, Bel_2 and m_1,m_2 be two belief functions and their corresponding $bpa's$, respectively. Then the bpa m of the new belief function Bel can be calculated as the orthogonal sum of m_1 and m_2:

$$m(A) = \begin{cases} 0 & \text{if } A = \phi \\ \frac{\sum_{X \cap Y = A} m_1(X) m_2(Y)}{1 - \sum_{X \cap Y = \phi} m_1(X) m_2(Y)} & \text{if } A \neq \phi \end{cases}$$

where X and Y are individually varied over all subsets of Θ. Applying Dempster's rule, DNESYS pools the multiple bodies of evidence obtained from the different tests to get new belief functions. Finally, a single hypothesis among others is selected, if and only if the belief function has a maximum value which is greater than a threshold specified in the knowledge base. When there are no remaining inconsistencies, the complete *PDG* contains the results of the reasoning process. Global consistency is achieved for two reasons: individual segments are consistent with one another and with ridge segments, and the *PDG* structure itself precludes the formation of global loops among the valley segments.

5 DATA STRUCTURE AND CONTROL MECHANISM

The high-level expert of DNESYS consists three major parts: the database, the knowledge base, and the inference procedures. Since the database written in PROLOG is accessed so frequently, the performance of DNESYS is critically dependent upon its data structures and its control mechanisms. Special efforts are required to make PROLOG's symbolic data structures work efficiently with multiband 2-D data arrays. The immediate difficulty is that of locating the spatially related valley segments which need to participate in each deduction. The data structure must be so designed that the time spent making deductions about spatially unrelated evidence is minimized.

5.1 The Data Structure and the Database

In the DNESYS database, all valley segments and ridge segments are structured as hierarchical lists. These lists are treated as PROLOG fact rules asserted in the system database. This database consists the following components:

1. an invertible index
2. the drainage primitive data
3. the intermediate data generated by the reasoning process.

To access all relevant events efficiently, an inverted indexing method is used for the spatial information. The DEM coordinate space is partitioned into blocks, and for each block an endpoint index is constructed for those segments having endpoints in that block. While a hierachical partitioning method such as quadtrees can be used, in this work only a single level of partitioning was used. In addition to the endpoint index, a space spanning index is used to indicate which blocks are spanned by each segment. The space spanning index is useful for the ridge intersection test and for the merging that may occur when an inconsistent segment is split. This method of indexing makes it possible to locate spatially related segments quickly and reject unrelated events at an early stage.

The index table is a linear list of elements. Each element has the form

```
((INDEX
 *NODE-TYPE {SR: source node, SN: sink node, IN: inconsistant node}
 *INDEX-TYPE {1: spanning index, 0: open-end index}
 *SEARCH-FLAG {1: done, 0: to be tested}
 *END-TYPE {1: start point, 0: end point}
 *HORIZONTAL-INDEX
 *VERTICAL-INDEX
 *SEGMENT-NUMBER-IN-TREE
*TREE-NUMBER))
```

Each open endpoint of a segment has one index element with *INDEX-TYPE=0. However, each indexing element with *INDEX-TYPE=1 corresponds to one block which the segment spanned. For example, if a segment spans N blocks, then there will be N index elements for that segment. By using the PROLOG matching mechanism, it is easy to locate the interesting indices given both of the segment number and the tree number. On the other hand, it is easy to locate all spatially related segment numbers and tree numbers by specifying both vertical and horizontal index numbers. It is efficient because the whole search and matching process involves only indices and not the segment data.

The data structure for drainage primitives includes four parts: a unique segment label, a segment parameter list, a list of sample points at each end, and an extended segment chain code which includes the elevation and direction. The data structure has the form

```
( ( SEGMENT-NUMBER *s
    TREE-NUMBER *t
   ( (START-TYPE *sp)
    (STOP-TYPE *tp)
    (SEGMENT-LENGTH *sl)
    (START-MEAN-ELEVATION *sme)
    (STOP-MEAN-ELEVATION *tme)
    (START-ELEVATION *se)
    (STOP-ELEVATION *te)
    (START-MEAN-CURVATURE *smc)
    (STOP-MEAN-CURVATURE *tmc)
    (START-ELEVATION-SLOPE *ses)
    (STOP-ELEVATION-SLOPE *tes)
    (START-DIRECTION *sd)
    (STOP-DIRECTION *td)
   )
   ((START-SAMPLING-POINTS *x11 *y11 *x12 *y12)
    (STOP-SAMPLING-POINTS *x21 *y21 *x22 *y22)
   )
   ((COMPOSED-SEGMENT-TRAIN-CODE
     (POINT 1    *x
                 *y
                 *elevation
                 *direction
     POINT 2    *x
                 *y
                 *elevation
                 *direction
                .
```

```
                  .
                  .
     POINT N      *x
                  *y
                  *elevation
                  *direction)
    ))
   ))
```

This database contains all the information required to reconstruct a raster image after reasoning has been completed. The data structure for drainage primitives including valley lines and ridge lines is a two-level hierarchical list. As soon as the segment number and the tree number have been chosen in the deduction process, any one of the three elements in the first-level list can be moved to a global area called the blackboard [18]. Then according to the END-TYPE in the index, corresponding elements in the second-level list can be asserted as the independent fact rules with their own predicates in the blackboard. Thus these data are temporarily shared by the hypothesis generation and verification processes. They are erased immediately from the blackboard before the data for another segment are moved in.

The knowledge base consists of a test name list, a threshold table, a table of evidence strengths (*bpa*) for confirmed or unconfirmed τ_i's, a spatial search distance, and inference procedures. The test name list contains the symbolic names for $\tau_1, ..., \tau_N$. Through this list, DNESYS specifies how many tests and which tests should be carried out in the system. The threshold table contains all threshold values $T_1, ..., T_N$ for their corresponding tests. According to this table, DNESYS maps the evidence that is extracted from the observed data into a Boolean logic set. The table of *bpa*s contains the evidential strengths, $p(h|\tau_i)$. The spatial search distance provides an upper limit to the spatial neighborhood involved when the algorithm focuses attention on a particular segment. This distance specifies how many neighboring blocks should be searched to generate hypotheses. Finally, the inference procedures are used to implement the evidential reasoning using the D-S model. The basic data structure in the knowledge base is the frame [19]. Each frame has a set of associated slots, which are empty buffers that will hold particular types of information. Some of these are filled with the corresponding values assigned by experts or obtained from training data, and the rest are filled as the the reasoning process takes place. Through the knowledge base frames, the user can specify or modify the knowledge base freely except for the inference network, which is relatively fixed.

5.2 The Control Mechanism in the High-level Expert

The main controller in the high level expert of DNESYS is based upon the database and knowledge base structures just described and consists of the following parts:

1. the spatial relevancy finder
2. the hypothesis generator
3. the evidence pooler
4. the decision maker or hypothesis selector

The spatial relevancy finder performs the task of focusing attention. First, it selects an open node (either a source node or a sink node) by searching an indexing element which has zero values both in the search flag and in the index-type flag in the index. As soon as the finder picks up an element, the search flag in its index element is set to 1. This indicates that the node has been examined and may be subsequently skipped in the top-down selection process. Thus, the segment number, tree number, node type, and spatial indices of the node in horizontal and vertical directions are obtained from the index element simultaneously. Second, the finder calculates all possible search indices which fall in a search radius specified in the knowledge base. As a result, a search index list is generated which contains all N index pairs spatially related to the selected segment.

For all N index pairs returned by the search, the hypothesis generator uses the search index list to generate hypotheses one by one. There are five types of hypotheses which form the frame of discernment described in Section 4. For each pair of indices in the list, the hypothesis generator promptly picks up the corresponding segment number and tree number from the index. However, only the node which belongs to the tree number that differs from the one selected by the relevancy finder is accepted. This prevents the generation of any loops in the PDG. The data for these two selected nodes participating in the hypothesis are brought to the blackboard by decomposing the hierarchical data list from the database. The resultant data formats in the blackboard are separate fact rules labeled by their parameter names. By consulting the test name list in the knowledge base, the tests described previously are computed. All the parameters concerned with a test are accessed, and then the corresponding rules are erased from the blackboard.

Next, the evidence extractor determines the Boolean values of $\tau_1, \tau_2, ..., \tau_8$ according to the threshold values imbedded in the knowledge base for each test. Then the initial *bpas*, $p(h|\tau_i)$, are extracted from the knowledge base by table look-up. These values are treated as the evidence strength and are propagated into the built-in inference network. This is done by filling an evidence frame which contains the evidence names and the corresponding slots for the evidence strengths. For each hypothesis, a belief function which accumulates all the evidence can be computed recursively from the evidence frame following the inference network. Finally, N belief functions are calculated for N hypotheses generated according to the N pairs of indices. These are put into a hypothesis frame for hypothesis selection. The decision maker or the hypothesis selector examines that frame to find the maximum belief function value and compares it to a threshold specified in the knowledge base. A hypothesis is accepted if and only if its belief function is above the threshold and if its value is maximal among the alternatives. The other hypotheses are rejected. The selected hypothesis invokes a curve fitting procedure to generate the digitization of a link between the endpoints of two stream segments. After that, the relevancy finder will pick up the next open node by searching the inverted index downward from the current position. Then the controller will repeat the above processes, until the inverted index is exhausted. A complete drainage *PDG* and its corresponding drainage networks can now be produced.

5.3 Curve Interpolation

If a hypothesis is selected that requires reconnections among the valley segments, then the next problem is to determine how the connections should be made at the pixel level. One way is to extend the end of one segment toward the other pixel by pixel. However, it is easy to become lost in the middle if the gap is wide, if the information in the gap is very ambiguous, or if there is missing information in the gap due to occlusion. To overcome this difficulty, the DNESYS system uses a parametric cubic spline to connect segments smoothly by maintaining continuous curvature at segment endpoints.

The expression of a single parametric cubic spline segment $\boldsymbol{\alpha}$, in terms of arc length S, is given by

$$\begin{aligned}\boldsymbol{\alpha}(S) &= \sum_{i=1}^{4} \mathbf{B}_i S^{i-1} \\ &= \mathbf{B}_1 S^0 + \mathbf{B}_2 S + \mathbf{B}_3 S^2 + \mathbf{B}_4 S^3\end{aligned} \tag{5.3.1}$$

where $0 \leq S \leq S_{max}$.

Let $\boldsymbol{\alpha}(S_k) = \boldsymbol{\alpha}_k$ and $\boldsymbol{\alpha}(S_{k+1}) = \boldsymbol{\alpha}_{k+1}$, where S_k and S_{k+1} are two adjacent point values within the range of parameter S. Then $\boldsymbol{\alpha}'_k$ is the value of the first derivative of $\boldsymbol{\alpha}(S)$ with respect to S at $S = S_k$, etc.. Four equations can be obtained by taking derivatives of Equation (5.3.1) with respect to S and setting $S = S_k$ and $S = S_{k+1}$ respectively.

$$\boldsymbol{\alpha}(S = S_k = 0) = \boldsymbol{\alpha}_k = \mathbf{B}_1$$

$$\boldsymbol{\alpha}(S = S_{k+1}) = \boldsymbol{\alpha}_{k+1} = \mathbf{B}_1 + \mathbf{B}_2 S_{k+1} + \mathbf{B}_3 S_{k+1}^2 + \mathbf{B}_4 S_{k+1}^3$$

$$\boldsymbol{\alpha}'(S = S_k = 0) = \boldsymbol{\alpha}'_k = \mathbf{B}_2$$

$$\boldsymbol{\alpha}'(S = S_{k+1}) = \boldsymbol{\alpha}'_{k+1} = \mathbf{B}_2 + 2\mathbf{B}_3 S_{k+1} + 3\mathbf{B}_4 S_{k+1}^2$$

The coefficient vectors $\mathbf{B}_i$ are then solved from above equations:

$$\mathbf{B}_1 = \boldsymbol{\alpha}_k \tag{5.3.2.a}$$

$$\mathbf{B}_2 = \boldsymbol{\alpha}'_k \tag{5.3.2.b}$$

$$\mathbf{B}_3 = \frac{3(\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k)}{S_{k+1}^2} - \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}} - \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}} \tag{5.3.2.c}$$

$$\mathbf{B}_4 = \frac{2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})}{S_{k+1}^3} + \frac{\boldsymbol{\alpha}'_k}{S_{k+1}^2} + \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}^2}. \tag{5.3.2.d}$$

Given N data points through which the curve must pass, to generate the curve position vector $\boldsymbol{\alpha}_i(S)$, $1 \leq i \leq N$, we need to know the tangent vector $\boldsymbol{\alpha}'_i$. In DNESYS, two points at the end of each segment are chosen for fitting the interpolating cubic as shown in Figure 7.

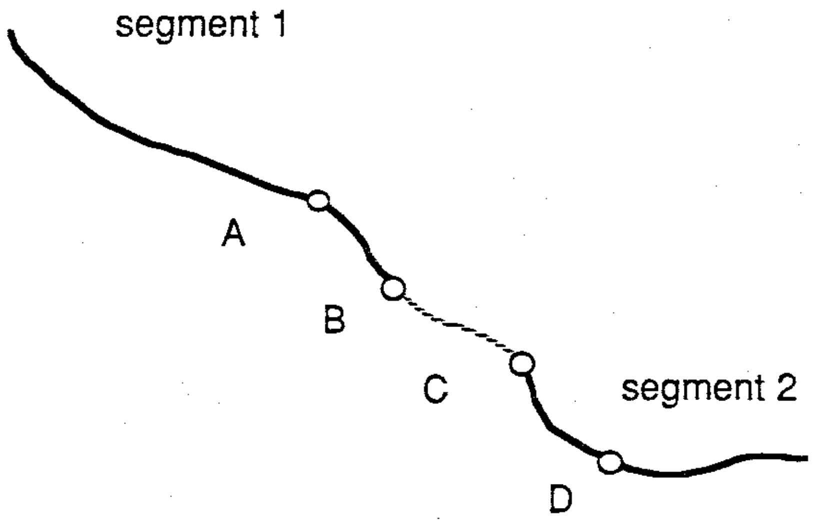


Figure 7. *Control points in the curve fitting.*

By using the natural end condition, $d^2|\boldsymbol{\alpha}|/dS^2 = 0$ [20], we have

$$\boldsymbol{\alpha}'_1 + \frac{1}{2}\boldsymbol{\alpha}'_2 = \frac{3}{2}\left(\frac{\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1}{S_2}\right) \tag{5.3.3.a}$$

$$S_3\boldsymbol{\alpha}'_1 + 2(S_3 + S_2)\boldsymbol{\alpha}'_2 + S_2\boldsymbol{\alpha}'_3 = \frac{3}{S_2S_3}(S_2^2(\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2) + S_3^2(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1)) \tag{5.3.3.b}$$

$$S_4\boldsymbol{\alpha}'_2 + 2(S_3 + S_4)\boldsymbol{\alpha}'_3 + S_3\boldsymbol{\alpha}'_4 = \frac{3}{S_3S_4}(S_3^2(\boldsymbol{\alpha}_4 - \boldsymbol{\alpha}_3) + S_4^2(\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2)) \tag{5.3.3.c}$$

$$2\boldsymbol{\alpha}_3' + 4\boldsymbol{\alpha}_4' = \frac{6}{S_4}(\boldsymbol{\alpha}_4 - \boldsymbol{\alpha}_3). \tag{5.3.3.d}$$

The above system can be written in matrix form as

$$\mathbf{C}\boldsymbol{\alpha}' = \mathbf{B}, \tag{5.3.4}$$

where $\mathbf{C}$ is a 4×4 matrix, $\boldsymbol{\alpha}'$ is a 4×1 variable matrix, and $\mathbf{B}$ is a 4×1 matrix. Then the matrix equation can be solved for $\boldsymbol{\alpha}'$ by matrix inversion. Substituting $\boldsymbol{\alpha}'$ into Equation (5.3.1), one obtains the coefficient matrix $\mathbf{B}$. Thus, the cubic spline segment is specified by

$$\begin{aligned}\boldsymbol{\alpha}(S) = \boldsymbol{\alpha}_k + \boldsymbol{\alpha}_k' S + (&\frac{3(\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k)}{S_{k+1}^2} - \frac{2\boldsymbol{\alpha}_1'}{S_{k+1}} - \frac{\boldsymbol{\alpha}_2'}{S_{k+1}})S^2 \\ &+ (\frac{2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})}{S_{k+1}^3} + \frac{\boldsymbol{\alpha}_1'}{S_{k+1}^2} + \frac{\boldsymbol{\alpha}_{k+1}'}{S_{k+1}^2})S^3\end{aligned} \tag{5.3.5}$$

where $0 \leq S \leq S_{k+1}$ and $1 \leq k \leq 3$.

5.4 *Verification*

Curve $\boldsymbol{\alpha}$ is a continuous curve superimposed on a discrete sampling grid. By tracing this curve, grid points near the curve are marked as component pixels for a new discrete segment connecting the endpoints of the original segments. Since most drainage networks are locally smooth, the fitted curve tends to be close to the shape of the true stream segment. A problem that arises from the computation of the spline independent of external constraints is that the generated curve may overlap with a few ridge points. This may occur if the resolution of the original DEM data is too coarse, if the generated curve bridges segments near the flow sources, or if the gap between the two segments is too wide. In fact, the real drainage network patterns are often more variable than those of the abstract model in our knowledge base. This is not surprising since natural streams are not so smooth and well behaved as the model.

The overlap problem can be solved by adding a verification stage to the reasoning process. Each hypothesis that has been selected by high level reasoning is verified by evidential reasoning at the pixel level under the guidance of the high level results. A belief function is computed for each pixel near those that are inserted as well as for those on all the selected valley segments.

The belief function selects between exactly two hypotheses for each selected pixel–stream or no stream, and the stream paths are corrected based on the results. If a ridge line intersects a valley segment, the entire segment may be removed. The evidence for the verification stage consists of:

1. The high-level reasoning results.

2. The V-shaped map.

3. The A-shaped map.

4. The flow-direction map.

5. The elevation data.

A pixel is verified as a component of a drainage network if its belief value is higher than a preset threshold. The hypothesis verification at the low level provides an opportunity to correct errors made at higher levels, and final decisions are based on reasoning both at high levels and at low levels. The reasoning process in the verification stage is similar to the high level reasoning process, and the details are omitted here due to space considerations.

6 EXPERIMENTAL RESULTS

Two sets of DEM data with different resolutions from the Stanardsville, Virginia and the Hamburg, Virginia USGS 7.5 minute topographic quadrangles were used in our experiments. The data for the Stanardsville area were derived by manual digitization of elevations at the centers of 200 meter grid cells, referenced to the UTM grid system, as shown on USGS 7.5 minute quadrangles. The data for the Hamburg area were collected as a digital elevation model (DEM) by the U.S. Geological Survey [1]. The 7.5 minute DEMs represent digital elevations for areas corresponding to USGS 7.5 minute quadrangles published at 1:24,000. Each DEM represents the earth's topography as a regular array of data referenced to the UTM grid system. Elevations for this grid are derived either by manual or automated compilation of stereo aerial photography at a scale of about 1:80,000. Values derived from the stereo models are then processed to represent elevations on the terrain spaced at intervals of about 30 meters. Each value is an integer representing topographic elevation in meters.

6.1 High Level Reasoning Results

In the first data set, peaks of the Blue Ridge mountains occupy the northwestern corner of the Stanardsville quadrangle, with drainage to the southeast to the upper piedmont surface through South River and Conway River. This drainage forms a dendritic pattern, with relatively steep channel gradients in the mountains, and relatively low gradients once the streams reach the dissected piedmont surface.

The original DEM data of size 70 by 55 pixels are shown in Figure 8.a. The topographic features plotted from the DEM data are shown in Figure 8.b. The V-map which contains the possible valley points marked by the V-shaped operator is shown in Figure 9.a. The A-map which contains possible ridge points marked by A-shaped operator is shown in Figure 9.b. From these two maps, it can be seen that the topography contains pits, saddle points (where valleys and ridges intersect), isolated stream segments, and other ambiguities. Figure 9.c shows the valley segment map after applying the A-map masked thinning algorithm. All the segments in this map satisfy the conditions described in Section 3.1. However, this map also contains inconsistent segments. An incomplete drainage parameterized directed graph is constructed from the this map by the network tracing and property measuring algorithm described in Section 3.2.

The statistics of this incomplete PDG are as following:

Number of drainage trees: 26
Number of drainage trees not terminated on the boundaries: 11
Number of stream branches: 106
Number of inconsistent branches: 3

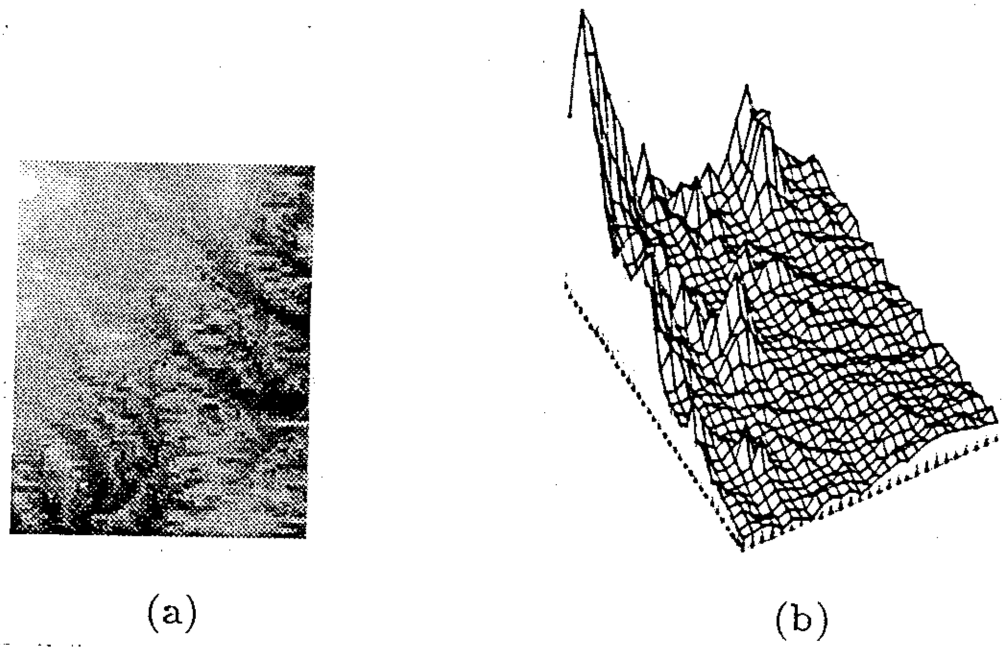


Figure 8. *(a) The original DEM data. Size:* 70 × 55. *(b) The surface plotted from the DEM in (a).*

Number of inconsistent nodes: 92 .

DNESYS then performed the spatial reasoning on the incomplete drainage graph. All inconsistent nodes and branches were examined and removed, and a complete drainage *PDG* was obtained. The resulting map produced by the high-level expert is shown in Figure 9.d.

The statistics for the resulting PDG are

Number of drainage trees: 10
Number of trees not terminated on boundaries: 0
Number of branches: 100
Number of inconsistent nodes: 0
Number of inconsistent branches: 0
Number of curves generated: 15 .

This result shows considerable improvement compared with the result obtained from low-level labeling shown in Figure 9.a. The only difficulty is a segment A that is introduced by a noise point. It will be shown later that this is removed when ridges are taken into account. The number of drainage trees which do not terminate on the boundaries are reduced from 11 to 0, and all nodes are consistent. The drainage network that is generated is consistent with the model of drainage networks described in Section 2 and therefore has the structure of a real drainage network. In particular, it is also comparable to the ground truth data in Figure 10.a. This ground truth was interpreted manually from the blue line

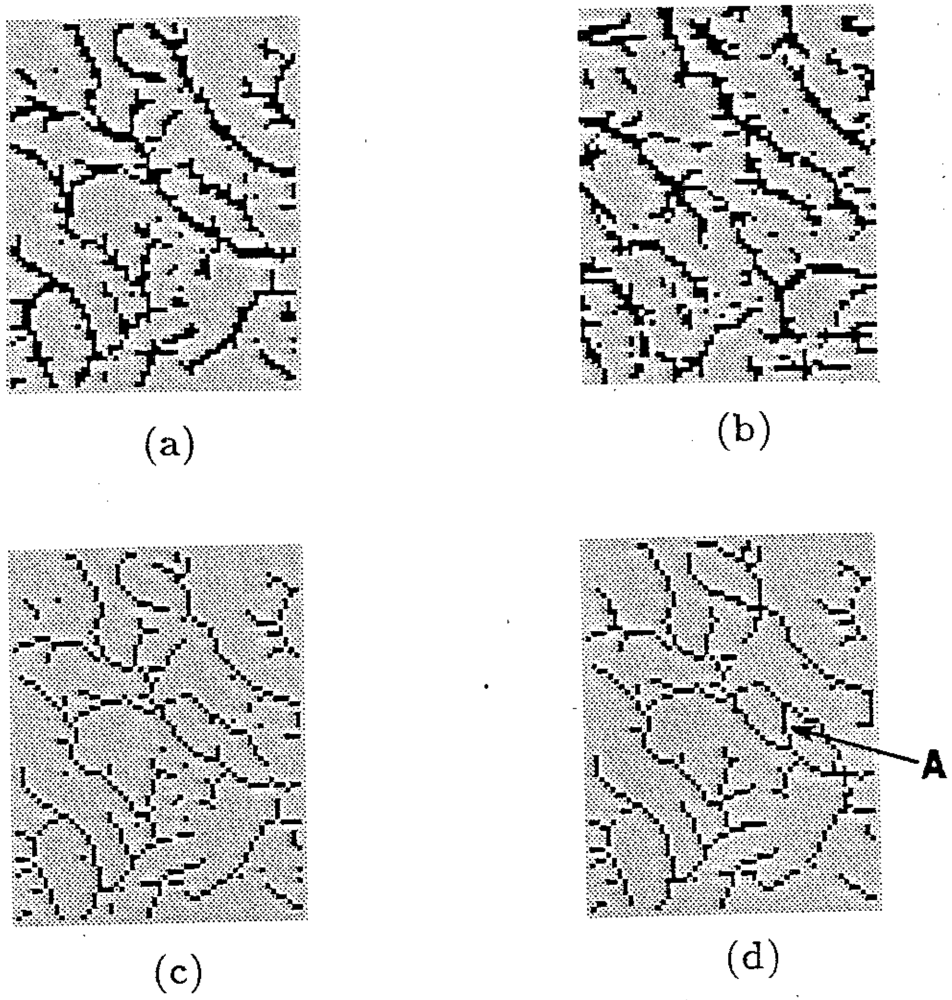


Figure 9. *(a) The V-shaped map for Figure 8.a. (b) The A-shaped map for Figure 8.a. (c) The valley segment map for Figure 8.a. (d) The drainage network after reasoning.*

drainage network of a USGS topographic map with a much finer resolution. The extracted drainage networks with the ridge lines are shown together in Figure 10.b.

The same improvements are obtained from the second set of test data. A 175 × 186 pixel region from the northeast quarter of the Hamburg, VA. quadrangle was tested using DNESYS (Figure 11.a). Topographic features plotted from this DEM data are shown in Figure 11.b.

The same low-level processing operations and the same inference procedures were applied to these test data. The parameters in the knowledge base, however, were slightly modified according to the geographic features in this area. The valley segment map that resulted from the low-level processing is shown in Figure 12.a. An initial PDG was deduced from this map by DNESYS.

The statistics for the initial PDG are:

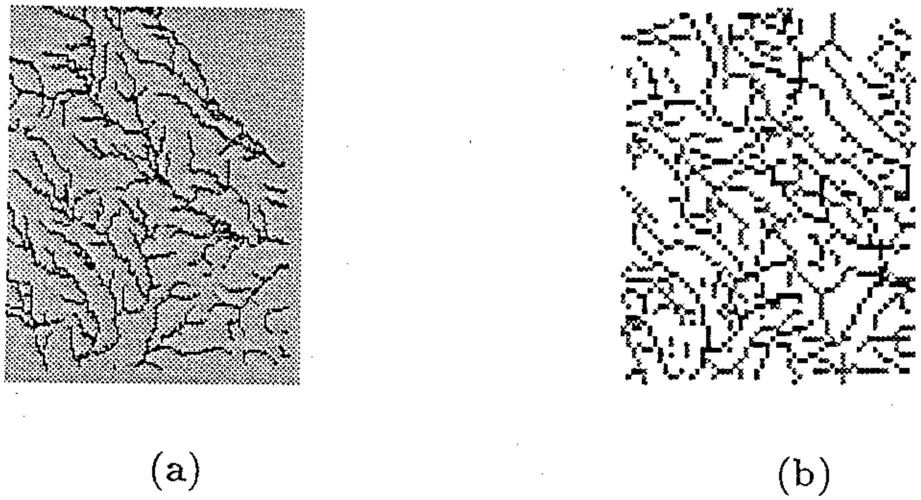


Figure 10. *(a) The ground truth data for Figure 8.a. (b) The extracted drainage network (dark) with ridge lines (light).*

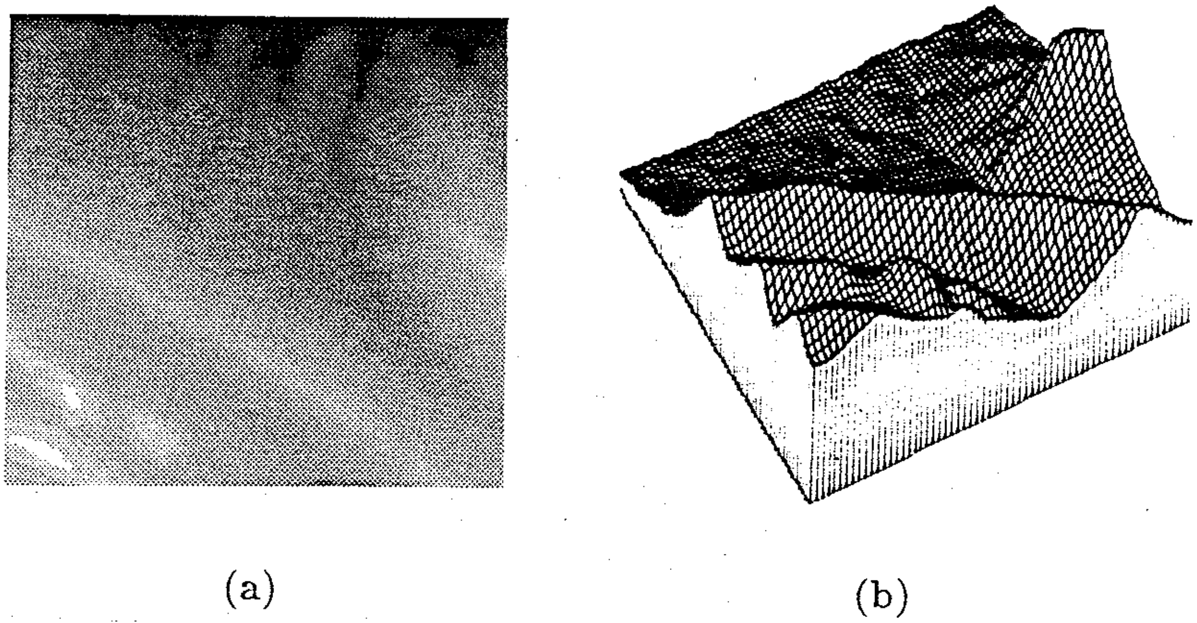
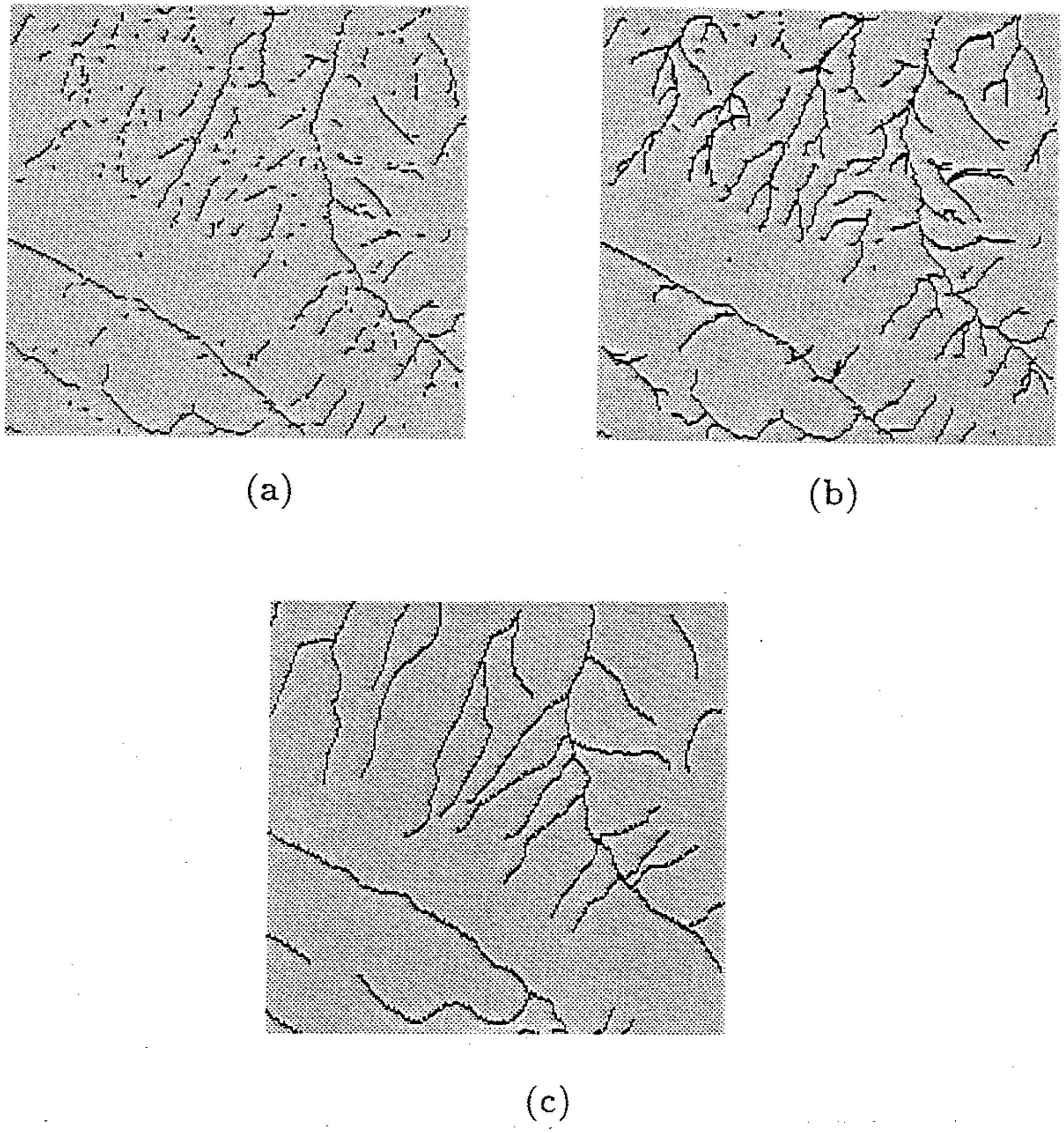


Figure 11. *(a) The DEM of the second test area. Size:* 175 × 186. *(b) The surface plotted from the DEM in (a).*

Number of drainage trees: 198
Number of drainage trees not terminated on boundaries: 180
Number of source nodes: 233
Number of inconsistent nodes: 168
Nunber of inconsistent branches: 21.

The statistics for the resulting PDG are:

Number of drainage trees: 24
Number of drainage trees not terminated on boundaries: 8
Number of source nodes: 124
Number of inconsistent nodes: 8
Number of inconsistent branches: 3.


(a)

(b)

(c)

Figure 12. *(a) The valley segment map of the second test area. (b) The corresponding reasoning result. (c) The ground truth data for Figure 11.a.*

The map corresponding to the resulting PDG is shown in Figure 12.b. From the comparision of the statistics of PDGs and the comparision of the extracted drainage map with the ground truth data (Figure 12.c), we see that DNESYS also works well for the finer resolution data. The number of drainage trees which do not terminate on the boundaries are reduced from 180 to 8. The number of inconsistent nodes are reduced from 168 to 8. The number of inconsistent branches are reduced from 21 to 3. The interpreted network shows more detail than the ground truth network derived from the USGS blue lines. Although these interpretations differ from the ground truth data, we do not consider them to be "errors" because they show more detail than the ground truth data, and the additional detail appears to depict true drainage.

A third area of 160×150 pixels from the northwest quarter of the Hamburg quadrangle was tested on DNESYS (Figure 13.a). The corresponding topographic features are shown in Figure 13.b. The same knowledge base as in the previous test was used. The experimental results are comparable to the results for the first DEM.

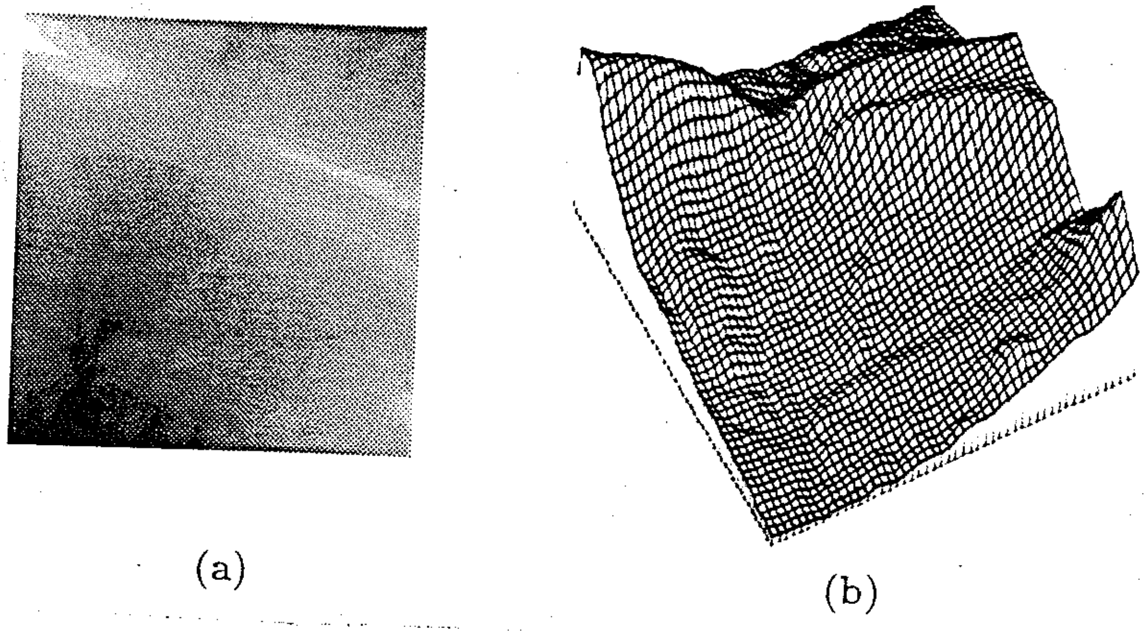


Figure 13. *(a) The DEM of the third test area. Size:* 160 × 150. *(b) The surface plotted from the DEM in (a).*

The statistics of the initial PDG are:

Number of drainage trees: 168
Number of drainage trees not terminated on boundaries: 153
Number of source nodes: 189
Number of inconsistent nodes: 148

Number of inconsistent branches: 16.

The valley segment map that corresponds to the initial PDG is shown in Figure 14.a.

The statistics of the resulting PDG are:

Number of drainage trees: 19
Number of drainage trees not terminated on boundaries: 6
Number of source nodes: 84
Number of inconsistent nodes: 6
Number of inconsistent branches: 1.

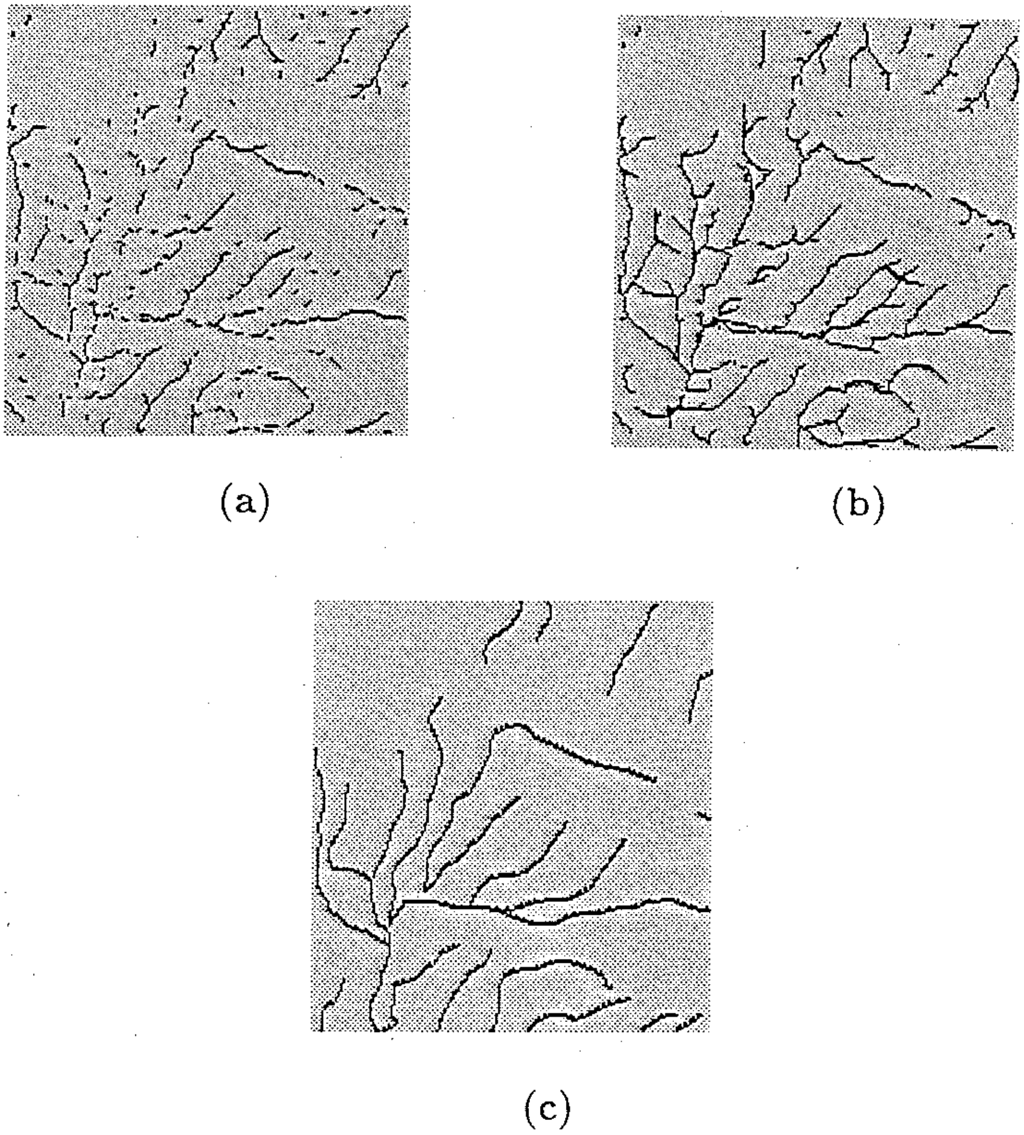


Figure 14. *(a) The valley segment map of the third test area. (b) The corresponding reasoning result. (c) The ground truth data for Figure 13.a.*

The drainage network after reasoning by DNESYS is shown in Figure 14.b. The ground truth data for this map is shown in Figure 14.c. The experimental results again demonstrate substantial improvement. The number of drainage trees which do not terminate on boundaries are reduced from 153 to 6. The number of inconsistent nodes are reduced from 148 to 6. The number of inconsistent branches are reduced from 16 to 1, a very significant improvement.

6.2 Comparison with Earlier Methods

In order to compare these results with other local operators, we have run Jenson's operator on the elevation data for the second test area shown in Figure 11.a. The results are shown in Figures 15.a and 15.b. Our experience is that it is difficult to extract the drainage networks from the result of Jenson's operator. In both figures, most drainage pixels labeled by the operator are connected together to form dense regions within which the stream segments are difficult to identify. Possible reasons for that are

1. Jenson's operator is sensitive to the data resolution.
2. Due to noise, quantization errors, and insufficient accuracy in the raw data, some drainage features are not directly supported by the local features of the raw data.

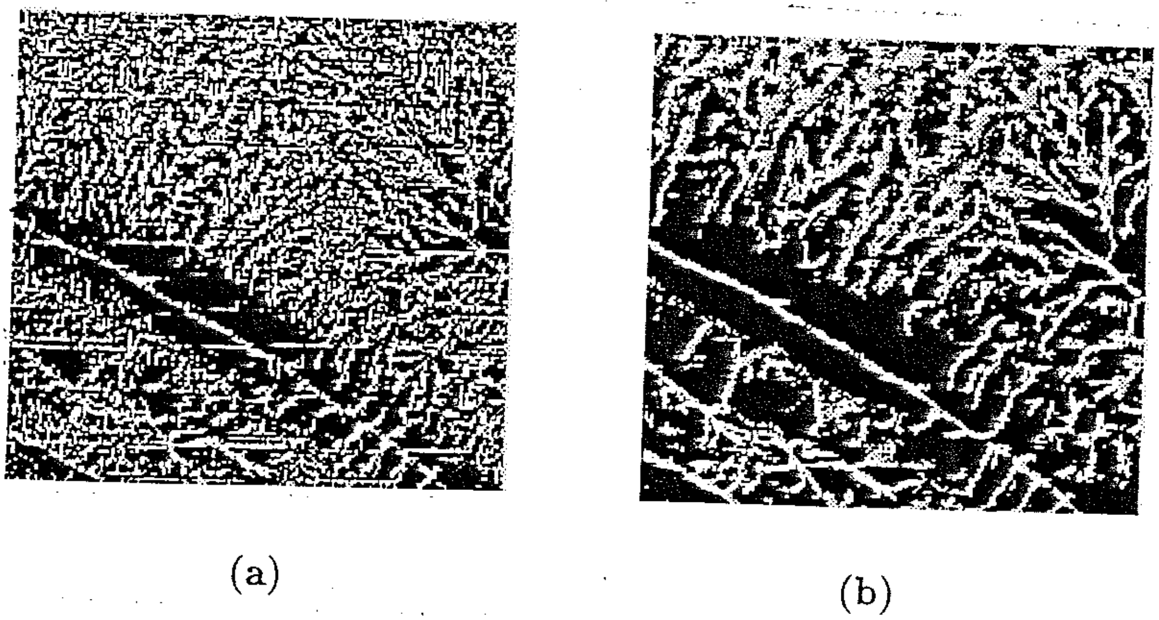


Figure 15. *(a) The drainage cells labeled by Jenson's operator without filtering. Black: drainage cells, White: others. (b) The drainage cells labeled by Jenson's operator with a Gaussian filter. Black: drainage cells, White: others.*

6.3 Results after Verification and Reinterpretation

It has been shown that high level reasoning significantly reduces the errors inherent in low level operators. The errors remaining after high level reasoning are relatively minor and may be categorized by comparing the extracted drainage networks with ground truth data. These errors include:

1. A few isolated segments, which are counted as trees which do not terminate on boundaries, are not connected to the overall system. These errors often occur in regions of low topographic slope where it is difficult to determine the correct drainage direction due to the lack of sufficient evidence.

2. A few true stream segments incorrectly connected to one another. These are counted as inconsistent branches.

3. A few interpreted segments that intersect ridge points. This is caused by the curve fitting algorithm explained in Section 5. However, the ridge points from the low-level labeling, especially the isolated one, are not always reliable.

The results can be further improved by adding the verification stage discussed in Section 5.4 and then making a second pass through the high level reasoning stage of DNESYS. The verification stage produces a new set of valley segments that replaces those from the V-shaped map that were used in the first pass.

To demonstrate the effectiveness of verification, the upper left portion of the second test area of Figure 11.a was processed further. The results are shown in Figure 16. The reinterpreted drainage networks are more accurate than the hypothesized ones and reveal more detail than the digital ground truth data in Figure 12.c. However, by detailed examination of corresponding topographic maps, almost all the additional details are confirmed to be extensions of stream segments into the valleys at their headwaters. To check this, Figure 16.d shows the extensions of the blue line stream map into the valleys, and the match is extremely good.

6.4 Drainage Basin Labeling

Earlier it was asserted that the drainage network was fundamental to the determination of the drainage basins. To confirm this a simple algorithm was tested for labeling the drainage basins using the network from Figure 16.c. The algorithm is a drainage tree thickening algorithm that is used to grow the drainage basins outward from the streams while maintaining the maximum possible consistency with the ridge line fragments that are located between the branches of the drainage trees. Basin labels are propagated from labeled pixels to adjacent pixels with higher elevations. The basins for the streams leaving the image area are shown in different colors in Figure 17.

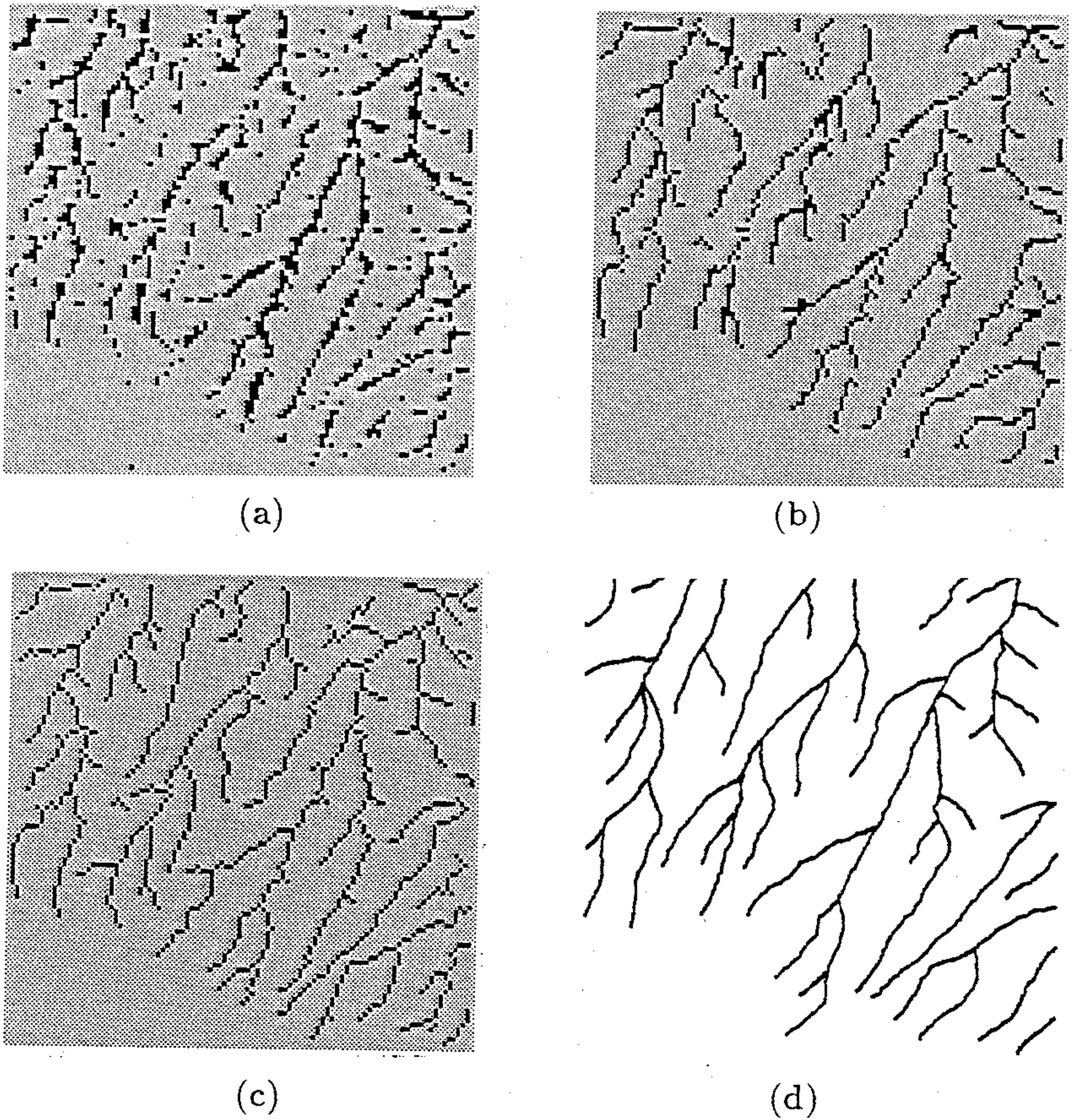


Figure 16. *(a) The low-level labeling. (b) The high-level hypotheses. (c) The networks after verification and reinterpretation. (d) The ground truth map.*

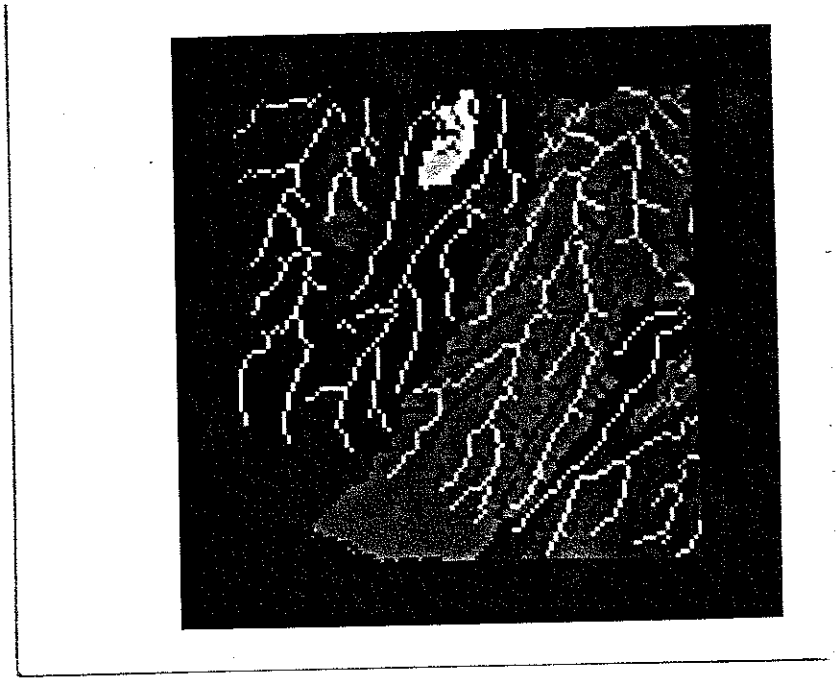


Figure 17. *Drainage basins with drainage network (white) and ridge line fragments (black).*

7 SUMMARY AND CONCLUSIONS

Significant problems are associated with the use of local point operators for extracting drainage systems from DEM data. Performance can be dramatically improved by making use of spatial reasoning. The DNESYS system described here is capable of performing this task at a high level based on organized expert knowledge. It is able to cope with broken stream segments, misconnected stream segments, conflicting flow directions, and other defects which usually result from the low-level labeling process. A drainage parameterized graph (PDG) is constructed to model the drainage networks. The initial drainage point marking and network tracing using spatial relation measuring provides an incomplete drainage parameterized graph for the uncertainty reasoning. By a hypothesis generation and verification process, the system produces a complete drainage parameterized directed graph (PDG) by discarding erroneous information and supplying missing information based the drainage system model. The experimental results show that this expert system performs extremely well. Some additional useful features of the DNESYS system are:

1. it is trainable and the knowledge base is easy to modify;

2. it has the flexibility for the user to choose the desired reasoning object—the specific segment, the specific tree, the specific subarea, or the entire database after it has been constructed from the low-level processing results;

3. it can be used as a system for direct acquisition of drainage systems from DEM maps for use in a geographic information system. The results from the DNESYS system are both in 2-D map format and in database format with all required attributes and distinct labels;

4. the principles, algorithms, and data structures used in the DNESYS system may be extended to other line reasoning systems and shape completion systems, which are important components of computer vision systems.

5. results register to the original elevation data and will have the same resolution as the original data. Therefore the results can be superimposed on the original elevation data.

8. REFERENCES

1 Elassal, A. A. and V. M. Caruso, "Digital Elevation Models," *U. S. Geological Survey Circular 895-B,* 40 p., 1983.

2. O'Callaghan, J. F. and David M. Mark, "The Extraction of Drainage Networks from Digital Elevation Data," *Computer Vision, Graphics, and Image Processing* (28), pp. 323–344, 1984.

3. Jenson, Susan K., "Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Model Data," *USGS Publication*, November 15, 1984.

4. Garland, E. and Roger W. Ehrich, "A GIPSY Primer," Spatial Data Analysis Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, December 1987.

5. Haralick, R. M., J. B. Campbell, and S. Wang, "Automatic Inference of Elevation and Drainage Models from a Satellite Image," *Proceedings of the IEEE* (73), No. 6, pp. 1040–1053, June 1985.

6. Zerniz, E. R., "Drainage Patterns and Their Significance," *Journal of Theoretical Biology* (40), pp. 498–521, 1932.

7. Horton, R. E., "Erosional Development of Streams and Their Drainage Basins, Hydrophysical Approach to Quantitative Morphology," *Geological Society of America Bulletin* (56), pp. 275–370, 1945.

8. Howard, A. D., "Optimal Angles of Stream Junctions: Geometric Stability to Capture and Minimum Power Criterion," *Water Resources Research* (7), No. 4, pp. 863–873, 1971.

9. Roy, A. G., "Optical Angular Geometry Model of River Branching," *Geographical Analysis* (15), No. 2, pp. 87–96, 1983.

10. Lubowe, J. R., "Stream Junction Angles in the Dendritic Drainage Patterns," *American Journal of Science* (262), No. 3, pp. 325–339, 1964.

11. Arcelli, C. and G. Sanniti, "A Thinning Algorithm Based on Prominence Detection," *Pattern Recognition* (13), No. 3, pp. 225–235, 1981.

12. Shafer, G., "A Mathematical Theory of Evidence," Princeton, N. J.: Princeton Univ. Press, 1976..

13. Shortliffe, E., "Computer-Based Medical Consultations: MYCIN," American Elsevier Publishing, 1976.

14. Heckerman, D., "Probabilistic Interpretations for MYCIN's Certainty Factor," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemer (Eds.), New York: Elsevier Science Publishers B.V., pp. 167–195, 1986.

15. Baldwin, J. F., "Fuzzy Logic and Fuzzy Reasoning," in *Fuzzy Reasoning and its Application*, E. M. Mamdani and B. R. Gaines (Eds.), London: Academic Press, pp. 133–148, 1981.

16. Goodman,I. and H. Nguyen, "Uncertainty Model for Knowledge-Based Systems," New York: Elsevier Science Publishers B.V., 1985.

17. Duda, R. O., P. E. Hart, and N. J. Nilsson, "Subjective Bayesian Methods for Rule-based Inference Systems," in *Proc. 1976 National Computer Conf.*, pp. 1075–1082, 1976.

18. Lesser, V. R. and L. D. Erman, "A Retrospective View of the HEARSAY-II Architecture," in *Proc. Fifth International Joint Conference on Artificial Intelligence, IJCAI, Cambridge, MA.*, pp. 790–800, August 1977.

19 Minsky, M., "A Framework for Representing Knowledge," in *The Psychology of Computer Vision*, P. H. Winston (Eds.) New York: McGraw Hill, pp. 211–267, 1975.

20. Rogers, D. F. and J. A. Adams, "Mathematical Elements for Computer Graphics," New York: McGraw-Hill, pp. 116–155, 1976.