REMOTE SENSING OF ENVIRONMENT 18:163–175 (1985)

Extraction of Drainage Networks by Using the Consistent Labeling Technique

R. M. HARALICK

Machine Vision International, Ann Arbor, Michigan 48104

S. WANG

Department of Computer Science, George Mason University, Fairfax, Virginia 22030

L. G. SHAPIRO

Machine Vision International, Ann Arbor, Michigan 48104

J. B. CAMPBELL

Department of Geography, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

Automatic deduction of the drainage network direction from Landsat imagery is a problem in remote sensing. The problem can be formulated in the abstract as a consistent labeling problem which is given a set of units (stream segments), possible labels (flow directions), and constraints on the way adjacent stream segments must be labeled. The goal is to find a mapping from their units to the labels that satisfies the constraints. Consistent labeling problems can be solved by tree search algorithms. In this paper, the stream labeling problem is formulated as a consistent labeling problem. The extraction of stream and valley segments from the Landsat images is discussed, and constraints on segments which meet at junctions are given. The tree search algorithm, employing a method called forward checking, is given and is used to determine the flow direction of all the stream segments in a way that is globally consistent with the junction constraints.

I. Introduction

There is a wealth of information in spatial patterns on aerial imagery, but most computer data processing of remotely sensed imagery, being limited to pixel spectral characteristics, does not make use of it. It is a common task for a photointerpreter to examine the spatial pattern on an aerial image and be able to tell the elevation of one area relative to another and be able to interpret the stream network. The problem we address here is how can a computer do this task.

In this paper, we describe how to extract a drainage network from a Landsat scene of mountainous terrain. The problem is not only to locate the stream segments, but also to deduce the flow directions of these segments. In order to locate the stream segments, we use both spectral and spatial information from the Landsat imagery. In order to deduce the flow directions of these segments, we apply the consistent labeling technique (Haralick and Shapiro, 1979; 1980) which requires the defining of a set of constraints for related segments and applying these constraints to all the stream segments.

Before any meaningful spatial reasoning work can be applied to the Landsat imagery or any other kind of imagery, it is necessary to split the reflectance

0034-4257/85/$3.30

information and topographic information which are mixed in the original Landsat imagery. (Eliason et al., 1981; Wang et al., 1983). From the reflectance information, visible stream segments can be detected by procedures described in Sec. 2.

Flow directions of these stream segments can be deduced by defining constraints at junctions based on segment orientations and lengths and finding optimal flow directions satisfying these constraints. Due to the low resolution of Landsat imagery, few visible stream segments can be detected, and few junctions are available. We can increase the number of junctions by including the junctions where valley segments intersect the visible stream segments. Although these small valley segments do not necessarily carry sizeable channels of open water, they do serve as pathways for water flow, and therefore must be organized spatially according to the same logic as are channels of open water. The detection of valley segments is discussed in Sec. 3.

In Sec. 4, the constraints at stream junctions are given, and the model of deducing the flow directions of streams is formalized. This model is a particular instance of the general consistent labeling model which is introduced in Sec. 5. The implementation is described and experimental results are given in Sec. 6. The results are discussed and conclusions presented in Sec. 7.

2. Visible Stream Segments

We examine three areas in southeastern West Virginia. The original Landsat imagery for these three windows are shown in Fig. 1. This area was imaged by the Landsat-1 MSS on 13 April 1976 (scene ID: 5360-14502). Drainage in this region is through tributaries of the New (Kanawha) River, which flows west into the Ohio drainage system. The overall drainage pattern within this region is that of a relatively large channel superimposed over the finer texture of a dentritic pattern formed by smaller streams. Unfortunately, only the large channels can be located by the procedures stated below because 1 pixel in the Landsat imagery represents approximately a 57×80 m area on the ground.

As described in Wang et al. (1983), a four-band material reflectance image can be computed from the original Landsat imagery. Visible rivers can be located by applying the following Alfoldi and Munday process (1978) to this material reflectance image.

1. A band 4 green coefficient x of every pixel is calculated as a ratio: the radiance of band 4 over the radiance sum of bands 4, 5, and 6. Similarly a band 5 red coefficient y is calculated for every pixel. x and y are called Landsat chromaticity coordinates.
2. In this coordinate system, Munday (1974) has determined a curve (Fig. 2) which is the locus of the positions of chromaticity values of water bodies. If, for some pixels, the x, y values calculated in Eq. (1) are close to this curve, then those pixels can be identified as portions of water bodies. Thus, the chromaticity technique permits identification of those rivers and water bodies large enough to dominate the spectral properties of pixels. Because most streams will be much smaller than a Landsat

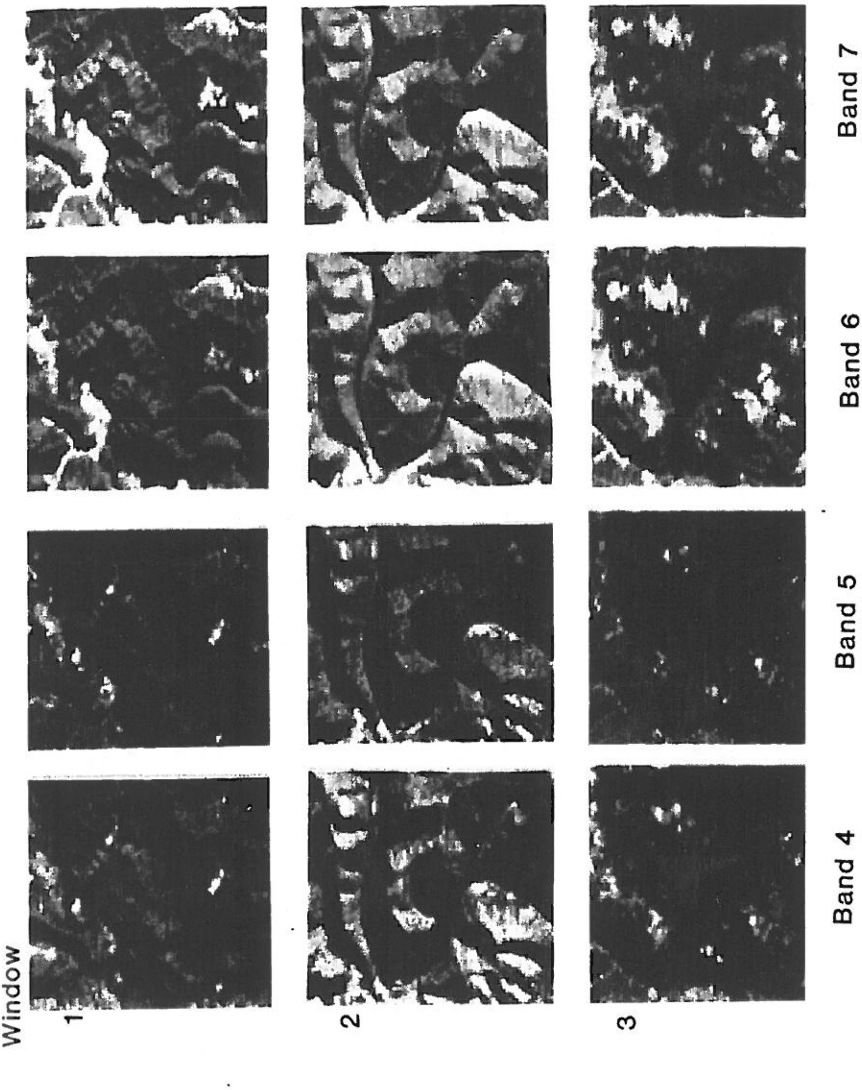

FIGURE 1. Windows of Landsat imagery.

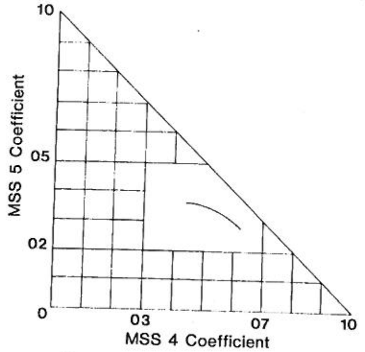

FIGURE 2. Chromaticity plot.

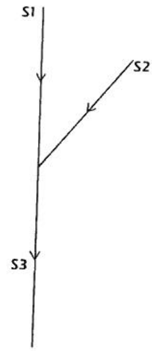

FIGURE 3. One pattern of a stream junction.

pixel, we must employ another method, described below, to identify smaller streams and rivers.

3. Valley Segments

In the well-known Waltz (1975) problem of labeling edges of polyhedra objects, there are only three kinds of edges: convex, concave, and boundary. Of all the possible ways three such edges can meet in a junction there are only 18 legal junction configurations. Similarly in our problem, we are interested in assigning labels of {upstream, downstream} to the visible stream segments by looking at the constraints at junctions. For example, as shown in Fig. 3, when a smaller stream s_2 flows into a larger stream which is composed of two segments s_1 and s_3 because of this intersection, very often the angle between s_2 and s_1 is less than 90°. The general rules about flow directions at junctions are given in Sec. 4. In the following, we describe why and how to detect valley segments.

As discussed before, only the large channels can be located due to the low resolution. In order to have more junctions, we need to include the first order

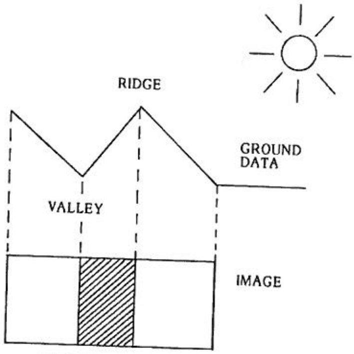

FIGURE 4. Ridge-valley detection.

streams which intersect the large channels. Even though we cannot see these smaller streams in the Landsat imagery, we can detect valley segments instead and assume that water flows through these valley segments which are very close to the large channels.

Valley segments can be detected by the knowledge that sides of hillsides facing the sun must be directly lit and sides of hillsides facing away from the sun must be indirectly lit or in shadow. A directly lit to indirectly lit transition in a direction moving away from the sun is a ridge. An indirectly lit to directly lit transition in a direction moving away from the sun is a

TABLE 1 Rules of Flow Directions at Junctions

PATTERN NUMBER	$A(s_1, s_3)$	$A(s_1, s_2)$	$A(s_2, s_3)$	UPSTREAM	DOWNSTREAM
	$=180°$	$=90°$	$=90°$		
1		$L(s_3) \geqslant \max(L(s_1), L(s_2))$		s_1 and s_2	s_3
2		$L(s_2) \leqslant \min(L(s_1), L(s_3))$		s_2	s_1 or s_3
3		$L(s_3) \leqslant \min(L(s_1), L(s_2))$		s_2 and s_3	s_1
4		$L(s_1) = L(s_2) = L(s_3)$		s_2	s_1 or s_3
5	$=180°$	$<90°$	$>90°$	s_1 and s_2	s_3
6	$=180°$	$>90°$	$<90°$	s_2 and s_3	s_1
7	$<180°$	$\geqslant 90°$	$\geqslant 90°$	s_1 and s_3	s_1
	$>180°$	$\geqslant 90°$	$\leqslant 90°$		
8		$L(s_1) = L(s_2) = L(s_3)$		s_2 and s_3	s_1
9		$L(s_2) \leqslant \min(L(s_1), L(s_3))$		?	
10		$L(s_1) \leqslant \min(L(s_2), L(s_3))$		s_2 and s_3	s_1
11		$L(s_3) \leqslant \min(L(s_1), L(s_2))$		?	
	$=120°$	$=120°$	$=120°$		
12		$L(s_1) = L(s_2) = L(s_3)$		?	
13		$L(s_2) \leqslant \min(L(s_1), L(s_3))$		s_1 and s_3	s_2
14		$L(s_1) \leqslant \min(L(s_2), L(s_3))$		s_1 and s_3	s_2
15		$L(s_3) \leqslant \min(L(s_1), L(s_2))$		s_1 and s_2	s_3

valley. This is illustrated in Fig. 4. Then valleys and ridges exist on the border between shadowed and directly lit areas. The shadowed areas can be located by calculating a binary shadow image based on clustering (Wang et al., 1983).

4. Constraints at Junctions

It is believed that when several stream segments join at a junction, there are constraints based on orientation and length patterns. The general rules about flow directions at junctions are given in Table 1 which is designed by J. B. Campbell. $A(s_1, s_2)$ indicates the clockwise angle between segments s_1 and s_2 and $L(s)$ indicates the length of a segment s.

We are interested in two kinds of junctions. Junctions of the first kind are vertexes at which three stream segments meet. The set of such junctions is called J_3. Junctions of the second kind are vertexes at which two stream segments and one valley segment meet. The set of such junctions is called J_2. Also we call S the set of all stream segments and V the set of all valley segments.

Let $J = J_3 \cup J_2$, X be the set of junction patterns in Table 1, and $L = \{\text{upstream, downstream}\}$. Then one can define $a\colon J \to X$ as the function that assigns junction patterns to junctions. As an example, for the junction j in Fig. 3, $a(j) = 5$ because $A(s_1, s_3) = 180°$, $A(s_1, s_2) < 90°$, and $A(s_2, s_3) > 90°$. For each pattern in Table 1, the flow directions of segments s_1, s_2, s_3 can be put in a triple which is an element in $L \times L \times L$. Thus the mapping from pattern numbers to flow directions can be defined a $b\colon X \to L \times L \times L$. For the junction j in Fig. 3, $b(a(j)) =$ (upstream, upstream, downstream).

If three stream segments meet at a junction, two constraint relations can be formally stated on the basis of Table 1 as follows. One is concerned with all triples of stream segments that constrain each other because they meet at this junction; the other is concerned with all triples of segment-label pairs where the stream segments meet in a junction and the labels are possible for that type of junction. For each x in X, we can define T_x and R_x as follows:

$$T_x = \{\{s_1, s_2, s_3\} | s_1, s_2, s_3 \in S, \text{ and } s_1, s_2, s_3$$

$$\text{meet at a junction } j \text{ of type } x, j \in J_3\},$$

$$R_x = \{\{(s_1, \ell_1), (s_2, \ell_2), (s_3, \ell_3)\} | \{s_1, s_2, s_3\} \in T_x, \text{ and } (\ell_1, \ell_2, \ell_3) \in b(x)\}.$$

If two stream segments and one valley segment meet in a junction, two similar constraint relations can be stated as follows. For each x in X,

$$T'_x = \{\{s_1, s_2\} | s_1, s_2 \in S$$

and there exists $s_3 \in V$ such that s_1, s_2, s_3 meet at a junction j of type $x, j \in J_2\}$,

$$R'_x = \{\{(s_1, \ell_1), (s_2, \ell_2)\} | \{s_1, s_2\} \in T'_x, (\ell_1, \ell_2, \ell_3) \in b(x)\}.$$

Now let

$$T = \left(\bigcup_{j \in J_3} T_{a(j)}\right) \cup \left(\bigcup_{j \in J_2} T'_{a(j)}\right)$$

and

$$R = \left(\bigcup_{j \in J_3} R_{a(j)}\right) \cup \left(\bigcup_{j \in J_2} R'_{a(j)}\right),$$

which means T consists of all triples or pairs of stream segments that constrain each other at junctions and R is the corresponding segment-label constraint relation.

Now the labeling problem of assigning {upstream, downstream} to all stream segments can be described by a compatibility model (S, L, T, R), which is a particular instance of the general consistent labeling problem. Because there are many spatial inference problems which are instances of the consistent labeling problem, the form of the general consistent labeling problem as given by Ullman et al. (1982) is reviewed in the next section.

5. Consistent Labeling

Let U be a set of objects called units, and L be a set of possible labels for those units. Let $T \subseteq \{f \subset U\}$ be the collection of those subsets of units from U that mutually constrain one another. That is, if $f = \{u_1, u_2, \ldots, u_k\}$ is an element of T, then not all possible labelings of $u_1, \ldots, u_k$ are legal labelings. Thus there is at least one label assignment $l_1, l_2, \ldots, l_k$ so that u_1 having label l_1, u_2 having label $l_2, \ldots$, u_k having label l_k is a forbidden labeling. T is called the unit constraint set. Finally, let $R \subseteq \{g | g \subseteq U \times L, \ g$ single-valued, and $\mathrm{Dom}(g) \in T\}$ be the set of unit-label mappings in which constrained subsets of units are mapped to their allowable subsets of labels. If $g = \{(u_1, l_1), (u_2, l_2), \ldots, (u_k, l_k)\}$ is an element of R, then $u_1, u_2, \ldots, u_k$ are distinct units, $\{u_1, u_2, \ldots, u_k\}$ is an element of T meaning $u_1, u_2, \ldots, u_k$ mutually constrain one another, and u_1 having label l_1, u_2 having label $l_2, \ldots$, and u_k having label l_k are all simultaneously allowed.

In the consistent labeling problem, one is looking for functions that assign a label in L to each unit in U and satisfy the constraints imposed by T and R. That is, a consistent labeling is one which when restricted to any unit constraint subset in T yields a mapping in R. In order to state this more precisely, the restriction of a mapping is first defined. Let $h: U \to L$ be a function that maps each unit in U to label in L. Let $f \subseteq U$ be a subset of the units. The restriction $h|f$ (read h restricted by f) is defined by $h|f = \{(u, l) \in h | u \in f\}$. With this notation, a consistent labeling is defined as follows.

A function $h: U \to L$ is a consistent labeling if and only if for every $f \in T$, $h|f$ is an element of R.

An example is given below. Suppose the inputs to the problem are as follows:

$U = \{1,2,3,4,5\}$,
$L = \{a, b, c\}$,
$T = \{\{1\}$, unary constraint
$\{1,2\}$, binary constraints
$\{2,5\}$,
$\{1,3,4\}\}$ ternary constraint,

$R =$
$\{\{(1, a)\}, \{(1, b)\}$, unary constraint
$\{(1, a), (2, a)\}$,
$\{(1, a), (2, b)\}$,
$\{(1, b), (2, b)\}$, binary constraints
$\{(2, a), (5, a)\}$,
$\{(2, b), (5, c)\}$,
$\{(1, a), (3, a), (4, c)\}$, ternary constraints
$\{(1, b), (3, a), (4, a)\}\}$.

Then $h = \{(1, a)\ (2, a)\ (3, a)\ (4, c)\ (5, a)\}$ is a consistent labeling. To see this, note that $h|\{1\} = \{(1, a)\}$, $h|\{1, 2\} = \{(1, a), (2, a)\}$, $h|\{2,5\} = \{(2, a), (5, a)\}$, and $h|\{1,3,4\} = \{(1, a), (3, a), (4, c)\}$ are all elements of R.

If having $l_1, \ldots, l_k$ applied to $u_1, \ldots, u_k$ when $\{(u_1, l_1), \ldots, (u_k, l_k)\}$ is not in R is allowed with a penalty, the process is called inexact consistent labeling (Shapiro and Haralick, 1981). In order to include these mappings, an error weighting function Ew is defined as $Ew: G \to [0,1]$, where $G \subset \{g | g \subseteq U \times L$, g single-valued and $\mathrm{Dom}(g) \in T\}$. $Ew(\{(u_1, l_1), (u_2, l_2), \ldots, (u_k, l_k)\})$ is the error which occurs when labels $l_1, l_2, \ldots, l_k$ are applied to $u_1, u_2, \ldots, u_k$.

If $\{(u_1, l_1), \ldots, (u_k, l_k)\}$ is in R, Ew is zero; otherwise, Ew is a constant ec and usually is defined as the reciprocal of the square of the size of U. The mapping $h: U \to L$ is an inexact consistent labeling if for all f in T, the sum of $Ew(h|f)$ is within some upper bound, usually 1.

6. Relational Reasoning Model and Flow Direction of Streams

In the last section, the very general consistent labeling model was introduced and the unit–label pairs in the elements of R were just assumed to be there. However, if one goes back to the flow direction problem and looks at Table 1, it is clear that one cannot talk about unit–label pairs without looking at the property values, such as angles and lengths, of these units. In the following, based on the very general consistent labeling model, a relational reasoning model is defined to explicitly include these properties. However, these properties are only related to the creation of elements in the set R; the basic tree searching technique is just the same for both the general consistent labeling model and the relational reasoning model. We first show that the relational reasoning model is applicable to the flow direction problem, and then discuss the tree searching strategies designed by Shapiro and Haralick (1981).

In relational reasoning problems, many spectral and geometrical properties can be computed for the locally detected units. Some frequently used properties are average gray level, size, and shape descriptors. For each unit, a list of property values can be computed. Considering all the units, these lists form an array which can be named P. Thus for a unit u, $P[u]$ gives the list of property values for u. For stream junctions, the line length of one segment and the clockwise angle from one segment to the next one can be detected so the $P[s]=$(angle, length) for a unit s. For example, in Fig. 3, $P[s_1]=(45,10)$, $P[s_2]=(135,10)$, and $P[s_3]=(180,15)$.

For each junction pattern in Table 1, the angles must be within certain ranges. With respect to pattern number 5, $P[s_1]$ must be in the property range $([0,89],[1,ub])$, $P[s_2]$ must be in the range $([91,179],[1,ub])$, and $P[s_3]$ must be in the range $([180,180],[1,ub])$ for some upper bound ub on the line length.

However, simply specifying a range for each unit is not enough. Sometimes one needs to compare the property values for different units. One example for the stream junctions is the pattern "$L(s_3) \geqslant \max(L(s_1), L(s_2))$." To handle this type of constraint, a relation $r(P[u_1],\ldots,P[u_k])$ must be defined on the property lists of the related units.

Now the relational reasoning model is a 6-tuple (U, P, L, T, R, Ew). U, L, T, Ew have the same meanings as before; however, the elements in R now have the form $\{(u_1, p_1, l_1),\ldots,(u_k, p_k, l_k), r(P[u_1], P[u_2]\},\ldots, P[u_k]\}$, where p_i is the list of the required ranges of property value for all the properties in P for unit u_i, $i=1$ to k. If the property values of u_i are within the ranges specified by p_i for $i=1$ to k, $\{u_1,\ldots,u_k\}$ is contained in T, and relation r is satisfied, then it is legal to assign label l_1 to $u_1,\ldots, l_k$ to u_k at the same time.

The relational reasoning model (U, P, L, T, R, Ew) can be applied to deduce the flow directions of visible rivers. U contains the units of visible rivers. P contains all the properties detectable from the stream segments. The most important properties are the length of a segment and the orientation of the segment at one end because they are used in Table 1. L is $\{\text{upstream}=1, \text{downstream}=2\}$. T contains the junction relations. R contains the relations of legal flow directions defined in Table 1. For $\{u_1,\ldots,u_k\}$ in T, if $\{(u_1,l_1),\ldots,(u_k,l_k)\}$ is in R, the error function $Ew(\{(u_1,l_1),\ldots,(u_k,l_k)\})$ is defined to be zero; otherwise, it is ec, the reciprocal of the square of the total number of stream segments.

To find the best possible labeling, four different tree searching strategies were described for the inexact consistent labeling problem (Shapiro and Haralick, 1981). Experiments were done to evaluate their performance. Forward checking was found the most efficient one. In the following, the idea of forward checking strategy is described first in English and then followed by mathematical equations. The detailed algorithm designed by Shapiro and Haralick (1981) is listed in the Appendix.

A tree search is performed to find a label for each unit. Each node of the tree represents a possible assignment of a label

l to a unit u. Associated with such a node is (1) the past error, (2) the error of this instantiation, and (3) the future error. Past error consists of the error of the partial mapping defined by the ancestors of this node in the tree. This error is the sum of $Ew(h|f)$ for all $f \in T$ involving past units. Error of instantiation is the error induced by the assignment of label l to unit u.

In a simple backtracking tree search, the error of instantiation is computed at the time l is assigned to u. In a tree search with forward checking, an error table keeps track of how much error the assignment of any label to an uninstantiated unit will generate. This is accomplished by constructing an updated table each time an assignment of l' to u' is made and propagating forward an error to each pair of as-yet-unassigned unit u'' and possible label l'' in that table. The error propagated is that error that would be caused by a simultaneous assignment of u' to l' and u'' to l''.

At any node of the tree, each as-yet-uninstantiated unit has a label in the error table with minimal propagated error. The sum of the minimum error for each such unit is the future error.

If at any node of the tree, the sum of the past error, error of instantiation, and future error is greater than the allowable threshold, then the assignment at this node is not made and backtracking occurs. Otherwise, the error of this assignment is propagated forward and the tree search continues. Details of these are given below.

The inexact consistent labeling problem can be solved by a brute force backtracking tree search. Before the bottom of the tree is reached, only some of the units are labeled, and thus only the error incurred against all units which have already been assigned labels can be calculated. Such a labeling is called a partial labeling; the labeled units are called past units, and the set of all past units is called Up. Similarly, the units which have not been labeled are called future units, and the set of all future units is called Uf. Also let Tl be the set of all sets composed of units which have already been assigned labels, i.e. $Tl = \{\{u_1, u_2, \ldots, u_k\} | u_1, u_2, \ldots, u_k \in Up$ and $\{u_1, u_2, \ldots, u_k\} \in T\}$. Thus the error for past units, ep, incurred in backtracking is

$$ep(Up, h) = \sum_{\substack{\{u_1, \ldots, u_k\} \\ \in Tl}} Ew(\{(u_1, h(u_1)), \ldots, (u_k, h(u_k))\}) \quad (1)$$

for a partial labeling h. If the error sum exceeds an error bound eb, the tree search must either try the next label for the current unit or if there is no next label, it must backtrack.

As a simple example, let $U = \{1, 2, 3, 4, 5\}$, $L = \{a, b, c\}$, $T = \{\{1\}, \{1, 2\}, \{1, 4\}\}$, $R = \{\{(1, a)\}, \{(1, a), (2, b)\}, \{(1, b), (2, b)\}, \{(1, b), (4, b)\}\}$, error constant $ec = 1/6$, and error bound $eb = 0.2$. In the tree search, label a is assigned to 1 first. Thus $Up = \{1\}$, $Tl = \{\{1\}\}$, $ep(Up, h) = 0$ because $\{(1, a)\}$ is in R.

Next, label a is assigned to 2 because backtracking is depth-first. Now $Up = \{1, 2\}$, $Tl = \{\{1\}, \{1, 2\}\}$, $ep(Up, h) = 1/6$ because $h = \{(1, a), (2, a)\}$ is not in R. Since ep is smaller than $eb = 0.2$, one can continue and assign label a to unit 3. As $\{1,3\}$, $\{2,3\}$ are not in T, Tl is not changed and $ep(Up, h)$ is not changed.

Next, label a is assigned to unit 4 which will cause $Up = \{1,2,3,4\}$, $Tl = \{\{1\}, \{1,2\}, \{1,4\}, ep(Up, h) = 1/6 + 1/6$

$= 1/3\}$ because $\{(1, a), (4, a)\}$ is not in R. At this point, $ep = 1/3$, which is larger than $eb = 0.2$, and one cannot continue with unit 5. Instead, next label b is assigned to unit 4. This is the trace for backtracking.

A technique called backtracking with forward checking can improve the speed of tree search. For the previous example, 3 units were assigned labels before a cutoff happened. Actually, by looking at sets T, R and doing some calculations described below, a decision about cutoff can be made even after the first unit is assigned a label. Thus the searching is more efficient. This technique is similar to the branch and bound technique except that a fixed bound value is used.

The speed of tree search can be improved if one also considers the minimum error that the current labeling must incur against future units which have not been assigned labels. Thus the set in T containing only one future unit is of interest; for a future unit u and label l, define

$$T(u, i, Up) = \{\{u_1, \ldots, u_k\} \in T \mid u_i = u \text{ and } n \neq i \text{ implies } u_n \in Up\}.$$

For example, when $Up = \{1\}$, $h = \{(1, a)\}$, $T(2, 2, \{1\}) = \{\{1, 2\}\}$, $T(4, 2, \{1\}) = \{\{1, 4\}\}$.

Using labeling h on all units except u and assigning label l to u, the error (epf, error for past and future units) is $epf(u, l; Up, h)$

$$= \sum_{i=1}^{k} \sum_{\substack{\{u_1, \ldots, u_k\} \\ \in T(u, i, Up)}} Ew(\{(u_1, h(u_1)), \ldots, (u_{i-1}, h(u_{i-1})), (u, l), (u_{i+1}, h), \ldots, (u_k, h(u_k))\}). \quad (2)$$

In the continuing example, if $u = 2$, $l = a$, then $epf(2, a, \{1\}, \{(1, a)\}) = Ew(\{(1, a), (2, a)\}) = 1/6$ because $\{(1, a), (2, a)\}$ is not in R.

To be complete, one should also consider the smallest error of the units in the nodes with higher level numbers in the tree created by backtracking or the units other than u in Uf. It is

$$\sum_{\substack{v \in Uf \\ v = u}} \min_{m \in L} epf(v, m, Up, h), \quad (3)$$

For the continuing example, when $v = 3$

$$\min_{m \in L} epf(3, m, \{1\}, \{(1, a)\}) = 0$$

because $T(3, i, \{1\})$ is always empty. When $v = 4$,

$$epf(4, a, \{1\}, \{(1, a)\}) = 1/6,$$
$$epf(4, b, \{1\}, \{(1, a)\}) = 1/6,$$
$$epf(4, c, \{1\}, \{(1, a)\}) = 1/6,$$
$$\min_{m \in L} epf(4, m, \{1\}, \{(1, a)\}) = 1/6.$$

When $v = 5$,

$$\min_{m \in L} epf(5, m, \{1\}, \{(1, a)\}) = 0$$

for the same reason as when $v = 3$. Now the sum in Eq. (3) is 0 1/6 0 = 1/6.

For current labeling h, if the summation of Eqs. (1), (2), and (3) exceeds an error bound for any label l for the current unit u, then one needs either try the next label for the current unit or backtrack. This is called backtracking tree search with forward checking. From the above calculations, for $u = 2$, $1 = a$, $0 + 1/6 + 1/6 = 1/3 > 0.2$, and one needs to try

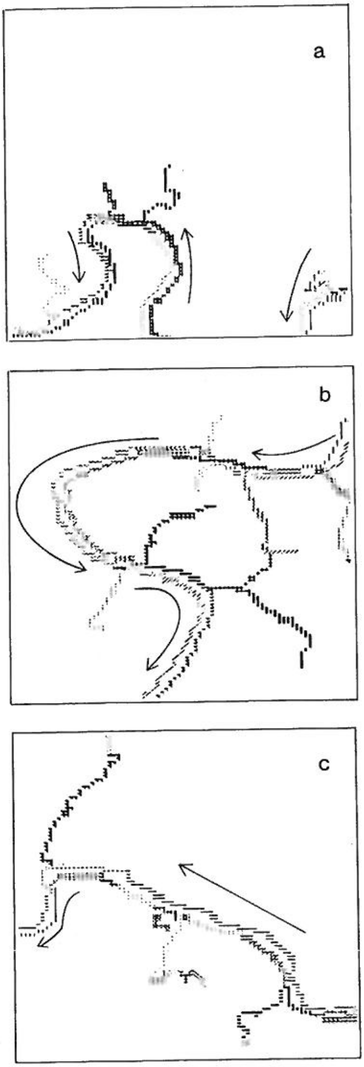

FIGURE 5. Flow directions of streams.

the next label b for current unit 2. Thus, it is clear that only one node is generated in the searching tree as opposed to three nodes in the case of backtracking.

Implementing the model (U, P, L, T, R, Ew) by using the algorithm of forward checking, the flow directions for the three test areas have been deduced. They are labeled as in Fig. 5. The valley segments that are used to help make the decision are also shown in Fig. 5. The flow directions are correct with respect to the ground truth.

7. Discussion

In this research, we have described a set of constraint rules for stream junctions and applied them to the units detected from real world imagery to prove the usefulness of the consistent labeling technique. Even though the numbers of units in the test are not large, the mathematical model of the spatial reasoning model is precise and useful for many applications. The reasons that the numbers of units in the test are not large are:

1. The resolution of the Landsat imagery is low so that few stream segments can be detected. This will not be as great a problem as sensors with higher spatial resolution, such as Thematic Mapper, become available.
2. Even if the resolution is improved and more stream segments can be detected, most flow directions can be determined by inspection of the drainage network. Thus, the only streams left with unknown flow directions would be the largest streams in the imagery and the total number of units suitable for the consistent labeling problem is very limited.

Instead, more challenging tasks for the consistent labeling process can be found in the domain of pattern recognition such as classifying ground objects in urban areas. Hundreds of elements in the unit constraint set T and unit-label constraint set R can be defined because of the diversity of ground objects.

The rules in Table 1 can be improved if we digitize a large set of stream networks from topographic maps and observe the junction patterns in these networks.

Appendix: Forward Checking Algorithm

```
  /* ULTAB and MINERR are stacks, one table per level;
CONTROL: = forward;
while CONTROL = forward or some units have been assigned labels
do begin
  if all units have a label then CONTROL := back;
  if CONTROL = back then back up one level;
  U := next unit to try;
  CONTROL := back;
  while there are labels to try for unit U
    do begin
      L := next label for U;
      PERR := error of partial labeling so far;
      BERR := FORER(ULTAB, U, L);
      FERR := FUTMIN(future units)
      if PERR + BERR + FERR ≤ e then
      begin
        ERRF := UPDATE(ULTAB, U, L, PERR + BERR);
        if UPDATE fails then try next label;
        CONTROL := forward;
        add (U, L) to the partial labeling;
        if all units have labels then stop;
        move forward one level;
      end
    end
  end
/*
procedure FUTMIN(future units);
FUTMIN := 0;
for each future unit UF do
  FUTMIN := FUTMIN + MINERR(UF)
end FUTMIN
/*
procedure UPDATE(ULTAB, U, L, PASTERR);
UPDATE := 0;
for each future unit UF
```

```
do begin
   SMALLERR := 99999.;
   for each label LF that is eligible for UF
   do begin
      if {(U,L),(UF,LF)} is in the unit-label
      constraint relation
      then ERR := 0
      else ERR := WEIGHT(U,UF);
      ULTAB(UF,LF) := ULTAB(UF,LF)+ERR;
      if ULTAB(UF,LF) < SMALLERR
      then SMALLERR := ULTAB(UF,LF)
   end
   UPDATE := UPDATE+SMALLERR
   if UPDATE+PASTERR > e then fail return;
   MINERR(UF) := SMALLERR
end
end UPDATE
/*
```

In the above algorithm, FORER calculates $epf(u, l; Up, h)$ of Eq. (2). MINERR calculate $\min epf(v, m; Up, h)$ of Eq. (3). e is the error bound for tree search, and WEIGHT is the error constant.

References

Alfoldi, T. T., and Munday, Jr., J. C. (1978), Water quality analysis by digital chromaticity mapping of Landsat data, *Can. J. Remote Sens.* 4(2):108–126.

Eliason, P. T., Soderblom, L. A., and Chauez, Jr., P. S. (1981), Extraction of topographic and spectral albedo information from multi spectral images, *Photogramm. Eng. Remote Sens.* 48(II):1571–1579.

Haralick, R. M., and Shapiro, L. G. (1979), The consistent labeling problem: Part I, IEEE *Trans. Pattern Anal. Machine Intelligence* PAMI-1(2):173–184.

Haralick, R. M., and Shapiro, L. G. (1980), The consistent labeling problem: Part 2, *IEEE Trans. Pattern Anal. Machine Intelligence* PAMI-2(3):193–203.

Munday, Jr., J. C. (1974), Lake Ontario Water Mass Determination from ERTSoı, Proc. 9th Intl. Symp. Remote Sensing of Environment, ERIM, Ann Arbor, pp. 1355–1368.

Roy, A. G. (1983), Optimal angular geometry models of river branching, *Geogr. Anal.* 15(2):87–96.

Shapiro, L. G., and Haralick, R. M. (1981), Structural description and inexact matching, *IEEE Trans. Pattern Anal. Machine Intelligence* PAMI-3(5):504–519.

Ullmann, I. R., Haralick, R. M., and Shapiro, L. G. (1982), Computer architecture for solving consistent labeling problem, Proceedings of SSST.

Waltz, D. (1975), Understanding line drawings of scenes with shadows, *In The Psychology of Computer Vision* (P. H. Winston, Ed.), McGraw-Hill, New York.

Wang, S., Haralick, R. M., and Campbell, J. (1983), Relative elevation determination from Landsat imagery, Spatial Data Analysis Lab Technical Report, Virginia Tech, Blacksburg, VA.

Received 5 July 1984; revised 9 January 1985.