An Automated Approach for Labeling Raster Digitized Contour Maps

Jayanta K. Sircar
College of Engineering, Room 1131E, Engineering Classroom Building, University of Maryland, College Park, MD 20742
Juan A. Cebrian
I.E.G.A., C.S.I.C., c/Pinar 25, 28006 Madrid, Spain

ABSTRACT: The substantive preprocessing invariably required by current methods to convert scan digitized contour maps into a digital elevation data base severely limits the use of raster scanning for many practical applications. This paper presents an efficient "raster based" approach to label contour lines digitized from topographic maps. In contrast to the predominantly "vector based" techniques currently in use to scan and capture topographic data from hard-copy contour maps, the proposed method uses a semi-automatic "batch-oriented" strategy. The key to the method is the use of graph theory to reconstruct the topological adjacencies of elevation regions in rasterized contour maps. The approach was successfully tested on a prototype scale in digitizing several map segments with terrain slopes ranging from flat to highly mountainous areas. In addition to a significant reduction in the time required to digitize and label scanned maps, from the order of weeks to a few hours, a major strength of the designed approach is the potential of using the many inexpensive raster scanners that are now easily coupled to desktop microcomputers. The proposed method is therefore not only well suited to reduce the associated costs of digitizing maps on a large scale within large organizations, but is also attractive to the many small and medium sized organizations that have so far not been able to take advantage of available digitizing technology.

INTRODUCTION

ENGINEERS AND EARTH SCIENTISTS involved in terrain analysis recognize the need to obtain digital topographic data in a timely and economic manner. The availability of hardcopy maps at scales suitable for practical engineering provides a basis for an innovative approach to create digital elevation models (DEMs). As summarized by Peuquet and Boyle (1983), the approach relies on "digitizing" or "capturing" the contour lines from topographic maps into a computer.

Most current digital cartographic efforts started by simulating the manual cartographic process. As a consequence, contour lines from maps, e.g., quadrangle maps of the United States Geological Survey (USGS), were digitally scanned and recorded as vectors (Allder *et al.*, 1982). In this process, the digital cartographic data output (e.g., points on a contour line) from a scanner is represented as a $[x,y]$ vector. A key to the creation of the DEM was the "tagging" or "labeling" of the digital contour vectors with the appropriate elevation. During tagging, the vector contour data are physically and logically associated with their analog attribute, elevation. As the tagging procedure requires the contour information in a "vector" data structure, it is frequently called a "vector" based method. Subsequently, interpolation of the tagged contours results in a DEM. Vector-based techniques of tagging invariably rely on a highly labor intensive manual operation and costly computer graphics resources; as a consequence, this constraint has limited its use.

Recent advances in low cost raster scanning techniques have resulted in a renewed interest in identifying less expensive alternatives for quickly digitizing maps. At this time, however, although raster scanning is being widely used to digitally "capture" the contour lines, attempts to solve the topological problem of automated "labeling" continue to rely on vector based methods. This is accomplished by using a conversion between raster and vector formats. Considering the significant organizational and time resources needed for vector based tagging methods, Selden (1985) stated, "...the successful development of automatic feature tagging capabilities in raster scanning systems will emerge as a significant factor in comparing raster and vector systems ... ." Meanwhile, the costly vector tradition of cartographic data processing continues to be a major constraint to the practical use of raster technology.

The ability to automate the task of tagging or labeling the digitized contours with the corresponding elevation in a raster format, without vectorizing the data, is the key to the design of a practical raster-based system. The last two decades of research in the area of automated map digitization have shown that, simple as it is in concept, the replication of the vector labeling operation using a raster format is a complex topological problem (Sircar, 1986).

Clearly, in order to fully utilize the power of a large range of relatively inexpensive raster-based scanning hardware, new topological insights are required in managing raster digitized contour information. In summary, an ability to interpret and manage scanned raster images of contour maps without the costly and time consuming vector-raster conversions is pivotal to the design of a practical contour labeling technique. Thus, the specific objective of the present research was to develop a fully raster-based method to label scan digitized contour lines. Such a method would significantly reduce the cost and effort of many practical applications in the Earth sciences and related fields of engineering, such as obtaining terrain data for hydrologic design models, land planning, and models for water quality assessment.

A GRAPH THEORETIC REPRESENTATION OF CONTOUR LINES IN TOPOGRAPHIC MAPS

THE CONTOUR MAP AS A GRAPH

The search for raster-based algorithms to manage spatial data has resulted in efforts to reconstruct the topology of contour maps using spatial relationships. The close association between spatially adjacent regions separated by contour lines and a graph data structure is intuitive. It parallels the intuitive use of the vector data structure for replicating the cartographic task of labeling a contour line. Boyell and Rushton (1963) first demonstrated the mapping of both contours into links, or edges, of a graph and intercontour spaces into nodes.

Since then, several researchers have made use of the topol-

0099-1112/91/5707-965$03.00/0
©1991 American Society for Photogrammetry and Remote Sensing

ogical equivalence of the nested nature of contours to a "free" tree or a graph structure with no cycles (Figure 1). Thus, paths crossing a sequence of contours move along a set of connected links in the corresponding tree, which provides rapid identification of the contours. One of the first uses of this concept was in the building of a search directory of terrain information.

Morse (1968, 1969) mapped contour lines into "nodes" and intercontour regions into "junctors." This format of notation appears to be a direct consequence of the more popular "vector" mode of map-to-image digitization, which was prevalent at the time. For a long time, the vector mode of digital output constrained the search for more economic strategies of map digitization. Only since the advent of less expensive raster technology for high resolution digitizing had there been some progress in the development of raster-based algorithms.

Mark (1978) reported the potential of the graph structure as a representation device of the shape of topographic surfaces and their parallelism with the surface networks (relating peaks, passes, and pits) of Pfaltz (1976). In the same paper, Mark suggests interesting applications of the concept to contouring problems. In a construct similar to that of Merrill (1973), Mark (1978) mapped contour lines into edges or links and intercontour regions into nodes of a graph. Later, Mark (1986) generated a graph or tree topology by considering the spatial adjacency of both contour lines and the intercontour spaces of the map. Figure 2a depicts a contour map and Figure 2b its free tree representation using Mark's (1986) approach. The contour lines are identified as L1, ..., L6 and the intercontour regions as R1, ..., R7.

Roubal and Poiker (1985) first reported the use of contour tree structures in automating the process of labeling the digitized contour line. Their method (Figure 2c) focused on defining a graph structure using the topological adjacency of only contour lines as nodes in the graph. However, the representation of Roubal and Poiker results in the loss of its tree structure for even very simple contour maps (Sircar, 1986). The approach described herein has used a representation of contour lines in a graph structure similar to that of Mark (1986).

An Oriented Tree Representation for Contour Maps

The present method uses insights gained by the free-tree representation of contour maps. The nested structure of intercontour regions or elevation regions enclosed by contour lines in a map is closely analogous to that of an "oriented tree." Given a set of vertices or nodes (V) in a graph, an oriented tree (Knuth, 1976) is defined as a "directed graph" with a specified vertex or root node, R, such that (i) each vertex (node) $V \neq R$ is the initial vertex (node) of exactly one edge (link or arc), denoted by e[V]; (ii) R is the intial vertex of no edge; and (iii) R is a root in the sense defined above (i.e., for each vertex $V \neq R$ there is an oriented edge or path directed from V to R). In a contour map the root may be formed by the lowest or highest elevation region.

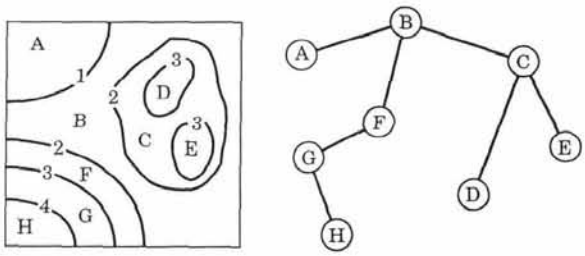


FIG. 1. An example of a contour map segment and the corresponding graph representation in the form of a free tree.

For the set of all intercontour regions within a map, there is a corresponding equivalent binary relationship ($=<$) created by decreasing or increasing elevations across contours. This is a partial ordering relation, i.e., reflexive, antisymmetric, and transitive. The set of intercontour regions or "elements" may be addressed as a partially ordered set, or poset. Using the free-tree concept as a basis, each element of the poset (or intercontour region) is initially made a node on the contour graph, and the link or edge between nodes or elements of the poset are the contour lines (values) of elevations that separate them.

Every poset has two subsets that are specially relevant: the set of maximals and the set of minimals. A maximal is an element that does not have an "upper" (higher in elevation in this case) element, while a minimal is an element that does not have a "lower" (lower in elevation) element in the set. By rooting the tree at a minimal or maximal and by imposing an "upwards" or "downwards" sense of direction in the graph, the poset formed by the intercontour regions in a map becomes analogous to an oriented tree. In the propsosed method, regions of lowest elevation in all nested sets of contours in a map (or minimals) are treated as root nodes of the contour graph.

It is possible that the contour graph be (i) an oriented tree with a global root (Figure 3a) equivalent to a poset with one minimal, region "A," or (ii) a number of trees sharing some nodes, in which case there are more than one global root, such as intercontour regions labled "1," "7," "10," and "13" (Figure 3b). Nodes of type "F," "G," and "I" in Figure 3a and "4" in Figure 3b are examples of maximals. In contrast, root node types "A" in Figure 3a and "1," "7," "10," and "13" of Figure 3b are minimals.

A COMPUTER REPRESENTATION OF THE LABELING PROBLEM

Given a map, it is relatively simple to use the insights developed by an oriented graph or tree representation of contour maps and manually construct and label the contour edges. However, the equivalent problem for the computer is three-

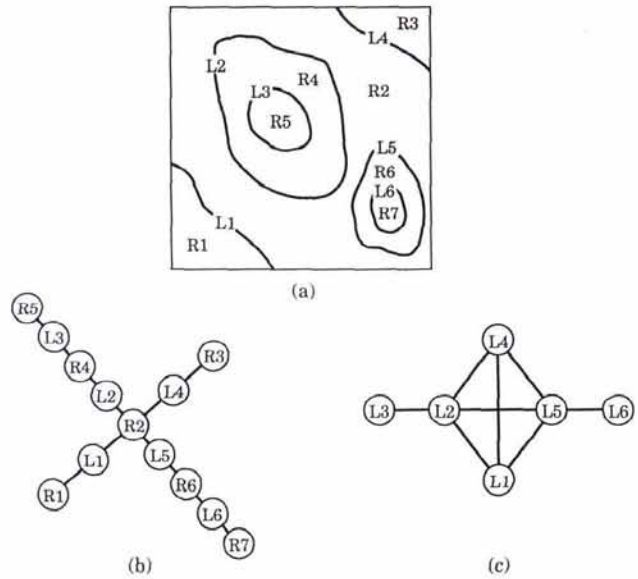


FIG. 2. A contour map and the corresponding free tree depicting the spatial adjacencies of intercontour regions and the contour lines. (a) A contour map segment. (b) The contour tree using the approach of Mark (1986). Both intercontour regions and contour lines are used as nodes in the graph structure. (c) The contour graph using Roubal and Poiker's approach (1985). Only contour lines are treated as nodes in the resulting graph.

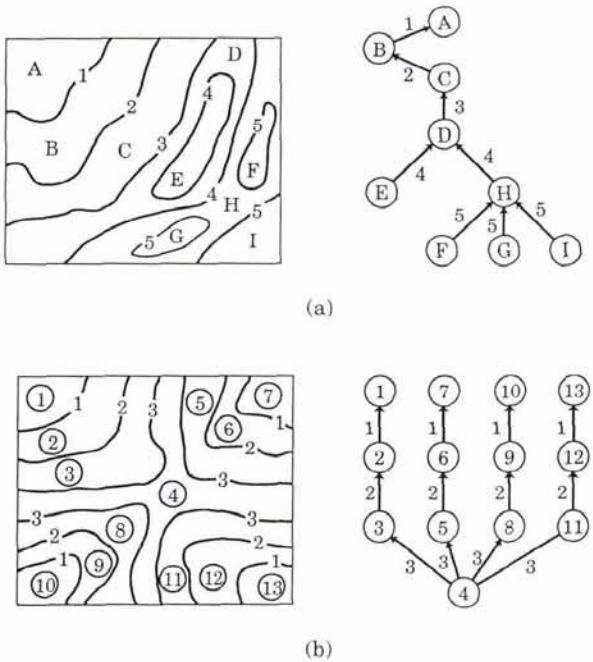


FIG. 3. Oriented tree representation of contour maps. (a) Map with a single global minimal (root). (b) An instance of a map with multiple nodes as distinct minimals (roots).

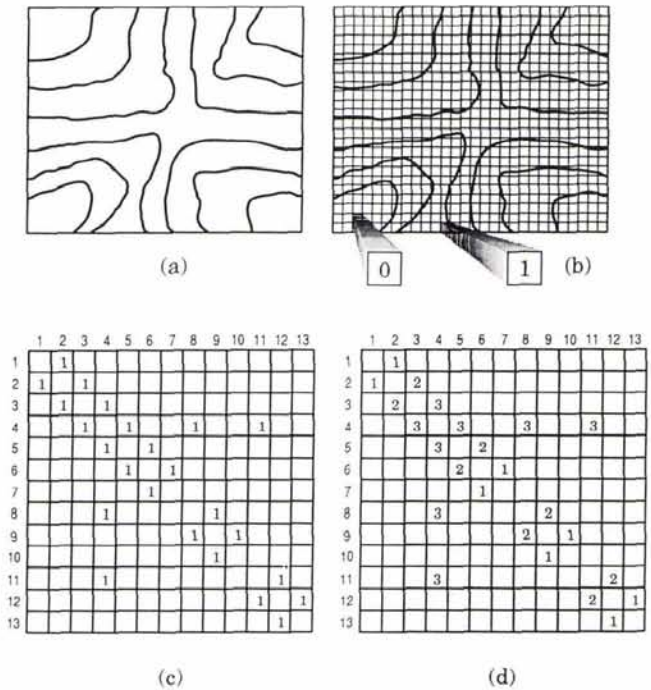


FIG. 4. Stages in the construction of a contour tree using as an example the map segment depicted in Figure 3b. (a) Example of black and white contour line drawing separate input to the scanning process. (b) Format of the corresponding binary output raster of scanned line drawing. (c) Represents the undirected adjacency matrix (free tree) representation of the intercontour regions, (d) The labeled adjacency matrix (oriented tree) representation.

fold: (1) to convert the information captured from a raster-scanned map into a computer representation of the graph or tree-like topology; (2) to provide a mechanism to label the edges or links with the correct value of the elevation; and (3) to reconstruct the digital raster, using the computer representation of the contour graph, with the value at each contour pixel replaced by the elevation of the contour line it represents.

CONSTRUCTING THE CONTOUR TREE

The input data for implementing the approach discussed in the present paper is a raster-scanned black-and-white contour separate (Figure 4a) of a topographic map with gaps and labels removed. The technology to produce a binary representation of the contour lines in a pixel-based image frame is well established (Sircar, 1986; Peuquet and Boyle, 1983). Each contour in the original hard copy or "analog" map is translated into a binary digital raster, as shown in Figure 4b. The output of the scanning process results in a raster with either an "absence (marked 0)" for a non-contour pixel or map background and a "presence (marked 1)" for a contour pixel. A set of connected "1" pixels represent a contour line.

Among the various methods available to logically represent the graph structure in a machine, an approach in routine use is that of the adjacency matrix (Sircar, 1986). The adjacency matrix, which is equivalent to the graph in Figure 3b, is illustrated in Figure 4c. The adjacency of the region in the map labeled "1" to the region labeled "2" is indicated by the presence of a "1" at the Row 1, Column 2 address of the matrix shown in Figure 4c. Stated in the terminology of graph theory, the "children" of node (region) "4" in Figure 3b are nodes (regions) "3, 5, 8, and 11."

Using the definitions stated above, the problem of "labeling" a contour line may be restated as the problem of labeling the edges of a graph with an appropriate value of the elevation. For example, if the elements of the matrix of Figure 4c were replaced with the value of the corresponding elevations, the labeling problem would be complete (Figure 4d). As shown in figure 4d, which is the computer representation of the labeled graph, the value "2" at the matrix element (2,3) indicates that region "2" and region "3" in the map are separated by a contour value of "2 ft."

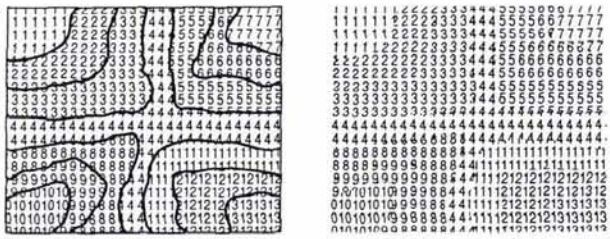
For the sake of simplicity, it is assumed that contour lines do not intersect and that there are no gaps along a line. Although in practical applications the assumption of contours without gaps and the absence of merged lines is a non-trivial one, it remains a requirement for both vector- and raster-based technology.

In practice, in the vector case, most commercially available systems attempt to provide only manual editing tools for those instances where the assumptions are not valid. In the raster approach, similar steps may be taken to reduce the time consuming burden of manual inspection and editing. Later in this paper, a discussion has been included on how the current approach may also enhance the efficiency of techniques in use to solve such problems.

There are three principal steps in constructing an adjacency matrix from a raster digitized image. First, the regions bounded by contours should be labeled as shown in Figure 5a. This can be done by applying a connected components algorithm to label every inter-contour pixel in the raster. Second, all of the contour boundaries ("1"s) of the raster output from the first step should be eroded such that intercontour region tags are rendered directly adjacent to each other (Figure 5b). Third, the raster output from the previous step should be traversed to examine adjacencies among pixels and generate an adjacency graph of all of the connected components in the image (Figure 4c); this is logically equivalent to the free tree.

LABELING THE CONTOUR ADJACENCY GRAPH

Given a set of minimals to be treated as roots, the intuitive idea now is to "pick up" the graph at the minimals and rearrange



(a) (b)

FIG. 5. (a) Results of labeling the connected components (inter-contour regions) in the raster image depicted in Figure 4b. (b) Result of eroding the contour pixels.

it. This is accomplished in the current method by applying a breadth-first graph traversal algorithm (Horowitz and Sahni, 1982) to the adjacency matrix. At every node visited during the traversal, a test is made to check for adjacency, and corresponding children, if present, are labeled with the appropriate elevation value. Thus, the adjacency matrix is rearranged from an undirected graph into a set of directed or rooted trees (Gold and Cormack, 1986) (Figure 4d); the sense of direction in the graph traversal is provided by labeling the edges with increasing elevation (by one contour interval) from the root (Figure 3b). Both the information on the minimals and the contour interval is input by the user. Figure 6 is another example of a map and a corresponding breadth-first traversal that correctly labels all the contour lines, with the user providing the information on the local minima only.

Detecting the Minimals

Initially, at the start of the labeling process, only the edges emerging from the minimals are labeled with the elevations that are manually input by the user. These minimum elevation regions are determined by a visual inspection of the map. At the outset, the task of visual inspection appears to be highly complicated, but, in practice, the task is rendered fairly simple by adopting a few heuristic rules that are true for most of the usual terrain types in the continental United States and for most other regions of the Earth.

The set of heuristics or guidelines to aid in detecting the regions of relative locally minimum elevation regions are as follows:

- Every region where a stream exits a map in the downstream direction is a region of local minima (Figure 7a); notice that most streams on a topographic map, unless very small or dry, are colored blue and are very easy to detect;
- If a contour line closes the boundary of a map such that neighboring contours around it increase in value, the inner most intercontour region is a locally minimum region (Figure 7b);
- Contour closures in maps that are hatched inwards, which indicates depressions, are also regions that are locally minimum (Figure 7c); and
- Lakes, reservoir boundaries, and wide reaches of a stream or river (blue regions in map) are locally minimum regions. (Figure 7d). As the traversal of the graph proceeds, the outgoing edges of succeeding nodes are automatically labeled. The elevation of each node is calculated as the sum of the elevation of the incoming edge incident to that node and the value of the contour interval of the map.

The manner in which the information on local minima is input may be either interactive or batch. Raster outputs of maps (e.g., USGS quadrangle sheets) with very high resolution scanners could generate image sizes as large as 18,000 pixels by 20,000 lines. In such cases, it is recommended that the list of "local minima" be input in a batch mode. The reduction of the task of "tagging" to a batch CPU load, in contrast to the manual procedures in

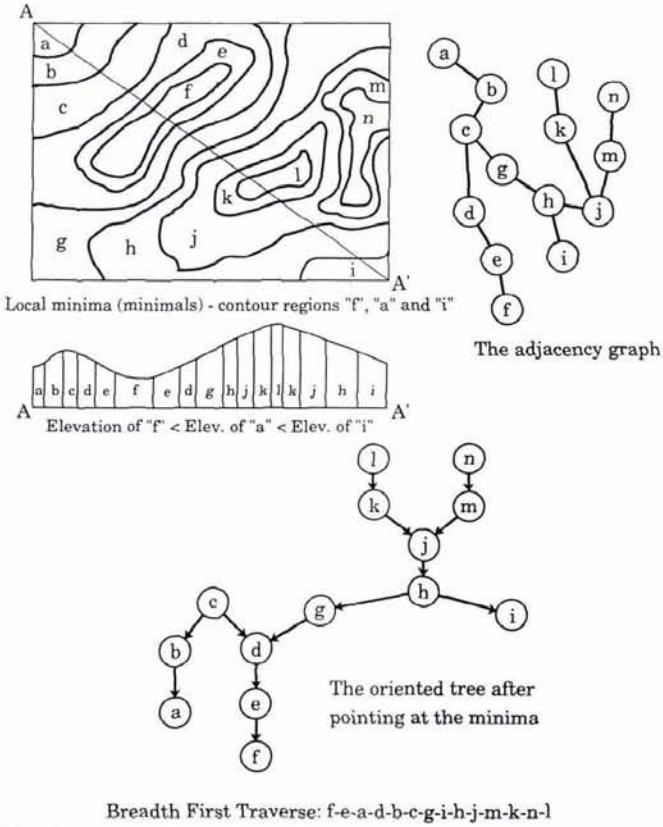


FIG. 6. An example of a breadth-first traversal path to label the adjacency graph of a map segment with multiple minimals.

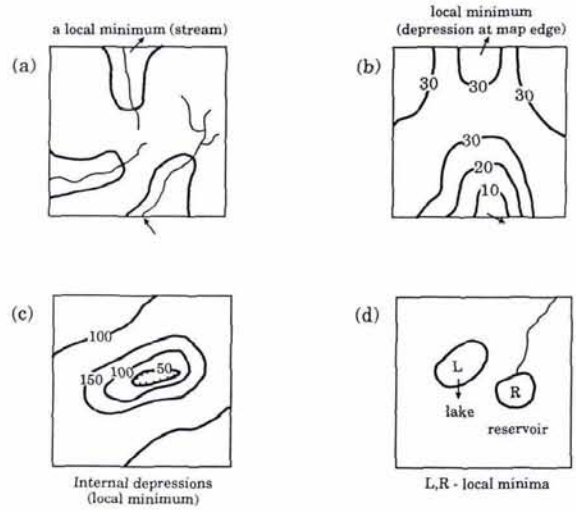


FIG. 7. Schematic of different types of regions forming local elevation minima in a rectangular map area.

the vector techniques, is perhaps the most important advantage of the raster approach.

Reconstructing the Labeled Digital Line Image

Once the adjacency matrix is completely labeled with the correct contour values, the remaining task is to reconstruct the labeled contour image illustrated in Figure 8b. The labeled adjacency matrix, such as in Figure 4d, that is output from the graph traversal contains the values of all edges or boundaries between

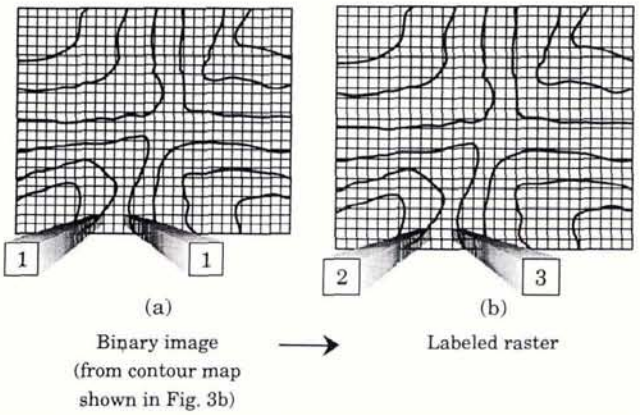


FIG. 8. Raster image output. (a) Binary image from scanned contour line separate. (b) Labeled contour pixels using the graph theoretic approach.

intercontour regions. Thus, a border pixel between region 2 and region 3 of a map, such as in Figure 3b, would be labeled with the value of the element in row 2 and column 3 of the corresponding adjacency matrix. The detection of the border pixels is accomplished in a single traversal of the raster.

Once a raster is available in the format of Figure 8b, the corresponding DEM can be constructed using any of a number of interpolation routines (Sircar, 1983; Sircar, 1986). The procedure of generating DEMs from a tagged digital contour raster is now fairly standard and is routinely used in many contour digitizing operations, including USGS procedures (Sircar and Cebrian, 1986).

RESULTS AND DISCUSSION

For the present research, high resolution scanner input images of contour maps have been simulated using video technology. The resolution of the video scanning technique used for the present study is approximately 50 lines/inch, and the raster sizes were limited to 512 by 512. High resolution scanners, on the other hand, operate at resolutions up to 1000 lines/inch. The difference in resolution affects only the ability to scan large map areas. All of the input documents were first edited in a manual preprocessing step to remove all gaps. In addition to removing gaps, intermediate contour lines were deleted for two of the test cases described herein. The larger contour interval makes it possible to maintain the spatial separability of the imaged lines by preventing contours from merging. In the majority of cases, the presence or absence of the intermediate lines does not affect the topology of the resultant graph; it only alters the number of nodes traversed. Although a few local peaks or passes may be eliminated, the structure of the algorithm for labeling remains unchanged.

Three map segments were tested to demonstrate that the labeling algorithm correctly labels a digital line image in raster mode with minimal user interaction. The first example used to test the approach was a 15-minute quadrangle sheet from a mountainous Himalayan region in Nepal. The second map chosen for the testing was the Damascus, Maryland, USGS quadrangle sheet. The mapped area represents a hilly terrain of moderate to steep slopes varying from about 2 percent to approximately 15 percent. In the third example, an attempt was made to extend the generality of the contour labeling algorithm to flat or moderately steep areas. In this case, an area was chosen from the Bay View, Maryland, quadrangle sheet in the eastern United States. In this area, the slopes vary from almost 0 to approximately 3 percent.

The processing time to label the map for all of the prototype tests was on the order of seconds. The short processing times are possible because the images are all very small compared to the 300 to 400 million pixels generated by high resolution scanning of full size USGS quadrangle maps. However, the present prototype tests have clearly demonstrated the success of the labeling algorithm. It has also enabled the development of a strategy to label raster-based images.

Unlike the vector case, the function of tagging in the current method involves a machine traversal of the adjacency matrix only. The number of elements traversed in the adjacency matrices in the three examples ranged from 70 to 90. For a full-sized quadrangle sheet, the number would probably average between 300 and 1000. However, because the matrix traversal is automatic and requires no user supervision, these large array sizes are of no operational significance.

Approximately 50 locally minimal regions were required on the average to correctly label the maps. However, this is dependent on the complexity of the terrain. The operator is required to visually inspect the map image on a screen and list the pixel coordinates corresponding to a point within each locally minimum region in the image plane for input to the labeling procedure that can be run in batch mode. The time required for visual inspection in the general case varies from 5 to 30 minutes, with obviously more time required for maps of complex terrain patterns.

For the most part, the time needed to process a map with the raster approach depends on the time consumed in disk-to-memory transfers. The times required to implement algorithms such as connected components labeling, border erosions, and traversals of adjacency matrices depend significantly on the performance of the host computer. At the average data transfer rates now available in commercially available, off-the-shelf workstation hardware, the times to process images even as large as 500 million pixels are on the order of minutes.

The only required interactive session is that necessary before the labeling operation to identify the minimal regions in the image and following the labeling to check for the correctness of the solution. In addition to verifying the results of automatic labeling, a certain amount of time is also needed to check for gaps and cliffs and to preprocess these before labeling can be undertaken. Compared to the time required to process vectorized contour maps—on the order of weeks for USGS quadrangle maps—the raster-based approach would require only a few hours.

TREATMENT OF GAPS AND MERGED CONTOURS (CLIFFS)

As stated previously, the assumption of no gaps along any contour line or the absence of intersecting or merged contours in the digitized raster image is unrealisitc. However, it remains a necessity if either the vector-based method or the present raster-based approach is to work successfully. However, one important advantage of the graph representation of contours is that the approach may also be used in a preprocessing step to detect either gaps or merges in contours. While the graph representation of Roubal and Poiker's (Figure 2b) was not found to be useful in efficiently labeling the contour graph, the representation of Mark (1986), shown in Figure 2a, appears to be very useful in detecting the location of gaps in contour lines.

Three different examples of contour trees for map segments where contours have been separated by gaps are shown in Figure 9 using Mark's approach. Both contour lines and intercontour regions are labeled with distinct identifiers L1, L2, L3, L4, and L5 and R1, R2 and R3, respectively, as nodes in the contour tree. For those lines without gaps, the corresponding nodes in the contour tree usually have a unique relationship within the graph. Each node representing a contour line will be connected to two intercontour region nodes. For contour lines with gaps, however, the corresponding nodes are found to be linked with only one intercontour region node. A search for all "contour line" nodes in the adjacency matrix thus formed with links to

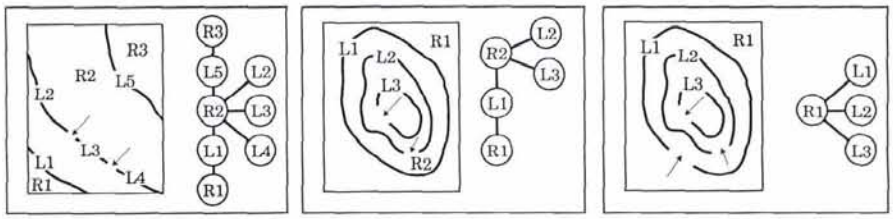


FIG. 9. Examples of contour maps with gaps (indicated by arrows) and their corresponding free tree representations. The cases shown in (a) and (b) are the more common occurences while the situation depicted in (c) is an extreme case with gaps in all of the contour lines in a map segment.

only one intercontour region results in the detection of all of the contour lines with gaps. Once the lines with gaps are identified, these may be processed by either automatic, semi-automatic, or manual editing procedures.

Although Mark's approach has been found to be useful in handling the case of contours with gaps, the graph structure used in the present study has been found to be relatively more efficient in the handling of cliffs or intersecting contours. Figure 10a is an example that illustrates the ease of locating intersecting regions in the resultant adjacency graph. The graph simply loses its tree structure.

The problem of correctly labeling the merged contours automatically can be stated as the problem of determining a correct traversal path from nodes R2 and R5 in Figure 10a. The graph traversal of the adjacency matrix from low to high elevations during the labeling process will generate a correct labeling for even intersecting contours if guided by the following criterion: accept only those nodes that result in a maximal path-length from the lowest to the highest region. Using this rule for the case shown in Figure 10a, the traversal paths R2–R7–R6–R5 and R2–R3–R4–R5 would be the selected paths for a correct traversal and labeling of all of the merged contour line segments.

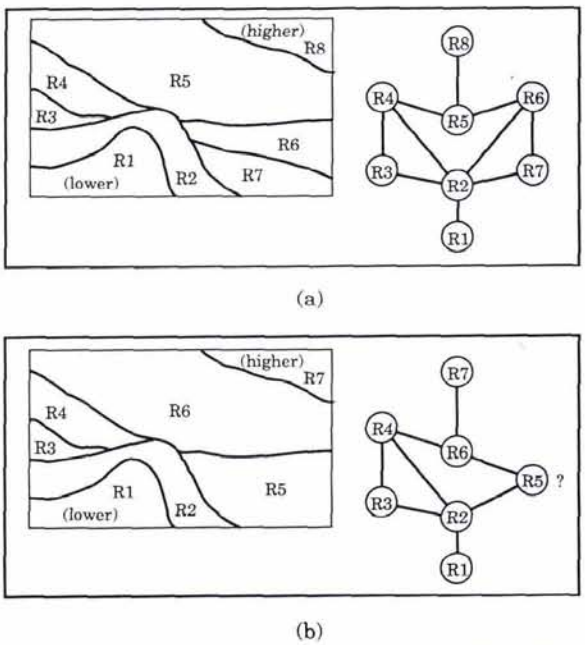


FIG. 10. (a) Example of a typical instance of merged or intersected contours caused normally by natural cliffs or man-made constructions on the surface, e.g., cuts, culverts, etc. (b) A case of merged contours resulting in an ambiguous labeling of the contours.

Although the detection and labeling of merged contours, or cliffs, is a logical outcome of the present approach, the method is constrained to only one category of cliffs. Intersecting contours detected as joining a cliff line at one point must also have a second contact with the same cliff at a different point within the raster. For cases that do not satisfy such a condition, there is no automatic labeling solution. An example of such an insoluble condition is shown in Figure 10b, where the region R5 cannot be uniquely labeled. In the majority of cases, however, most real maps rarely fall into this category. A second condition that needs to be satisfied for a practical graph traversal for merged contours is that the number of intercontour regions affected by a cliff is small, and is indeed most likely to occur in the majority of cases. The universal path-generator chosen for the present study cannot work when the numer of nodes is large.

The example of Figure 10 is a simple one. However, although there are a large class of possible scenarios for merged contours, it is possible to summarize cliffs into two main categories. These are shown in Figure 11a and 11b. In addition, it is possible for intercontour regions to be fragmented by more than one intersection or merging contours (Figures 11c and 11d). While the path-generator rule postulated above can solve the labeling problem for regions fragmented more than once by cliffs of the same type, it cannot solve the problem of one intercontour region involved in two cliffs of different types as shown in Figure 11d.

For the graphs shown in Figures 11a, 11b, and 11c, there is only one path, in each case, that fulfills all of the prerequisite conditions for a correct automatic traversal. These are respectively R1-R2-R3-R4, R1-R2-R3-R4, and R1-R2-R3-R4-R5, respectively. In Figure 11d, however, the paths R1-R3-R4-R5, R1-R2-R4-R5, R1-R4-R2-R5, and R1-R3-R4-R2-R5 are eligible, and, therefore, there is no unambiguous correct path. It is recommended that the nodes be left unlabeled for later manual editing in those cases where the graph traversal results in unambiguous situations. However, the automatic detection and marking of these cases are a natural by-product of the traversal and require no extra effort. In reality, situations as complex as the ones discussed above will be rare and, in the majority of cases, the present approach will lead to a correct labeling even for merged contours.

CONCLUSION

Currently available hardware and software capabilities, ranging from the use of virtual memory to parallel processing, together with large screen oriented interactive devices, offer, for the first time, a unique opportunity to use the described approach to label raster digitized contour images in practical engineering scenarios. The raster approach developed here leads to the replacement of the extremely time consuming elements

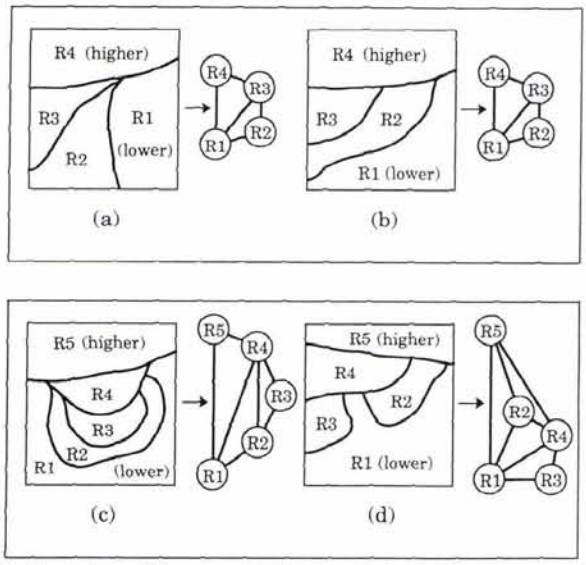


FIG. 11. Examples of the essential types of intersecting or merged contours and the corresponding categories of contour trees.

of contour tagging by efficient algorithms that can be implemented in a batch mode using general purpose workstations.

Because most of the user input information required is either before or after the labeling operation, the tasks of scanning and tagging no longer require investing in costly computer operation skill sets. Developers of scanning and tagging systems have long felt the need for raster-based algorithms to truly take advantage of the high speed of raster scanners. The developed methodology offers the potential of providing a solution to this long-standing problem.

Agencies from the level of individual consultants, to rural, municipal, county, state, national, and international may now find it attractive and practical to use maps to quickly acquire topographic information. Large organizations like the USGS have already invested significantly in creating in-house "mass-digitizing" facilities using scanning technology. Although most of these facilities rely on "vector-based" techniques for many of the digitizing tasks, and integrating the developed method within the existing infrastructure will require some initial cost, the designed approach now offers a means for an appreciable saving in cost and time in at least one of the most time-consuming tasks, that of labeling the contour elevations from scanned maps. Considering the magnitude of the scan digitizing task currently being undertaken by the USGS—approximately 48,000 quadrangle map sheets for the conterminous United States alone—the potential savings become significant.

For the many small- to medium-sized organizations involved in terrain analysis, such as water resources engineering or land resources planning firms, the proposed method provides a quick and efficient alternative to create elevation datasets from readily available maps using a desktop scanner and a microcomputer. The success of the developed strategy in the prototype tests clearly indicates its potential value for implementing the method at operational scales and should be attractive to designers of digital data acquisition systems in the Geographic Information Systems (GIS) industry.

ACKNOWLEDGMENTS

Funding for this research, received through NASA grants NAGW-33, SUP No. 4, and NAGW-512, is gratefully acknowledged. We are grateful to Professor Robert M. Ragan for his encouragement and guidance during the research. Our thanks to Professor Richard H. McCuen, University of Maryland, for his helpful suggestions.

REFERENCES

Allder, W. R., V. M. Caruso, R. A. Persall, and M. I. Troup, 1982. An overview of Digital Elevation Model Production at the United States Geological Survey. *Proceedings, Auto-Carto 5*, pp. 23–32.

Boyell, R. L., and H. Rushton, 1963. Hybrid Techniques for Real-time Radar Simulation. *Proceedings, Fall 1963 Joint Computer Conference*, Las Vegas, Nevada.

Gold, C., and S. Cormack, 1986. Spatially Ordered Networks and Topographic Reconstructions. *Proceedings, Second International Symposium on spatial data Haldling*, Seattle, Washington pp. 74–85.

Horowitz, E., and S. Sahni, 1982. *Fundamentals of Data Structures*. Computer Science Press.

Knuth, D. E., 1976. *The Art of Computer Programming*. Addison Wesley Publishing Company.

Mark, D. M., 1978. *Topological Properties of Geographic Surfaces: Applications in Computer Cartography*. Harvard Papers on Geographic Information Systems. Cambridge, Massachusetts, Vol. 5, 11 p.

———, 1986. Two Contour-Tagging Algorithms Based on the Contour Enclosure Tree. *Abstracts of the Canadian Cartographic Association Meeting*, Burnaby, B.C.

Merril, R. D., 1973. Representation of Contours and Regions for Efficient Computer Search. *Communications of The ACM* 16:2:69–82.

Morse, S. P., 1968. A Mathematical Model for the Analysis of Contour Line Data. *Journal of the ACM*. 15:2:205–220.

———, 1969. Concepts of Use in Contour Map Processing. *Communications of the ACM*. 12:3:147–152.

Peuquet, D. J., and A. R. Boyle, 1983. *Raster Scanning, Processing and Plotting of Cartographic Documents*. A Spad Systems Publication, Spad Systems, Ltd., Williamsville, New York.

Pfaltz, J. L., 1976. Surface Networks. *Geographical Analysis*. Vol. 8, pp.77–93.

Roubal, J., and T. K. Poiker, 1985. Automated Contour Labeling and the Contour Tree. *Proceedings, Auto-Carto 7*, pp. 472–481.

Selden, D., 1985. An Approach to Evaluation and Benchmark Testing of Cartographic Data Production Systems. *Proceedings, Auto-Carto 7*, pp. 499–509.

Sircar, J. K., 1983. *An Interactive Technique to Generate Digital Elevation Data*. Master's Thesis, Department of Civil Engineering, University of Maryland, College Park, Maryland.

———, 1986. *Computer Aided Watershed Segmentation for Spatially Distributed Hydrologic Modeling*. Doctoral Dissertation, Department of Civil Engineering, University of Maryland, College Park, Maryland.

Sircar, J. K., and J. A. Cebrian, 1986. Application of Image Processing Techniques to the Automated Labeling of Raster Digitized Contour Maps. *Proceedings, Second International Symposium on Spatial Data Handling*. Seattle, Washington, pp. 171–184.

(Received 19 March 1990; accepted 24 October 1990; revised 26 November 1990)

Erratum

In the Program for the ASPRS Annual Convention in Baltimore, Bon Dewitt, of the University of Florida, was incorrectly listed as the moderator for the Non-Conventional Applications on Wednesday, 27 March. We regret the error.