Automated Vectorization and Labeling of Very Large Raster Hypsographic Map Images Using Contour Graph

Xiuwen Liu and J. Raul Ramirez

ABSTRACT: This paper presents a very efficient method for vectorizing and labeling very large raster hypsographic map images. By extending the contour tree (Freeman and Morse, 1967) to a contour graph, open contour lines and carrying contour lines can be handled. The ambiguities of the topological relationships among contour lines are resolved using map structure information, i.e., elevation tags and index contour lines. Contour graph is constructed and processed efficiently by introducing index contour graph. By using a sequential extraction algorithm (Agrawala and Kulkarni, 1977) for vectorization, attributes for contour lines, such as line width, and elevation tag(s), are extracted efficiently. Although it was developed primarily for USGS 1:24,000 hypsographic maps as part of the GISOM project at The Ohio State University Center for Mapping, the method also is applicable to other scale USGS hypsographic maps. We have applied the algorithm successfully to raster images with pixel size of 19,320 by 25,400. On average, it can correctly vectorize more than 98 percent of the contour lines and correctly label more than 90 percent.

Introduction

The Ohio State University Center for Mapping has a grant from the U.S. Geological Survey (USGS) to convert 1:24,000 quadrangles into digital files for the state of Ohio. This is known as the GISOM project (Generating Information from Scanning Ohio Maps). One of the main goals of this project is to develop a cost-effective production model for conversion (Ramirez, 1994). Conversion is the process of converting line drawings of raster map images into vector format with proper attributes attached.

Theoretically, raster and vector maps carry the same information in different forms; however, they differ in many respects. Raster maps employ symbols to represent different types of map features. The neighboring relationships contain structure information and are expressed implicitly by relative locations. Vector maps, on the other hand, use an absolute coordinate system to represent geographic features; proper attributes are attached to the vector representation to indicate feature types and other information. The structural information can be explicitly represented by topological relationships.

In order to automate the conversion process, map symbol patterns must be recognized. The structural information conveyed implicitly by neighboring relationships must be employed, in order to attach proper attribute codes to vector representation and overcome ambiguities inherent in map images. The general-purpose vectorization solutions, such as line-following, treat the lines as connections of "dumb" pixels;

Xiuwen Liu is a Ph.D. student in the Department of Computer Science and a research associate at the Center for Mapping, The Ohio State University. **J. Raul Ramirez** is a research scientist at the Center for Mapping and an Adjunct Assistant Professor in the Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH 43210.

they can not take advantage of the information indicated by map symbol patterns and their neighboring relationships. In other words, these general-purpose solutions can not be directly used in an automated conversion method.

At The Ohio State University Center for Mapping, we are investigating the automated, more cost-effective conversion of the hypsographic layer. Raster hypsographic map images contain many contour lines. Each contour line may contain many points, and elevations must be carried with contour lines. A prototype system based on a contour tree model (Liu, 1995) was developed, which works well for contour maps containing few open and carrying contours. System performance declines greatly when many open and carrying contour lines are present in contour maps. The primary reason is that the contour tree model assumes that all contours are closed; thus, it can not handle open contours directly. This work tries to overcome these limitations by extending the tree structure to a contour graph.

Existing Approaches

The topological properties of contour maps are a generalization of adjacency relationships between contour lines. Boyell and Reston (1963) demonstrated that the adjacency relationships between contour lines can be represented by a tree structure, where contour lines are mapped as edges and interstitial spaces as nodes. Shortly afterwards, Freeman and Morse (1967) applied the contour tree concept to the profile search problem. In their approach, a slightly different graph notation was used, where contour lines are mapped into nodes and inter-contour regions into edges. Theoretically, these two representation schemes are equivalent. But computationally, one may be more efficient and convenient than the other with respect to data structures and techniques in an implementation.

Roubal and Poiker (1985) followed Freeman and Morse's representation to automate the process of tagging raster contours. Adjacency relationships are identified using region growing from contour lines. In their model, gaps are allowed and the length of boundaries between different regions is used to identify the correct neighbors fitting the contour tree. They assumed "normal" terrain; the method may not work properly, even for some closed contour cases, as shown in Figure 1. After region growing, the true adjacency relationship between contours ***a*** and ***c*** can not be identified. This problem is due to the fact that the adjacency relationships among regions do not correspond to the adjacency relationships fitting the contour tree one by one.

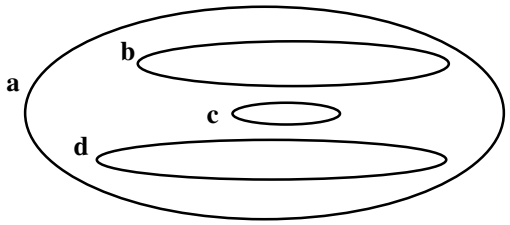

Figure 1. A simple contour map where region growing from contours does not work.

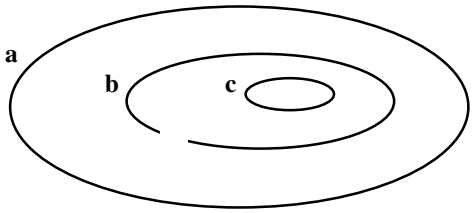

Figure 2. A small gap makes the contour tree construction procedure invalid.

Sircar and Cebrain (1986) used the representation model by Boyell and Reston (1963). The adjacency relationship is extracted using region growing for regions defined by contours followed by line eroding. Although the authors claimed that their method is more efficient, it is very sensitive to noise. Even a small gap may cause the entire processing to be invalid. As in the example shown in Figure 2, even though contours ***a*** and ***c*** are not adjacent to each other, the region growing

process will give a wrong result by detecting their adjacency relationship.

In addition to the problems described above, both methods have difficulty handling large images because they need a huge amount of memory to hold tags for region growing. These problems were solved by Liu (1995) using parity testing along raster scan lines to construct the contour tree. The algorithm has been implemented efficiently, using a sequential extraction algorithm (Agrawala and Kulkarni, 1977).

As we can see, the contour tree model is based on the assumption that a contour line defines a closed region and the adjacency between contour lines can be identified through the relationship between the corresponding regions. This assumption is no longer valid for open and carrying contour lines. In order to overcome the limitations, the contour tree is extended to a contour graph. Contour elevations are used to recover the topological relationships for those problematic contour lines.

The Contour Graph Model

Before we describe the contour graph model, we will review some definitions and contour tree construction using a parity-checking test. These definitions and properties are from Freeman and Morse (1967) and Liu (1995).

Definition 1. Contour lines are the lines of intersection between the terrain, defined by an elevation function and the family of parallel horizontal planes, with given elevation values from e_1 to e_n.

Definition 2. Two contour lines are said to be adjacent if a line, not necessarily straight, can be drawn that connects them without intersecting any other contour lines.

Definition 3. The interior set of a given closed contour line is the set of all contour lines contained within the interior of the region defined by the contour line.

Definition 3 only applies to closed contour lines because open contour lines do not define a region.

Definition 4. A contour line L_1 is called an interior contour line of a contour line L_2 if L_1 is adjacent to L_2 and belongs to the interior set of L_2.

Definition 5. The interior adjacency set of a given contour line L is the set of all contour lines adjacent to L and belonging to the interior set of L.

Property 1. If two different contour lines are adjacent, then a line, not necessarily straight, can always be drawn to connect any two points on the two given contour lines, respectively, without intersecting any other contour lines.

Property 2. The contour lines in the interior adjacency set of a contour line with value e_i may have only values e_i and e_{i-1}, or e_i and e_{i+1} ; but never e_{i-1} and e_{i+1} at the same time (Freeman and Morse, 1967).

Property 3. Interior adjacency set relation defines a strict partial ordering and can be represented by one or more tree structure components.

Property 4. The path from a node to the root in the contour tree can be determined by even-odd tests along any semi-infinite ray whose origin is within the inter-contour region defined by the given contour line and the contour lines in its interior adjacency set.

Based on property 4, we can actually construct the contour tree only along raster scan lines provided all contour lines are closed, and do not intersect with each other. For open and carrying contour lines, property 4 holds only along those semi-infinite rays that define a labeling path in definition 7. In order to handle the open and carrying contour line cases, we first define a general graph where contour line segments are nodes and adjacency relationships are edges. Here, the adjacency relationship is as that in definition 2, which may not correspond to a parent-child relationship in the contour tree.

Definition 6. A contour graph is a general graph. Its nodes are contour segments. There is an edge between two nodes if they are adjacent.

Carrying contour lines should be broken into segments before insertion into the graph.

Definition 7. A labeling path in a contour graph is the path that has a corresponding path in the contour tree constructed, given that all open contours are closed correctly.

Property 5. A path in the contour graph is a labeling path only if the elevation sequence for the path is increasing or decreasing linearly by the contour interval.

Property 6. A peak is a node from which all labeling paths are decreasing locally. A depression is a node from which all labeling paths from it are increasing locally.

Based on properties 5 and 6, a contour graph can be partially converted into a tree structure using known elevations. Also, when a labeling path is found, all contour lines along that path can be labeled; this may further provide information for other contour lines.

For USGS hypsography maps, labeling paths may be difficult to define, because of the feathering lines. However, conventionally, index contours are included. Instead of searching through the whole contour graph, first an index contour graph is constructed and labeled. Here, index contour graph is defined as the contour graph after deleting all secondary contours. In addition to that it can be constructed correctly, index contour graph also provides computational advantages by localizing the possible labeling paths for secondary contours. Please refer to the experimental results section for details.

Implementation

The system is implemented as an extension of the prototype system (Liu, 1995). Currently, two preprocessing stages, noise and gap removing, and elevation tag recognition, are done before vectorization and labeling.

1. Noise and Gap Removal

During this phase, gaps and noise will be processed based on user-supplied area criteria and surrounding features. For example, a region smaller than a given threshold will be eliminated if there is no other feature in its neighborhood.

Uncertainty and ambiguities exist during this stage. If ambiguities cannot be resolved, they will be skipped simply at this phase.

2. Elevation Tag Recognition

Elevation labels are recognized using the Fourier descriptor (Grandland, 1972), which is independent of scale, position, and orientation. Individual numbers will be assembled based on the geometric constraints between numbers. Ambiguities such as "6" and "9" are resolved based on orientation conventions of USGS hypsography maps.

For special cases where digits and spot elevation markers overlap with contour lines, template matching is used in a possible elevation tag area to recognize the missing digits and segment the images accordingly. These templates for digits and spot elevation markers are scale dependent.

3. Vectorization and Elevation Labeling

A very efficient sequential extraction algorithm (Agrawala and Kulkarni, 1977) is used to derive polygonal approximation of boundaries of contour lines. Centerlines are estimated from the boundaries using the minimum distance method (Shapiro and Sklansky, 1981). During this process, nodes in the contour graph are defined as centerline segments with proper attributes associated (such as closed or open, index or secondary contour based on average line width) and elevation tags attached. Special cases such as carrying contour lines are broken into segments so that each segment can be processed.

To facilitate the parity test, the Center for Mapping Database Format (Ramirez et al., 1992) is used to find the neighboring contour lines along a semi-infinite ray. This is achieved by using grid structure with coarser resolution to store the line ID's of those contours which pass through the cells. For each cell, a linked list is used where ID's of index contour lines are in front of those of secondary contour lines.

a. Index Contour Graph Construction

After the image is processed scan line by scan line, all nodes in the contour graph are defined. Then, for each index contour, we detect the corresponding adjacent index contours using vertical and horizontal semi-infinite rays. Currently, this is only done for those rays from peaks, pits, and spot elevations, which detects most of the relationships correctly.

b. Index Contour Graph Labeling

After we construct the contour graph, we search through all possible labeling paths by traversing all paths. A labeling path must have elevations attached at its both end nodes. Generally, we do not need to search through the whole graph because elevation tags are redundant.

When a valid labeling path is found, all nodes along the path can be labeled correctly, which provides information for unlabeled contours.

c. Secondary Contour Labeling

After index contours are labeled, secondary contour lines can be labeled based on index contours adjacent to them, using the parity test as we did for index contour lines. In order to label contours correctly, we also need to search for labeling paths and label along these paths.

Because of the feathering and carrying lines, we may find out that labeling paths give inconsistent labels for some contour lines. In these cases, we need to analyze the graph structure and sort the nodes along index contours in order to determine the correct labels for them. Those contours with inconsistent labels are marked and will be checked manually.

d. Final Vectorization Result

The final results are currently written into an Intergraph DGN file with a corresponding alphanumeric attribute file. The contour lines are smoothed with a threshold within half of the line width so that the topological relationship is consistent and no crossing contour lines are possible.

Experimental Results

By extending the tree structure to a general graph form, the system performance has been improved greatly for maps with many open contour lines. Because the elevation tags in the hypsographic map images provide redundant information, labeling paths can be easily found. By incorporating the conventions of USGS 1:24,000 hypsographic maps, the system can correctly label most of the contours. The following example demonstrates how this is achieved in detail.

Figure 3 shows the original raster images. It contains two contour elevations and one spot elevation. These numbers are first recognized and their images are removed. The index contour graph for this map is constructed as shown in Figure 4. The single digit is the number in the index contour graph. The contour interval for the map is 10 and so the interval between index contour lines is 50. When searching through all paths, only the path {1,2,3,6} gives a valid labeling path, and so contours 2 and 3 can be labeled with elevations of 1000 and 1050, respectively.

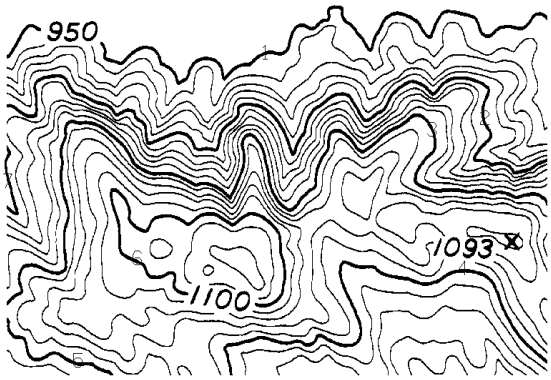

Figure 3. Original raster hypsographic map image.

Based on USGS convention, node 6 defines a peak, so node 4 is labeled as 1050. Also path {2,3} gives an increasing elevation sequence, with respect to nodes 2 and 7, so

node 3 is a local peak and node 7 is labeled as 1000.

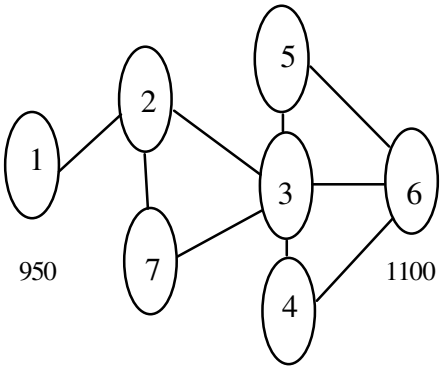

Figure 4. Index contour graph for the contour map in Figure 3.

After index contours are labeled, other secondary contours can be labeled, based on index contour elevations. Except for several contours which are not enclosed by index contours, all other contours can be labeled. The final results are shown in Figure 5.

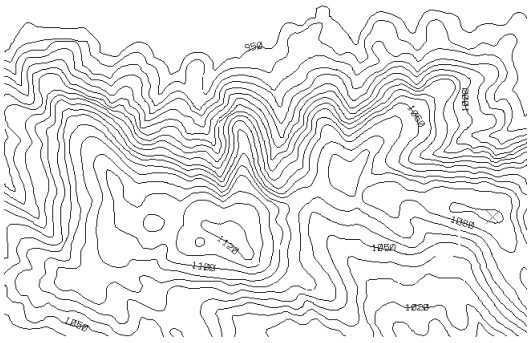

Figure 5. Final vector contour map.

We have applied the method to full-size images of USGS 1:24,000 quadrangles. As an example, the quadrangle of Amanda, Ohio was processed. Ninety-nine percent of contour lines were vectorized correctly and 95 percent of contour lines were labeled correctly. We also applied the method to USGS 1:62,500 hypsographic maps from Alaska; it works as well as it does on the 1:24,000 maps.

General Conclusion

In this paper, an efficient method for converting raster hypsographic images into vector format is presented. The major advantages are that it works directly on raster images, even with open and carrying contour lines, and it is highly automated. Also, the consistency of the results can be checked using redundant map information. By doing parity checking, the contour graph is constructed efficiently. Known elevations are used to resolve ambiguities for open and carrying contours. The graph-based approach provides a general framework to automate the conversion process from raster images to vector format as demonstrated in this paper. Although the work done here is primarily for hypsographic maps, this model can be applied to other USGS layers too. The only change needed would be to incorporate the semantics for map symbols and process the graph using appropriate models for that kind of map.

ACKNOWLEDGMENTS

The authors gratefully acknowledge The Ohio State University Center for Mapping for supporting this research. They wish to thank Dr. Alan Saalfeld for his valuable suggestions and comments. They thank Mr. Eric Cordle and Mr. Joseph Szakas for helping implement and test the system, and Ms. Beverly Henderson for making the paper more readable.

References

Agrawala, A. K. and Kulkarni, A. V., A Sequential Approach to the Extraction of Shape Features, *Computer Graphics and Image Processing*, Vol. 6, 1977, pp. 538-557.

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

Freeman, H. and Morse, S. P., On Searching a Contour Map for a Given Terrain Elevation Profile*, Journal of the Franklin Institute*, Vol. 248, No. 1, 1967, pp. 1-25.

Grandland, G. H., Fourier Preprocessing for Hand Print Character Recognition, *IEEE Transactions on Computers*, Vol. 21, No. 2, 1972, pp. 195-201.

Liu, X., *A Prototype System for Automatic Conversion from Raster Hypsography Maps to Vector Format,* M.S. Thesis, The Ohio State University, 1995.

Ramirez, J. R., Fernandez-Falcon, E., Schmidley, R., and Szakas, J., *The Center for Mapping Database Format*, unpublished paper, The Ohio State University Center for Mapping, 1991.

Ramirez, J. R., *Automatic Elevation Tagging as Part of Hypsography Data Conversion from Scanning Data*, Internal Report, The Center for Mapping, The Ohio State University, 1994.

Roubal, J. and Poiker, T. K., Automated Contour Labeling and the Contour Tree, *AutoCarto 7 Proceedings*, Washington D.C., March 1985, pp. 472-481.

Shapiro, B., Pisa, J., and Sklansky, J., Skeleton Generation from x, y Boundary Sequences, *Computer Graphics and Image Processing*, Vol. 15, 1981, pp. 136-153.

Sircar, J. K. and Cebrian, J. A., Application of Image Processing Techniques to the Automated Labeling of Raster Digitized Contour Maps, *Proceedings of Second International Symposium on Spatial Data Handling*, Seattle, Washington, July 5-10, 1986, pp. 171-184.