Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Extracting Topographic Features for Outdoor Mobile Robots

In So Kweon and Takeo Kanade

The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Some applications such as the autonomous navigation in natural terrain and the automation of map making process require high-level scene descriptions as well as geometrical representation of the natural terrain environments.

In this paper, we present methods for building high level terrain descriptions, referred as topographic maps, by extracting terrain features like "peaks", "pits", "ridges", and "ravines" from the contour map. The resulting topographic map contains the location and type of terrain features as well as the ground topography.

We develop new definitions for those topographic features based on the contour map. We build a contour map from an elevation map and generate the connectivity tree of all regions separated by the contours. We use this connectivity tree, called Topographic Change Tree, *to extract the topographic features. Experimental results on a Digital Elevation Model (DEM) supports our definitions for topographic features and the approach.*

1 Introduction

Extracting topographic features from elevation maps has traditionally been studied in two research areas: in robotics, autonomous underwater vehicle (AUV) route planning and self localization [10] and in cartography, the automation of the map making process [5, 9, 12].

Recently, Seemuller [12] presented a method for producing drainage networks from terrain elevation data. In his method, he determined drainage points by finding grids located at a local minimum in elevation for the horizontal or vertical direction in a 3×3 neighborhood. He then traced and linked drainage points into a list of drainage nets. As he described in his paper, it suffers with the locality of the method. Moreover, he did not justify why a local minimum in elevation only for the horizontal or vertical direction can be a drainage point.

There also has been a variety of techniques to detect pits, peaks, ridges, and ravines as a means of characterizing topographic structures in a digital image. Many of the methods, however, use various heuristics without rigorous mathematical justification [6].

Haralick *et. al* [4], PoncePonce85, and Besl [1] defined and extracted topographic features using concepts from differential geometry. There exists a problem: If we take a peak and then rotate it about a horizontal axis, it soon stops being a peak, even if curvature, being intrinsic to the shape, does not change.

In this paper, we will present a method to overcome those problems found in previous research works.

We first examine the definitions of topographic features. We then present our basic representation, we call it as the contour tree, from which topographic features will be extracted. In Section 5, we describe an algorithm for extracting peaks and pits from the contour tree. This is followed by the extraction of ridges and ravines from the contour tree in Section 6.

2 Definitions of topographic features

Many topographic feature definitions are ambiguous. Traditional natural language definitions of topographic features have a substantial drawback that such definitions either use terms which are not exactly defined, or end up in circular definitions.

However, a 3-D vision system cannot rely on these definitions for extracting the topographic features. To avoid the problem of ambiguous definitions, there has been a wide variety of techniques to define features based on a procedure for the identification of a feature [4].

Seemuller [12] defined ravines and pits as all the locations at which water would flow or collect if water were poured on the terrain. According to this definition, ravines or pits correspond to points lying at a local minimum in elevation. Ridges and peaks are duals to ravines and pits, respectively. In other words, ridges or peaks correspond to points lying at a local maximum in elevation.

More recently, many researchers have used concepts from differential geometry to classify the points on a surface into basic classes of several features [1, 2]. From the theory

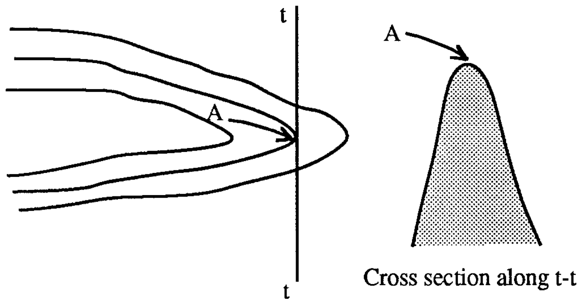


Figure 1: Contours of ridges and a cross-section along the direction of gravity

of differential geometry, combinations of values of the two principal curvatures uniquely define features.

2.1 Our definitions based on contour lines

Those definitions described in the previous section may have a sound mathematical justification. However, human beings are capable of recognizing topographic features from a contour map without knowing the detailed elevation of the terrain. Based on this observation, topographic features can be qualitatively defined using the shape and relationship of contour lines on contour maps [8].

Peaks or pits can be defined as a series of closed contours. From the elevations of the contour lines shown, a peak or pit is defined as having either ascending or descending elevations. In terms of elevations, peaks and pits are corresponding to local maxima and minima, respectively. Therefore, we define peaks and pits as:

$$\text{Peak:} \quad \max \quad z(x,y)$$
$$\text{Pit:} \quad \min \quad z(x,y)$$

Ravines can be defined as the contour lines forming V's pointing upstream. Similarly, ridges can be defined as the contour lines forming V's pointing downstream. The contour-based definitions for ridges and ravines are still very heuristic and qualitative. We present a new definition for ridges and ravines, and show that the new definition is equivalent to the contour-based definition.

Let's assume that we are at a point on a ridge (a point A in the left figure of Figure 1). The right figure of Figure 1 shows the cross-section of the terrain by a vertical plane along the tangential direction t. Why do we believe that we are at a ridge? If a ball is displaced slightly from this point, it rolls away along the direction t. Similarly, if water were poured at the point, water would naturally flow along the same direction (along the steepest direction). We can formally define ridges and ravines as

$$\text{Ridge:} \quad \max_{(x,y)\ \text{on}\ C} \quad \frac{z_t''}{z_n'}$$
$$\text{Ravine:} \quad \min_{(x,y)\ \text{on}\ C} \quad \frac{z_t''}{z_n'}.$$

where z_t'' is the second derivative of z along the direction tangent to the contour at (x,y) on C and z_n' is the first derivative of z along the direction normal to the contour.

The above equation states that a ridge point has the maximum change in the gradient of elevation in the tangential direction (z_t'') and has the minimum elevation change in the normal direction (z_n') along the contour.

Using the implicit function theorem and differential geometry [3], we can show that the local extrema in z_t''/z_n' occur at the point on the contour where the shape of contour forms V (for the details of the derivations, see [7]).

3 Constructing contour maps from elevation maps

Building a contour map from an elevation map is straightforward: imagine a series of parallel cuts which intersect the profile of the terrain. Next, define successive plateaus of constant elevation by planar cuts located at a particular distance below or above a reference plane.

The algorithm for building a contour map works in five steps:

1. Start a planar cut of the elevation map, specified as an elevation H_{ij} at each grid point ij, at a particular elevation, H_c;

2. Create a binary image B_{ij} by assigning

$$B_{ij} = \begin{cases} 1 & \text{if } H_{ij} \geq H_c \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

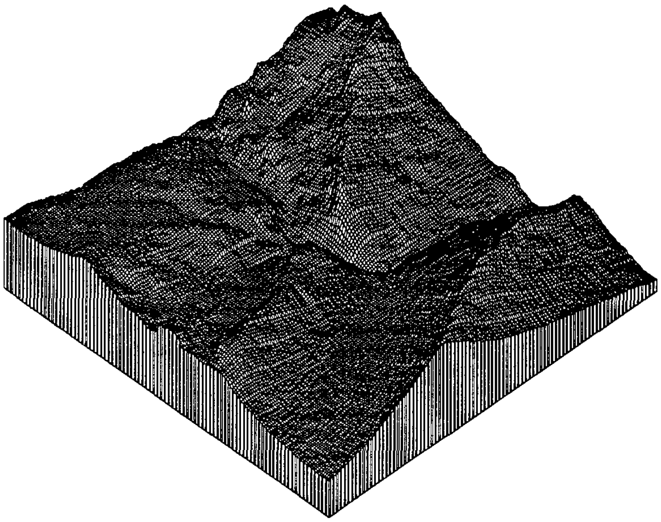
3. Extract the connected components;

4. Compute the contour lines by fitting polygons to the connected components;

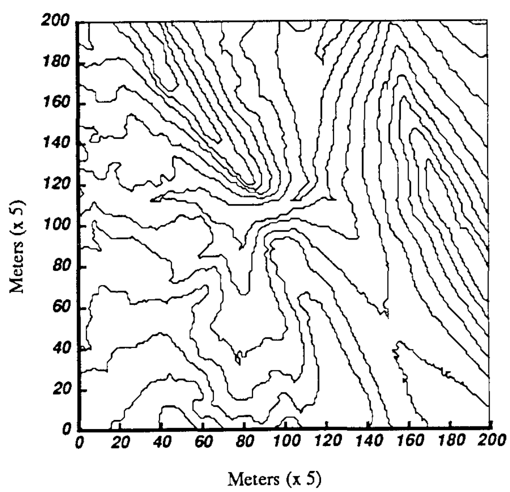
5. Repeat steps from (2) to (4) at a new elevation

$$H_c = H_c + \Delta H. \tag{2}$$

We applied this algorithm to a Digital Elevation Map (DEM). Figure 2 shows an elevation map and the corresponding contour map.



(a) A DEM



(b) Extracted contour map

Figure 2: Result of constructing a contour map from an elevation map.

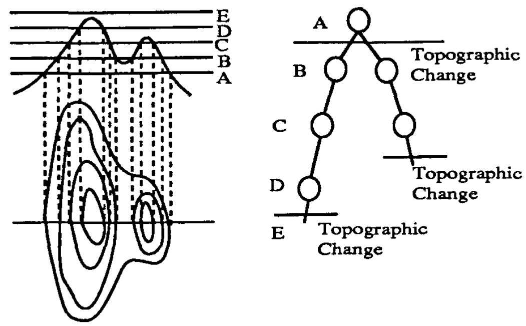


Figure 3: An example of Topographic Change Tree for 1-D.

4 Constructing the contour tree from a contour map

We present a tree data structure, the contour tree, which can represent the nesting of contour lines on a continuous topographic surface. The contour tree represents the relationships among contour lines on a contour map. We call this contour tree representation a *Topographic Change Tree* (TC tree).

In TC tree, each node represents a cross-sectional area of terrain intersected by a planar cut, and links between nodes represent parent-child relationships (i.e. a path from one area to another). Every node has a list of descendants, its corresponding elevation value, a list of edge points approximated by a polygon, and its parent.

In general, we can have three kinds of links connecting the two nodes. First, single parent-child relationship between two nodes indicates no topographic change, noted as no-change (e.g., the planar cut A in Figure 3). Second, multiple connections among nodes or a *branch* indicates topographic changes among them (e.g., the planar cut B in Figure 3). Third, no child at a particular node also means a topographic change has occurred between the two elevations. (e.g., the planar cut C in Figure 3).

Starting from the root node with minimal elevation, we recursively create the contour tree in a depth-first fashion. A depth-first tree generation expands the most recently generated node first. When it reaches a node that has no descendants, it visits an unexplored node at the nearest depth.

5 Extracting peaks and pits from the contour tree

Peaks and pits are extracted by finding a series of closed contours from the TC tree. The TC tree is recursively traversed for extracting a series of closed contours. The algorithm is a graph traversal procedure to label each region

```
extract-connected-closed-contours (origin-node, cnode)
  M ← the number of children of cnode
  if M > 1

  then
        origin-node ← cnode
  else if M = 0
        get-contours()
  end if
  for j = 1 to M do
        cnode ← child of cnode
        extract-connected-closed-contours ()
  endfor j
```

Table 1: An algorithm for extracting a series of closed contours.

encircled by a contour (Table 1 shows the algorithm in pseudocode).

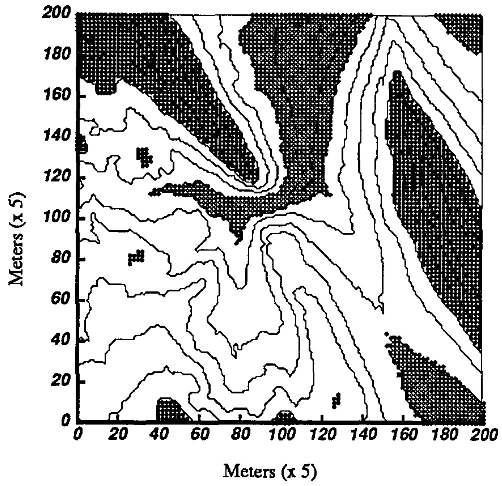
In 3-D, the relationship between contour lines of the same elevation can be divided into the two cases: (1) two contours are completely separated with each other, (2) one contour is enclosed by the other contour (for example, we can observe this case from a volcano in real terrain). The correct labels for the self-included contours can be obtained by creating two contour trees: one from downward the other from upward direction [7].

We apply the algorithm to extract peaks and pits from real terrain data, a DEM shown in Figure 2a. Figure 4 shows extracted peaks and pits and the corresponding contour tree.

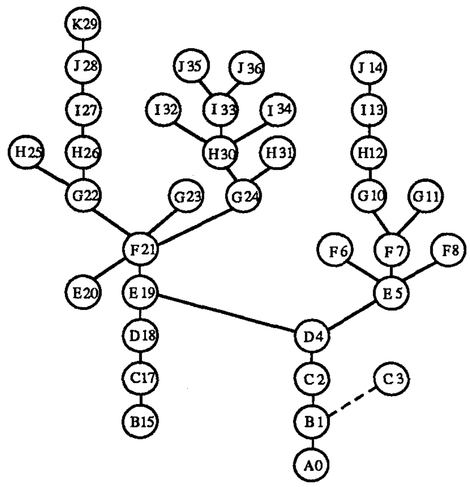
6 Extracting ridges and ravines from the contour tree

According to the definitions of ridges and ravines based on contour lines, the contour lines consisting of ridges or ravines have V shape. We also showed that the point whose contour shape forms V has the local extrema in z_t''/z_n' along the tangential direction on the contour. Also note that the shape of V corresponds to the local maximum in curvature on the contour. Therefore, ridges and ravines can be extracted if we extract those local maxima of the curvatures from the contour, and group them together.

We first represent the contour lines by chain codes. Each element in a chain code represents the curvatures of their respective curve segments. We extract local maxima of the curvatures by using the scale space approach. We then describe an algorithm to group those extracted features into ridge- and ravine-lines by using the topological relationship



(a) Peaks and pits.



(b) The contour tree.

Figure 4: Results for the extraction of peaks and pits from a DEM.

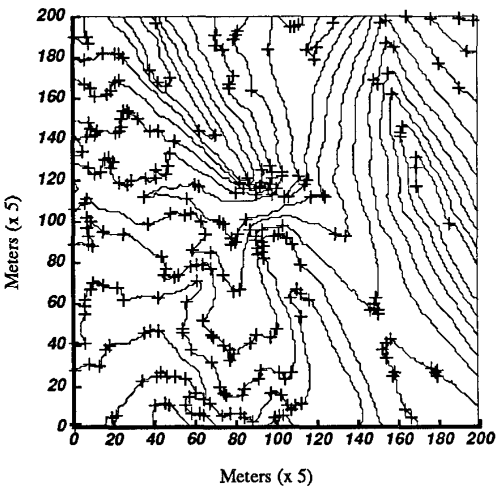


Figure 5: Results for extracting the local extrema in curvature.

among the features, represented by the contour tree.

6.1 Extracting extrema points

The algorithm for extracting the local extrema in curvature along contour lines works in two steps: (1) the contours are represented by eight-directional chain-codes. (2) we then divide the contours into line segments by detecting the extrema curvature points on the contours.

The chain code representation is calculated by computing the angle change at a particular point along the contour. We use the arm method, introduced by Rosenfeld [11], to compute the angle change [1].

To detect the extreme points at different scales, we analyze the contours in the scale space. At each scale, we extract the candidate areas of the local extrema in curvature as a rough estimate of feature positions. A coarse-to-fine strategy is then applied to these candidate areas in the multiple curvature curves. During this coarse-to-fine tracing, the arm length becomes short and the candidate area is narrowed down. Peaks in the resulting corner areas are picked up as high curvature points.

Figure 5 shows the resulting high curvature points , denoted as +, overlayed on the contour map. These results demonstrate that the multi-arm method provides accurate and reliable high curvature points on the contour lines.

6.2 Tracing and linking ridge and ravine points

We described how to extract ridge points in the above section. The next important step is to convert the extracted ridge points into a list of lines. We present a method for tracing and linking ridge (or ravine) points by using the contour tree. As we explained in Section 4, the contour tree provides the topological relationships among contour lines on a contour map. Assume that we have a known (a starting) ridge point in the current node of the contour tree. Then we first look for the ridge points in the children nodes of the current node because the topological relationship embedded in the contour tree representation indicates that those points in the children nodes are the only topologically possible connections to the current point. If no points are found in the children nodes, it recursively visits the children of its children. The traversal of the contour tree is done in a breadth-first fashion. A breadth-first search visits all the children of a node before visiting any of the children of its children.

For each ridge point of each child node, we compute some attributes which are used to search for the best ridge point connecting to the current ridge point. The attributes to be computed include:

- The current global direction of a ridge, determined by the slope of the line drawn from the previous ridge point to the current point (α);
- The current local direction of a ridge, determined by the slope of the line drawn from the center of the curvature to the current point (β);
- The Euclidean distance from the previous ridge point to the current point (ρ).

Only points within $\pm\ 90^o$ of the global and local directions are considered candidates for the next point. These restrictions eliminate radical direction changes in the ridge. The candidates are further reduced by finding those having the minimum distance from the current point. If there is still more than one, the points resulting in the least angle change in the ridge are found.

In addition to angle and distance constraints, we use the similarity of the shape to find the ridge lines. The idea is to choose the feature point with the most similar curvature value with the known ridge point. To solve the ambiguity of the feature point positions in different contours, we use the neighbor points of the local extreme point to compute the correlation of curvature values between the two high curvature points. We determine the size of neighbor by using the most appropriate arm length which is determined by the contour analysis. We then define the curvature similarity between two feature points, C_i and P_j, by

$$S_{ij} = \sum_{k=-l}^{k=l} \sqrt{\frac{(C_{i+k} - P_{j+k})^2}{2l}}, \qquad (3)$$

where l refers to the arm length.

[1]The angle change can be considered to be the curvature because the definition of curvature is given as $\kappa = \frac{\delta\theta}{\delta s}$.

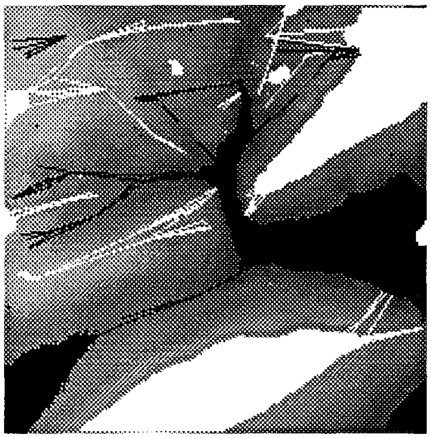


Figure 6: Extracted topographic features from a DEM.

By following the same procedure for all feature points along contours, we can obtain the ridge and ravine lines.

Figure 6 shows the extracted topographic features from the DEM. White lines indicate ridge lines, black lines indicate ravine lines. White and black areas indicate peaks and pits, respectively. The features are overlaid onto on the DEM in which lighter values indicate higher elevations.

7 Conclusion

In this paper, we have developed algorithms for extracting topographic terrain features from contour maps, represented as the contour tree.

First, we have have shown that a point with a local extreme in curvature on the contour has a local extreme in z_t''/z_n' on the contour, which is our definition of ridges and ravines.

We have also developed an algorithm for computing the contour tree from an elevation map. The contour tree provided a description of the spatial topology of terrain from which peaks and pits were extracted reliably.

In converting ridge (or ravine) points into a list of ridge (or ravine) lines, we have presented a new algorithm to trace and link ridge (or ravine) points. The algorithm uses the topological relationships among the contour lines, represented by the contour tree, to search for the best linking points.

The method for extracting topographic features can be extended to a more general and robust method by using: 1) a more stable method for extracting high curvature points, and 2) a multi-resolution approach for linking the ridge (or ravine) points to the ridge (or ravine) lines.

References

[1] P. J. Besl and R. C. Jain. Segmentation Through Symbolic Surface Descriptions. In *Proc. of CVPR*, May 1986.

[2] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing Surfaces. *CVGIP*, 32:1–28, March 1985.

[3] I. Faux and M. Pratt. *Computational Geometry for Design and Manufacture*. Ellis Horwood, 1979.

[4] R. Haralick, L. Watson, and T. Laffey. The Topographic Primal Sketch. In *The International Journal of Robotics Research*, Spring 1983.

[5] S. K. Jenson. Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Model Data. In *Proceedings of ASP/ACSM Conference*, March 1985.

[6] E. G. Johnston and A. Rosenfeld. Digital Detection of Pits, Peaks, Ridges, and Ravines. *IEEE Trans. Syst. Man Cybern.*, pages 472–480, July 1975.

[7] I. Kweon. *Modeling Rugged 3-D Terrain by Mobile Robots with Multiple Sensors*. PhD thesis, Carnegie Mellon University, The Robotics Institute, August 1990.

[8] F. H. Moffitt and H. Bouchard. *Surveying*. Harper Row, 1982.

[9] J. F. O'callaghan and D. M. Mark. The Extraction of Drainage Networks from Digital Elevation Data. *Computer Vision, Graphics, and Image Processing*, 28:323–344, 1984.

[10] D. Orser and M. Roche. The Extraction of Topographic Features in Support of Autonomous Uncerwater Vehicle Navigation. In *Proc. Fifth International Symposium on Unmanned Untethered Submersible*, pages 502–514, Robot Systems Division, National Bureau of Standards, Gaithersburg, Md 20899, 1987. Unknown.

[11] A. Rosenfeld. Digital Straight Line Segments. *IEEE Trans.*, C-23:1264, 1974.

[12] W. W. Seemuller. The Extraction of Ordered Vector Drainage Networks from Elevation Data. *Computer Vision, Graphics, and Image Processing*, 47:45–58, July 1989.