Lineament Extraction from DEM Using Raindrop Tracing Algorithm

Sang-ho Yun
Department of Earth System Science
Seoul National University
Tel : 02-872-2713
FAX : 02-888-8696
E-mail : blitz@snu.ac.kr

ABSTRACT

Lineament extraction from mountain area often provides valuable geological information. In many cases, the lineaments correspond to a series of continuous large valleys. This paper introduces a new lineament extraction method from Digital Elevation Model (DEM) using Raindrop Tracing Algorithm (RTA). The main advantage of this algorithm over conventional Segment Tracing Algorithm (STA) is that it utilizes DEM directly unlike the STA which utilizes the shaded relief of DEM. The RTA simulates the real life of raindrops that converge into a large valley. The simulation has been done by sprinkling the randomized raindrops over DEM and counting the number of raindrop path that follows the negative gradient of the DEM. The large counting number indicates the location of a big valley where the raindrops converge. With the help of the counting number array (accumulator array) recording the flowing path information, RTA can produce perfectly unbiased binary image of the lineament.

INTRODUCTION

In order to extract lineaments from raster image data sets we have to follow two major processes – 1. Enhancing the lineaments into a binary image, 2. Applying Hough Transform. The Segment Tracing Algorithm (STA, Koike and others, 1995) has recently been used and is very reasonable for lineament extraction from the images of mountainous terrain acquired with optical sensor like Landsat TM. When using Digital Elevation Models (DEMs) instead of such an optical sensor image, many researchers are making shaded relief by illuminating the DEM with an imaginary sun and then using an algorithm such as the STA to make a binary image. The only reason for making shaded relief is because the STA needs it. This process, however, deprives the DEM of a certain accurate information that the STA struggles to regain. In other words, making a shaded relief is the course of giving bias to the DEM by the sunlight azimuth, and the STA is the course of eliminating such a bias.

There should be a better solution for the case of DEM. I suggest a more straightforward and reasonable algorithm. The Raindrop Tracing Algorithm (RTA) is the simulation of raindrops flowing on the surface of mountain area. As they flow along the negative gradient direction of the DEM, the trace of the flow path is recorded on an accumulator array that has the same size of the DEM. The droplet flows until it encounters a boundary of the image or the zero gradient regions such as reservoirs. Once the droplet flowing come to an end, another one is released on an arbitrary point of the image and the trace of the new droplet is added on the existing accumulator array. The total number of raindrops to be released is assigned by a user. After proper number of raindrops finish their journey, we have a new image, the accumulator array, which contains a number of valleys with various intensities. The intensity of a specific valley indicates how large and dominant the valley is. What we want to find is a series of such valleys because in many cases a lineament that represents faults and fractures appears in the form of straight stream valleys and aligned segment of valleys. At least such lineaments can represents the weak zones of upper most crust.

With the prepared binary image using the RTA Hough transform is used to detect the dominant lineament. In this paper, I used Hierarchical Hough Transform (HHT, Princen and others, 1990) to extract line segments from the binary image.

EXPLANATION ON DATA SET

The DEM used in this paper is produced from topographical map with 1:250000 scale by Defense Mapping Agency (DMA) in United States. The pixels of this raster format are apart 3 seconds from each other. For this study area, 3 seconds is equivalent to 92.5m and 75m in the direction of latitude and longitude respectively.

I selected a part of the DEM image for the mountainous area of Hwacheon the northern Kangwon district in the middle of the Korean peninsula. The dimension of the test area is 400 by 400 pixels. Figure 1 shows the DEM of the study area.

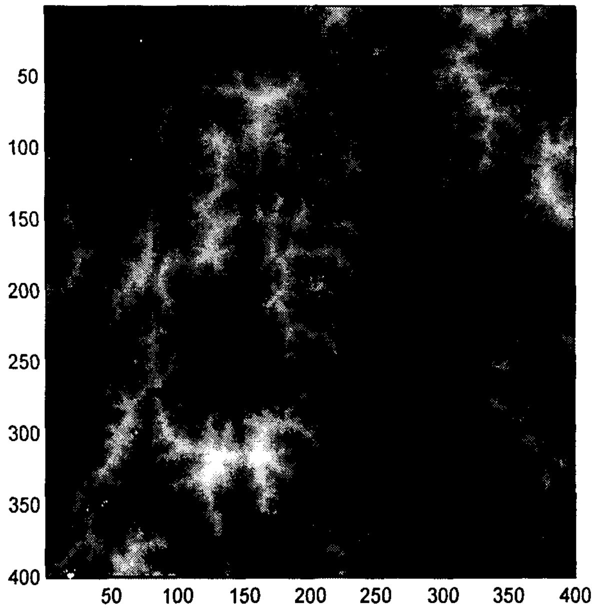

Figure 1. Digital Elevation Model of study area

DATA PROCESSING

Raindrop Tracing Algorithm

The DEM contains elevation information in each pixel. To locate valleys in this image, we must have the information of slope at each point by performing a gradient operator. Then multiply −1 to each element of the gradient vector field to make it represent the steepest downward direction. To produce the gradient, I took forward differences on the boundary points of the image and central differences on interior points.

$$g_1 = \frac{f_2 - f_1}{h} \qquad g_n = \frac{f_n - f_{n-1}}{h}$$

$$g_i = \frac{f_{i+1} - f_{i-1}}{h} \qquad (2 \le i \le n-1)$$

And each of the gradient vectors is normalized to have the unit length for a droplet to move only one pixel at a time. After this, an accumulator array with the dimension of the original DEM is initialized with zeros. The accumulator array is then initialized with zeros.

A raindrop is released as a seed on the normalized negative gradient vector field. Then let the droplet flow one pixel in the direction of the vector of the point where the droplet currently is. This is performed over and over again. Whenever the droplet makes an advance to some direction, the number 1 is added to the corresponding pixel in the accumulator array.

The tracing of one droplet comes to an end when:

1. The length of current gradient vector becomes zero – this happens in large plain surface of water in a reservoir.
2. Newly directed point happens to run across the former trace of the same droplet.
3. The newly directed point goes beyond the boundary of the image.

When the one drop tracing is finished, another raindrop is chosen to released at random point in the image until the number of raindrop reaches the assigned value.

Once a raindrop encounters a former drop's trace, it follows exactly the former drop's path. This convinces us that the larger valley will be highlighted more brightly

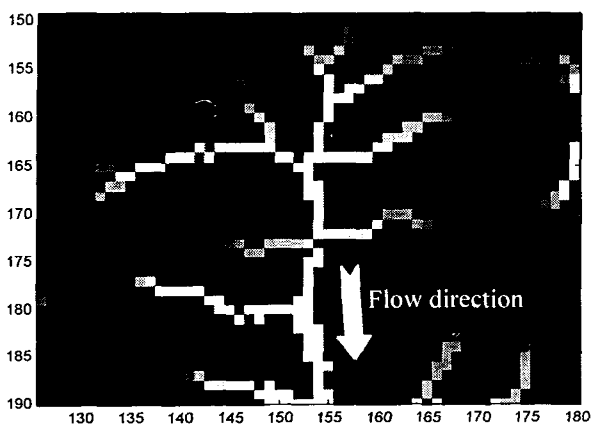

Figure 2. Highlighting process

with the increasing number of drops that passed by the same path. Figure 2 shows the process of highlighting. I cut out a small section of the study area, and magnify it. Obviously several small streamlets are flowing into large stream. At last the largest stream flows away in the direction as indicated with a white arrow.

When the tracing is forced to stop due to the case 2 above, we need to decide what to do next. This is necessary because the droplet sometimes arrive in a pseudo local minimum. This kind of minimum is also called as a "sink". This is caused by the real sink in nature and by the resolution limit of the image. To overcome this problem, I permit the encircled droplets to overflow above the barrier. To this end I set a small search window. Then start searching for another minimum in a small local window with the center of which located on the current point. If a minimum value found are smaller than the value of the current point, the current point is dragged to the new minimum point and the linear dragging path is also recorded on the accumulator array.

On this stage of work, I found that there appeared more sinks in the lower part of the streams than in the upper part of the streams, which made it difficult for a droplet to advance farther. In this case, larger search window is used in proportion to the ratio of the raindrops in the stream to the total number of raindrop to be sprinkled. That is, larger amount of water can overflow a larger barrier. By doing this, the raindrops tend to keep flowing until they meet the image boundary.

$$W_{size} = W_{\min} + (W_{\max} - W_{\min}) \times \frac{N_{current}}{N_{total}}$$

where W_{size} is the current search window size

$W_{min} = 7$ $W_{max} = 31$

$N_{current}$ is the number of droplets at current point

N_{total} is the number of total raindrops sprinkled

Hierarchical Hough Transform

Hierarchical Hough Transform (HHT, Princen and others, 1990) starts to use Hough transform in each 4×4 subimage and apply the Hough transform again at a higher level with combined larger subimage. Through this procedure detected lineaments are inherited from 4 child subimages to a parent subimage. By using subimages at each level the HHT can extract line segments instead of lines, which are the output of Standard Hough Transform (SHT, Dada and others, 1972).

RESULTS

Figure 3(a), and 3(b) show shaded relieves produced from the original DEM – Figure 1. Two images exhibit dramatic difference. Figure 3(a) was produced using ER-Mapper 6.0 with sun's azimuth of 288° and inclination of 30°. This image seems excellent for extracting the lineament which lie in the direction indicated by Direction 1, but it looks impossible to extract a lineament of perpendicular to Direction 1 especially in the center of the image. Likewise, Figure 3(b) was produced with sun's azimuth of 198° and inclination of 30°. This image shows lineament of Direction 2 best, but how can we detect the indistinct but existing lineament along Direction 1? In this situation we had better use the original DEM directly. It is hard to expect that one can extract a lineament that is not even visible to bare eyes

With Figure 4, on the contrary, it looks reasonable to remark the lineament of both Direction 1 and Direction 2. Figure 4 is obtained with 32000 raindrops and the scale of [5 30], which means assigning black color for less than and equal to 5 and white color for more than and equal to 30, and for the rest linear gray scale is applied from 5 to 30. Note that the result is totally unbiased and therefore may thought to represent lineament of any direction. Care must be taken when choosing the threshold of scaling to extract the proper lineaments.

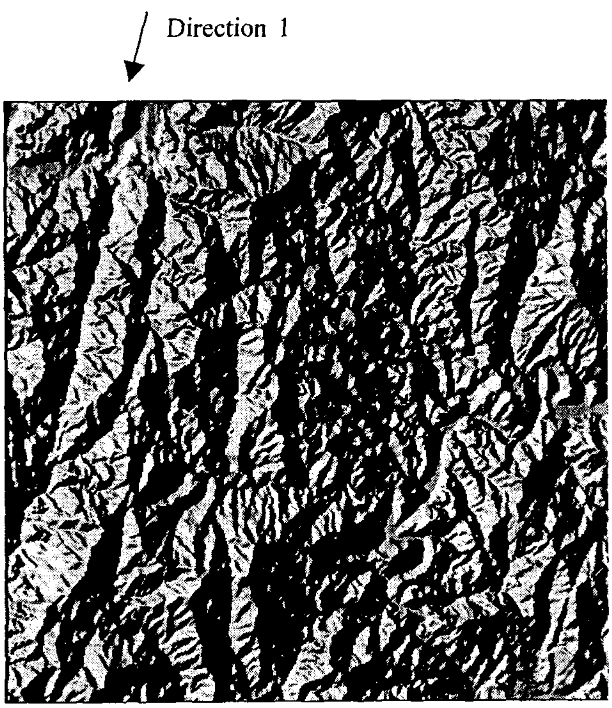

Figure 3(a). Shaded relief derived from the DEM
Sun's azimuth : 288°, inclination : 30°

Figure 3(b). Shaded relief derived from the DEM
Sun's azimuth : 198°, inclination : 30°

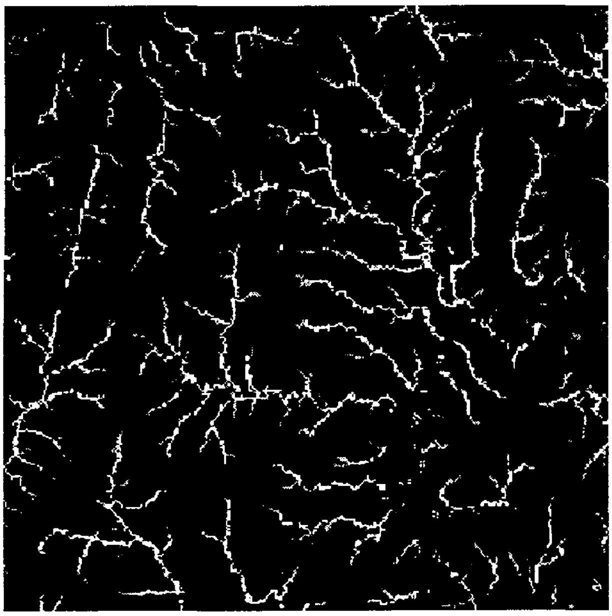

Figure 4. The result of the RTA with scaling [5 30]

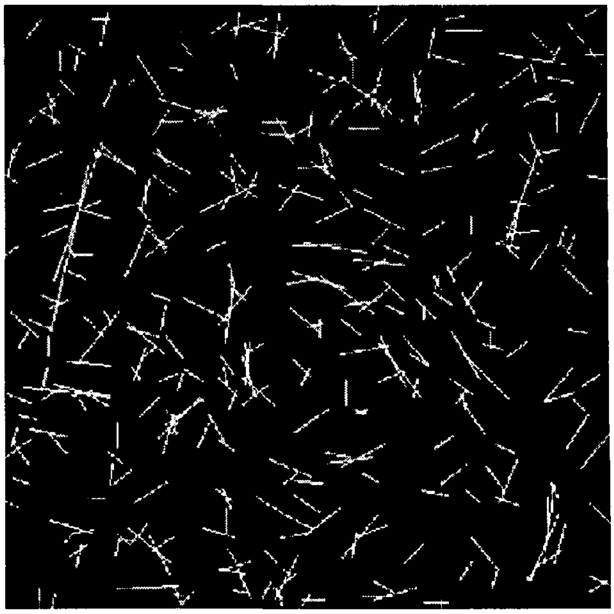

Figure 5. The result of the HHT

To be used as an input data of the HHT, this image should be made as to have only two intensity values – binary image. The number 32 was given as the threshold to make the binary image, and Figure 5 is the result of the HHT. Note the detection of both the directions.

The faults of this area can be divided into three types, thrust which has north-south trend with east to west transport direction, strike-slip fault which is the northeast-southwest trend of dextral sense, and strike-slip fault which is the E-W direction of dextral sense (Park and others, 1997). This can be checked with HHT, which helps us to classify the lineament in terms of its predominant direction. There is large variety of Hough transforms. I have checked several Hough transforms and decided to use the HHT, which is thought to be efficient for detecting line segments from complex image like Figure 4.

SENSITIVITY AND ERROR ANALYSIS

When using the RTA, there is another advantage of flexible calculation time. I used random sprinkle instead of starting from every pixel in the image. Now let's look into this point in detail. The acute reader may ask the sensitivity of the RTA to the number of raindrop. What happen if we use a different number of raindrops? If we assume that the result from raindrop-on-every-pixel (or full saturation) is exact solution, the RTA is 97.5% accurate when the number of raindrop is 2000, which is 1/80 of the number of pixels of the DEM. This means that we can spend only 1/80 of the exact calculation to have a reasonable solution. The key point is that we should keep the same ratio of the threshold level to the number of raindrop. If we reduce the total number of sprinkling raindrop, we can expect that there will be half as many droplets in a particular valley as before. Figure 6 shows the sensitivity of the RTA to the number of raindrop. The ratio of the threshold level to the number of raindrops is constant as 1/1000.

CONCLUSIONS

The main advantage of the RTA is that it is simple, straightforward, and unbiased. Only if we are to seek

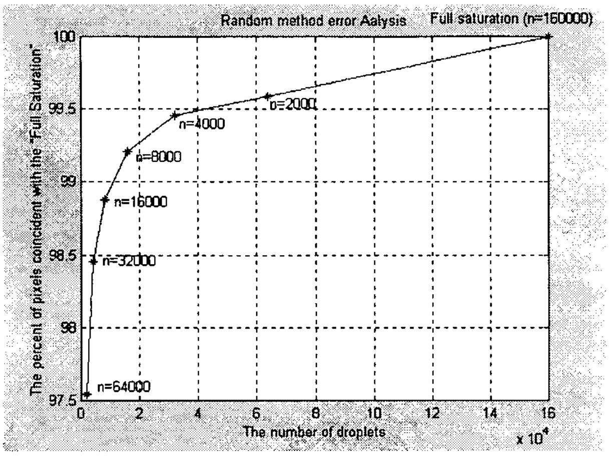

Figure 6. The sensitivity of the RTA to the number of raindrop sprinkled

lineament in the form of valley with DEM, the RTA seems robust and time efficient. This algorithm might be helpful to calculate the amount of flow in a specific stream when the intensity of precipitation varies spatially. For that case, the RTA can adopt the variance as a function of position instead of just sprinkling randomly.

REFERENCES

Costa, L. F., Mark B. Sandler, 1992, Effective Detection of Digital Bar Segment with Hough Transform, *CVGIP : Graphical Models and Image Processing*, 55(3), May, pp. 180-191.

Duda, R. O. and P. E. Hart, 1972, Use of the Hough transform to detect lines and curves in pictures, *Communications ACM*, 15(1), pp. 11-15.

Koike, K., S. Nagano, and M. Ohmi, 1995, Lineament Analysis of Satellite Images Using a Segment Tracing Algorithm (STA) : *Computers & Geosciences*, 21(9), pp. 1091-1104.

Kim, S., J. Won, W. Kim, and K. Min, 1999, Lineament Analysis in the Euiseong Area Using Automatic Lineament Extraction Algorithm, *Econ. Environ. Geol.*, 32(1), pp. 19-31.

Kim, W., J. Won, and S. Kim, 1996, A Technique Assessing Geological Lineaments Using Remotely Sensed Data and DEM : Euiseong Area, Kyungsang Basin, *Journal of the Korean Society of Remote Sensing*, 12(2), pp. 139-154.

Leavers, V. F., 1993, Survey Which Hough Transform?, *CVGIP : Image Understanding*, 58(2), September, pp. 250-264.

Lee, B., U. Chwae, and P. Kang, 1997, Lineaments in the Southeastern Part of the Korean Peninsula : *Journal of the Geological Society of Korea*, 33(1), pp. 18-26.

Lee, K., K. Chi, 1995, Spatial Integration of Multiple Data sets regarding Geological Lineaments Using Fuzzy Set Operation : *Journal of the Korean Society of Remote Sensing*, 11(3), pp. 49-60.

Park, K., B. Lee, D. Cho, and C. Kim, 1997, *Geologic Report of the Hwacheon Sheet* : Korea Institute of Geology, Mining & Materials.

Princen, J., J. Illingworth, and J. Kittler, 1990, A hierarchical approach to line extraction, *Computer Vision, Graphics, and Image Understanding*, 52, pp. 57-77.

Raghavan, V., S. Masumoto, and K. Koike, 1995, Automatic Linement Extraction from Digital Images Using a Segment Tracing And Rotation Transformation Approach, *Computers & Geosciences*, 21(4), pp. 555-591.