Computers & Geosciences 33 (2007) 229–237

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

A visual basic program for ridge axis picking on DEM data using the profile-recognition and polygon-breaking algorithm☆

Yet-Chung Chang[a,*], Gaurav Sinha[b]

[a]*Department of Network and Communication, Diwan College of Management, 87-1 Nanshih Li, Madou Tainan 72141, Taiwan, Republic of China*

[b]*Department of Geography University at Buffalo, The State University of New York—Buffalo, NY, 14261 USA*

Received 4 July 2005; received in revised form 11 May 2006; accepted 14 June 2006

Abstract

For many scientists working with digital topographic data, extracting lineaments or linear features is an important step in structuring and analyzing raw data. A ridge axis, which represents the top a mountain ridge, is one of the most important topographic features used in a wide variety of applications. Algorithms and software for automating the extraction of ridges or ridge axes from DEMs are, however, still not easily available or not widely acceptable. In this paper, we present a user-friendly Visual Basic program that automates the extraction of the ridge axis system from DEM data, based on the profile-recognition and polygon-breaking algorithm (PPA). An important feature of PPA is that it takes a global approach, as opposed to the local neighborhood operators used in many other algorithms. Each segment detected by PPA considers not only relations with contiguous neighboring grid points, but also strives to preserve the continuity of the global trend. This is an attempt to simulate human operators, who always factor in the overall trend of the lineament before delineating its local parts. PPA starts by connecting all points in a neighborhood that can possibly lie on the ridge axis, thus forming a belt of polygons in the first step. Next, a polygon breaking process eliminates unwanted segments according to the assumption that a ridge segment cannot be the side of any closed polygon, and that the result should be a purely dendritic line pattern. Finally, a branch-reduction process is executed to eliminate all parallel false ridges that remained due to the conservative approach taken in the first step. Results indicate that PPA is reasonably successful in picking out ridges that would have been identified manually by experts. In addition to providing a detailed user interface for executing PPA, several modifications were made to significantly improve the computational efficiency of PPA, as compared to the original version published in 1998. The source codes are provided for free download on the website listed above.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Ridge; Polygon; DEM; PPA; Visual basic; Algorithm

1. Introduction

Linear features that can be extracted from DEMs, satellite images or aerial photographs often provide useful information for scientists. Manual extraction of such features is labor-intensive and subject to

☆Code available at URL: http://ycc.dwu.edu.tw/Research/RidgePicker.htm.

*Corresponding author. Tel.: +886 6 5718888; fax: +886 6 5722829.

E-mail address: ycc@dwu.edu.tw (Y.-C. Chang).

0098-3004/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2006.06.007

variations in individual expertise and conceptions. With the availability of computers to scientists everywhere, automating the extraction of lineament features is being studied intensively by scientists to promote consistent and standardized lineament detection methods (Koike et al., 1995; Knappertsbusch, 1998; Mugglestone and Renshaw, 1998; Casas et al., 2000; Costa and Starkey, 2001; Székely and Karátson, 2004).

Although most extraction algorithms profess to be widely applicable and often suggested as the standard, they have, in reality, remained ad-hoc, idiosyncratic methods limited by the context for which they were first designed. Consequently, most researchers still tend to design their own lineament extraction algorithms. One of the reasons for the lack of an acceptable standard is that the lineament concept is too diffuse to have a generic solution. There are several kinds of lineaments that scientists have to identify and deal with individually: ridges, valleys, fault lines, water-land boundary, vegetation zone boundaries, soil-type changes, and many others. Each one of them is modeled differently. Even after lineament segments are identified and extracted, connecting them together into one *Gestalt* whole to maintain the global trend is a difficult task. For example, a ridge or drainage system tends to be dendritic; on the other hand, a shoreline should not have any branches. Even in these two cases, when the concept can be clearly stated, the process of axis thinning (Choy et al., 1995) and segment connecting (Lu and Cheng, 1990; Raghavan et al., 1995) are still too complicated to have a universal solution.

To address the heterogeneity in lineament detection, the authors of this paper suggest beginning from scratch with a simple model that can be applied to diverse lineament types. In this paper, the ridge axis is chosen as a simplified, geometric abstraction for all observable linear topographic features. By simplifying the problem thus, all the attention can be paid to the problems of axis thinning and lineament continuity. As many researchers have mentioned (Chorowicz et al., 1992; Band, 1986; Tarboton et al., 1991), both DEM resolution and production errors cause many of the problems for automated feature recognition algorithms. Such problems are minimized in manual feature detection, because of the human ability to overrule local inconsistencies in favor of the overall observable trend. Clear preference for preserving the global trend preempts false truncations and fragmented lineaments during manual extraction of lineaments. The algorithm for detecting the ridge axis, discussed in this paper, is inspired by the human focus on maintaining the global linear trend.

The original ridge axis algorithm that this paper seeks to improve was presented in Chang et al. (1998) for dendritic pattern recognition. It was also successfully implemented later as part of the popular open-source GIS-GRASS (Chang and Frigeri, 2002). This ridge axis detection program presented here can simulate the trend awareness that human operators display through its Profile-Recognition algorithm. A polygon-breaking, algorithm is then used to thin down the linear feature without introducing improper truncations. In this paper, the authors extend the previous work, by adopting a visualization based approach to ridge axis detection. This "visualized" approach to the complicated process of polygon-breaking is expected to make the ridge axis detection and extraction easier than the original algorithm proposed by Chang et al. (1998). The algorithm has also been enhanced computationally, to increase the efficiency by as much as ten times in some cases.

2. Data preparation

In this paper, the DEM for a small area from Nanto County in Central Taiwan is used for illustrative purposes (Fig. 1a). First, the DEM is read by our program and transformed into a gray-level image. The user can exaggerate the image scale for visualization purposes. Fig. 1b shows an example. The original data set is 100×100 pixels, with samples on a 40 m grid.

3. Profile-recognition

For most lineament extraction processes, target feature recognition based on contiguous neighboring pixels is unavoidable. For example, Chorowicz et al. (1992) proposed a "profile scan" method to recognize ridge, valley and others features in the DEM. Local pixel comparison methods suffer greatly, however, from scale effects. For instance, there is no reason to assume that a ridge top must always be a local maximum. Ridge tops can be relatively flat in many cases, in which case local measures will fail to detect them correctly (Miliaresis and Argialas, 1999). Detection failure can also occur when the DEM resolution is too coarse to capture small ranged variations. To make the ridge axis detection robust against spurious truncations,

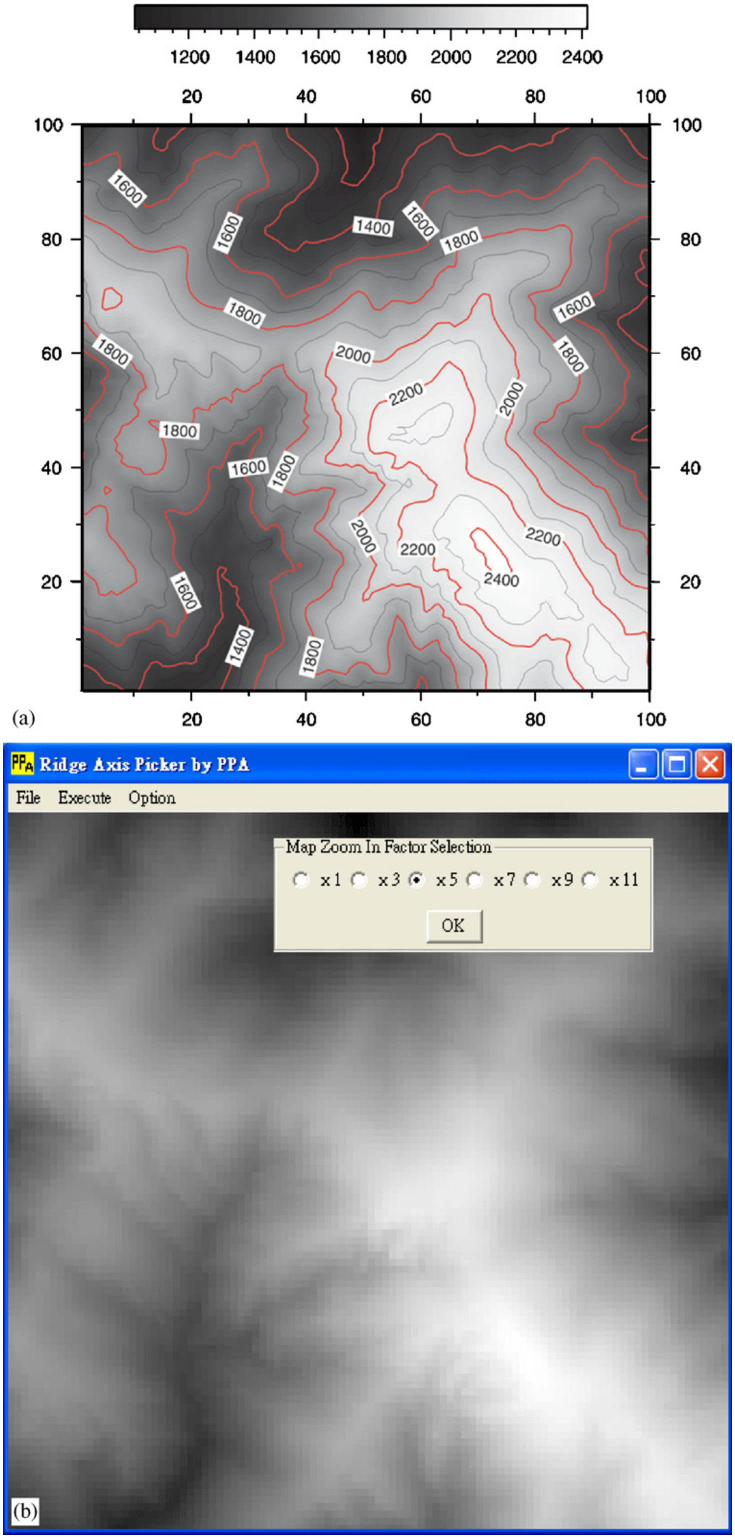

Fig. 1. (a) Contour map of the test area and (b) gray level elevation image.

Chang et al. (1998) used a profile length of five or more grid points, instead of the conventional three-pixel wide neighborhood, to decrease the probability of spurious breaks in the ridge axis. In the first profile-recognition step, therefore, if there is at least one point lower than a profile point on each side of the profile, the point is flagged as a ridge axis candidate.

As shown in Fig. 2, if the length of the evaluation profile is chosen as 5, then any point, which has a lower point *within two grid cells* on both sides, is recognized as a ridge axis candidate. The process is further repeated in all four grid directions, i.e. the N–S, E–W, NE–SW and NW–SE cardinal directions. This does make the ridge axis wider than a grid dimension—but the payoff is preservation of the continuity of the ridge axis as a connected system. Note that the choice of profile length is not crucial, unless it is short enough to endanger the continuity of the ridge because of occasional noises. Longer profiles ensure greater continuity, but also increase many unnecessary computations in the following cleaning processes. A length of five is recommended by the authors of the original paper.

Next, a Profile Recognition algorithm is used to simulate the human operator's tendency to ignore local perturbations that go against the overall trend of the linear feature. The linear segments that result from the profile recognition step are connected to form complex polygons. First, only the diagonally crossing segments are compared, and the one with lower elevation is eliminated (Fig. 3a). Then, the segments are compared with their existing parallel neighboring segments. If any segment is identified as higher than segments on both of its sides, it is assumed to be more "valid", while its parallel neighbors are eliminated. Segment heights are calculated as the average height of the two end points. These two steps simplify the polygon belts without breaking the continuity of the ridge system. The later step is an addition to the original version of PPA (Chang et al., 1998), and has the potential to significantly reduce the processing time by eliminating many spurious ridge axes.

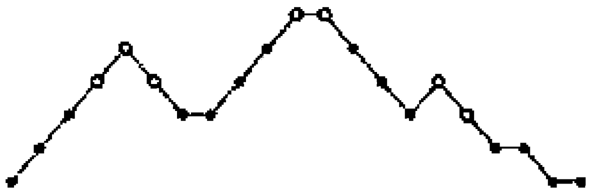

Fig. 2. Under five-point profile-recognition criterion, empty circles are picked as possible ridge points.

4. Polygon-breaking

The main purpose of the polygon-breaking method is to simplify the segment groups without breaking the continuity of the main ridge axes. It is very similar to the thinning process popular in optical character recognition (OCR). O'Gorman (1990) stated five important requirements for all thinning processes; it can be proven that the polygon-breaking process, presented in this section meets four of O'Gorman's requirements, whereas the fifth is achieved by the branch-reduction step, which is the next step of PPA. The polygon-breaking phase consists of the following steps:

1. Sort the segments following to their elevations.
2. Start from the lowest segment to check whether it is a side of a close polygon—if it is, the segment should be deleted, else preserved.
3. Shift the focus segment to next lowest one and repeat step 2, until all segments are processed.

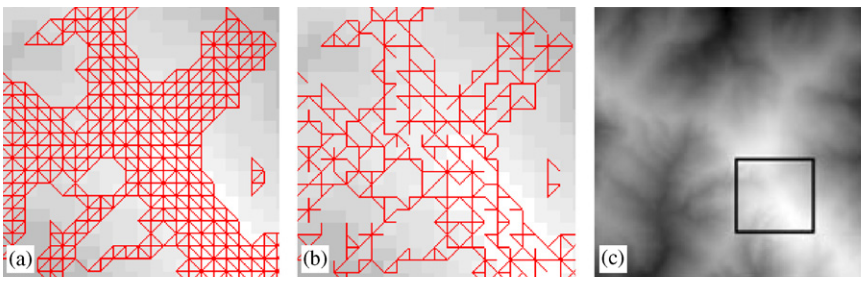

Fig. 3. Zoomed in images of part of ridge segment group. (a) Segments connected by all neighboring targets, only crossing segments with lower elevations are excluded. (b) Segments closely parallel to ridge segments are eliminated. (c) Zoom area of (a) and (b) in Fig. 1.

The polygon breaking process preserves branches and line continuity, but eliminates closed polygons. The overall resulting pattern of ridge axes appears dendritic at this stage. In Fig. 4a, the focus segment in yellow is apparently crossing a topographic valley and should be deleted. In Fig. 4b, this is not so apparent, but deletion of the segment is a reasonable step in thinning of the ridge axis. Both figures show that the Polygon-Breaking algorithm was executed correctly, even for complex polygons.

As shown in Fig. 4, the tracing process to identify a close polygon may be long and consume significant computational resources. A technique of ''dead-end detection'' has been designed in this paper to preempt many unnecessary tracing steps: when the segment tracing process encounters a dead-end of a branch, all the segments in this branch are marked as ''route disabled''. The tracing process then regresses to the root of the dead-end branch and the system flags the branch to be avoided for future tracings.

Another necessary but time-consuming process is the sorting of segment elevations. In this paper, the sorting algorithm is designed to especially take advantage of the fact that the number of possible elevations that the points on the segments can take is finite. The sorting process here is designed as a ''collecting process'' that identifies all segments with the same elevation. Generally, there are several for each elevation level, but identifying segments to be the same elevation, prevents them from being ordered by the sorting algorithm. In this program, only arbitrary sequential numbers are assigned to segments of the same (average) height to significantly speed up the segment sorting process, when compared to original PPA algorithm. In addition, segments higher than both the surrounding parallel segments are identified as better ridge axis candidates, and are consequently ranked higher (more reliable) than other segments at similar heights.

The result of the polygon-breaking process is shown in Fig. 5. The pattern is dendritic, which is

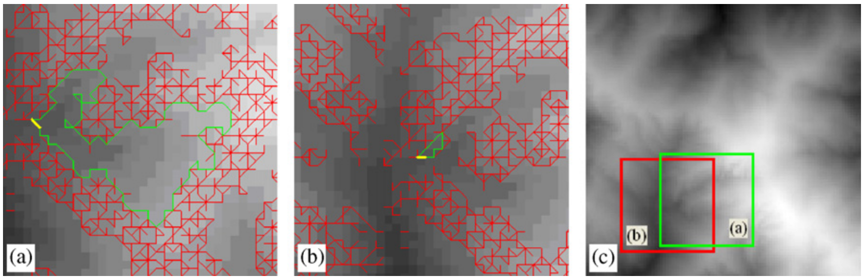

Fig. 4. Two examples of polygon tracing result. Yellow thick segment represents focus segment being checked, green ones represent polygon traced. Relative locations of (a) and (b) are shown on (c).

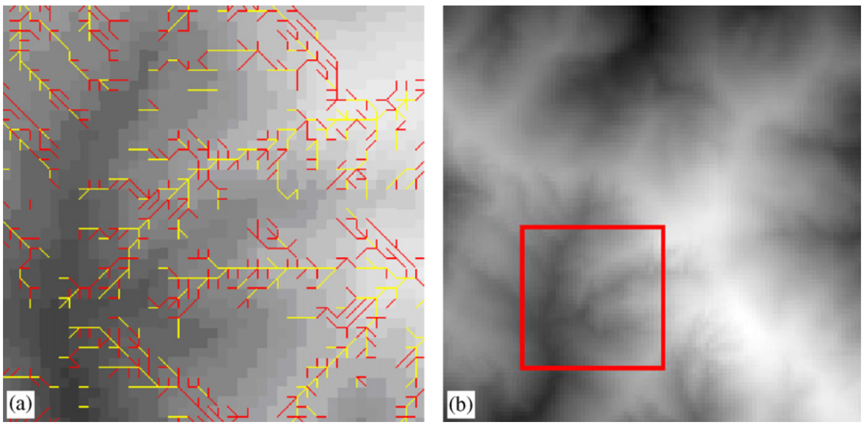

Fig. 5. (a) Result of polygon-breaking. (b) Area represented in (a).

typical of topographic ridges and valley axes systems. The yellow segments represent the "reliable" ridge axes segments. We can see that they comprise the major part of the ridge axis, but cannot always connect to each other. This is why we need a more flexible criterion to identify ridge targets in the earlier profile-recognition process.

5. Branch-reduction

The ridge axes identified in Fig. 5 are still too noisy for interpreters who are accustomed to manually prepared maps. branch-reduction is thus proposed as the next step to further reduce noise in ridge axis detection. Two kinds of features are designated as noise elements in this program: (i) branches with "unreliable" end-segments and (ii) branches too short to be assigned any semantic importance.

The end segments of each branch are continually deleted till the branch has a "reliable" terminal segment. In Fig. 5, all end segments depicted in red mark unreliable segments, which will need to be pruned. The assumption behind this step is that many false ridge axis segments are identified due to over-selection in the profile-recognition process. For the purpose of maintaining continuity, we allow unreliable segments around the joint of ridge lines, but consider the long tails of branches as unnecessary and misleading the semantic interpretation process. The result of the branch-reduction process is shown in Fig. 6.

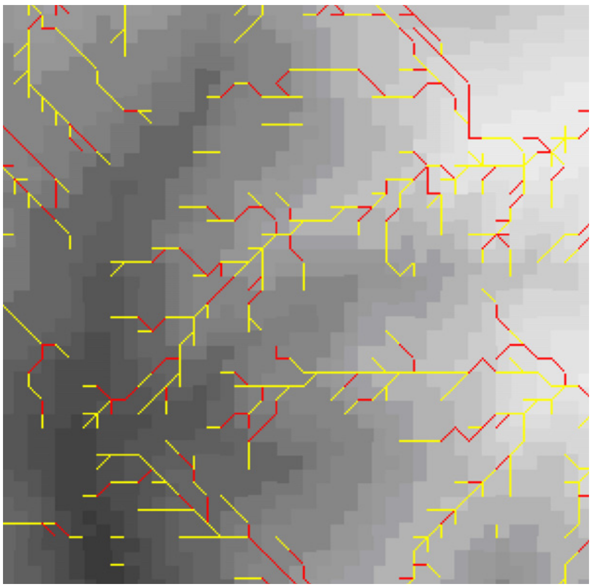

Fig. 6. Result of branch-reduction according to segment's ridge certainty.

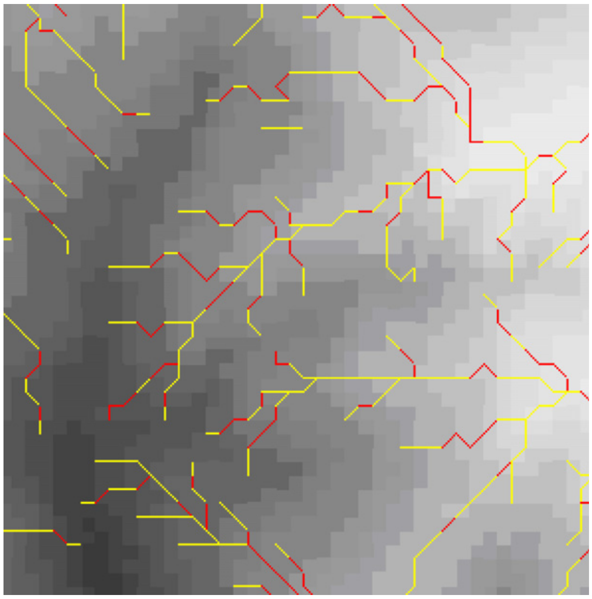

Fig. 7. After elimination of branches shorter than two segments.

After the first stage of branch-reduction, all ridge axes can be assumed to correspond to real-world linear features. However, there still probably remain many short branches or isolated small ridges, which may not have semantic importance. An optional clearance process can follow the branch-reduction process to weed out small, isolated segments. The tolerance size for weeding segments is decided by the user. Fig. 7 shows the result after elimination of branches shorter than three segments. The final result of ridge axis extraction is shown in Figs. 8 and 9. For visual comparison and evaluation purposes, Fig. 10 shows the ridge axes segments overlaid on a contoured background of the study area.

It can be seen that some unreasonable parallel branches still persist. They are mostly oriented in the diagonal (NW–SE or NE–SW) directions. The main reason for this is the difficulty in defining the adjacent parallel segments of a diagonal segment. As shown in Fig. 10, if the segment BF is considered, it is difficult to decide whether AE or DH is the proper parallel segment. They are both parallel to BF. However, AE is not aligned beside BF. On the other hand, DH is aligned with BF, but is farther away than AE. In this program, we use the

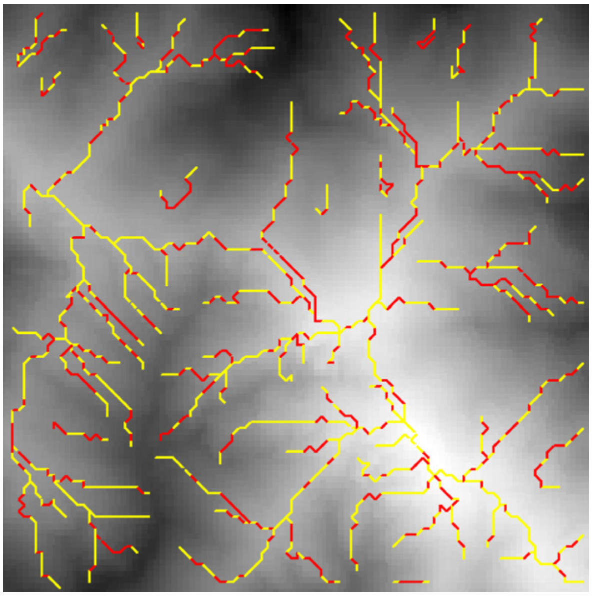

Fig. 8. The ridge axes of Fig. 1. extracted by PPA after elimination of branches with less than four segments.

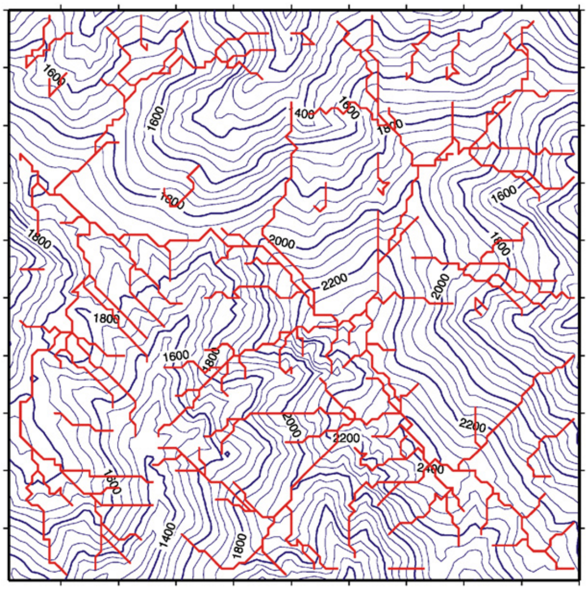

Fig. 9. Ridge axes overlaid on contour map of study area.

elevations of E and C to decide whether BF is a ridge segment or not. If they are both lower than BF, then BF is accepted as a valid ridge segment. Note that when segments are aligned parallel along either of the axis, the average height of the segments can be used to choose reliable segments.

Some, small hook-shaped features can also be seen in Fig. 9. This is a result of the compromise

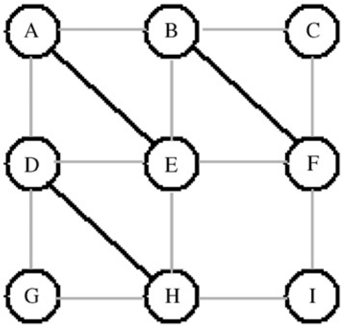

Fig. 10. Diagram of diagonally parallel segments.

achieved between strictly following the mathematical definition of the ridge and the desire to uphold the continuity of the ridge system. As can be seen in Fig. 8, the mathematically well-defined ridge segments in yellow have elevations higher than their parallel segments. On the other hand, the red colored segments are not so perfectly defined. They are necessary for continuity in most cases, but do have the chances to make some unreasonable parallel lines or small features.

6. Visual feedback

As mentioned earlier, for better understanding of the PPA based ridge axis detection and extraction, the lead author of this paper designed a Visual Basic program to display on-screen, in real time, the results from each of the steps: profile-recognition, polygon-breaking, and branch-reduction. Constant visual feedback reduces the efficiency of the program, but is considered crucial for better results as real time feedback enables the user to intervene at any step of the ridge extraction process. The earlier version of the PPA algorithm was focused on efficiency and hence lacked the interactive component that is at the core of the enhancements suggested in this paper. The authors hope that the stress on visualization of results from successive steps will promote better understanding of the procedure, which will in turn make it more accessible and become more sophisticate in future versions. The actual source code, which can be freely downloaded from URL: http://ycc.dwu.edu.tw/Research/RidgePicker.htm is also well commented to promote proper usage and modification by advanced users, who would like to improve the program.

7. Discussion and conclusion

The main contribution of the PPA method is that ridge axis continuity can be maintained successfully to a large extent. It should be noted that the polygon-breaking and the branch-reduction methods are capable of using a ‘‘whole map’’ context for each single segment. The traditional approach to preserving feature continuity is to increase the size of the window from 3×3 to 11×11 or more grid points (Choy et al., 1995; Koike et al., 1995; Costa and Starkey, 2001). However, such methods have failed to capture the holistic trend because grid window or mask-based morphometric evaluations can only detect localized grid patterns, which are difficult to connect into a meaningful ‘whole’. The PPA algorithm, on the other hand begins by building an over-connected whole and pruning it down to identify meaningful components. There is a fundamental philosophical difference, therefore, between PPA and local neighborhood based feature detection methods.

In addition to the method and techniques proposed by Chang et al. (1998), a new local recognition criterion of ridge ‘‘segments’’ (not gridded points) is also introduced in this paper to help eliminate some unwanted parallel false ridges. This makes polygon-breaking and branch-reduction mathematically reasonable and efficient. Moreover, the segment sorting process is simplified and executed much faster than before, based on the fact that no real sorting of segments with the same height are needed. It is an important improvement because: in the former version, the targets to be sorted were proportional to the total number of segments identified, which increases with the size of the study area. However, following the new sorting strategy, no matter how large the study area is, the numbers of targets for sorting are always proportional to the range of elevation found in the study area, since all segments with the same (average) height are identified first and grouped together. This computational efficiency enhancement, therefore, offsets to a certain extent the inefficiency introduced due to constant visual feedback provision.

While the discussion here has focused on ridge axis detection, it can also be used to detect valley axes or drainage channels. This can be done by transforming the original elevation values into negative numbers or subtracting them from a very high value and then using the PPA algorithm to detect valley axes and drainage channels. This was suggested by Chang et al. (1998) in their original discussion of PPA. However, there are drawbacks to applying PPA directly for valley bottom or channel detection. As most geologists and hydrologists know, there are significant topographic differences between ridge and drainage systems. For example, the bottom of a valley tends to be flat; on the other hand, the ridge top tends to be sharp edged and not as wide. Such differences arise because valleys and channels collect water and are continuously being modified by fluvial action, whereas ridge tops generally act as drainage divides and lose water. Most automatic drainage extraction algorithms therefore pay much attention to conquering the problem of flat valley bottoms (Chorowicz et al., 1992; Band, 1986; Tarboton et al., 1991). The ridge axis detection PPA algorithm will need to be modified to become an efficient valley bottom detector, therefore. Nonetheless, the PPA algorithm seems more promising in other areas. It has been applied for detecting shorelines from satellite images, shelf breaks on bathymetric maps and the skeletons of seismic profiles. Results of such efforts and many others will be reported elsewhere.

Appendix A. Supplementary Materials

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.cageo.2006.06.007.

References

Band, L.E., 1986. Topographic partition of watersheds with digital elevation models. Water Resources Research 22 (1), 15–24.

Casas, A.M., Corte’s, A.L., Maestro, A., Soriano, M.A., Riaguas, A., Bernal, J., 2000. LINDENS: a program for lineament length and density analysis. Computers & Geosciences 26 (9), 1011–1022.

Chang, Y.C., Frigeri, A., 2002. Implementing the automatic extraction of ridge and valley axes using the PPA algorithm in Grass GIS. In: Open Source Free Software GIS GRASS Users Conference, 2002.

Chang, Y.C., Song, G.S., Hsu, S.K., 1998. Automatic extraction of ridge and valley axes using the profile-recognition and polygon-breaking algorithm. Computers & Geosciences 24 (1), 83–93.

Chorowicz, J., Ichoku, C., Riazanoff, S., Kim, Y.J., Cervelle, B., 1992. A combined algorithm for automated drainage network extraction. Water Resources Research 28 (5), 1293–1302.

Choy, S.S.O., Choy, C.S-T., Siu, W-C., 1995. New single-pass algorithm for parallel thinning. Computer Vision and Image Understanding 62 (1), 69–77.

Costa, R.D., Starkey, J., 2001. PhotoLin: a program to identify and analyze linear structures in aerial photographs, satellite images and maps. Computers & Geosciences 27 (5), 527–534.

Knappertsbusch, M.W., 1998. Short note: a simple FORTRAN 77 program for outline detection. Computers & Geosciences 24 (9), 897–900.

Koike, K., Nagano, S., Ohmi, M., 1995. Lineament analysis of satellite images using a segment tracing algorithm (STA). Computers & Geosciences 21 (9), 1091–1104.

Lu, S.Y., Cheng, Y.C., 1990. An iterative approach to seismic skeletonization. Geophysics 55 (10), 1312–1320.

Miliaresis, G.C., Argialas, D.P., 1999. Segmentation of physiographic features from the Global Digital Elevation Model/GTOPO30. Computers & Geosciences 25 (7), 715–728.

Mugglestone, M.A., Renshaw, E., 1998. Detection of geological lineations on aerial photographs using two-dimensional spectral analysis. Computers & Geosciences 24 (8), 771–784.

O'Gorman, L., 1990. $k \times k$ thinning. Computer Vision Graphics Image Process 51, 195–215.

Raghavan, V., Matsumoto, S., Koike, K., Nagano, S., 1995. Automatic lineament extraction from digital images using a segment tracing and rotation transformation approach. Computers & Geosciences 21 (4), 555–591.

Székely, B., Karátson, D., 2004. DEM-based morphometry as a tool for reconstructing primary volcanic landforms: examples from the Börzsöny Mountains, Hungary. Geomorphology 63, 25–37.

Tarboton, D.G., Bras, R.L., Rodriguez-Iturbe, I., 1991. On the extraction of channel networks from digital elevation data. Hydrological Processes 5, 81–100.