Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Pattern Recognition Letters 25 (2004) 743–751
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Detection of ridges and ravines using fuzzy logic operations

Kyoung Min Kim [a,d,*], Joong Jo Park [b], Myung Hyun Song [c], In Cheol Kim [a], Ching Y. Suen [a]

[a] Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1455 de Maisonneuve Blvd. West, Suite GM606, Montreal, Canada H3G 1M8

[b] Department of Control and Instrumentation Engineering, Gyeongsang National University, 900, Gazwa-dong, Gyeongnam, Chinju 660-701, South Korea

[c] Department of Electric Control Engineering, Sunchon National University, 315, Maegok-dong, Chonnam, Suncheon 540-742, South Korea

[d] Department of Electrical Engineering, Yosu National University, San 96-1, Dunduk-dong, Chonnam, Yeosu 550-749, South Korea

Received 5 June 2003; received in revised form 5 January 2004

Abstract

In object analysis, line and curve finding plays a universal role, and it can be accomplished by detecting ridges and ravines in digital gray scale images. In this paper, we present a new method of detecting ridges and ravines by using local min and max operations. This method uses erosion and dilation properties of fuzzy logic operations and requires no information of ridge or ravine direction, so that the method is simple and easy in comparison with the conventional analytical methods. Experimental results from different types of images show that the proposed technique is both effective and efficient.

Keywords: Ridge; Ravine; Local min and max operation; Opening; Closing

1. Introduction

Line and curve finding plays an important role in object analysis, and this can be accomplished through the detection of ridges or ravines in the gray scale images. Therefore, one crucial part of the computer vision lies in the detection of ridge and ravine pixels (Haralick and Shapiro, 1992; Moon et al., 2002; Peucker and Douglas, 1975; Johnston and Rosenfeld, 1975; Haralick and Watson, 1981; Haralick, 1983). A digital ridge (ravine) occurs in a digital image when there is a simply connected sequence of pixels with gray level intensity values that are significantly higher (lower) than those neighboring the sequence, where there is a local maximum (minimum) in the direction across the ridge (ravine)

* Corresponding author. Address: Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1455 de Maisonneuve Blvd. West, Suite GM606, Montreal, Canada H3G 1M8. Tel.: +1-514-848-7956; fax: +1-514-848-4522.

E-mail address: kkm@cenparmi.concordia.ca (K.M. Kim).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.01.015

line. To determine ridges and ravines analytically (Haralick and Shapiro, 1992; Haralick and Watson, 1981; Haralick, 1983), we need to use the neighborhood of a pixel and compute directional derivatives, i.e., the direction across the ridge or ravine line is obtained by computing the extremum of the second directional derivatives, and the ridge peak or the ravine bottom can be determined by checking the values of the first directional derivatives in the same direction. However due to the computational complexity, this method has some practical complications when applied to the digital image; using the second directional derivatives for direction of ridge or ravine requires a time-consuming mask operation with 10 masks to obtain cubic surface for each pixel.

This paper presents a simple method of detecting ridges and ravines in the digital image by using fuzzy logic operations (local min and max operations). Nakagawa and Rosenfeld (1978), inspired by fuzzy set theory, suggested the operations of local min and max. These operations behave analogously to erosion and dilation on gray scale images, which form the basis for most of the gray scale morphological operations. Therefore, the combinations of the local min and max operations can be used to approximate various gray scale morphological operations (Giardina and Dougherty, 1988). Of the morphological operations, *opening* generally breaks narrow isthmuses and eliminates thin protrusions, and *closing*, conversely, fuses long thin lakes and eliminates small holes (Chen and Haralick, 1995; Nadadur and Haralick, 2000). These properties can be used to detect ridges and ravines in gray scale images. In this paper, opening and closing are implemented by combining local min and max operations to detect ridges and ravines in gray scale images. This method needs no information on the direction of the ridge or ravine line, and can easily detect ridges and ravines of various pixel widths.

2. Definition of image by fuzzy set theory

Since a gray scale image possesses some ambiguity within the pixels due to the possible multi-valued levels of brightness, it is justified to apply the concept and logic of fuzzy set rather than ordinary set theory to solve image processing problem (Gupta and Knopf, 1992). Let $\Pi_\phi = \{\mu_\phi^{ij};\ i = 1,2,\ldots,I,\ j = 1,2,\ldots,J\}$ be the fuzzy set representation of the pattern corresponding to an $I \times J$, L-level image array, where each element μ_ϕ^{ij} is a fuzzy singleton that represents the degree of membership of spatial coordinates (i,j) to the fuzzy subset given by ϕ. The various sets ϕ used in image processing generally involve notions such as gray level, color, texture, and shape.

In this paper the brightness property of the image is used. Therefore the given image is represented as an array of $I \times J$ fuzzy singletons, each with a membership value equal to the normalized gray level of the image at that point.

3. Local min and max operations

Shrinking and *expanding* operations on a binary image are useful for noise removal and segmentation. In such an image consisting of 0's and 1's, shrinking the 1's is equivalent to computing the logical AND of each pixel with its neighbors, and expanding the 1's is equivalent to logically ORing each pixel with its neighbors. One limitation of these operations is that they can be applied to binary images only. Therefore all real (gray scale) images have to be thresholded before they can be processed. In trying to cope with the multi-valued membership function, fuzzy logic has been introduced to extend the Boolean concepts of AND and OR to min and max respectively. For more general operations which behave the same way as the shrink/expand pair does, Nakagawa and Rosenfeld (1978) suggested local min and local max operations. The local min operation replaces the membership value at each spatial location (i,j) by the minimum membership value of itself and all immediate neighboring pixels, and the local max operation replaces the membership value at (i,j) by the maximum value of itself and its neighbors. These operations can be used to erode and dilate the multi-valued image.

Mathematically, erosion operation for a pixel at (i,j) is written as

$$\mathrm{EROSION}\{\mu_{\phi}^{ij}\} = \min_R \{\mu_{\phi}^{ij}\} \tag{1}$$

where $\min_R$ is the notation for the local min operator over the region R. The dilation operation is mathematically given by

$$\mathrm{DILATION}\{\mu_{\phi}^{ij}\} = \max_R \{\mu_{\phi}^{ij}\} \tag{2}$$

where $\max_R$ is the notation for the local max operator over the region R. Local min and max operations belong to the neighborhood operation. In the neighborhood operation, the output at a given pixel position is determined by the values of the input pixels in some neighborhood around the given input position. A neighborhood might be small or large. If we use a symmetric neighborhood centered around the given position, the neighborhood operation with a large neighborhood could be achieved by performing the neighborhood operation with a small neighborhood recursively.

Fig. 1 shows the range of neighborhood which affects the given center pixel when the local min or max operation is performed recursively. Fig. 1(a) is the case where a 4-connected neighborhood is used as the region R in $\min_R$ or $\max_R$ operator. If these operators are applied n times to the image, the resulting range of neighborhood which affects the given center pixel is the area composed of the pixels within the 4-neighbor distance equal to n shown in Fig. 1(b). Similar comments hold for 8-connected neighborhood as shown in Fig. 1(c) and (d). Therefore, instead of using the different sizes of neighborhood, the required erosion and dilation

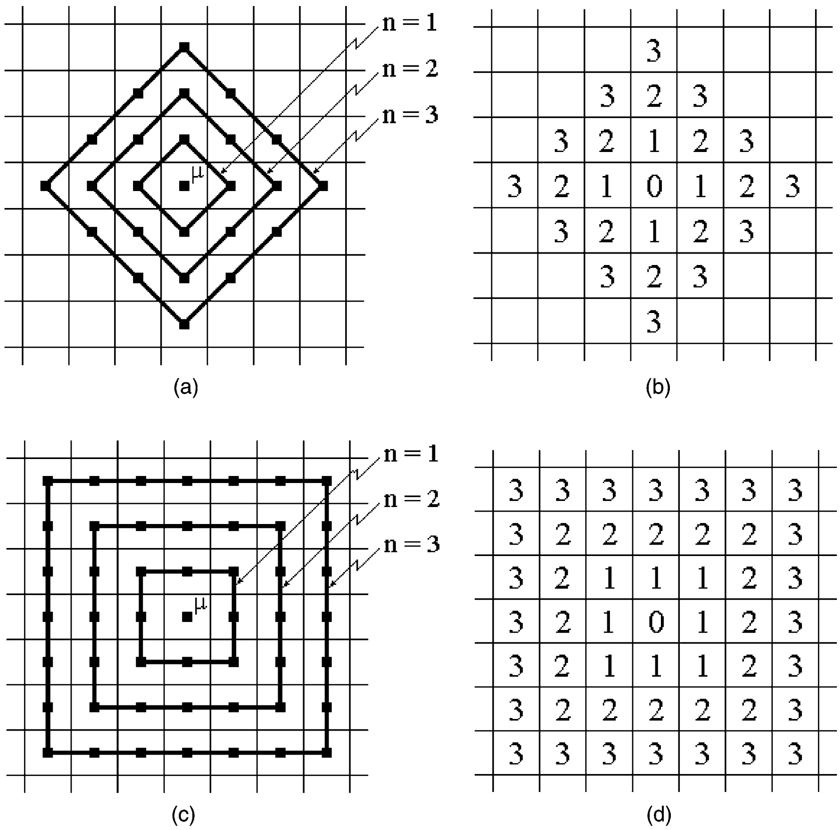

Fig. 1. Recursive applications of local min and max operations and its influenced range: (a) local min and max operations with 4-connected neighborhood and iteration no. n; (b) 4-neighbor distance; (c) local min and max operations with 8-connected neighborhood and iteration no. n; (d) 8-neighbor distance.

could be achieved by determining the local min and max operations with some minimal size of neighborhood and applying these operations recursively to the image.

4. Detection of ridges and ravines

The local min and max operations behave analogously to erosion and dilation on gray scale images (Chen and Haralick, 1995; Nadadur and Haralick, 2000; Nakagawa and Rosenfeld, 1978; Gupta and Knopf, 1992; Goetcherian, 1980), which form the bases for most of the morphological operations (Haralick and Shapiro, 1992; Gonzalez and Woods, 1992; Giardina and Dougherty, 1988). Therefore, we can extend the application areas of local min and max operations by referring to the properties of several basic gray scale morphological operations.

In gray scale morphological methods, the gray scale dilation of image f by image (structuring element) b, denoted $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y) + b(x,y) \mid (s-x,t-y) \in D_f; (x,y) \in D_b\} \tag{3}$$

And, the gray scale erosion of f by b, denoted $f \ominus b$, is defined as

$$(f \ominus b)(s,t) = \min\{f(s+x,t+y) - b(x,y) \mid (s+x,t+y) \in D_f; (x,y) \in D_b\} \tag{4}$$

where D_f and D_b are the domains of f and b, respectively.

Opening and closing are two other important morphological operations for this process. The gray scale opening of image f by b, defined $f \circ b$ is defined as

$$f \circ b = (f \ominus b) \oplus b \tag{5}$$

And the gray scale closing of f by b, denoted $f \bullet b$, is defined as

$$f \bullet b = (f \oplus b) \ominus b \tag{6}$$

Generally, the effects of the gray scale opening are to remove small bright details from an image, while leaving the overall gray levels and larger bright features relatively intact. The gray scale closing tends to remove dark details from an image, while leaving bright features relatively undisturbed. From these properties, the gray scale opening and closing can be used to extract image components such as ridges and ravines, i.e., by using the combinations of the local min and max operations.

For an image

$$\Pi_G = \{\mu_G^{ij}; \quad i = 1,2,\ldots,I, \ j = 1,2,\ldots,J\}$$

where μ_G^{ij} is a membership function, which represents the brightness property of the image, the effect of gray scale opening can be obtained by

$$\mathrm{OPENING}\{\mu_G^{ij}\} = \max_R \left\{ \min_R \{\mu_G^{ij}\} \right\} \tag{7}$$

And the effect of gray scale closing can be obtained by

$$\mathrm{CLOSING}\{\mu_G^{ij}\} = \min_R \left\{ \max_R \{\mu_G^{ij}\} \right\} \tag{8}$$

where the region R in $\min_R$ and $\max_R$ operators corresponds to the structuring element in the morphology. Then, the positions of ridge pixels in the image can be detected by computing the difference between the original image Π_G and its opened version, i.e., the ridge image Π_{RIDGE} is obtained by

$$\Pi_{\mathrm{RIDGE}} = \{\mu_{\mathrm{RIDGE}}^{ij}; \quad i = 1,2,\ldots,I, \ j = 1,2,\ldots,J\} \quad 0 \leqslant \mu_{\mathrm{RIDGE}}^{ij} \leqslant 1 \tag{9}$$

where

$$\mu_{\mathrm{RIDGE}}^{ij} = \begin{cases} 1 & \text{if } \mu_G^{ij} - \max_R \left\{ \min_R \{\mu_G^{ij}\} \right\} > 0 \\ 0 & \text{otherwise} \end{cases}$$

In Eq. (9), if we use a 4- or 8-connected neighborhood as the region R in $\min_R$ and $\max_R$ operators, ridges not exceeding 2-pixel width can be detected. Wider ridges can be detected by extending Eq. (9) as follows:

$$\Pi_{\mathrm{RIDGE\_}n} = \{\mu_{\mathrm{RIDGE\_}n}^{ij}; \ i = 1,2,\ldots,I, \ j = 1,2,\ldots,J\} \quad 0 \leqslant \mu_{\mathrm{RIDGE\_}n}^{ij} \leqslant 1 \tag{10}$$

where

$$\mu_{\mathrm{RIDGE\_}n}^{ij} = \begin{cases} 1 & \text{if } \mu_G^{ij} - \max_R^{(n)} \left\{ \min_R^{(n)} \{\mu_G^{ij}\} \right\} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of iterations of $\min_R$ and $\max_R$ applied to the image. In Eq. (10), if we use a 4- or 8-connected neighborhood as the region R in $\min_R$ and $\max_R$ operators, ridges not exceeding $(2 \times n)$ pixel width can be detected. If we need not only the positions but also the height information of ridges, we can obtain them by modifying Eq. (10) to

$$\Pi_{\mathrm{RIDGE}\_n} = \{\mu^{ij}_{\mathrm{RIDGE}\_n};\quad i = 1,2,\ldots,I,\quad j = 1,2,\ldots,J\}\quad 0 \leqslant \mu^{ij}_{\mathrm{RIDGE}\_n} \leqslant 1 \tag{11}$$

where

$$\mu^{ij}_{\mathrm{RIDGE}\_n} = \begin{cases} \mu^{ij}_G & \text{if } \mu^{ij}_G - \max^{(n)}_R \left\{ \min^{(n)}_R \{\mu^{ij}_G\} \right\} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the position of ravine pixels in the gray scale image can be detected by computing the difference between the original image Π_{G} and its closed version, i.e., the ravine image Π_{RAVINE} is obtained by

$$\Pi_{\mathrm{RAVINE}\_n} = \{\mu^{ij}_{\mathrm{RAVINE}\_n};\quad i = 1,2,\ldots,I,\quad j = 1,2,\ldots,J\}\quad 0 \leqslant \mu^{ij}_{\mathrm{RAVINE}\_n} \leqslant 1 \tag{12}$$

where

$$\mu^{ij}_{\mathrm{RAVINE}\_n} = \begin{cases} 1 & \text{if } \min^{(n)}_R \left\{ \max^{(n)}_R \{\mu^{ij}_G\} \right\} - \mu^{ij}_G > 0 \\ 0 & \text{otherwise} \end{cases}$$

In Eq. (12), if we use a 4- or 8-connected neighborhood as the region R in $\min_R$ and $\max_R$ operators, ravines not exceeding $(2 \times n)$ pixel width can be detected. If we need not only the positions but also the height information of ravines, we can obtain them by modifying Eq. (12):

$$\Pi_{\mathrm{RAVINE}\_n} = \{\mu^{ij}_{\mathrm{RAVINE}\_n};\quad i = 1,2,\ldots,I,\quad j = 1,2,\ldots,J\}\quad 0 \leqslant \mu^{ij}_{\mathrm{RAVINE}\_n} \leqslant 1 \tag{13}$$

where

$$\mu^{ij}_{\mathrm{RAVINE}\_n} = \begin{cases} \mu^{ij}_G & \text{if } \min^{(n)}_R \left\{ \max^{(n)}_R \{\mu^{ij}_G\} \right\} - \mu^{ij}_G > 0 \\ 0 & \text{otherwise} \end{cases}$$

In this case, zero valued ravine pixels in the original image are removed, therefore some labeling on them are required. Fig. 2 shows the procedures of detecting ridges and ravines in a scan line of a gray scale image by using Eqs. (11) and (13) with $n = 1$, respectively.

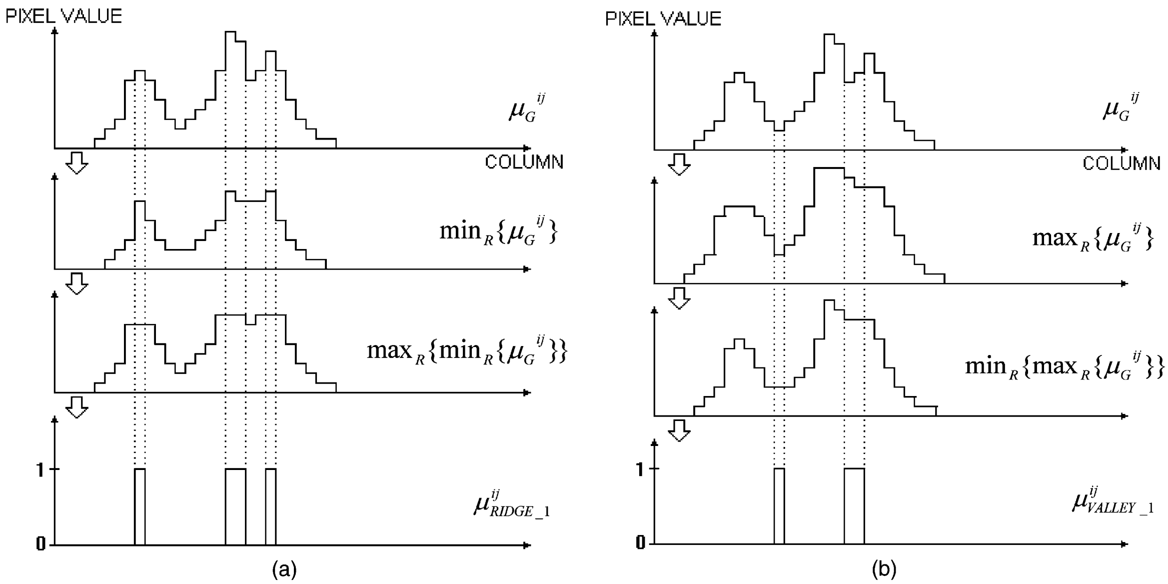

Fig. 2. The procedures of detecting ridges and ravines: (a) ridge detection; (b) ravine detection.

5. Experimental results

To illustrate the power of the new ridge–ravine detection scheme, we show its results on digital patterns and real images. Fig. 3 presents the results on digital patterns and their 3-D view. Fig. 3(a) is the input pattern and Fig. 3(b) and (c) depict the detected ridges and ravines, respectively. Fig. 3(d) is another pattern and Fig. 3(e) and (f) show the detected ridges and ravines, respectively. To obtain these results, Eqs. (11) and (13) are used with $n = 1$.

In order to verify the correctness of our method, we compare the detected ridges and ravines of the digital pattern shown in Fig. 3 with their corresponding real ones. By overlapping each other as shown Fig. 4, we can see that the detected ridges and ravines are almost same as their real ones.

The real image of an airphoto is illustrated in Fig. 5(a). Fig. 5(b) shows the ridges alone obtained by using Eq. (11) with $n = 1$ and appropriate thresholding, and Fig. 5(c) shows the ridges overlaid on the airphoto scene.

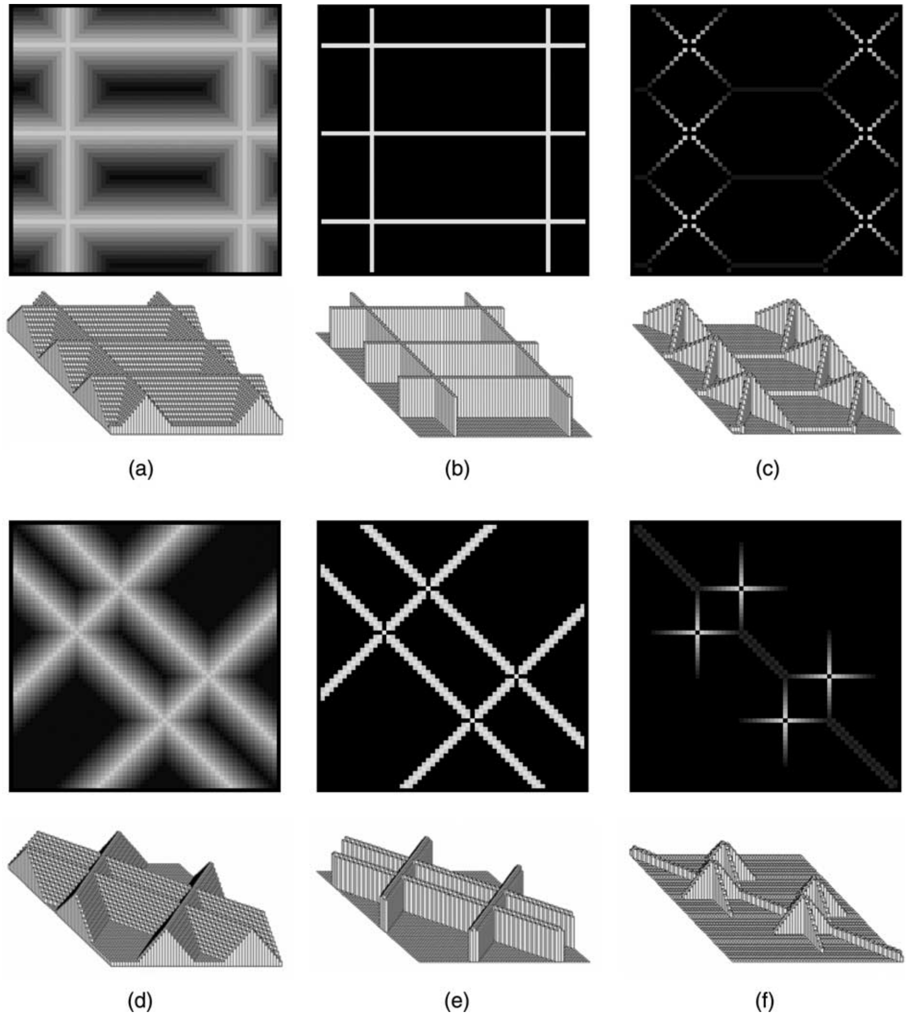

Fig. 3. The computed ridges and ravines for the pattern images: (a) original image; (b) ridges of (a); (c) ravines of (a); (d) original image; (e) ridges of (d); (f) ravines of (d).

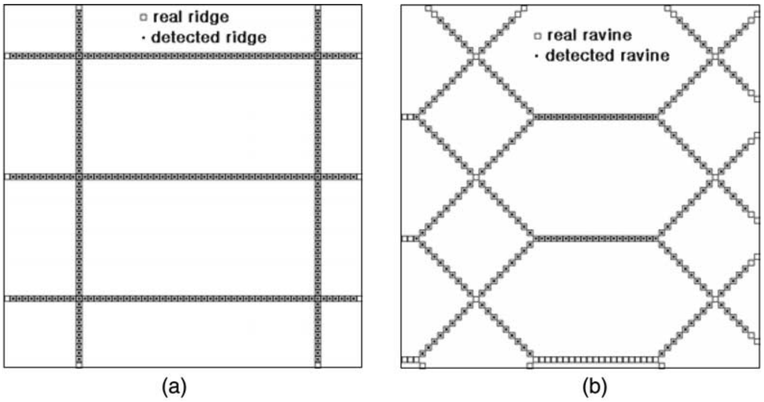

Fig. 4. The real ridges and ravines overlaid with detected ones (a) ridges and (b) ravines.

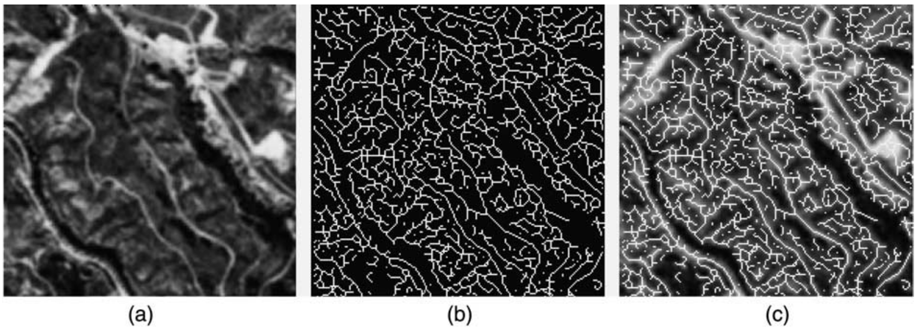

Fig. 5. The computed ridges for the airphoto scene: (a) original image; (b) ridges of (a); (c) ridges overlaid on original image.

Lastly, we measure the processing time of the proposed method and compare it with those of existing methods. In experiments on 256×256 of the lenna edge image, our method requires about 101 ms of CPU time to obtain ridge image on a PC of Pentium III-900 MHz with Windows 2000 operating system. This processing time is quite faster than that of Haralick method (1185 ms) but somewhat slower than that of Johnston and Rosenfeld method (16 ms). However, from the resulting ridges shown in Fig. 6, it can be seen that the ridge information is more clearly preserved by our method when compared to Johnston and Rosenfeld method.

Fig. 7(a) contains an image composed of different characters on different backgrounds in gray levels. We can segment characters of different widths from the background by using our ridge detection method. Fig. 7(b) brings out the characters obtained from Eq. (10) with $n = 2$.

6. Conclusion

We have presented an efficient method of detecting ridges and ravines in gray scale images by using local min and max operations. This method stems from the idea that, of the gray scale morphological operations, the opening operation tends to remove small bright details and the closing operation removes dark details from the gray scale image. In this method, the properties of opening and closing are implemented by combining the fuzzy logic operations (local min and max operations). This method needs no information of ridge or ravine direction, and has been shown to be

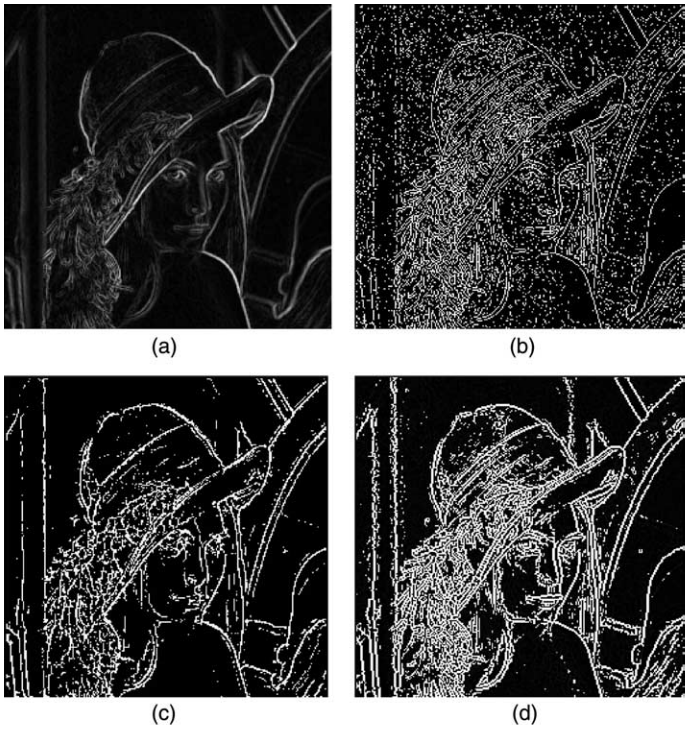

Fig. 6. (a) Gray scale lenna edge image and its ridges detected by using (b) Johnston and Rosenfeld method; (c) Haralick method, and (d) proposed method.

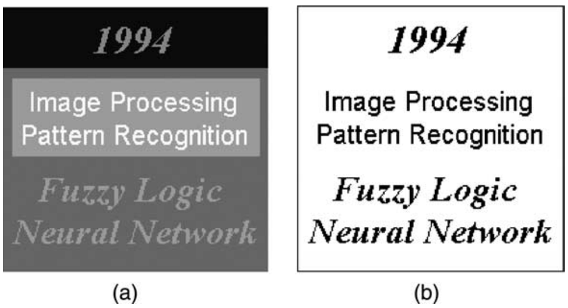

Fig. 7. The computed ridges for the character image: (a) original image; (b) ridges of (a).

capable of detecting ridges and ravines of various pixel widths.

References

Chen, S., Haralick, R.M., 1995. Recursive erosion, dilation, opening, and closing transforms. IEEE Trans. Image Process 4 (3), 335–345.

Giardina, C.R., Dougherty, E.R., 1988. Morphological Methods in Image and Signal Processing. Prentice-Hall, Englewood, Cliffs, NJ.

Goetcherian, V., 1980. From binary to grey tone image processing using fuzzy logic concepts. Pattern Recognit. 12, 7–15.

Gonzalez, R.C., Woods, R.E., 1992. Digital Image Processing. Addison-Wesley, Reading, MA.

Gupta, M.M., Knopf, G.K., 1992. Fuzzy logic in vision perception. SPIE Intell. Robots Comput. Vision XI 1826, 300–316.

Haralick, R.M., 1983. Ridges and valleys on digital images. Comput. Vision Graphics Image Process 22, 28–38.

Haralick, R.M., Shapiro, L.G., 1992. Computer and Robot Vision. Addison-Wesley, Reading, MA. vol. 2.

Haralick, R.M., Watson, L., 1981. A facet model for image data. Comput. Vision Graphics Image Process 15, 113–129.

Johnston, E.G., Rosenfeld, A., 1975. Digital detection of pits, peaks, ridges and ravines. IEEE Trans. Systems Man Cybernet SMC-5 (4), 472–480.

Moon, H., Chellappa, R., Rosenfeld, A., 2002. Optimal edge-based shape detection. IEEE Trans. Image Process 11 (11), 1209–1227.

Nadadur, D., Haralick, R.M., 2000. Recursive binary dilation and erosion using digital line structure elements in arbitrary orientation. IEEE Trans. Image Process 9 (5), 749–759.

Nakagawa, Y., Rosenfeld, A., 1978. A note on the use of local min and max operations digital picture processing. IEEE Trans. Systems Man Cybernet SMC-8 (8), 632–635.

Peucker, T.K., Douglas, D.H., 1975. Detection of surface-specific points by local parallel processing of discrete terrain elevation data. Comput. Vision Graphics Image Process 4, 375–387.