Algorithm to Compute the Depth of a Basin using 8-Connected Skeleton

G. Guzmán, R. Menchaca, M. Moreno, M. Torres and R. Quintero

Geoprocessing Laboratory (GEOLAB)
Centre for Computing Research (CIC)
Mexico City, Mexico
{jguzmanl, rmen, marcomoreno, mtorres, quintero}@cic.ipn.mx

Abstract. In this paper, we present a method to compute the depth of a basin using its 8-connected skeleton. The proposed algorithm consists of two stages: First, the preprocessing stage, which is based on transforming the source image into a binary format. The second stage is denominated quantifying that is used to find the skeleton and its variables to determine the depth of the basin. Although in most cases the algorithm provides good results, we found several pathological situations. However, these cases can be solved by means of a crossing point fusion in conjunction with morphological opening algorithms. Our approach uses as input geo- image that contains only information about the studied basin. Moreover, the image contains the entire trajectory of the basin. This algorithm can be used in diverse areas of Geocomputation; similarly, by making some modifications in the parameters to obtain the features. In addition, the algorithm can be applied to other problems related to the identification and quantification of cartographic objects.

Keywords: Image Processing, Skeleton, Mathematical Morphology, Geo-Image.

1 Introduction

Previous works [1-3], propose algorithms to extract specific information of high resolution raster images, for instance; highways, rivers, populations, among others. Usually, these algorithms (automatic or semiautomatic) are used to process geo-images. In addition, the output image only contains the pixels that compose the desired objects, denominated raster to vector algorithms. These works avoid a manual digitalization of the data. Nevertheless, in Geocomputation it is frequently required to know quantitative information about the geographic data set, hence, having data in raster or vector format is commonly not enough.

The determination of the depth of a basin is a typical example in hydrology, geomorphology and cartography; because from this datum other parameters can be obtained, e.g. flood risk, bifurcation ratio [9], water flow, etc.

We develop an algorithm to determine, the depth of a basin, following the method formulated by Horton [4]. The advantages of our technique are a minimum computing cost and the final result is the same that obtained using the Horton method. Up-to-date there is not an automatic method or algorithm to compute this.

The paper is organized as follows: in section 2 the mathematical foundations of the algorithm are sketched out, section 3 explains in detail the proposed algorithm, in section 4 we present the obtained results, and finally in section 5 we outline the conclusions and future work of the present method.

2 Mathematical Background

In this section, we present some important concepts of digital geometry. Let $p=(x,y)$ and $q=(u,v)$ be points of Z^2. The following metrics are commonly used:

$$\begin{aligned} d_4 &= |x-u|+|y-v| \\ d_8 &= máx(|x-u|,|y-v|) \\ d_{6R} &= máx(|x-u|,|y-v|,|x-u+y-v|) \\ d_{6L} &= máx(|x-u|,|y-v|,|x-u-y+v|) \end{aligned} \quad (1)$$

Each one of those metrics $d_k, k \in \{4,6L,6R,8\}$, lead to a none reflective symmetric relation N_k defined by $(p,q) \in N_k \Leftrightarrow d_k(p,q)=1$. The structure (Z^2, N_k) usually is named *k*-connected graph. We say that points p,q are *k*-neighbors if $(p,q) \in N_k$, and $N_k(p)$ denotes all k – neighbors of *p*. In this paper, we use a 8-connectivity, that is, $k=8$.

A subset $R \subseteq Z^2$ is *k*-connected if for any $p,q \in R$, exists at least one sequence of points in $R, p=a_1,a_2,...,a_n=q$, where a_i and a_{i+1} are *k*-neighbors for $i=1, ..., n-1$. Let *S* the 8-connected skeleton of a region *R*, and *p* any point that satisfies the condition $p \in S$, we have:

1. If $N_8(p)=1$, p is a terminal point, TP.

2. If $N_8(p)=2$, p is an internal point, IP.

3. If $N_8(p)\geq 3$, p is a cross point, CP.

4. If $N_8(p)=3$, p is a strict triad, TE.

3 Proposed Algorithm

As we stated previously, the developed method consists of two stages: one for preprocessing and another for quantification. The main purpose of the *preprocessing* stage is to take a color (RGB values) or gray level image and generate as output a binary image using this convention: an intensity of 1 is assigned to object pixels; whereas an intensity of 0 is used for none object pixels. In the *quantification* stage, we compute two variables to determine the depth of the basin, the number of terminal points and the number of crossing points. Both stages are shown in Figure 1.

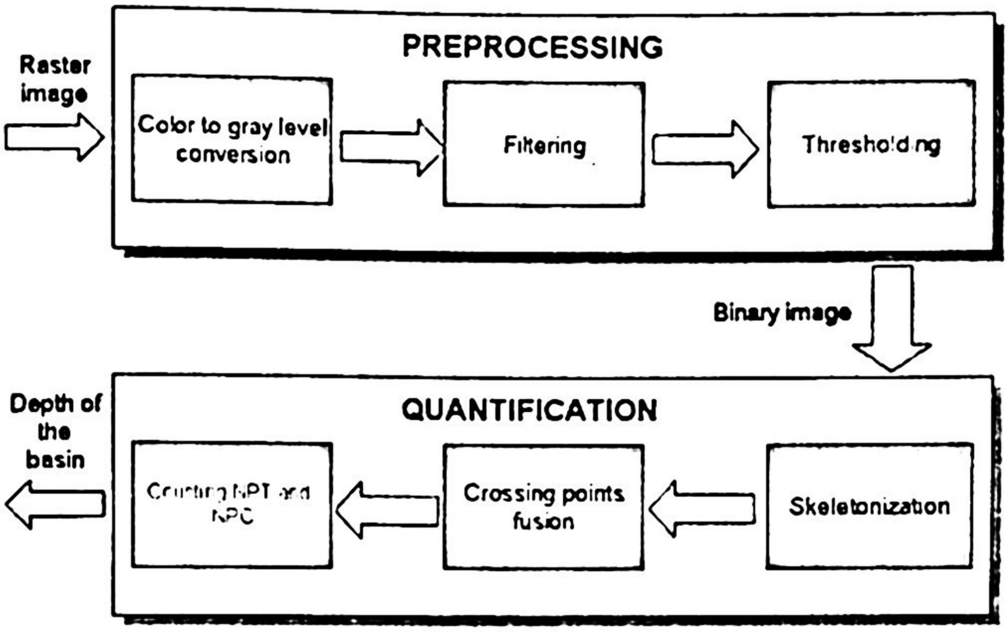


Fig. 1. Preprocessing and quantification stages

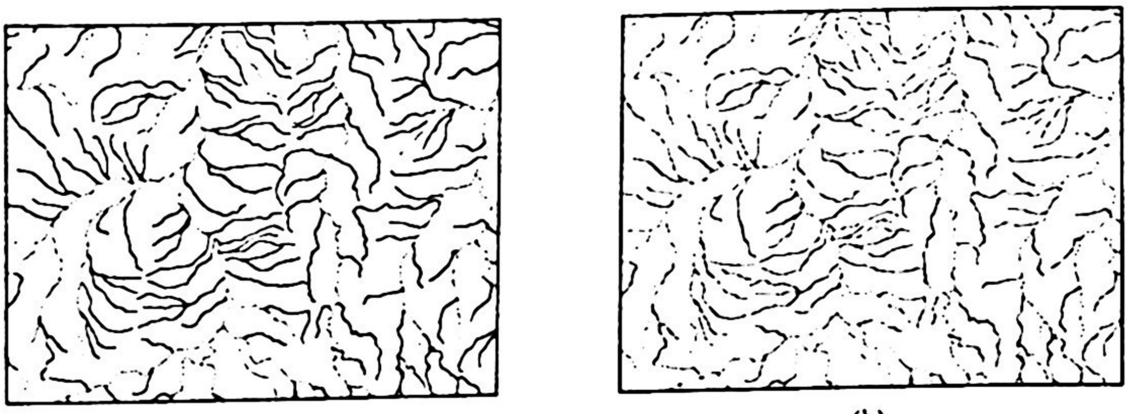
1. *Color to gray level conversion.* If the pixels are in RGB format, we apply a conversion using Equation 2 for each pixel of the original image.

$$gray-level = \frac{299*R+587*G+114*B}{1000}. \quad (2)$$

Where R, G, B are the red, green and blue components respectively of a pixel *p* in an image.

We use a conversion, based on the YIQ model because the loss of descriptive information is minimal in this case. Figure 2 shows an example using this formula.

2. *Filtering.* To decrease the *probable* noise contained in the digital image, we apply a smoothing algorithm, for example, the median filter with a mask of 3 x 3 pixels. It is not required to remove all noise in the contour of the *basin* (watershed), since the performance of the skeletonization algorithm is not affected by the presence of structural noise and this algorithm guarantees removing any possible parasite branches. Also, some other technique of smoothing can be used, such as ordering or average filter [8].



(a) (b)

Fig. 2. (a) Input color image. (b) Gray level image generated using YIQ model

3. *Thresholding*. The purpose of this part is to accurately identify all object regions in the image. We can use one of different techniques, which generally are based on the Otsu method [6]. The selected algorithm [7] makes use of probability techniques, which allow us to choose a suitable threshold for the thresholding algorithm. The binary image that we obtained at the output of the *preprocessing* stage is used as input for the *quantification*.

4. Skeletonization. This process generates the 8-connected skeleton of the source image. To do this, we use the thinning algorithm developed by Díaz [5]. This method removes all contour pixels without breaking the connectivity of the region. This algorithm is based on two conditions: terminal point rule and an own set of templates that decides if a pixel can be or not removed from the image. Basically, this algorithm consists of the next steps:

   - Let R be a region, compute the contour of this region using the next criterion: a pixel $p \in R$ is a contour pixel if at least one k'-neighbor q, where $q \notin R$, exists. It is very important to use dual metric; otherwise parasites braches or ruptures in the connectivity may exist in the final skeleton. As we need an 8-connected skeleton the dual metric used is 4-connected.

   - Using the template shown on Figure 3, each contour pixel is examined to determine if it can be removed or not. If the pixel and its neighbors don't match any of the templates we proceed to eliminate it.

   - If no pixel was removed in the previous step, the region is the desired skeleton, otherwise, return to first step.

$$A_1 = \begin{bmatrix} 1 & 0 & a \\ 0 & p & b \\ e & d & c \end{bmatrix} \qquad A_5 = \begin{bmatrix} X & 0 & X \\ 1 & p & 1 \\ X & 0 & X \end{bmatrix}$$

$$A_2 = \begin{bmatrix} e & 0 & 1 \\ d & p & 0 \\ c & b & a \end{bmatrix} \qquad A_6 = \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c & d & e \\ b & p & 0 \\ a & 0 & 1 \end{bmatrix} \qquad A_7 = \begin{bmatrix} 0 & f & 0 \\ i & p & g \\ 0 & h & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} a & b & c \\ 0 & p & d \\ 1 & 0 & e \end{bmatrix} \qquad \begin{matrix} a+b+c+d+e>1 \\ f+g+h+i=1 \end{matrix}$$

Fig. 3. Templates set to generate 8-connected algorithm

5. *Crossing points fusion.* Due to basin topology, it is possible to find cases in which the skeleton algorithm breaks a single cross point into two cross points. This problem can be easily solved using the crossing points fusion algorithm [10].

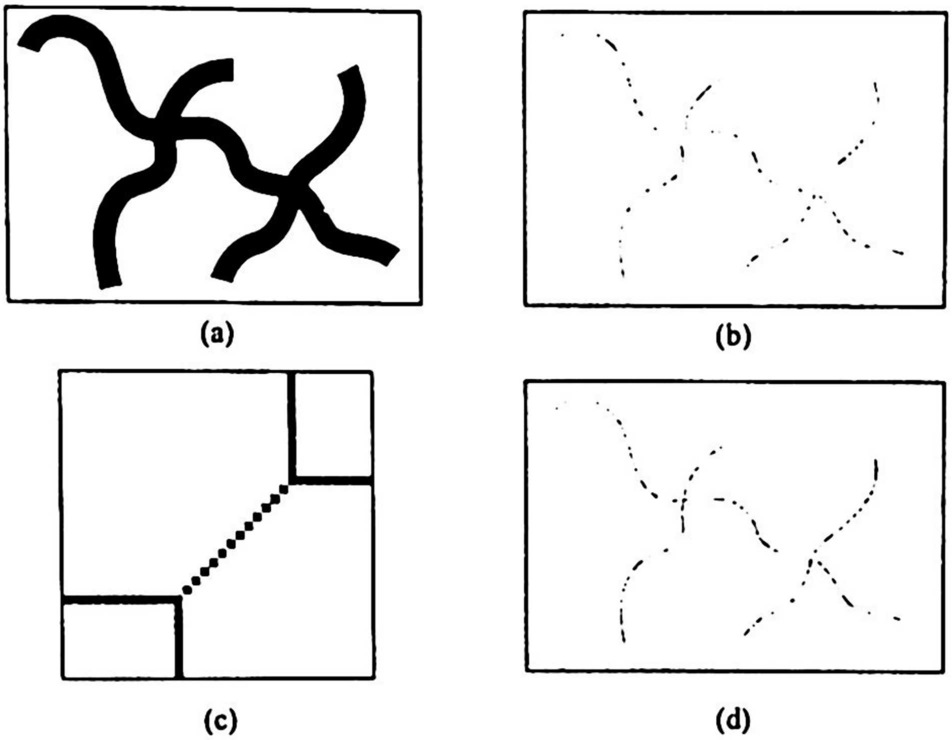


Fig. 4. Description of crossing points fusion. (a) Original image. (b) Initial skeleton (c) Amplified region of the skeleton showing the two crossing points generated by the algorithm. (d) Result of fusion algorithm

Consider the objects presented in the Figure 4(a), its corresponding skeleton that is shown in Figure 4(b), and an amplified section of the image that is shown in Figure 4(c). It is possible to appreciate that we obtain two cross point instead of one. This pathological case can be corrected using this criterion: if two cross points

(CP) are separated be at most $\pm(0.5)W$ pixels[1], applying the Euclidean distance, these points and arcs of the image are joined in one cross point. The result appears in Figure 4(d).

6. *Terminal and cross points counting.* Here, we proceed to count the two variables that allow us calculate the depth of a basin. The algorithm makes a top – down analysis and proceeds to classify object pixel according to Equation 3.

$$\begin{aligned} TP &= \{p \in S \mid N_8(p) = 1\} \\ CP &= \{p \in S \mid N_8(p) \geq 3\} \end{aligned} . \tag{3}$$

Where *TP* denotes a terminal point, and *CP* a cross point. An arc is a k-connected sequence of pixels delimited by two terminal points. Finally, the steps involved in our algorithm are:

a. Find the starting terminal point: First, associate the binary image to a Digital Elevation Model map for each basin pixel. Second, find the longest arc in the image, the length of the arc is the number of pixels contained in it. For the largest arc, the initial point is the terminal point of this arc with higher altitude; the other terminal point is the end terminal point.

b. If *NCP* (number of cross points) is zero, then finish, the depth of the basin is 1.

c. Review each arc contained in the skeleton, starting with the arc at the terminal point to which it belongs.

d. Assign a weight of 1 to all terminal points (*TP*), except to the end terminal point.

e. Find the weight for each cross point using the next rule: In a basin, all cross points are denominated bifurcations. These bifurcations define the intersection of two or more rivers. First, we have to infer the directions of the rivers. It is calculated as follows. If a river acts as sink of the water flow the arc that represents that river is considered as an outgoing arc. On the other hand, if a river delivers water flow to a bifurcation, the corresponding arc is considered as an incoming arc. Then we use altitude values of the arcs to determine if an arc is an incoming or an outgoing arc.

f. In a bifurcation, if all the weights of the incoming arcs are equal, the weight of the outgoing arc is the arithmetic sum of these weights, otherwise, the greatest weight is the value assigned to the outgoing arc. This condition is resumed in Equation 4.

g. Process the remaining arcs, until reaching the end terminal point, the weight of this point is the desired depth.

---

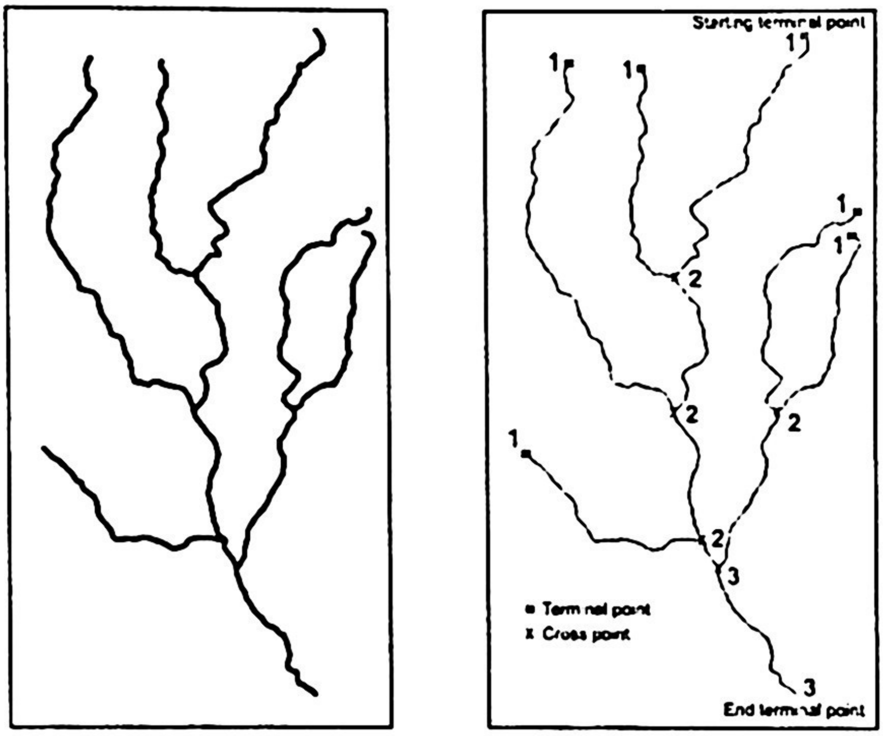
[1] *W* denotes the width in pixels of the object.

$$w(p) = \begin{cases} w(a_1)+1, & \text{if } w(a_1) = w(a_2) = \ldots = w(a_n) \\ Greatest(w(a_1), w(a_2), \ldots, w(a_n)) & \text{otherwise} \end{cases}. \quad (4)$$

This algorithm does not require demonstration due the fact that we use the technique proposed by Horton [4].

4 Tests and Results

In this section, we show the results obtained by applying our algorithm. The application was tested with a bank of 100 raster images. In about 97% of all processed images the results have been satisfactory. Figure 5(a) shows the original image and 5(b) the processed image.



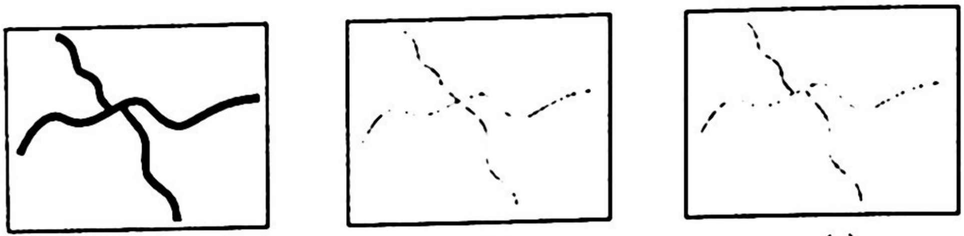
(a) (b)

Fig. 5. Algorithm result. (a) Source image (b) Output image

4.1 Pathological cases

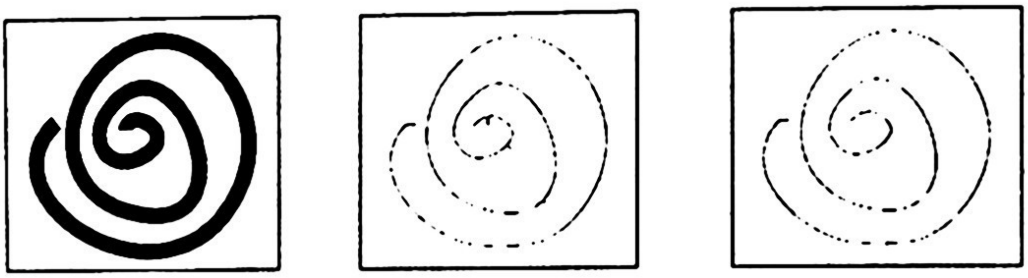
We identified at least two pathological cases, which can produce erroneous results. The first case is when two independent rivers touch in a bifurcation; see Figure 6. The cross point fusion processes the skeleton and proceeds to make a fusion as we mentioned before. However in this case, it is not necessary to carry out this fusion that represents clear algorithm disadvantage. We are working on the algorithm improvement to solve this case by another (semantic-based approach).

The second case is when the parasite branches are presented. We solved this case by applying a morphological operation (opening with 3x3-structured element); see Fig. 7. Obviously a river rarely takes this form, but is useful to show the pathologic case.



(a) (b) (c)

Fig. 6. Pathological case. (a) Source image. (b) 8-connected skeleton. (c) Output of cross point fusion



(a) (b) (c)

Fig. 7. Second pathological case. (a) Original image. (b) 8-connected skeleton without applying morphological operator. (c) 8-connected skeleton in an original image with morphological operation

5 Conclusions

In this paper, we have presented an algorithm that correctly determines the depth of a basin. The main advantages of the presented algorithm are:

- Due to is low complexity it can be applied to real-time process
- It uses only two variables to determine the depth of a basin.
- Use DEM allows us to take into consideration the elevation characteristics as well as flow direction of the basin.

However, one pathological case (section 4.1) remains unsolved in our approach. We believe that it can be solved by adding another technique based on semantic analysis of "river's independence". Our future research will be concerned with this approach.

Acknowledgments

The authors of this paper wish to thank to the Centre for Computing Research (CIC), the General Coordination of Postgraduate Study and Research (CGEPI) and to the National Polytechnic Institute (IPN) for their support. Additionally, the authors wish to thank the reviewers of this work for their pertinent comments.

References

1. A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12(9):855-867, 1990.
2. A. Gruen and H. Li. "Semi – automatic road extraction by dynamic programming". ISPRS Internation Archives of Photgrammetry and Remote Sensing. 30(3/1):324-332, 1994.
3. D. Craig, N. Olaf, and R. Dianne "Semi-Automated Extraction of Rivers from Digital Imagery". GeoInformática, Vol. 6, Number 3 (2002).
4. Horton, R. E.: Drainage basin characteristics. Transactions American Geophysical Union, Vol. 13 (1932) 350-361.
5. J. L. Díaz, C. Yánez, G. Guzmán, "Algoritmos para generar esqueletos k-conectados basados en criterios de conectividad". XII International Congress in Computation. ISBN: 970-36-0098-0.
6. N. Otsu, "A threshold method from gray-level histograms". IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62-66.
7. F. Jiulun, X. Winxin, "Minium error thresholding: A note". Pattern Recognition Letters, 18:705-709, 1997.
8. R. González, E. Woods, "Digital Image Processing". Second Edition, Prentice Hall.
9. http://www.physicalgeography.net/fundamentals/10ab.html.
10. G. Guzmán, "Metodología Extendida para el Conteo de Objetos en Imágenes". Tesis de Maestría, CIC – IPN, 2003.