Available online at http://scik.org
J. Math. Comput. Sci. 6 (2016), No. 6, 1047-1057
ISSN: 1927-5307

EXTRACTION OF THIN NETS IN GREY-LEVEL IMAGES USING COMPUTATIONAL DIFFERENTIAL GEOMETRY

NASSAR H. ABDEL-ALL[1], M. A. SOLIMAN[2], R. A. HUSSIEN[2], AND WADAH M. EL-NINI[2,*]

[1]Department of Mathematics, Faculty of Science and Arts, Qassim University, Oniza, Saudi Arabia

[2]Department of Mathematics, Faculty of Science, Assiut University, Assiut, Egypt

Abstract. In this paper, we describe a new approach for extracting thin nets in $2-D$ grey-level images. The key point is to model thin nets as the crest lines of the image surface. Crest lines are the lines where one of the two principal curvatures is locally extremal. We define these lines using first, second and third derivatives of the image. We compute the image derivatives using recursive filters approximating the Gaussian filter and its derivatives. This paper presents an algorithm to extract thin nets from $2-D$ images and we apply this method to the extraction of roads in satellite images.

Keywords: thin nets; differential geometry; curvatures.

2010 AMS Subject Classification: 53C21.

1. Introduction

In many images, Thin Nets (TN) correspond to important features [12, 13]. For instance, in aerial and medical images, TN are attached respectively to roads and blood vessels. TN are formed by the points where the grey-level is locally extremum in a given direction. This direction is the normal to the curve traced by the TN at this point. Classical edge detection

[*] Corresponding author

Received Aug 3, 2016

algorithms [4, 9, 10, 15, 27] are not able to detect TN. In this paper, we propose a new method to detect TN using differential geometry.

The authors in [1, 2, 3, 4, 6] provided a new approach to extract ridges (as local maximum) and ravines (as local minimum) in images by the gradient, Hessian matrix and its derivatives of this images, and they provided a new approach in [24], a new algorithm in [7] to extract height ridges (generalized local maximum) in 2-D images and [25] provide generalized of this algorithm in n-D Riemannian geometry.

An important point of our approach in this paper is the ability to identify TN by crest lines on the surface defined by the image [22]. We use the definition of crest lines proposed in [19] for $3-D$ volumic images, i.e. the points where the maximum curvature is maximum along the maximum principal curvature direction. In the present case, we come up with different expressions of the surface differential properties, using partial derivatives of the $2-D$ images. The principal curvatures and principal directions of the surface defined by the image are expressed using first and second order partial derivatives of this image. These are the same as those given in reference [22]. We propose a new criterion which uses, in addition to the first and second order partial derivatives, a third partial derivative for characterizing the crest points. We stress that this criterion is different from the one proposed in [19] for $3-D$ volumic images, although it characterizes the same differential property. To compute the partial derivatives, we use an extension to the third order of the recursive filters approximating the Gaussian and its derivatives [11, 18]. We have tested this method on satellite data in which TN correspond to roads. The quality of the results obtained clearly shows the flexibility and the pertinence of our approach.

2. Computing partial derivatives of a 2-D image using the Gaussian filter and its derivatives

Let $I(x,y)$ be a $2-D$ image. We look for the partial derivatives of $I(x,y)$ using the following formula :

$$I_{x^m y^p} = \frac{\partial^n (I(x,y))}{\partial x^m \partial y^p}, \quad n = m + p. \tag{2.1}$$

Here, we use the subscript notation $I_{x^m y^p}$ to indicate for the derivative orders. If $g(x,y) = g_1(x)g_2(y)$ is the impulse response of a smoothing filter, the restored image I_r is equal to $I * g$, where $*$ is the $2-D$ convolution product. Typically, when the properties of the convolution product are used, we obtain :

$$\frac{\partial^n I_r}{\partial x^m \partial y^p} = \frac{\partial^n (I * g)}{\partial x^m \partial y^p} = I * (\frac{\partial^n g}{\partial x^m \partial y^p}) \tag{2.2}$$

where, the impulse response of the filter which computes $I_{x^m y^p}$ can be defined by : $\frac{\partial^n g}{\partial x^m \partial y^p}$. Using the separability property, we use the Gaussian smoothing filter and its derivatives up to the third order described in [20] to compute the derivatives of a $2-D$ image. We use these filters because all the results of the forthcoming algorithms strongly depend on the way for which the partial derivatives are computed. We come up with the following algorithm for computing $\frac{\partial^n I}{\partial x^m \partial y^p}, m+p \leq 3$ [13] :

$$for\,(m,p)\,such\,that\,(m+p) \leq 3\,do \begin{cases} I_{x^m} = I * g_m(x) \\ I_{x^m y^p} = I_{x^m} * g_p(y) \end{cases}$$

where the convolution products are implemented using the recursive filters. In the next section, we estimate first, second and third-order partial derivatives which estimate using this algorithm.

Lemma 2.1. *In digital image processing, convolutional filtering* [16, 23] *plays an important role in many important algorithms in edge detection and related processes.*

3. Using differential properties of the image surface to extract the TN

3.1. TN and crest lines of an image surface

In the grey level images, the Thin Nets (TN) are formed by the points where the luminance is locally extremum in a given direction. This direction is the normal to the curve traced by the TN at this point. Classical edge models [4, 5, 9, 10, 15, 27] are not able to characterize TN. One way to tackle this problem is to define TN, using the differential properties of the image surface $I(x,y)$ which defines a regular surface in the Mong form $(x,y,I(x,y))$. Some work on corner and vertex detection [14, 21] use the same paradigm. We use the definition of crest lines proposed in [20] for $3-D$ volumic images, i.e. the points where the maximal curvature is locally maximal in the corresponding principal direction. These lines can be characterized as the zero-crossings of a coefficient, noted e_m, which they called *extremality*. the vector e_m

is the directional derivative of the maximum principal curvature (DMC) in the corresponding principal direction defined by $e_m = \nabla \kappa_{max} \cdot t_{max}$, where $\cdot$ means the scalar product. We use this expression ($e_m = 0$) in order to extract TN, but our explicit equation of e_m is completely different because it must be obtained from a $3-D$ surface defined by a $2-D$ image.

We can also extract the TN by the zero-crossings of

$DDI = \nabla I(x,y) \cdot t_{max}$, where DDI is the directional derivative of the 2-D grey-level image function $I(x,y)$ in direction t_{max}. The explicit expression of DDI shows that DDI can be computed directly from the first and second derivatives of the image which an interesting aspect of this method. But unfortunately, the quality of the results obtained by this algorithm is not acceptable for various reasons including noise. Moreover conceptually speaking, this method does not stem from the differential geometry properties of the surfaces. This is why we have chosen $e_m = 0$ using the third derivative of the image.

3.2 From partial derivatives to crest lines of an image surface

3.2.1 Computing the directional Derivative of the Maximum principal Curvature (DMC)

Let us consider the surface $S(x,y)$ associated to the grey-level intensity of a $2-D$ image $I(x,y)$ described by the Mong form $S(x,y) = (x,y,I(x,y))^t$.

We define the tangent plane $S_T(x,y) = \{S_x, S_y\}$ of the surface $S(x,y)$ at each point P by :

$$S_x = (\frac{\partial S}{\partial x}) = (1,0,I_x)^t, \quad S_y = (\frac{\partial S}{\partial y}) = (0,1,I_y)^t \tag{3.1}$$

and we have

$$S_{xx} = (0,0,I_{xx})^t, \quad S_{xy} = (0,0,I_{xy})^t, \quad S_{yy} = (0,0,I_{yy})^t \tag{3.2}$$

The first and second fundamental quantities $g_{\alpha\beta}$ and $L_{\alpha\beta}$ respectively, can be used to compute the two principal curvatures and two principal directions at each point P of the surface $S(x,y)$ [8] as the following

$$\begin{gathered} g_{xx} = 1 + I_x^2, \quad g_{xy} = I_x I_y, \quad g_{yy} = 1 + I_y^2, \quad g = Det(g_{\alpha\beta}) = 1 + I_x^2 + I_y^2 \\ g^{xx} = \frac{g_{yy}}{g} = \frac{1+I_y^2}{g}, \quad g^{xy} = \frac{-g_{xy}}{g} = \frac{-I_x I_y}{g}, \quad g^{yy} = \frac{g_{xx}}{g} = \frac{1+I_x^2}{g} \end{gathered} \tag{3.3}$$

$$L_{xx} = \frac{I_{xx}}{\sqrt{g}}, \quad L_{xy} = \frac{I_{xy}}{\sqrt{g}}, \quad L_{yy} = \frac{I_{yy}}{\sqrt{g}}, \quad L = Det(L_{\alpha\beta}) = \frac{1}{g}[I_{xx}I_{yy} - I_{xy}^2] \tag{3.4}$$

where $N = \frac{S_x \wedge S_y}{\sqrt{g}} = \frac{1}{\sqrt{g}}(-I_x, -I_y, 1)$.

By definition, $\kappa_g = \kappa_1 \kappa_2$ is the Gaussian curvature and $\kappa_m = \frac{1}{2}(\kappa_1 + \kappa_2)$ is the mean curvature at each point of the surface $S(x,y)$. Therefore, the expressions of κ_g and κ_m can be given by the following forms:

$$\begin{aligned} \kappa_g &= \frac{L}{g} = \frac{1}{g^2}(I_{xx}I_{yy} - I_{xy}^2) \\ \kappa_m &= \frac{1}{2} g^{\alpha\beta} L_{\alpha\beta} = \frac{1}{2g^{\frac{3}{2}}}(I_{xx} - 2I_xI_{xy}I_y + I_{xx}I_y^2 + I_{yy} + I_{yy}I_x^2) \end{aligned} \tag{3.5}$$

Finally, the principal curvatures κ_1 and κ_2 are the solutions the equation given as follows:

$$K^2 - 2k_mK + k_m = 0$$

$$\kappa_{1,2} = \kappa_m \pm \sqrt{\kappa_m^2 - \kappa_g} \tag{3.6}$$

The explicit expression can be written as (using (3.5)):

$$\begin{aligned} \kappa_{1,2} =&[I_{xx} - 2I_xI_{xy}I_y + I_{xx}I_y^2 + I_{yy} + I_{yy}I_x^2 \pm (I_{xx}^2 + 4I_{xy}^2 - 2I_{xx}I_{yy} + 4I_x^2I_{xy}^2 + \\ &I_x^4I_{yy}^2 - 2I_{xx}I_y^2I_{yy} - 4I_xI_{xx}I_{xy}I_y^3 - 2I_x^2I_{xx}I_{yy} - 4I_xI_{xy}I_yI_{yy} + 2I_{xx}^2I_y^2 + \\ &4I_{xy}^2I_y^2 + I_{xx}^2I_y^4 + I_{yy}^2 + 2I_x^2I_{yy}^2 - 4I_x^3I_{xy}I_yI_{yy} - 4I_xI_{xx}I_{xy}I_y + 2I_x^2I_{xx}I_y^2I_{yy} + \\ &4I_x^2I_{xy}^2I_y^2)^{1/2}]/(1 + I_x^2 + I_y^2)^{3/2} \end{aligned} \tag{3.7}$$

The expressions of κ_1 or κ_2 show that the principal curvatures of a surface can be computed directly from the first and second derivatives of the image. Once κ_1 and κ_2 are computed, the principal directions t_1 and t_2, which are the two eigenvectors of the matrix $(L_{\alpha\beta}\, g^{\alpha\beta})$, may be represented in the basis $\{S_x, S_y\}$ of the tangent space to the image with the two vectors

$$t_i = \frac{1}{g^{3/2}} \begin{pmatrix} I_{xy} + I_y^2 I_{xy} - I_xI_yI_{yy} \\ g^{3/2}\kappa_i - (I_{xx} + I_y^2I_{xx} - I_xI_yI_{xy}) \end{pmatrix}, i = 1, 2.$$

Let $\kappa_{max} = \kappa_1$ be the maximum curvature in absolute value ($|\kappa_1| \geq |\kappa_2|$) and also $t_{max} = t_1$. The gradient of κ_{max} in the associated direction t_{max} , called the directional derivative of the maximum curvature (DMC), is given by :

$$DMC = \nabla\kappa_{max} \cdot t_{max} = \nabla\kappa_{max} \quad t_{max}^{tr} \tag{3.8}$$

where

$$\nabla\kappa_{max} = \begin{pmatrix} \frac{\partial\kappa_{max}}{\partial x} \\ \frac{\partial\kappa_{max}}{\partial y} \end{pmatrix}, t_{max} = \begin{pmatrix} \xi \\ \kappa_{max} - \zeta \end{pmatrix}, \xi = \frac{I_{xy} + I_y^2 I_{xy} - I_x I_y I_{yy}}{g^{3/2}},$$

$$\zeta = \frac{I_{xx} + I_y^2 I_{xx} - I_x I_y I_{xy}}{g^{3/2}} \; and\; tr\; means\; the\; matrix\; transpose.$$

The expression of DMC can also be written as

$$DMC = (\frac{\partial\kappa_{max}}{\partial x})\xi + (\frac{\partial\kappa_{max}}{\partial y})(\kappa_{max} - \zeta) \tag{3.9}$$

The expression of the DMC shows that the directional derivative of the maximum principal curvature at each point of the surface can be computed directly from the first, second and third derivatives of the image. Once the DMC is computed, we extract the points where the DMC=0 (the zero-crossings of the DMC).

3.2.2. Extraction of the zero-crossings of the DMC

Here, we present how to extract the zero-crossings of the computed DMC, in the previous section. Theoretically, the absolute value of the DMC is invariant but its sign depends on the orientation of the vectors t_1 and t_2. That is one reason why Thirion [26] introduced the Gaussian extremality which is an Euclidean invariant but having a different geometrical meaning than crest lines. However, in our case, assuming that (N, t_1, t_2) is always a direct orthogonal frame, where N is the surface unit normal, we can ensure the coherence of the orientation of t_{max}. Therefore, the zero-crossings of the DMC can be characterized, in most cases, as the points where the sign of the DMC changes.

Using the expression 3.9 and separating the flat and umbilical points (the points which $\kappa_1 = \kappa_2 = 0$ and $\kappa_1 = \kappa_2 \neq 0$ respectively) [8], we compute the DMC image. The sign image (S_{dmc}) can be built from the DMC image as follows :

$$\begin{cases} S_{dmc} = 1 \quad \text{if } DMC > 0 \\ S_{dmc} = 0 \quad \text{if } DMC \leq 0 \end{cases} \tag{3.10}$$

The zero-crossings of the DMC are the transitions $0 \uparrow 1$ and $1 \downarrow 0$. We select the zero-crossings of the DMC such that the maximum curvature κ_{max} is greater than a given positive threshold and remove in image obtained the points where κ_{max} is smaller than the other one. This thresholding image allows us to remove the valley points having one principal curvature which is negative with a high absolute value and the other having a small value near zero. Moreover, this thresholding stage yields to hold only the points where the maximum curvature is locally maximum (the zero-crossings of the DMC correspond to both the maximum and the minimum of the maximum curvature).

4. Algorithms

The proposed extraction algorithm of TN can be split as follows :

(1) Computation of the partial derivatives of the image $I(x,y)$ up to 3d order. We estimate these derivatives using a recursive Gaussian filter and its derivatives [11, 20].

(2) Computation of the two principal curvatures κ_1, κ_2, as well as the two principal directions t_1, t_2.

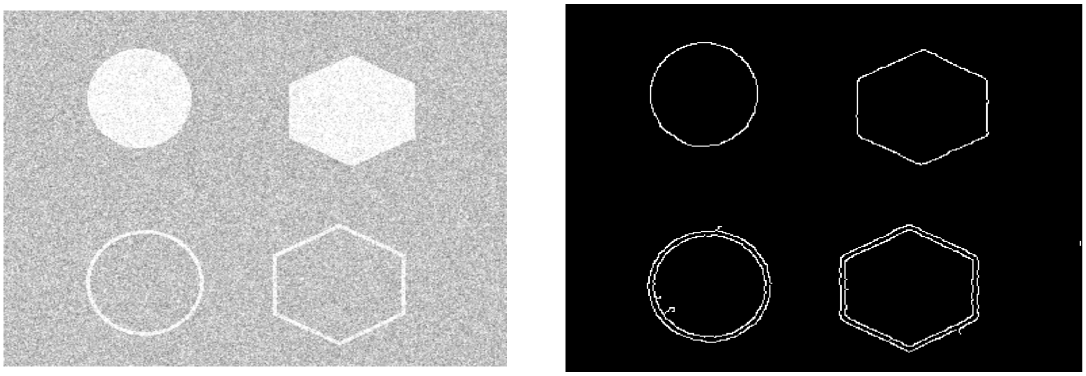
(3) Separating the flat and umbilic points and for the other points:

$$\begin{cases} if\,|\kappa_1| > |\kappa_2| & \text{do } \kappa_{max} = \kappa_1, \quad t_{max} = t_1 \\ else & \text{do } \kappa_{max} = \kappa_2, \quad t_{max} = t_2 \\ end. & \\ DMC = \nabla \kappa_{max} \cdot t_{max} & \end{cases} \tag{3.10}$$

(4) Extraction of the zero-crossings of the DMC.

5. Experimental results

In this section, we discuss some experimental results obtained on synthetic and real data in order to extract the TN. We illustrate the results on synthetic data in Fig.(1) and Fig. (2). In Fig.(1a), we have added a Gaussian noise to picture representing circle and a form hexagonal in grey-level. This is done in order to make complicated noisy data. Fig.(1b), present the results of the edge detection with a Gaussian filter [10] of the original image. This image show that a classic edge detection algorithm is not able to detect TN picture.

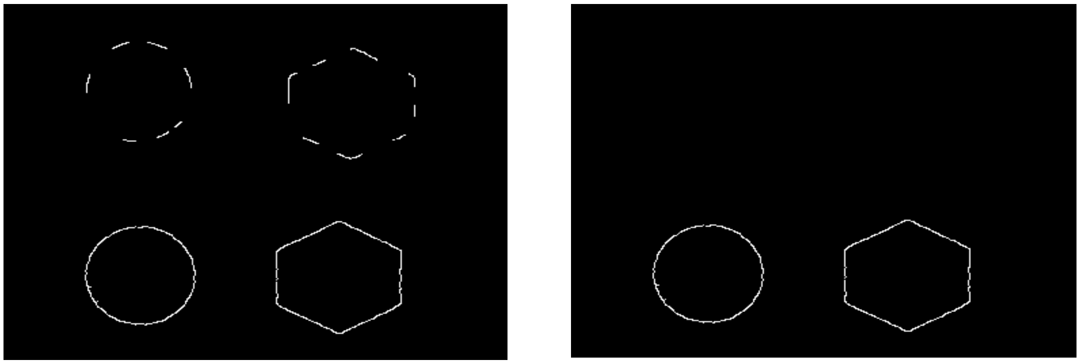


(A) Original image with Gaussian noise.

(B) Edge detection by Gaussian filter.

FIGURE 1. Edge detection

Fig. (2) is the results of our extraction algorithm of TN for two values of the threshold, which we illustrate the extraction of these thin nets.



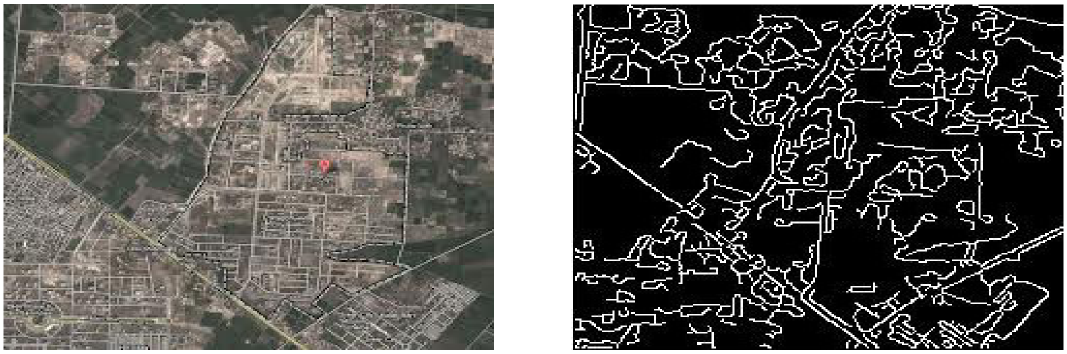
(A) Extraction of TN. Threshold =0.5

(B) Extraction of TN. Threshold =0.8

FIGURE 2. Extraction of TN by our algorithm.

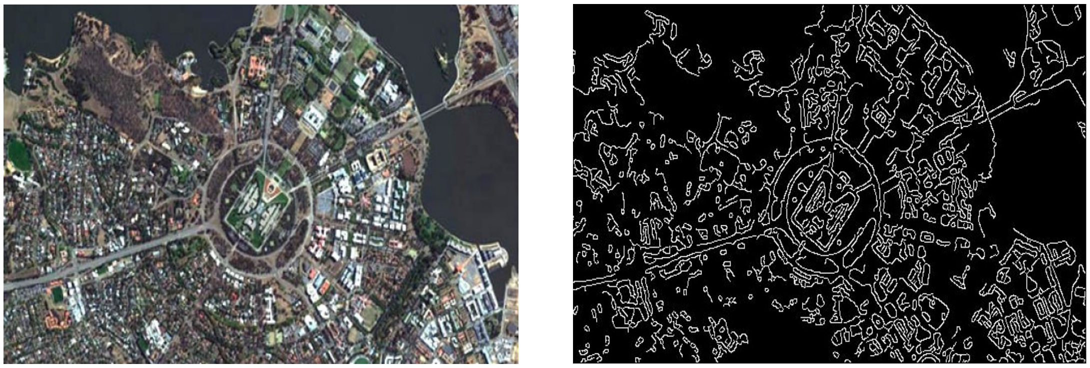
We have also compared in Figs.(2a) and Figs. (2b) the response fidelity of our method using two pictures in which edge detection and TN are presented. The ability to successfully extract the thin net yet exclude the edge detection shows the pertinence and fidelity of the TN extraction algorithm.

Fig.(3) to Fig.(5) present the results of our algorithm on satellite data. For these images, we have extracted the zero-crossings of the DMC image and have thresholded with κ_{max} image.



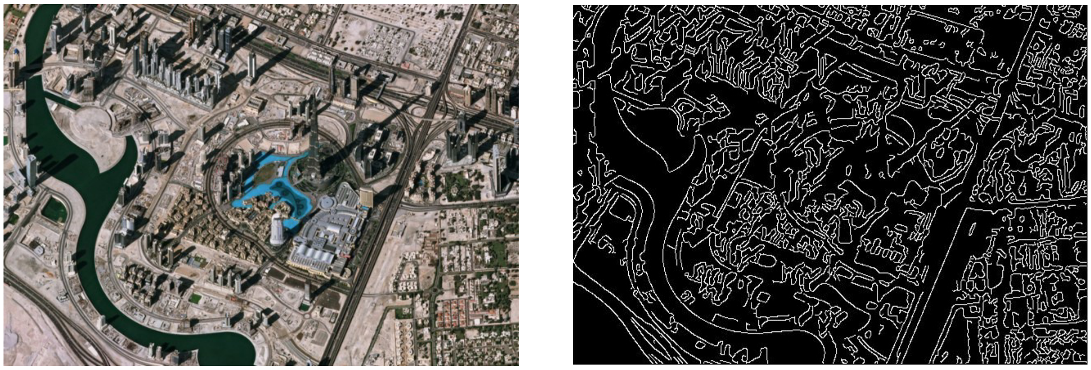
(A) Satellite Image of the roads. (B) Extraction of TN

FIGURE 3. Extraction of TN for Satellite Image



(A) Satellite Image (B) Extraction of TN

FIGURE 4. Extraction of TN by our algorithm.



(A) Image of Dubai City

(B) Extraction of TN

FIGURE 5. Dubai City.

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] Nassar. H. Abdel-All and A. A. Al-moneef, Ridges and singularities on hypersurfaces. *J. of Inst. of maths.& Comp. Sci.,* 21 (2008) 109-116.

[2] Nassar. H. Abdel-All and A. A. Al-moneef, Generalized ridges and ravines on an equiform motion. *Studies in Mathematical sciences,* 4 (2012) 76-85.

[3] Nassar. H. Abdel-All and A. A. Al-moneef, Local study of singularities on an equiform motion. *Studies in Mathmatical sciences,* 5 (2012) 26-36.

[4] Nassar H. Abdel-All, M. A. Soliman, R. A. Hussien and Wadah M. El-Nini, Geometric image edge detection techniques. *Int. J. Math. Comput. Applics. Res.,* 3 (2013) 1-14.

[5] Nassar H. Abdel-All, M. A. Soliman, R. A. Hussien and Wadah M. El-Nini, The circle-point mapping between image space and parameter space, *Int. J. Math. Comput. Applics. Res.,* 3 (2013) 53-62.

[6] Nassar H. Abdel-All, M. A. Soliman, R. A.Hussien and Wadah M. El-Nini, Extract Ridges and Ravines Using Hessian Matrix of 3D Image, *J. Math. Comput. Sci.,* 3 (2013) 1252-1270.

[7] Nassar H. Abdel-All, M. A. Soliman, R. A.Hussein and Wadah M. El-Nini, On an Algorithm for Extracting Height Ridges on 2-D Images, *J. Math. Comput. Sci.,* 6 (2016), No. 1, 86-102

[8] M. P. D. Carmo. Differential Geometry of Curves and Surfaces, *Prentice Hall,* (1976) 153-175.

[9] J. F. Canny, Finding Edges and Lines in Images. *Technical Report 720. MIT, Artificial Intel ligence Laboratory, Cambridge, Massachusetts,* June (1983).

[10] R. Deriche, Optimal edge detection using recursive filtering, *In Proc. First International Conference on Computer Vision, London,* June (1987) 8-12.

[11] R. Deriche, Recursively implementing the Gaussian and its derivatives, *INRIA Research Report,* (1993).

[12] D. Geman and B. Jedynak, Tracking Roads in Satellite Images by Playing Twenty Questions, *INRIA Research Report, August,* (1994).

[13] D. Geman and B. Jedynak, Detection of roads in SPOT satellite images, *Proc. IGRASS 91, Helsinski,* (1991).

[14] G. Giraudon and R. Deriche, On Corner and Vertex Detection, *In Conference on Computer Vision and Pattern Recognition, Hawaii (USA),* June (1991).

[15] R. M. Haralick, Digital step edge from zero-crossing of second directional derivatives, *IEEE Transactions on Pattern Analysis and Machine Intel ligence,* 6 (1984) 58-68.

[16] P. M. Mather, Computer Processing of Remotely-Sensed Images An Introduction Third Edition, *West Sussex. John Wiley, Sons Ltd.,* (2004).

[17] P. Montesinos, H.V. Blanc, Perceptual Grouping of Continuation : Application to Satellite Images, *CARI 94, 2nd African Conference on Reasearch in Conputer Science, Ouagadougou, Burkina Faso,* October (1994).

[18] O. Monga, R. Lengagne, R. Deriche, Extraction of the zero-crossings of the curvature derivative in volumic 3D medical images : a multi-scale approach, *In IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA,* June (1994).

[19] O. Monga, S. Benayoun and D. Faugeras, Using partial derivatives of 3D images to extract typical surface features, *In IEEE Conference on Vision and Pattern Recognition, Urbana Champaign, Illinois,* July (1992).

[20] O. Monga, R. Lengagne, R. Deriche, Crest lines extraction in volumic 3D medical images : a multiscale approach, *In International Conference on Pattern Recognition (ICPR), Jerusalem, Israel,* October (1994).

[21] J. A. Noble. Finding Corners, *In Image and Vision Computing,* 6 (1988) 121-128.

[22] J. Ponce and M. Brady, Towards a surface primal sketch, *In Proceedings, IJCAI,* (1985).

[23] S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, *California Technical Publishing San Diego, California,* (1999).

[24] M. A. Soliman, Nassar H. Abdel-All, R. A. Hussien and Wadah M. El-Nini, Extraction of Ridges on 2-Dimensional Images. *Ciencia e Tecnica Vitivinicola Journal,* 30 (2015) 311-325.

[25] M. A. Soliman, Nassar H. Abdel-All, R. A.Hussein and Wadah M. El-Nini, Generalized Algorithm to the Extraction of Height Ridges in Riemannian Geometry, *Int. J. Manage. Infor. Technol. Eng.,* 4(2) (2016) 23-36.

[26] J. P. Thirion, The Extremal Mesh and Understanding of 3D Surfaces, *INRIA Research Report 2149,* December (1993).

[27] V. Torre and T.A. Poggio, On edge detection. *IEEE Transaction on Pattern Analysis and Machine Intel ligence,* 8(1986) 163-187.