ISSN 2277-3061

3D from 2D for Nano images using image's processing methods

Z. Z. AbouSalem

I . Arabaev Kyrgyz State University

Deeb2222@hotmail.com

Salman bin Abdulaziz University, Community College, Computer Department, KSA

ABSTRACT

The scanning electron microscope (SEM) remains a main tool for semiconductor and polymer physics but TEM and AFM are increasingly used for minimum size features which called nanomaterials. In addition some physical properties such as microhardness, grain boundaries and domain structure are observed from optical and polarizing microscope which gives poor information and consequentially the error probability of discussion will be high.

Thus it is natural to squeeze out every possible bit of resolution in the SEM, optical and polarizing microscopes for the materials under test. In our paper we will tackling t[1]his problem using different image processing techniques to get more clarify and sufficient information.

In the suggested paper we will obtain set of images for prepared samples under different conditions and with different physical properties. These images will be analyzed using the above mentioned technique which starting by converting the prepared sample's images (gray scale or colored images) to data file (*.dat) in two dimensional using programming. The 2D data will convert to 3D data file using FORTRAN programming. All images will subject to the generate filter algorithm for 3D data file. After filtering the 3D data file we can establish histogram, contours and 3D surface to analysis the image. Another technique will be prepared using Visual FORTRAN for steepest descent algorithm (SDA) which gives the vector map for the obtained data. Finally the depth from one single still image will be created and determine using OpenGL library under Visual C++ language, as well as, perform texture mapping.

The quality of filtering depends on the way the data is incorporated into the model. Data should be treated carefully. From our paper we can analysis any part from any image without reanalysis the image, all size of the image as in this paper we take three samples with different size (256 * 256), (400 * 400), (510 * 510), this method decrees the cost of hardware and sample.

Keywords

Image processing; Nanomaterials; converting data; Steepest descent method (SDA).

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 14, No. 2

www.ijctonline.com , editorijctonline@gmail.com

INTRODUCTION

Digital image processing analysis and computer visions have exhibited an impressive growth in the past decade in terms of both theoretical development and applications. They constitute a leading technology in a number of very important areas, for example in digital telecommunication, broadcasting medical imaging, multimedia systems, biology, material sciences, Robotics and manufacturing, intelligent sensing systems, remote sensing, graphic arts and printing.

Spectral enhancement relies on changing the gray scale representation of pixels to give an image with more contrast for interpretation. It applies the same spectral transformation to all pixels with a given gray scale in the image. However, it does not take full advantage of human recognition capabilities even though; it may allow better interpretation of the image by the user. Interpretation of the image includes the use of brightness information, and the identification of features in the image [1] and [2].Spatial enhancement is the mathematical processing of the image pixel data to emphasize spatial relationships. This process defines homogeneous regions based on linear edges. Spatial enhancement techniques use the concept of spatial frequency within the image. Spatial frequency is the manner in which gray-scale values change relative to their neighbors within the image. If there is a slowly varying change in gray scale in the image from one side of the image to the other, the image is said to have a low spatial frequency. If pixel values vary radically for adjacent pixels, the image is said to have a high spatial frequency [2],[3].

In the present research any of the normally dealt with image files with extensions (*.bmp) or tiff are first changed to one with extension (*.dat). Thus, each image is incorporated as a data matrix. Practically, we apply our method on samples in Nanomaterials.

The steepest descent algorithm is an old mathematical tool for numerically finding the minimum value of a function, based on the gradient of that function. Steepest descent uses the gradient function (or the scalar derivative if the function is single-valued) to determine the direction in which a function is increasing or decreasing most rapidly. Each successive iteration of the algorithm moves along this direction for specified step size and the recomputed the gradient to determine the new direction to travel [4].

Digital image processing analysis and computer visions have exhibited an impressive growth in the past decade in terms of both theoretical development and applications. They constitute a leading technology in a number of very important areas, for example in digital telecommunication, broadcasting medical imaging. multimedia systems, biology, material sciences, Robotics and manufacturing, intelligent sensing systems, remote sensing, graphic arts and printing [2], [4] and [5] and [6].

The rest of the paper is organized as follows. In Section 2, the required computations are presented steepest descent algorithm .and contains the description of our algorithm (SDA). the implementation of the algorithm is illustrated in Section 3. Statistical analysis is illustrated in Section 4. The conclusions and further work are given in Section 5.

Steepest descent algorithm (SDA)

The gradient is a vector operator denoted ∇ : $\nabla f = grad(f)$ the gradient is given by:

$$\nabla f(x,y,z) = \frac{1}{h_1}\frac{\partial f}{\partial x}\hat{x} + \frac{1}{h_2}\frac{\partial f}{\partial y}\hat{y} + \frac{1}{h_3}\frac{\partial f}{\partial z}\hat{z} \qquad (1)$$

The direction of ∇f is the orientation in which the directional derivative has the largest value and $\left|\nabla f\right|$ is the value of that directional derivative. Furthermore, if $\nabla f \neq 0$, then the gradient is perpendicular to the level curve through $(x_0, y_0)\ if\ z = f(x,y)$ and perpendicular to the level surface through $(x_0, y_0\ z_0)\ if\ f(x,y,z)=0$.

We can generalized the equation (1) for x_k where $k = 1,2,3,\dots\dots\dots,n$

$$\nabla f(\vec{x}) = \left\{ \frac{\partial f(\vec{x})}{\partial x_1}\ ,\ \frac{\partial f(\vec{x})}{\partial x_2}, \frac{\partial f(\vec{x})}{\partial x_3} + \dots + \frac{\partial f(\vec{x})}{\partial x_n} \right\} \qquad (2)$$

We applying this function on file of data for the piping image of $f(x_1, x_2, x_3)$, recall that the gradient vector in (2) points locally in the direction of great rate of increase of $f(\vec{x})$. Hence $-\nabla f(\vec{x})$ points locally in the direction of greatest decrease $f(\vec{x})$. Start at the point $\vec{p}_0$ and search along the line through $\vec{p}_0$ in the direction $\vec{s}_0 = -\nabla f(\vec{p}_0) / \left\| -\nabla f(\vec{p}_0) \right\|$. you will arrive at a point $\vec{p}_1$, where a local minimum occurs when the point $\vec{x}$ is constrained to lie on the line $\vec{x} = \vec{p}_1 + v\vec{s}_0$. Since partial derivatives are accessible, the minimization process can be executed using either the quadratic or cubic approximation method [3], [6] and [7].

Next we compute $-\nabla f(\vec{p}_1)$ and move in the search direction $\vec{s}_1 = -\nabla f(\vec{p}_1)/\left\|-\nabla f(\vec{p}_1)\right\|$. you will come to $\vec{p}_2$, where a local minimum occurs when $\vec{x}$ is constrained to lie on the line $\vec{x} = \vec{p}_1 + v\vec{s}_1$. Iterative will produce a sequence, $\{\vec{p}_k\}_{k=0}$, of points with the property $f(\vec{x}_0) \succ f(\vec{x}_1) \succ \ldots \succ f(\vec{x}_k) \succ \ldots$ if $\lim_{k\to\infty} \vec{\mathrm{p}}_{\mathrm{k}} = \vec{\mathrm{p}}$ then $\mathrm{f}(\vec{\mathrm{p}})$ will be a local minimum $f(\vec{x})$.

Outline of the steepest descent algorithm (SDA)

Suppose that $\vec{p}_k$ has been obtained.

Step 1: Evaluate the gradient vector $\nabla f(\vec{p}_k)$.

Step 2: Compute the search direction $\vec{s}_k = -\nabla f(\vec{p}_k)/\left\|-\nabla f(\vec{p}_k)\right\|$.

Step 3: Perform a single parameter minimization of $\Phi(v) = f(\vec{p}_k + v\vec{s}_k)$ on the interval [0,b], where b is large. This will produce a value $v = h_{\min}$ where a local minimum for $\Phi(v)$. the relation $\Phi(h_{\min}) = f(\vec{p}_k + h_{\min}\vec{s}_k)$ shows that this is a minimum for $f(\vec{p}_k)$ along the search line $\vec{x} = \vec{p}_k + v\vec{s}_k$.

Step 4: Construct the next point $\vec{p}_{k+1} = \vec{p}_k + h_{\min}\vec{s}_k$.

Step 5: Perform the termination test for minimization, as: Are the function value $f(\vec{p}_k)$ and $f(\vec{p}_{k+1})$ Sufficiently close and the distance $\left\|\vec{p}_{k+1} - \vec{p}_k\right\|$ small enough ?

Step 6: Repeat the process.

Methods of conversion

Figure 1) The following block diagram, (Fig.1), presenting new method for converting images with different format as *.bmp, *.tif, *.jpg, *.jpeg, *.gif *.png, *.pcx …etc to digital numbers represent the intensity of colors in the image.

Using different programming codes and Visual Fortran for making transformation on data files as well as using filters, on these data files for denosing the data. In this method, generally the images are converted to digital matrix and processed it by using filter, for filtering.

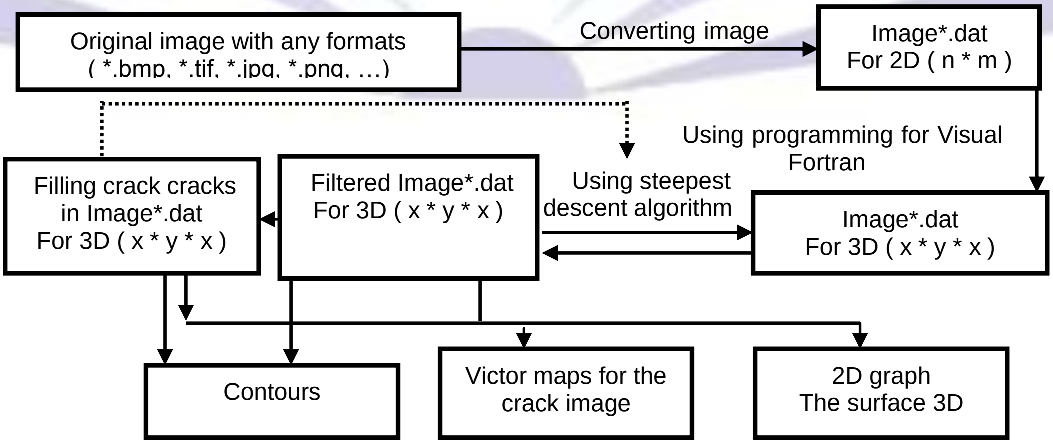

Original image with any formats
(*.bmp, *.tif, *.jpg, *.png, …)
Converting image
Image*.dat
For 2D (n * m)
Using programming for Visual Fortran
Filling crack cracks in Image*.dat
For 3D (x * y * x)
Filtered Image*.dat
For 3D (x * y * x)
Using steepest descent algorithm
Image*.dat
For 3D (x * y * x)
Contours
Victor maps for the crack image
2D graph
The surface 3D

Fig.1

Figure 1) An integrated methodology for the detection and removal of cracks on digitized images is presented in (Fig. 2), using the steepest descent algorithm (SDA), which is an old mathematical tool for numerically finding the minimum value of a function, based on the gradient of that function. Steepest descent uses the gradient function (or the scalar derivative if the function is single valued) to determine the direction in which a function is increasing or decreasing most rapidly, each successive iteration in the algorithm moves along this direction for specified step size and recomputed gradient determine the new direction to travel

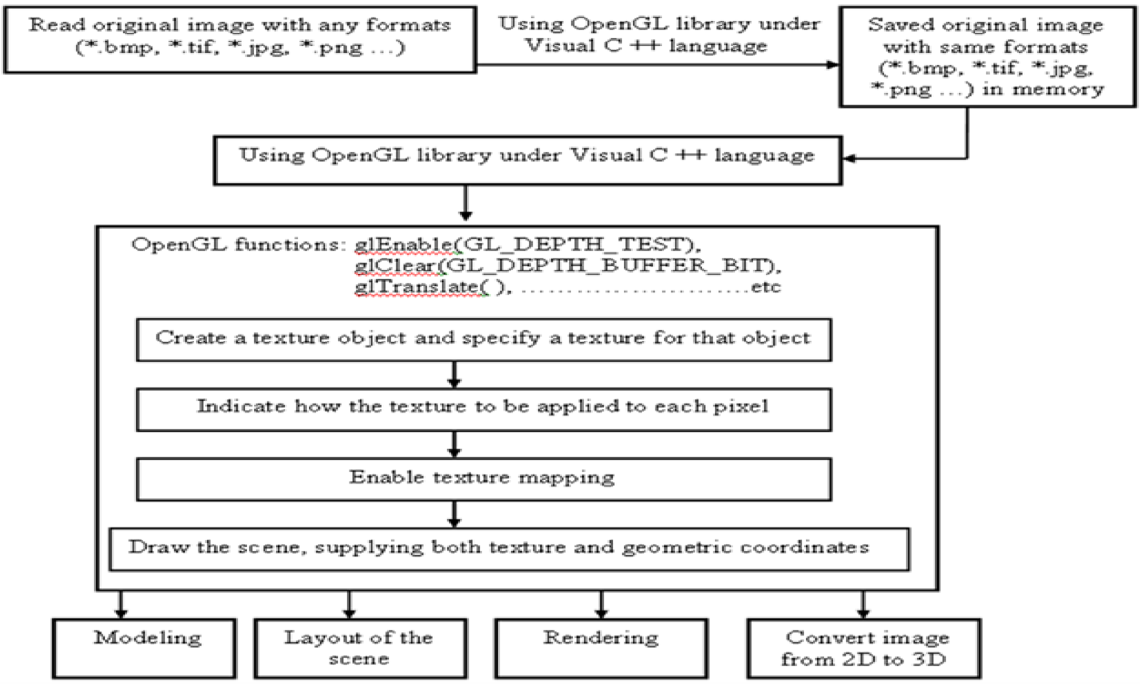

Read original image with any formats (*.bmp, *.tif, *.jpg, *.png ...)
Using OpenGL library under Visual C ++ language
Saved original image with same formats (*.bmp, *.tif, *.jpg, *.png ...) in memory
Using OpenGL library under Visual C ++ language
OpenGL functions: glEnable(GL_DEPTH_TEST), glClear(GL_DEPTH_BUFFER_BIT), glTranslate( ), ...etc
Create a texture object and specify a texture for that object
Indicate how the texture to be applied to each pixel
Enable texture mapping
Draw the scene, supplying both texture and geometric coordinates
Modeling
Layout of the scene
Rendering
Convert image from 2D to 3D

Fig.2

Figure 2) The aim of the method, presented in (Fig. 2), is finding the depth from one single still image using OpenGL library under Visual C++ language, as well as, gives an overview of the steps necessary to perform texture mapping, and also presents a relatively simple texture-mapping program. The normal process of creating 3D computer graphics scene can be sequentially divided into three basic phases: Modeling, Layout of the scene, and Rendering. Typically 3D objects are approximated by polygonal meshes onto which texture images are mapped. Finally the resulting 2D view of this 3D scene is rendered taking into account the viewing position, projection parameters, lighting conditions.

The steps are using programming to convert image from any extension to *.dat file this file is 2D data (n * m).

Getting the mages from any resources as digital camera, scanners, and normal- electronic- microscopic or any Nanomaterials, after that converting by our source code for gray scale or colored images as:

Converts the image x with color map to an intensity image I. ind2gray removes the hue and saturation information while retaining the luminance, B= ind2gray (xx , map);

Converts the matrix X and corresponding color map to RGB (true color) format, RGB = ind2rgb (xx , map)

Step 0: (Input image any formats)

Generate an image rows and columns

Step1: (image file *.dat, 2D data (n * m))

Using Fortran programming for converting 2D data (n * m) to 3D data (x , y , z)

Step2: Repeated Step 0 and Step1 for any part of the image. (or using programming in Visual Fortran for any algorithms)

Step3: Take the 3D data file after filtering and finding the histogram, contours, 3D surface to analysis the image.

after we get the two types of data files 2D data (n * m) , 3D data (x , y , z) we can make compression, transformation …etc.

IMPLEMENTATION

We implemented our algorithm for two-dimensional filter using Visual Fortran and the figures have been plotted using, Origin. The filtering looks of high quality since it seems to recover the original sine wave with the add noise totally removed.

For two-dimensional, any of the normally deal with image files with extensions. (*.bmp) or (*.tiff) are first changed to one with extension. (*.dat). Thus each image is a data matrix. Practically,

Figure 3 and Figure 4 illustrate a) original image b) Histogram for the original image c) The surface 3D d) contours for image for where it looks from the filtered one above that most of the noisy patches have been removed. e) The SDA f) The 2D, the relation between the Distance (pixels) and Gray value for the original image. g) Applying of OpenGL programs on image a second running for the

It should be born in mind that this filtering could be repeated more than one time to obtain the

looked for filtering levels.

THE DESCRIPTION OF THE RESULTS:

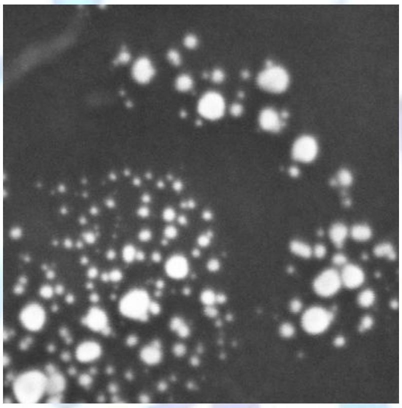
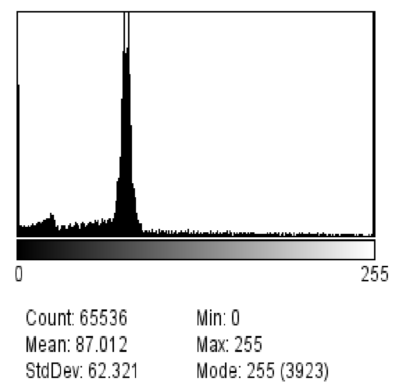


Fig 3.a. Original image (400 * 400 pixels)



Count: 65536 Min: 0
Mean: 87.012 Max: 255
StdDev: 62.321 Mode: 255 (3923)

Fig 3. b. Histogram for the original image

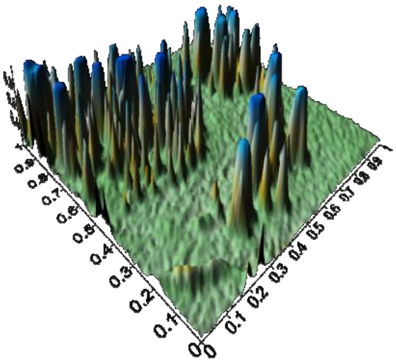


Fig 3. c. The surface 3D for the original image (400 * 400 pixels)

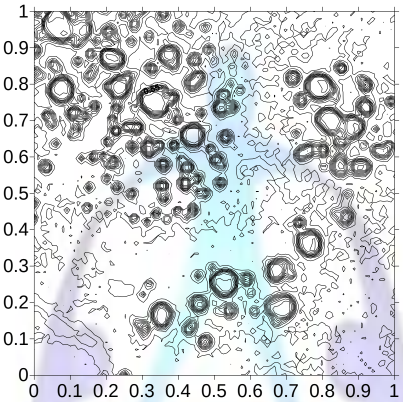


Fig 3. d. The Contour lines for the original image (400 * 400 pixels)

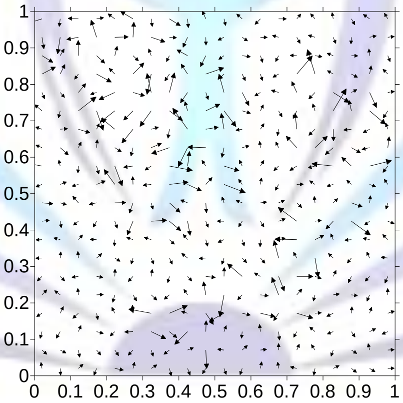


Fig 3. e. The SDA and the direction for the original image (400 * 400 pixels)

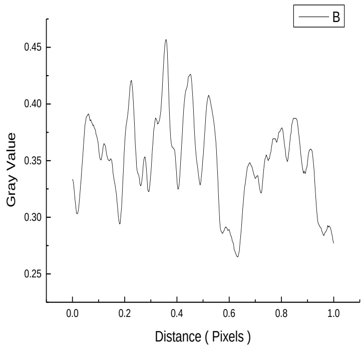


Fig 3. f. The 2D, the relation between the Distance (pixels) and Gray value for the original image (400 * 400 pixels)

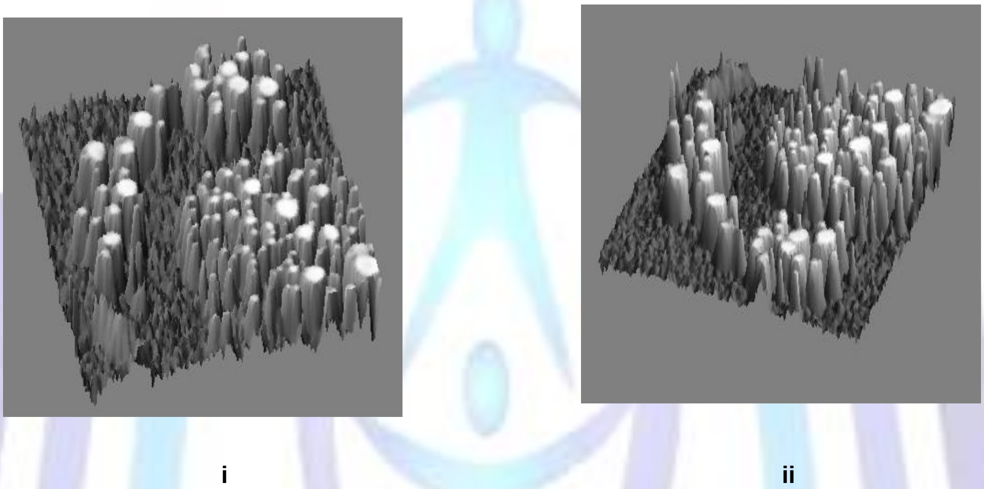


Fig 3. g. Applying of OpenGL programs on image a second running for the programs

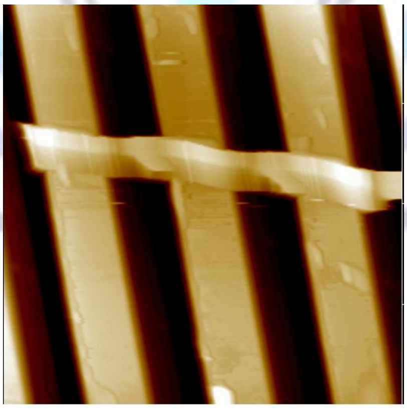


Fig 4.a. Original image (510 * 510 pixels)

ISSN 2277-3061

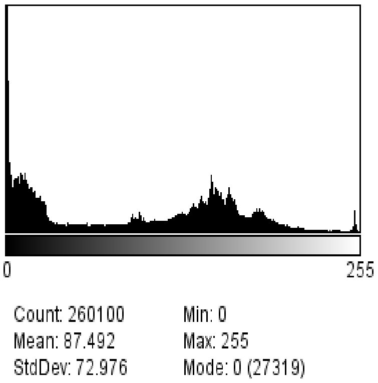


Fig 4. b. Histogram for the original image

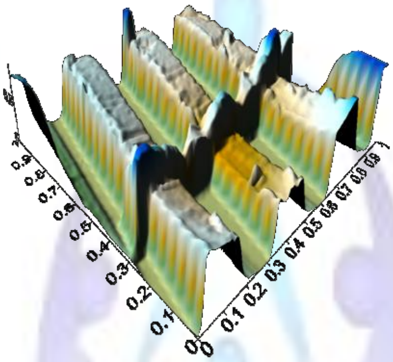


Fig 4. c. The surface 3D for the original image (510 * 510 pixels)

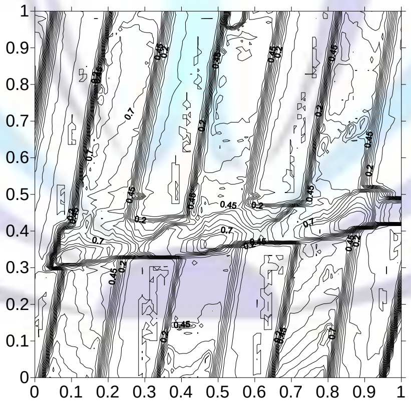


Fig 4. d. The Contour lines for the original image (510 * 510 pixels)

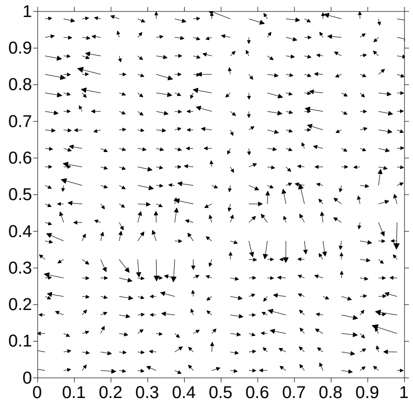


Fig 4. e. The SDA and the direction for the original image (510 * 510 pixels)

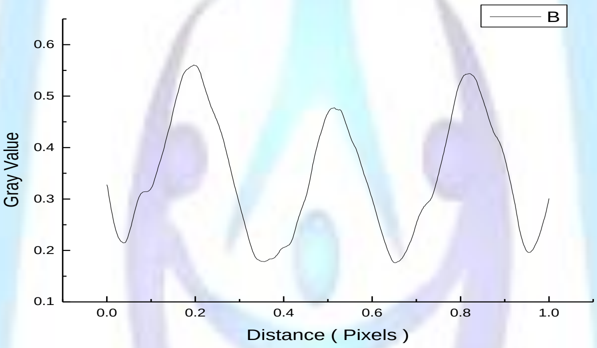
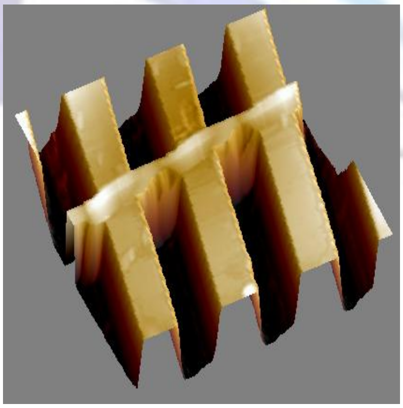
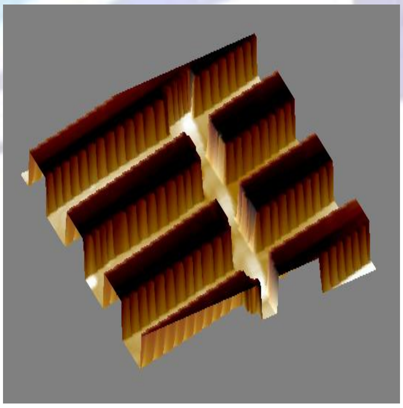


Fig 4. f. The 2D, the relation between the Distance (pixels) and Gray value for the original image (510 * 510 pixels)



i



ii

Fig 4. g. Applying of OpenGL programs on image a second running for the programs

After seeing the up figures we find that the algorithms is the best and reduction the costs for processing, its depends on the programming using Visual Fortran, Surfer.

From the statistical analysis report we get the mean, median, mode, the linear regression equation to the pixels for the matrix for the data files, getting the Sum of Squares (SS) and Mean Square (MS).

After the analysis we get the distribution for the three images as below:

Univariate Statistics for image a:

Planar Regression: Z = AX+BY+C

	X	Y	Z
Mean	0.501	0.504	0.357
Median	0.501	0.511	0.278
Standard Deviation	0.289	0.288	0.179
Variance	0.084	0.083	0.032
Coef. Variation			0.501
Coef. of Skewness			2.192

Fitted Parameters

	X	Y	Z
Parameter Value	- 0.037	0.130	0.310
Standard Error	0.002	0.002	0.001

Univariate Statistics for image b:

Planar Regression: Z = AX+BY+C

	X	Y	Z
Mean	0.499	0.500	0.344
Median	0.499	0.499	0.366
Standard Deviation	0.289	0.289	0.286
Variance	0.083	0.084	0.081
Coef. Variation			0.830
Coef. of Skewness			0.161

Fitted Parameters

	X	Y	Z
Parameter Value	- 0.011	0.022	0.339
Standard Error	0.002	0.002	0.001

CONCLUSION AND FURTHER WORK

In this paper we introduce method for converting the image formats to digital images. And using this method for samples in Nanomaterials. However, this method can be applied to the image patterns in different ways. Giving different quality of image in each graph. We use the transformation method also. This method can be applied in the all Nanomaterials.

We have presented an integrated strategy for crack detection and filling in digitized piping images are detected by using steepest descent algorithm (SDA), Crack interpolation is performed by appropriately modified also by using steepest descent algorithm (SDA). We have a statistical result as mean, median, standard deviation, variance and sum of squares (

SS) and mean square (MS) after detecting and filling the cracks are shown in tables above. Then the steepest descent algorithm is a useful tool for digital image processing because it can be applied iteratively.

REFERENCES

[1] M. A. Ashabrawy, E. E. Elbehadi. " 3-D from 2-D Using Warping Transformations" , International Journal of Computers & Technology, .(01.05) , 2014.

[2] I. A. Ismail, El-Sayed M. El-Horbaty, M. A. Ashabrawy, "Autoregressive Filter Algorithm for Sifting Digital Images" , I. J. Comput. Appl. 9(4): 200-204 (2002).

[3] Mohri, M., "Finite-state transducers in Language and speech processing" Computational Linguistics, 23 {2) (1997) . 269-311

[4] Shashua, A. and Navab, N., "Relative affine structure: canonical model for 3D from 2D geometry and application", IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(9) (1996)873-883. [5] Sid-Ahmed, M. A., "Image processing: theory, algorithms, and architectures, New York: Mc Graw Hill, l995.

[6] Bellanger, M. Digital Processing of Signals, Wiley, New York, 1984.

[7] Castleman, K.R., "Digital image processing", Englewood Cliffs. NJ; Prentice Hall, l995.

Author biography with Photo

Zib Ziab Abousalem .born on 10th July 1968. He did his B Tech in Computer Science from Salman bin Abdulaziz University, Community College, Computer Department, KSA