International Research Journal of Applied and Basic Sciences
© 2014 Available online at www.irjabs.com
ISSN 2251-838X / Vol, 8 (10): 1510-1515
Science Explorer Publications

Removing Noise in Texture using Opening Optimization in Grayscale Images

Ehsan Sadeghipour[1], Nasrolla Sahragard[2], Zeinab bahmanzadeh[3]

1.Sama technical and vocational training college, Islamic Azad University, Bandar Abbas Branch, Bandar Abbas, Iran, ehsan61.ai@gmail.com
2.University of Hormozgan, Assistant Professor of Faculty of Power and Computer Engineering, sahragard@yahoo.com
3.Sama technical and vocational training college, Islamic Azad University, Bandar Abbas Branch, Bandar Abbas, Iran, z.bahmanzadeh@gmail.com

Corresponding Authoremail: ehsan61.ai@gmail.com

ABSTRACT: In opening applications, which is one of the basic actions in morphology, some unwanted appendices remain in images. This problem can be solved using a grayscale constituent, which is similar to the objects of a target image. Since texture images are made of small and similar elements, a grayscale constituent can be estimated for the objects in it. Estimating method optimizes a grayscale element in "*Primitive, Grain, and Point Configuration (PGPC)*" textures. After estimating shape of a texture, the estimation can be used as a constituent to remove noise in the texture. Such noise removing method is often superior to the classical ones.
***Keywords*:** morphological operation; noise removing; texture analysis, texture element estimation

INTRODUCTION

'Morphology' is one of the major applied fields in image processing. An wide range of non-linear processes can be performed in this field. 'Opening' is one of the major actions in morphology, which is used in many applications including noise removing. In morphology, we use a constituent, which is the same as a sliding window in image processing. Determining the shape of a sliding window is the major problem in morphology applications. In some applications, two images –one corrupted and one ideal - are prepared. Using one of the optimization algorithms - such as simulated annealing (SA), genetic algorithm (GA), etc - it is possible to choose a constituent to minimize the differences between the ideal image and the obtained one. However, such methods are not so common and a constituent is usually considered using the known shapes such as a line or a square. It is very difficult to determine the amount of grayscale in a constituent for grayscale images. Research in this field is under way. In case a constituent with an inappropriate shape is used while using opening in grayscale images, some appendices will be formed in the image and if we use a binary element for this purpose, the image will be spotty. These problems will be solved by using a constituent that is similar to the objects of a target image. However, the target image is generally formed of many objects and it is impossible to consider a constituent for each object. Thus, we limit ourselves to texture images consisting similar objects.

MORPHOLOGY OPERATION

A morphological conversion is specified by a set called *A* and a set of smaller points called *B*. The basic morphological operation includes *expansion* and *erosion* [4]. Definitions of binary expansion and erosion are respectively as follows:

$$A \oplus B = \{z: z = r + s \ \forall\, r \in A \ and \ s \in B\} \qquad (1)$$

$$A \ominus B = \{z: z + s \in A \ \ \forall \ \ s \in B\} \qquad (2)$$

Expansion and erosion actions are dual to each other. Other two major actions in morphology processing are opening and closing, which are defined as follows:

$$A \circ B = (A \ominus B) \oplus B \qquad (3)$$

$$A \cdot B = (A \oplus B) \ominus B \qquad (4)$$

Opening action creates a version of an image with fewer details. This conversion makes surroundings smoothers and breaks thin strips. Applying basic actions of expansion and erosion to grayscale images is defined in another form. The definitions are respectively as follows:

$$[f \oplus b](x,y) = \underbrace{max}_{(s,t)\in b} \{f(x-s, y-t) + b(s,t)\} \qquad (5)$$

$$[f \ominus b](x,y) = \underbrace{min}_{(s,t)\in b} \{f(x+s, y+t) - b(s,t)\} \qquad (6)$$

It binary modes, *opening* action removes all white spots smaller than the available constituents do. In grayscale images, it removes the points, which are brighter than their surroundings and keeps the remaining parts of the image. As noise has some features that are removed while opening, opening conversion can be used for removing noise. Noise removing operation is performed in a specific group of textures called *PGPC* [2]. They are those types of textures that include the elements, which is much smaller than the main image and they are placed beside each other regularly and/or irregularly. Figure 1 shows a sample of the textures.

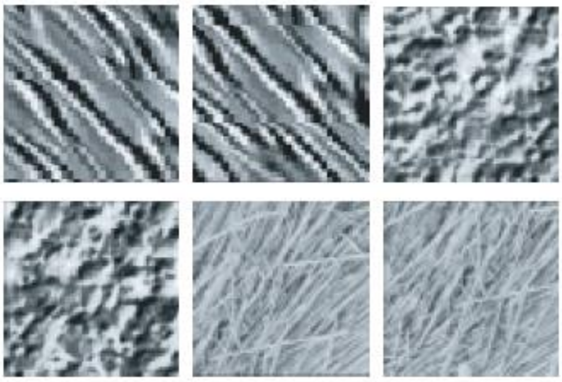

Figure 1: A sample of PGPC texture selected for removing noise

PATTERN SPECTRUM

Pattern Spectrum is a morphologic method to obtain size of objects of an image. An *n*-size pattern spectrum for a certain image and a constituent is defined as the difference between an image that opened using a constituent enlarged as *n*-size symmetrically and an image opened using a constituent enlarged as *n+1* size symmetrically [5].

$$nB = \underbrace{B \oplus B \ldots \oplus B}_{n-1\ additions}$$

$$0B=\{0\}. \qquad (7)$$

Relation 7 stated how a constituent could be extended symmetrically. Based on this, the *n*-size pattern spectrum for *X* image and *B* constituent is as follows:

$$PS(X,B,n) = \sum_{x\in whole\ image}\{X_{nB}(x) - X_{(n+1)B}(x)\} \qquad (8)$$

As opening action removes all the parts smaller than a constituent, the difference of the opened image as large as *n* and *n+1* exactly includes the *n*-size sections. *Normal pattern spectrum* is usually used in our applications, which is defined as the ratio of an n-size image pattern spectrum to total value of the image's pixels that is called Size Density [3].

$$F(X,B,n) = \frac{PS(X,B,n)}{\sum_{x\in wholeimage} X(x)} \qquad (9)$$

TEXTURE ELEMENT ESTIMATION METHOD

[1], [2], and [3] present a new method for texture analysis. In this method, a texture is defined by estimating its element and it is mainly aimed at defining a texture. Our method for texture element estimation is defined by an example. Figure 2 shows a natural texture formed by similar and line-like elements. Most of the lines were extended from top right corner to bottom left side. The figure also shows size density for 4 directions and shape of the constituent is as a line in each of the directions. As shown in the figure, value of size density function has an even form in the direction, which is similar to the texture elements, i.e. top right side to bottom left. This example shows that if shape of a constituent is close to the one of a texture element, density function will create an even amount. Mean and variance of size density function is defined as follows:

$$E[F(X,B,n);l] = \frac{\sum_{n=0}^{l} F(X,B,n)}{l+1} \qquad (10)$$

$$V[F(X,B,n);l] = \frac{\sum_{n=0}^{l}\{F(X,B,n) - E[F(X,B,n);l]\}^2}{l+1}$$

In Relation 10, *l* value is related to the maximum limit that we enlarge a constituent. In accord with the described example, this idea is directly found that in case of having a texture, the constituent that is able to provide an estimation for the shape of texture element will be the constituent whose related size density function has the minimum variance.

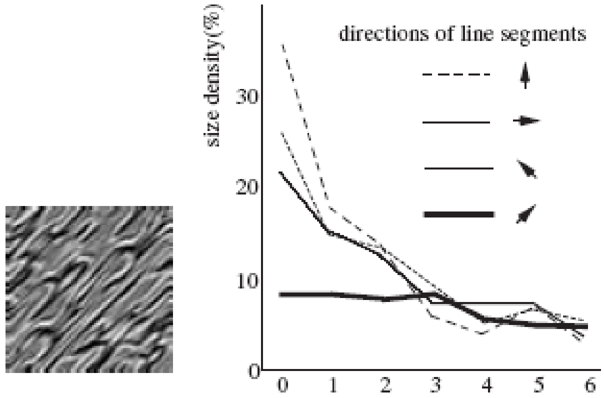

Figure 2: A sample image of texture and value of size density in 4 directions

Therefore, it is possible to start from a primitive constituent and evaluate variance of size density function by applying some changes in the constituent and/or changing grayscale of its pixels and obtain the constituents with minimum variance value. This can be done using an optimization method, e.g. simulated annealing. A proper estimation for a primitive constituent, like the above example, may be like a line in 4 directions and our primitive estimation is the shape of a line in a 5*5 window that creates the minimum variance. This operation is performed for a binary and grayscale image as follows:

1- After obtaining the primitive estimation of the constituent, values of size density function and variance are calculated for *k8*.
2- One of the pixels available in a 5*5 window is changed. The change may be as grayscale or constituent size (Figure 3).
3- Values of size density and variance are changed. If the variance was small, the change is accepted and fixed. If the variance was bigger, the change is accepted with a probability to prevent encountering local conditions. Value of the probability is high at the beginning and it becomes lower in the following iterations. (SA).
4- Step 2 and 3 are repeated as long as no change leads to improving variance of size density function and it would not be possible to change it.

EXPERIMENTAL RESULTS

ER evaluation function is used in our experiments, which is equal to variance of size density function. *l* value is assumed equal to 3, i.e. maximum value of enlargement of a constituent was assumed to be 3 times bigger. The function, which is assumed as acceptance probability of a variable, is as follows:

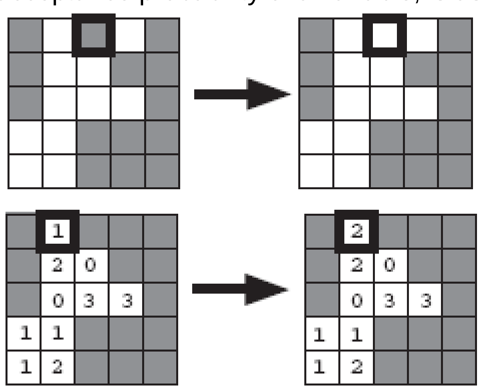

Figure 3: Method of changing limits of constituents and grayscale value

$$P(\Delta ER) = \begin{cases} 1 & if\ \Delta ER < 0 \\ \frac{1}{1+exp\left(\frac{\Delta ER}{T_i}\right)} & if\ \Delta ER \geq 0 \end{cases} \qquad (11)$$

In Relation 11, ΔER value is the difference of *ER* or the same variance before and after applying changes. *Ti*, which is called temperature, is defined in terms of a function of number of iterations as follows:

$$\begin{cases} T_0 = 10^5 \\ T_{i+1} = 0.98 T_i \end{cases} \qquad (12)$$

In Formula 12, *i* value is in fact number of iterations of steps 2 and 3 in the algorithm. If this approach is used, the unfavorable constituents will be accepted at the beginning in order not to encounter with local conditions; however, acceptance probability of the unfavorable elements is reduced gradually. Number of iterations in this experiment was assumed as 1000 times. Several textures are observed as binary in Figure 4. *A', B',* and *C'* are related to other parts of A, B, and C textures. This figure also shows estimations obtained from the texture constituent using the stated algorithm. As it is noticed, this algorithm showed shapes of constituents while our images are binary and it created similar shapes under similar conditions.

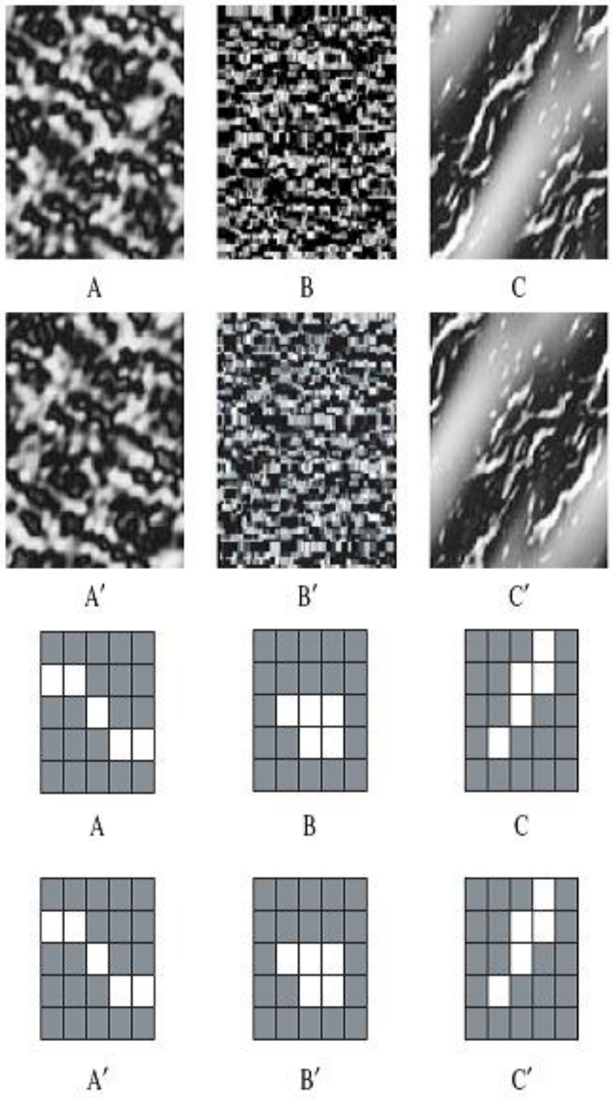

Figure 4: A multi-texture image as binary resulting from texture element estimation using the proposed method

It should be stated on grayscale images - that is mainly focused on here - that if it is intended to estimate texture element in such images using the abovementioned method and the morphology-related formulas in grayscale images, we reach the elements whose entire elements are zero within the constituent limit. Therefore, size and shape of a constituent and its grayscale values should be estimated in two stages. The earlier steps of optimization algorithm are performed in two steps. In the first step, similar to binary images mode, shape of a constituent is estimated. In the second step, the grayscales of the same shape is obtained with respect to the shape of the earlier step. Figure 5 shows the estimations of these two steps for the grayscale texture.

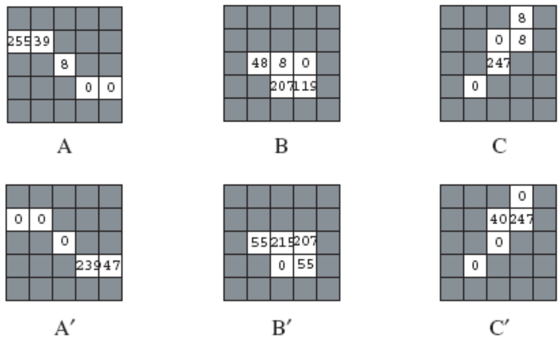

Figure 5: Shape and values of the relevant textures using the algorithm described in this section

As shown in Figure 5, some estimations are obtained with many zero values and large grayscales values. These are unreal estimations for texture element, as the images of two parts of a texture are different. The texture shapes also prove that the estimations are unreal. Formation of such estimations is due to the fact that *opening* in grayscale images using a constituent with a big peak value and small values around is also the same for all objects. It reduces variance of size density function, even makes it zero, and forms an unreal estimation of a constituent for us. Therefore, another evaluation function is used in the second step of optimization that includes optimization of grayscale values of constituents.

$$ER = \frac{F(X,B,0)}{F(X,B,1)-F(X,B,0)} \qquad (13)$$

Here is our final estimation of the elements of different textures as per the evaluation function.(Figure 6)

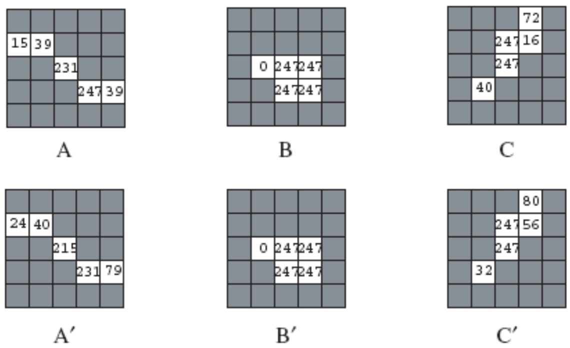

Figure 6: The shape and values obtained for the relevant textures using the new evaluation function discussed for optimizing grayscale values

NOISE REMOVING IN TEXTURE

After obtaining an estimation for the element of a texture, it can be considered as a constituent and the spots brighter than the local background, which are mostly noises, can be removed using opening conversion. Figure 7 shows it. Section *a* shows the relevant texture, Section *b* shows the estimation obtained from texture element, and Section *c* shows noise texture. Values between 0 and 255 were added randomly to 1000 pixels of the image. Section *d* shows the image obtaining from applying *opening* with a constituent obtaining from estimating texture element. Section *e* shows the same action with a 5*5 binary constituent, and Section *f* shows the result using a 3*3 constituent. As it is noticed, *MSE* value in Figure 7, which is the pixel-to-pixel difference of the image obtained in each step and the main figure, is minimum for Section *d*, which confirms the theory of using a constituent as a texture element for removing noise and its effect. However, it is not the case for all textures. Figure 8 shows that other methods obtained lower *MSE* values. It seems that our estimation of the texture element in these figures was not correct.

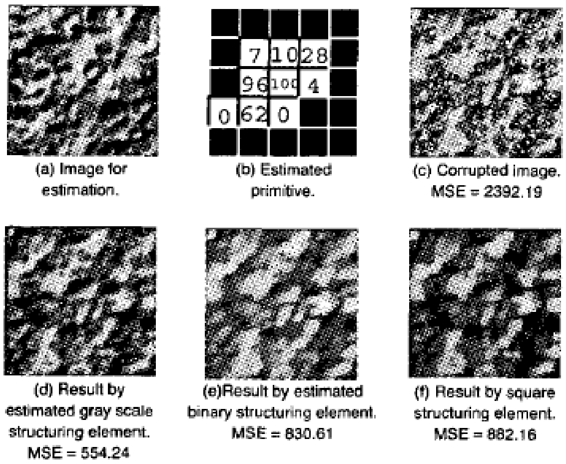

Figure 7: The texture image in which the proposed method – as compared with other methods - could remove noises well.

CONCLUSION

This paper presented a method for estimating texture element. This method was based on morphology process of images and one of the actions in this field, i.e. *opening*. This estimation was presented for specific textures called *PGPC*, which are the textures with the same elements and forms. After obtaining these estimations for a texture, they can be used as a constituent in an *opening* conversion. It can also be used for removing noises. This provides favorable results for some textures. However, it does not provide acceptable results for other textures; one reason might be inaccuracy of primitive estimation.

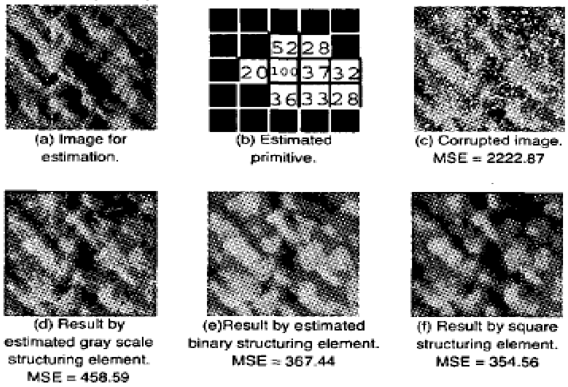

Figure 8: The texture image in which the proposed method – as compared with other methods – could not remove noises well.

REFERENCES

A. Asano, Y. Kobayashi, C. Muraki, and M. Muneyasu, "Optimization of gray scale morphological opening for noise removal in texture images," 47th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS2004), Hiroshima, Japan, 1, 313-316 (2004). T. Sun and Y. Neuvo, “Detail-preserving median based filters in image processing”, Pattern Recognition Lett ,Vol. 15,pp.:341–7, 1994.

A. Asano, "Morphological texture analysis using optimization of structuring elements," 11th Workshop "Theoretical Foundations of Computer Vision" Geometry, Morphology, and Computational Imaging, Dagstuhl-Seminar 02151,Wadern, Germany (2002). W. Luo, “A new impulse detector based on order statistics”, Int. J. Electron. Commun. (AEU) , Vol. 60, pp. 462-466,2006.

A. Asano, T. Ohkubo, M. Muneyasu, and T. Hinamoto, "Primitive and Point Configuration Texture Model and Primitive Estimation using Mathematical Morphology," 13th Scandinavian Conference on Image Analysis, Göteborg, Sweden; Springer LNCS 2749, 178-185 (2003). S. Hsia,“A fast efficient restoration algorithm for high-noise image filtering with adaptive approach”, J. Vis. Commun. Image, Vol. 16, pp. 379-392,2005.

A. Asano, M. Miyagawa, and M. Fujio, "Texture Modelling by Optimal Gray Scale Structuring Elements using Morphological Pattern Spectrum," 15th International Conference on Pattern Recognition, Barcelona, Spain, 3, 479-482 (2000).

S.Yashima,1 A. Asano,2 A.Taguchi1” modeling of mono-feature texture images using the structuring element of morphology filters” Electronics and Communications in Japan, Part 3, Vol. 88, No. 2, 2005