Using Threshold Methods
to Segment Highly Images beans without overlapping

Saba Q. H. Al-Hashimy
Technical Institute in Mosul, Mosul - Iraq

Abstract

This paper presents a proposed method for segmenting images of beans into separated objects without overlapping, based on the area of ideal grain in (perfect) beans by thresholding methods. The global thrshoding for the entire image is used at first. Then local thresholding was used to segment into sub-images, that contain overlapping between grains. Segmentation operations for the overlapping objects into separated objects without any overlapping depend on the minimum value of the histogram at first in sub-image, by repeating local thresholding until getting a value (valley between two peaks) to split two objects. Each object is compared with the ideal grain to put it in the result image.

The proposed method is compared with the technique adopted in Matlab environment. By computing the average ratio of erosion for overlapping edges objects after separating in two methods, the proposed algorithm is 5% better than the traditional method of 19% that means increase the accuracy by 14%, additional clarity image (shape of grain) and the given number of objects is (44), which is exactly too.

Received: 8/1/2014 Accepted: 6/5/2014

استخدام طرق العتبة

لتقطيع صور حبوب الفاصولياء الكثيفة بدون تداخل

صبا قاسـم حسـن الهاشمي

المعهد التقني بالموصل، موصل – العراق

المستخلص

يقدم هذا البحث طريقة مقترحة لتقطيع صورة حبوب الفاصولياء الى حبوب منفصلة من دون اي تداخل، اعتمادا على مساحة الحبة المثالية الموجودة بالصورة باستخدام طرق العتبة. في البداية استخدمت طريقة العتبة العامة التي تطبق على الصورة كاملة وبعدها اعتمدت طريقة العتبة الخاصة التي استخدمت لتقطيع الصور الصغيرة التي تحتوي على الحبات المتداخلة الى حبات منفصلة بدون اي تداخل. إن عملية تقطيع الحبات المتداخلة تعتمد على اقل قيمة للـhistogram كبداية للصورة الصغيرة. بتكرار عملية العتبة الخاصة الى ان نحصل على اقل قيمة (منطقة منخفضة بين مرتفعين) لفصل حبتين عن بعضهما. كل حبة يتم مقارنتها بالحبة المثالية (perfect) ووضعهما بالصورة الناتجة.

بعدها أجريت مقارنة الطريقة المقترحة بالتقنية الموجودة في نظام الـ Matlab وتم حساب معدل نسبة تآكل حافات الحبات المتداخلة بعد الفصل بالطريقتين و كانت الطريقة المقترحة (٥٪) كونها افضل من الطريقة التقليدية (١٩٪)، بنسبة تحسين زادت على (١٤٪) اضافة الى وضوح معالم الصورة (شكل الحبوب)، وكذلك اعطت الطريقة عدد الحبوب بالصورة إذحيث كان (٤٤) حبة بشكل دقيق ايضاً.

1. Introduction:

The problem of image segmentation has been known and addressed for the last 30 years. Because it plays an important role in image, it is important in image processing to select an adequate threshold of gray level for extracting objects from their background. A variety of techniques have been proposed in this regard. In an ideal case, the histogram has a deep and sharp valley between two peaks representing objects and background respectively, so that the threshold can be chosen at the bottom of this valley [1]. However, for most real images, it is often difficult to detect the valley bottom precisely, especially in such cases as when the valley is flat and broad, imbued with noise or when the two peaks are extremely unequal in height, often producing no traceable valley. There have been some techniques proposed in order to overcome these difficulties [2].

Many previous methods have been proposed and applied on blood cells, rice, and fish images, to Otsu's and the proposed methods [3]. Several thresholding techniques have been developed. Some of them are based on histogram; others based on local properties, such as local mean value and standard deviation, or local gradient. The most intuitive approach is global thresholding [4]. When only threshold is selected for entire, based on the image histogram, thresholding is called ***global.*** If the threshold depends on local properties of some image region, for example local average gray value, thresholding is called ***local***. if the local thresholds are selected independently for each pixel (or groups of pixels), thresholding is called **dynamic or adaptive** [5].

The Canny edge detection and Otsu thresholding are used with a variety of color images resulting from these algorithms binary images [2]. In medical application color segmentation is used in color image by K-means clustering method for detection of malaria parasites [6]. Otsu method for gray level segmentation and Median filter hybrid method are used to remove noise without blurring edges [8].

2. Theoretical Aspects

2.1 Pre-processing

Image pre-processing is a necessary step to improve the image quality. In general, the pre-processing can be divided into: denoising, enhancement of structure, and enhancement of contrast. The methods of denoising refer to mean filters, Laplacian filters and Gaussian filters, the methods of enhancing the edge of image structure include un sharpening and wave late transform and the method of enhancing image contrast can be histogram equalization. The Pre-processing refers to the enhancement the intensity and contrast manipulation, noise reduction, background, removal, edges sharpening, filtering, etc. [8].

2.2 Global Thresholding

Global thresholding is one of the widely used methods for image segmentation. It is useful in discriminating the foreground from the background [9].

Global thresholding is oldest method that is based on result value from Otsu's method which utilizes discriminate analysis. Every possible threshold value evaluates the benefit of this value if used as threshold. This evaluation uses either the heterogeneity of both classes or homogeneity of every class [10].

By selecting an adequate threshold value (T), the gray level image can be converted in to a binary image. A binary image should contain all of the essential information about the position and shape of the objects of foreground. The advantage of obtaining a first binary image is to reduce the complexity of the data and simplify the process of recognition and classification. The most common way to convert a gray level image to a binary image is select a single threshold value (T). All the gray level values below this (T) will be black (0), and those above (T) will be white (1). The segmentation problem becomes one of selecting the proper value for the threshold (T). A frequent method used to select (T) is analyzing the histograms of the images type that are to be segmented. The ideal case occurs when the histogram presents only two dominant modes and clear valley (biomdal) Fig. 1. In this case the value of (T) is selected as the valley point between the two modes. In real applications histograms are more complex, with many peaks and unclear valleys and it is not always easy to select the value of (T) [9].

In general, the threshold should be located at a deep valley of the histogram. Especially for a well-defined image, its histogram has a deep valley between two peaks. Therefore, the optimum threshold value can be found in the valley region [11]. The thresholded image values are: [5].

$$g(x,y) = \begin{Bmatrix} 1 & if\ (x,y) > T \\ 0 & if\ (x,y) \le T \end{Bmatrix} \quad \text{.....(1)}$$

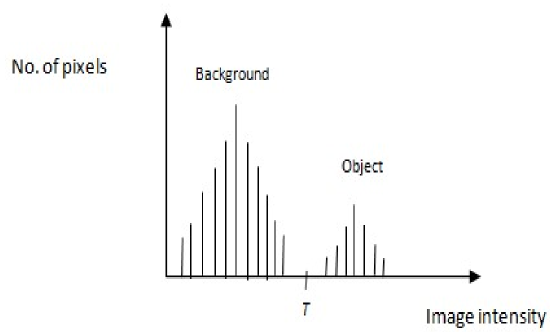


Fig. (1)
An example of bimodal histogram with selected threshold T

2.3 Local (adaptive) Thresholding

In many applications, global threshold cannot be found in a histogram, on the other hand single threshold cannot give good segmentation results over an entire image. For example, when the background is not constant and the contrast of objects varies across the image, thresholding may work well in one part of the image but may produce unsatisfactory results in other areas. One of the solutions is apply ***local (adaptive) thresholding*** [12].

Local thresholds can be determined by splitting an image into sub-images and calculating thresholds for each one. An image is divided into rectangular overlapping sub-images and the histograms are calculated for each one. The sub-image includes at least both object and background pixels. If the sub-image has bimodal histogram, then the minimum between the histogram peaks should be determined as a local threshold. If a histogram is unimodal, the threshold can be assigned by increasing the local thresholds found to nearby sub-images. Iterative increasing is necessary to find correct thresholds at each pixel [13]. In general, local thresholding is computationally more expensive than global thresholding. It is very useful for segmentation objects from varying background [5].

2.4 Finding peaks and valleys

The process of finding peaks and valleys is done using the function (Imextendedmax). Imextendedmax computes the extended-maxima transform, which is the regional maxima of the (H-maxima) transform. (H) is a nonnegative scalar. Regional maxima are connected components of pixels with a constant intensity value, and all external boundary pixels have a lower value. By default, imextendedmax uses 8-connected neighborhoods for 2-D images and 26-connected neighborhoods for 3-D images [14].

2.5 Proposed Algorithm

Input data: Capture beans image by the camera.

Output data: segmentation image.

Step I: Initialization of original image after preprocessing.

Step II: Global thresholding.

Step III: Calculate the minimum the histogram for overlapped sub-image, and computing the local thresholding, as shown in Fig. (2).

Step a: Choose the maximum area of single object as reference.

Step b: Compute the number of objects in the image that result from Global threshold (I).

Step c: Repeat the following steps until all separated objects are put in Iseg (image)

Step d:

- Compute the area for that object, Check the area of object less than or equal to reference.
- Updating Imseg by Iseg OR image contains object index.
- Otherwise (overlapping), compute the histogram and remove all zero values.

Step e:

- Apply local threshold after m= min (histogram), and compute t=m/255
- Apply local thresholding and remove all very small objects
- Compute the number of objects and Check the number of object greater than 1

Step f: Repeat the following steps until all objects are separated

- Check the size for an object less than or equal to reference
- Show the object in the Iseg by index-list =true
- Hide the object in old image by index-list =false

Step g: Repeat the following steps until all overlapping are finished**.**

- Check that image (I6) is not empty from objects**.**
- Compute the area in image (I6) and update I6 to gray scale**.**

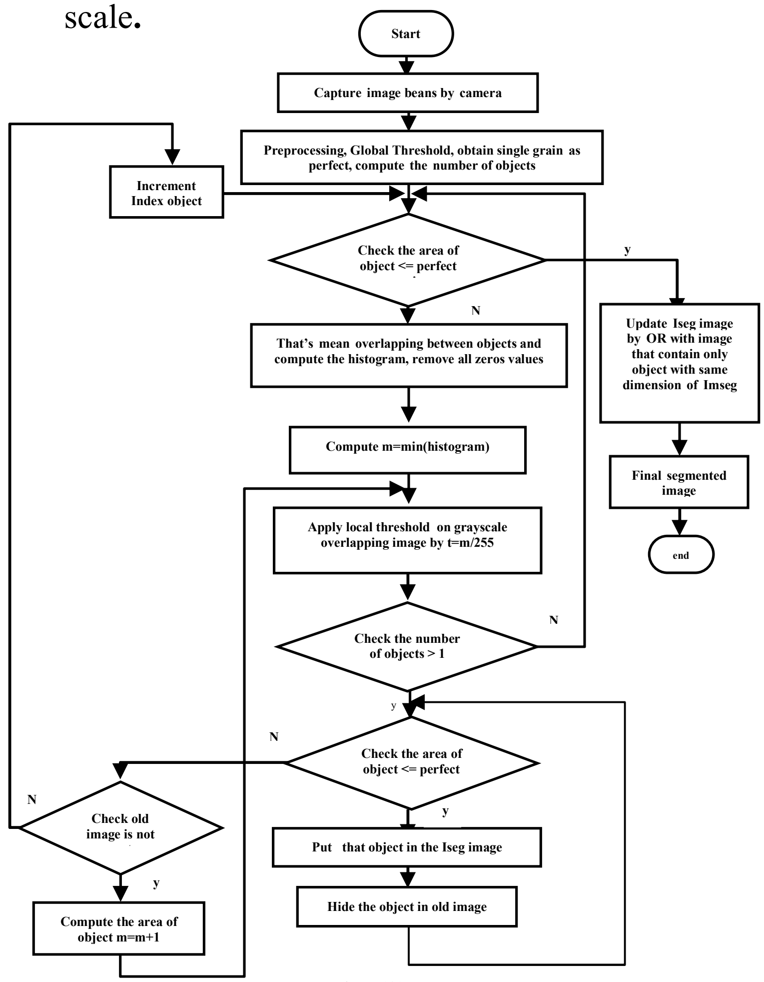


Fig. (2)
Flowchart of the proposed algorithm

Experimental work & Results

The proposed algorithm has been implemented using (Matlab 12a). Applying many steps on original image with color image size (608x1080) Fig. (3).

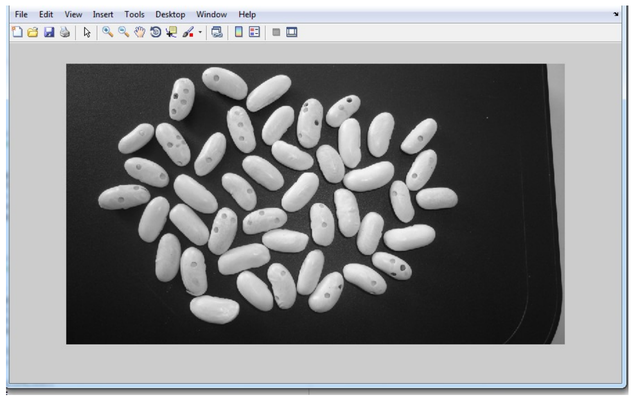


Fig. (3)
Original image

I) preprocessing is an essential step to realize good segmentation results; it includes many steps after reading the original image.

1. Converting RGB to gray
2. Showing the histogram for image
3. Adjustment of the histogram shown in Fig. (4- A), in maps the intensity values in grayscale image to increase the contrast of the output image
4. Clearing the boarder Fig. (4- B).

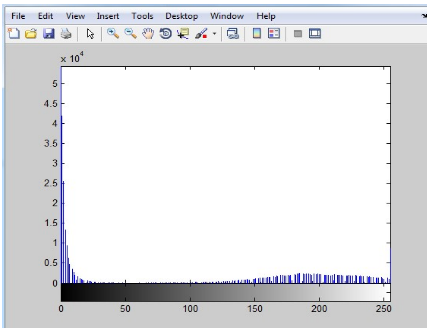


Fig. (4) A **Fig. (4) B**

Fig. (4)
Result preprocessing operation

II) Use global threshold:

1. By *graythresh (I)* command computes a global threshold (*level*) that can be used to convert an intensity image to a binary image with im2bw command. This technique is Otsu's threshold to automatically compute an optimum threshold T by *im2bw* to find black and white image and separate the edge pixels from the region interiors
2. Remove all very small objects from image by *bwareaopen command* shown in Fig. (5).

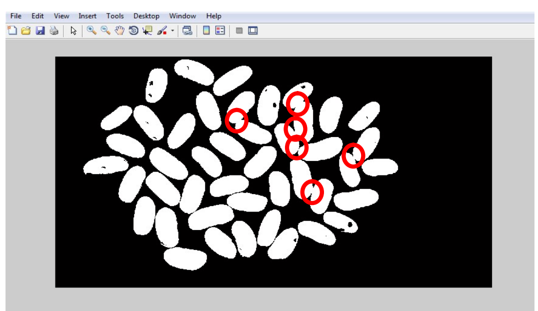


Fig. (5)
After apply global threshold appear overlapping in objects

III) Local threshold method includes:

1. Choose the maximum area for separated object as reference in the image is *ref*=5330
2. *Iseg* the result image must be black at first it takes the same size of the original image *BW* that result from previous step global threshold.
3. Compute the number of objects in BW image at first equall 38 because of overlapping
4. Repeat for all objects in *BW* image *(5, 6)*
5. *I4 (indexlist) = true*
6. Compute the area in *grain_data* for that *indexlist* object by *regionprops* command
7. Check the area less than or equal *ref*
8. Updating *Iseg* by *Iseg* OR *I4* shown in Fig. (6) and Fig. (7).

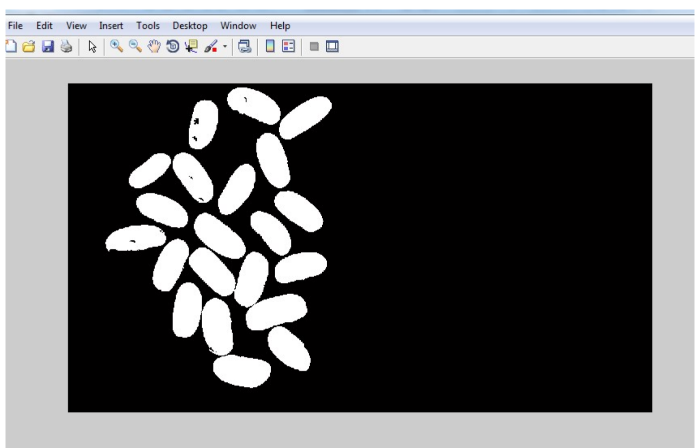


Fig. (6)
Iseg image is containing separated objects

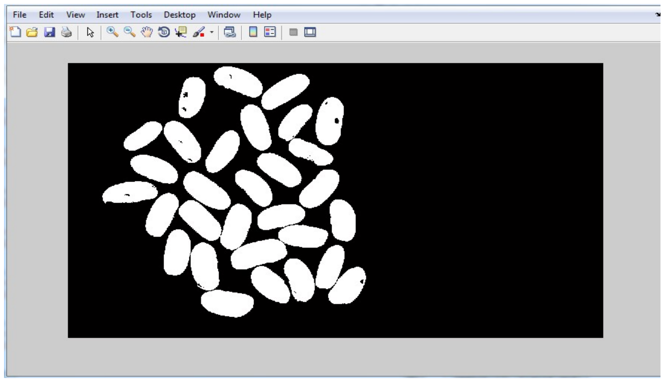
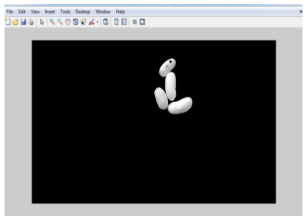


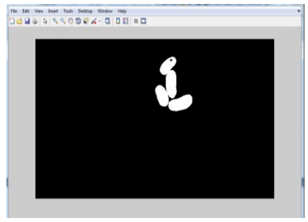
Fig. (7)
Add other single objects in Iseg image and open one of overlapping

IV) Otherwise that means overlapping:

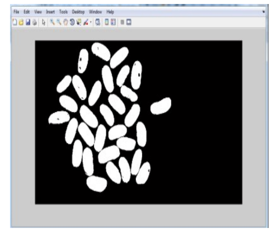
1. Update *I4* by multiplied *I3* with *I4* to convert *I4* to grayscale image Fig. (8 - A, E, I).
2. Compute the histogram in image *I4* in *z*
3. Remove all zeros in *z* to obtain the minimum value in the histogram in *m.*
4. Memorize *I4* in *I6*
5. Repeat the following until the number of objects is terminated
   5.1 Check the area whereas greater than *ref*,
   5.2 Compute *(t*) by dividing *m* into *255* Fig. (8 – B, F, J).
   5.3 Update by local threshold in *I6* by *im2bw*
   5.4 Remove all very small objects by *bwareaopen* command.
   5.5 Compute the number of objects in image *I6.*
   5.6 Check the number of objects greater than (1).



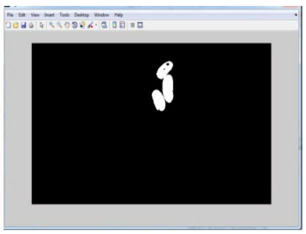
(A)
Another overlapping,I6



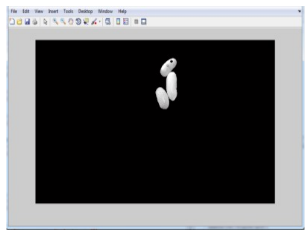
(B)
m=121 , t=0.4745



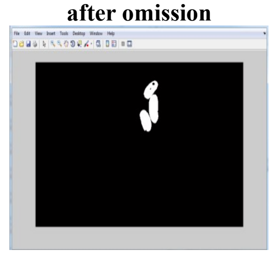
(C)
Iseg image take object after omission



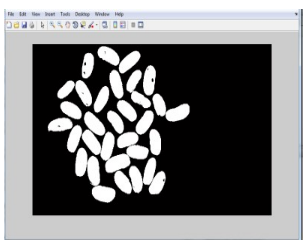
(D)
I6 Image after omission



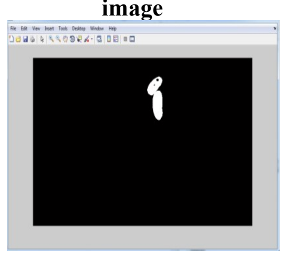
(E)
I6 is Converting to gray image



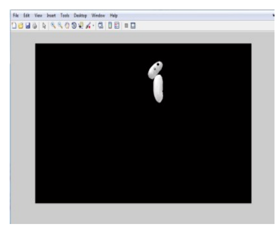
(F)
m=143, t=0.5608



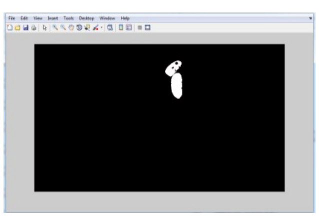
(G)
Iseg image takes object after separated



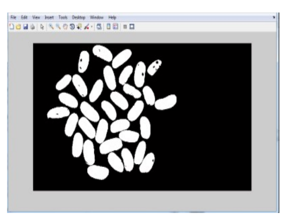
(H)
I6 Image after omission



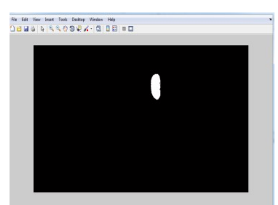
(I)
I6 is Converted to gray



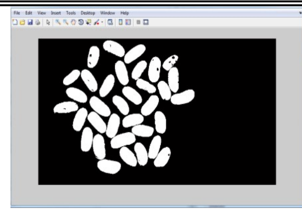
(J)
m=168, t=0.6588



(K)
Iseg image take object after separated



(L)
I6 Image after omission



(M)
Iseg image takes object after being separated

Fig. (8)
Results of each stage at high overlapped for the proposed algorithm

5.7 Apply for all objects in *I6* image
5.8 Check the area for that object which is less than or equal to *ref*
5.9 Apply *Iseg* by *indexlist =true* show Fig. (8 - C, G, K, M).
5.10 Hide (omit) that object from *I6* by *indexlist =false* shown in Fig. (8 – D, H, L).
6. Repeat until the iterative is completely that means all objects are separated (not found overlapping), obtain all objects are segmented and put them in image as result in Iseg image show Fig. (9,10) Apply for all object in *I6* image
5.8 Check the area for that object are is less than or equal *ref*
5.9 Apply *Iseg* by *indexlist =true* shown in Fig. (8 - C, G, K, M).
5.10 Hide (omit) that object from *I6* by *indexlist =false* show Fig. (8 - D, H, L).
6. Repeat until the iterative is completely that means all objects are separated (not found overlapping), obtain all objects are segmented and put them in image as result in Iseg image

Thus, *Iseg* adds other separated objects (not overlapping) the result will be show as in Fig. (9, 10) which represent the final result.

Fig. (9)
Add other single objects in Iseg image and open overlapping

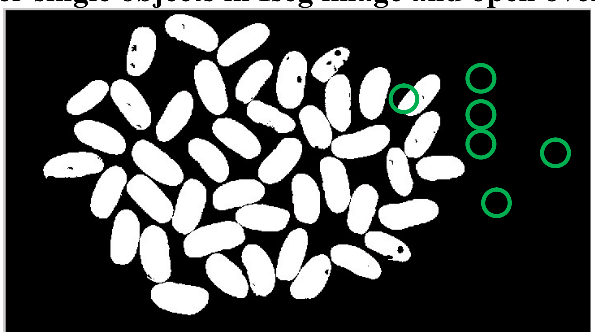


Fig. (10)
The final Result of Imseg image

Finally, the threshold values for all overlaps are shown in Table (1).

Table (1)
Shows the values of the local thresholding used in the image to solve all the overlapping problems

No. overlap	m	T
1	173	0.6824
2	121	0.4745
	143	0.5608
	168	0.6588
3	124	0.4863
4	145	0.5686

The number of objects is (44) which is the exact number. Apply the technique in Matlab in the same image with a threshold equaling (76 is the best threshold) shown in table (2) and Fig.

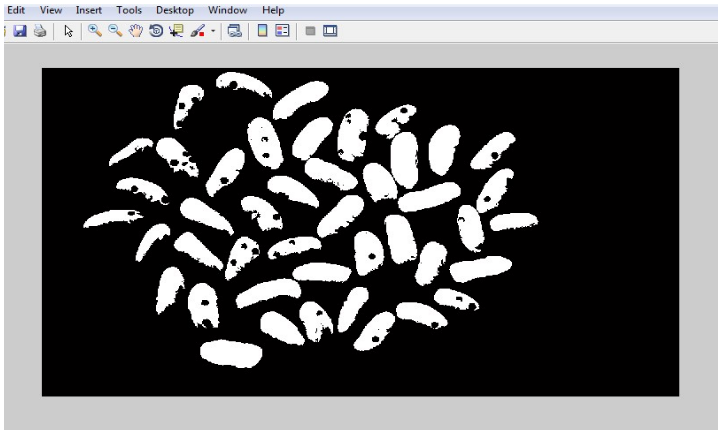


Fig (11)
Image resulting from Mat Lab based technique, and segmentation compare between proposed algorithm and technique in Mat Lab.

Table (2)
Relation Threshold value with the number of objects

Threshold value	Number of objects
50	39
51	40
52-54	41
55-59	42
60-62	43
63-76	44
77-78	43
79-80	42

The number of objects also equals (44). Table (3) shows the area of objects in the technique in Mat Lab.

Table (3)
When threshold (76) in image Fig (11)

Number of Objects	The area of Object	Number of Objects	The area of Object
1.	1827	23.	2828
2.	1319	24.	2543
3.	1968	25.	2941
4.	1658	26.	3230
5.	2480	27.	3615
6.	2506	28.	2303
7.	1798	29.	2648
8.	2478	30.	3106
9.	2918	31.	2976
10.	3235	32.	2078
11.	4225	33.	3621
12.	2881	34.	4077
13.	2096	35.	2534
14.	2225	36.	3495
15.	3351	37.	2764
16.	2365	38.	3539
17.	3848	39.	1760
18.	2812	40.	3366
19.	2646	41.	2850
20.	2052	42.	2480
21.	3283	43.	2606
22.	3166	44.	2004

Fig. (12) show the relation between the region area and threshold values

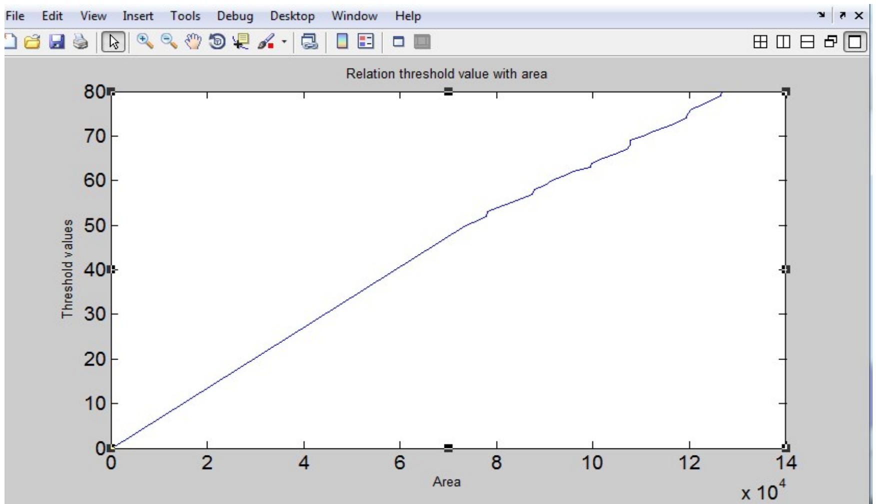


Fig. (12)
Relation between region area and threshold values

Table (4) show the new index of grain after separated.

Table (4)
Explains the new index of grains after separated

Overlapping no.	Number of grains	New index of grains
1	2	22 25
2	4	31 32 34 36
3	2	33 37
4	2	41 42

Table (5) show, table (6), and table (7) show the erosion ratio for overlapping objects after separated in proposed method and technique in Matlab respectively

Table (5)
Explains the ratio of erosion by proposed method

No.	A	B	A.B	A xor B	A Or B	A-A.B	B-A.B	(A-AB)+(B-AB)	Ratio
22	1624	1446	1445	180	1625	179	1	180	0.110
25	1584	1382	1376	214	1590	208	6	214	0.135
31	1329	1089	1086	246	1332	243	3	246	0.128
32	1945	1874	1873	73	1946	72	1	73	0.037
34	1559	1533	1531	30	1561	28	2	30	0.019
36	2090	2126	2087	42	2129	3	39	24	0.011
33	1924	1934	1920	18	1938	4	14	18	0.009
37	1671	1696	1669	29	1698	2	27	29	0.017
41	1435	1394	1391	47	1438	44	3	47	0.032
42	1662	1607	1605	59	1664	57	2	59	0.035

Table (6)
Explains the ratio of erosion by technique in Matlab

No.	A	B	A.B	A xor B	A Or B	A-A.B	B-A.B	(A-AB)+(B-AB)	Ratio
22	1624	1407	1406	219	1625	218	1	219	0.134
25	1584	1306	1300	290	1590	284	6	290	0.183
31	1329	927	927	402	1329	402	0	402	0.302
32	1945	1812	1812	133	1945	133	0	133	0.068
34	1559	1321	1321	238	1559	238	0	238	0.152
36	2090	1554	1554	536	2090	536	0	536	0.256
33	1924	1588	1588	336	1924	336	0	336	0.174
37	1671	1229	1229	442	1671	442	0	442	0.264
41	1435	1258	1258	177	1435	177	0	177	0.123
42	1662	1150	1150	512	1662	512	0	512	0.308

A is represent the original image for each grain overlapped; B is representing the logical image for each grain overlapped.

Table (7)
Explains the comparison of the ratios between two methods

No. of grains	The Ratio by Proposed Algorithm	The Ratio by Technique in Matlab
22	0.110	0.134
25	0.135	0.183
31	0.128	0.302
32	0.037	0.068
34	0.019	0.152
36	0.011	0.256
33	0.009	0.174
37	0.017	0.264
41	0.032	0.123
42	0.035	0.308

3. Conclusion

A proposed algorithm for segmentation was applied and gave a good result, and the accuracy of the methods was increased by 14% in compression to the traditional technique used in Matlab environment. Applying this method, it has been noticed that the objects have not overlapped; also, the erosion was reduced of edges from an average 19% to 5% of the grain size. It could be added that this method has showed more obvious outlines of the shape of the grain. The algorithm has given the accurate number of the objects which is (44), and as a future work, the proposed technique is recommended in medical image processing, especially in tumors detection, blood cells diseases detection and GIS (Geography Information System) images.

References

[1] Nobuyuki Otsu (1979), "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on systems, Vol. Smc-9, No. 1, Jan.

[2] Saif, Jamil, Al-Kubati, Ali (2012), "Image Segmentation Using Edge Detection and Thresholding", Dec.

[3] R. C. Gonzalez & R. E. Woods (1993), Digital Image Processing. Addison-Wesley Publishing Company.

[4] Castleman KR. (1996), "Digital Image Processing", Upper Saddle River Prentice, Hall.

[5] Adwiga Rogowska (2009), "Handbook and Fundamentals of Medical IMAGE Segmentation", Harvard Medical School, P. 73-90.

[6] ABDUL–NASIR A. S. & MASHOR, M., Y (2013),"Colour Image segmentation Approach for Detection of Malaria Parasites Using Various Colour Models and Means Clustering", University Malaysia, EISSN: 2224-2902 Vol. 10, Jan.

[7] Firas Ajil Jassim & Fawzi H. Altaani (2013), "Hybridization of Otsu Method and Median Filter for Color Image Segmentation", University of Baghdad, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Vol. 3, May.

[8] Mohanty A. K., Champati P. K., Swain S. K. & Lenka S. K (2011), "A review on Computer Aided Mammography for Breast Cancer Diagnosis and Classification Using Image Mining Methodology", International Journal of Computer Science and Communication, Vol. 2, No. 2, Dec., pp. 531-538.

[9] Salem Saleh Al-amir, N.V. Kalyankar & Khamitkar .S .D, (2010), "Image Segmentation by Using Threshold Techniques, Journal of Computing, Vol. ISSUE 5, MAY.

[10] Arifin A. Z. & Asano A. (2006), "Image Segmentation by Histogram Thresholding using Hierarchical Cluster Analysis", Pattern Recognition letters, Feb.

[11] Arifin A. Z. & Asano A. (2004), "Image Thresholding by histogram Segmentation Using Discriminant Analysis" In: Proceeding of Indonesia-Japan joint Scientific Symposium: IJJSS'04, pp. 169-174.

[12] Bernsen J. (1986), "Dynamic Thresholding on Gray-Level Images", Proc 8th Int Conf Pattern Recognition, Paris, France. Oct.: 1251-1255

[13] Fu. Ks. & Mui. JK. (1981), "A Survey on Image Segmentation", Pattern Recognition, 13(1); 3-16.

[14] The Math Work Inc. (2012), Image Processing toolbox for use with Matlab.