*Techniques Emerging From Classical Adaptive Thresholding Techniques*

Edris Amin; Kris Armstrong

edris.amin@rockets.utoledo.edu; kris.armstrong@rockets.utoledo.edu

EECS5330 Image Processing Fall 2011

Department of Electrical Engineering and Computer Science

The University of Toledo

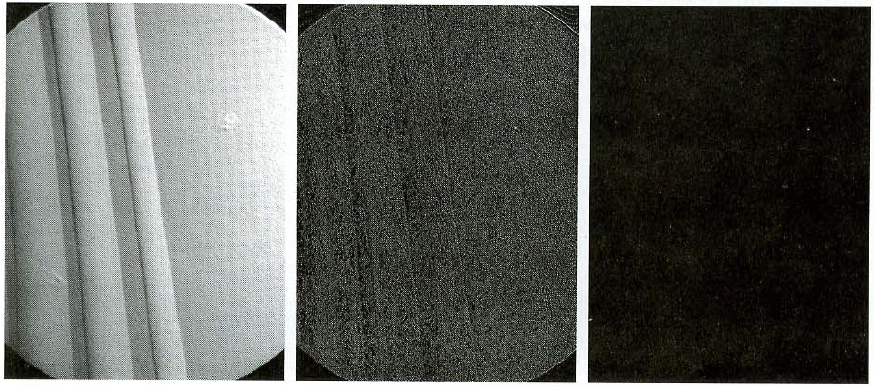
2801 West Bancroft Street, Toledo, OH 43606-3390

**Abstract – In this paper we present an overview of the concept of image thresholding. We offer a comprehensive explanation of various classical methods, modern techniques, and applications of adaptive thresholding techniques.**

1. ***Introduction***

A major goal of digital image processing is identification of objects. Identification is made difficult when an image was produced by mediocre imaging devices, in a poor environment, or an original document has suffered environmental degradation.

Various techniques have been derived for detecting variations in digital images; variations like points, lines, and edges [6]. Of the three, point detection is the simplest technique to explain. Point detection involves scanning a digital image and identifying points significantly different from the surroundings, the image in the figure below is an example of point detection taken from Gonzalez’s text [6].



In the figure above the original image in the far left has had the common Laplacian-8 mask applied to it providing the middle image. The middle image has been run through a filter only selecting pixels above a certain intensity providing the point detection result image in the far right.

Techniques for detecting lines in images is similar to point detection but instead of looking for specific pixel values when scanning an image line detection involves looking for differences in intensity levels of neighboring pixels. The techniques used for detecting lines and edges can be extended to identifying objects and the background in an image [Gonzalez].

Identification of objects in an image is a very important topic, the primary step in identification involves thresholding an image. Thresholding in short is the process in which the pixels of an image are reassigned either the value of a background, or object intensity [6]. Thresholding has various applications in fields like medical imaging, astronomical imaging, and degraded document restoration. The last area is the most common technique against which new thresholding techniques are compared against.

In this paper we aim to provide an overview of some classical image thresholding techniques in section II, some newer modern thresholding techniques in section III, and a brief overview of some applications in section IV.

1. ***Classical Techniques*** 
   1. ***Global Thresholding***

Global thresholding is summarized in Gonzalez’s text in the following. A user selected threshold level is chosen as the intensity boundary dividing an object with the background of an image [6]. The image pictured below is one produced in MATLAB, in which the chosen intensity boundary was iteratively selected to be the maximum gray level intensity of 20% of the pixels in a 256 gray level image. The MATLAB code is a modified version of code written for an assignment and is provided in the appendix.

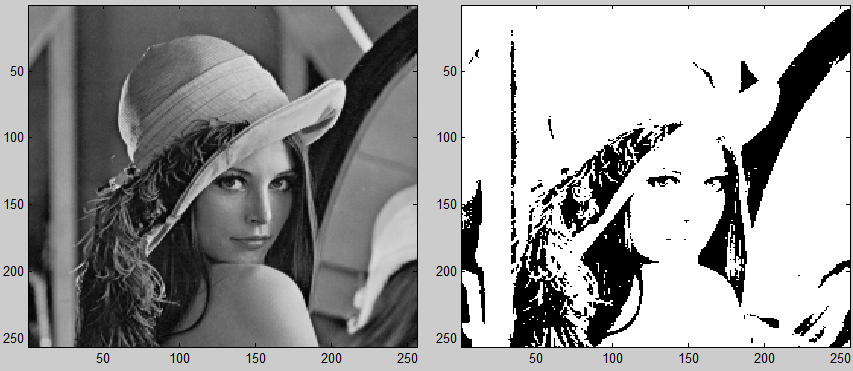


Image with global threshold applied for 20% intensity boundary

The image above has demonstrates a direct application of thresholding to a complete image. Adaptive and local techniques are presented in the following sections.

Thresholding is one of the most common techniques for separating objects in an image from the background. This is known as segmentation. Although this paper is aimed primarily at local adaptive thresholding, there are many other thresholding algorithms. An overview of these techniques should be informative and will show how local adaptive thresholding compares to these other thresholding techniques. The various forms of thresholding can be sub-divided into several categories. In their paper [3], Sezgin and Sankur, divide the types of thresholding into the following areas:

Shape - Histogram shape information

Cluster - Measurement of space clustering

Entropy - Histogram entropy information

Attribute - Image attribute information

Spatial - Spatial information

Local - Local characteristics

They compared 40 thresholding methods. They used 2 classes of non-destructive tests NDT (e.g. ultrasounds) and text images.

**Table 7** Thresholding evaluation ranking of 40 ***NDT*** images according to the overall average quality score.[1]

**Rank Method Average score (AVE) Rank Method Average score**

1 Cluster–Kittler 0.256 21 Shape–Ramesh 0.460

2 Entropy–Kapur 0.261 22 Spatial–Cheng 0.481

3 Entropy–Sahoo 0.269 23 Attribute–Tsai 0.484

4 Entropy–Yen 0.289 24 Local–Bernsen 0.550

5 Cluster–Lloyd 0.292 25 Spatial–Pal–a 0.554

6 Cluster–Otsu 0.318 26 Local–Yasuda 0.573

7 Cluster–Yanni 0.328 27 Local–Palumbo 0.587

8 Local–Yanowitz 0.339 28 Entropy–Sun 0.588

9 Attribute–Hertz 0.351 29 Attribute–Leung 0.590

10 Entropy–Li 0.364 30 Entropy–Pun–a 0.591

11 Spatial–Abutaleb 0.370 31 Spatial–Beghdadi 0.619

12 Attribute–Pikaz 0.383 32 Local–Oh 0.619

13 Shape–Guo 0.391 33 Local–Niblack 0.638

14 Cluster–Ridler 0.401 34 Spatial–Pal–b 0.642

15 Cluster–Jawahar–b 0.423 35 Entropy–Pun–b 0.665

16 Attribute–Huang 0.427 36 Local–White 0.665

17 Shape–Sezan 0.431 37 Local–Kamel 0.697

18 Entropy–Shanbag 0.433 38 Local–Sauvola 0.707

19 Shape–Rosenfeld 0.442 39 Cluster–Jawahar–a 0.735

20 Shape–Olivio 0.458 40 Entropy–Brink 0.753

**Table 8** Thresholding evaluation ranking of 40 ***degraded document*** images according to the overall average quality score.[1]

**Rank Method Average score (AV) Rank Method Average score**

1 Cluster–Kittler 0.046 21 Cluster–Yanni 0.300

2 Local–Sauvola 0.066 22 Attribute–Tsai 0.308

3 Local–White 0.08 23 Attribute–Hertz 0.317

4 Local–Bernsen 0.09 24 Spatial–Cheng 0.320

5 Shape–Ramesh 0.093 25 Local–Yasuda 0.336

6 Attribute–Leung 0.110 26 Entropy–Sun 0.39

7 Entropy–Li 0.114 27 Local–Kamel 0.391

8 Cluster–Ridler 0.136 28 Entropy–Pun–a 0.463

9 Entropy–Shanbag 0.144 29 Local–Niblack 0.475

10 Shape–Sezan 0.145 30 Local–Oh 0.514

11 Entropy–Shaoo 0.148 31 Spatial–Abutaleb 0.515

12 Entropy–Kapur 0.149 32 Spatial–Pal–a 0.533

13 Entropy–Yen 0.156 33 Spatial–Beghdadi 0.539

14 Entropy–Brink 0.17 34 Attribute–Huang 0.566

15 Cluster–Lloyd 0.18 35 Entropy–Pun–b 0.593

16 Local–Palumbo 0.195 36 Shape–Guo 0.596

17 Cluster–Otsu 0.197 37 Spatial–Pal–b 0.605

18 Cluster–Jawahar–b 0.251 38 Shape–Rosenfeld 0.663

19 Attribute–Pikaz 0.259 39 Shape–Olivio 0.711

20 Local–Yanowitz 0.288 40 Cluster–Jawahar–a 0.743

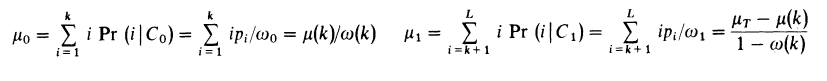
As can be seen from the data supplied by Sezgin and Sakur, the local thresholding methods do well on text images. Sauvola ranks 2nd and Bersen ranks 4th. On the other hand they fare poorly on NDT images with Sauvola ranking 38th and Bernsen coming in at 24th. For this data at least, the clear winner is Kittler, a clustering method, which ranks 1st in both image categories.

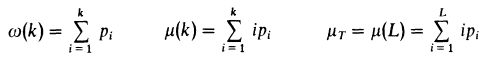
* 1. ***Otsu’s Technique***

Nobuyuki Otsu presented a thresholding technique in 1979. His technique was revolutionary in that the parameters of a thresholded image to be produced was calculated for all possible threshold levels before a specific threshold level was selected [7].

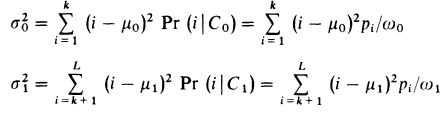
Otsu describes his algorithm’s procedure in [7] by first taking a digital image and representing the total number of gray levels as [1, 2, 3, …, L]. The total number of pixels N is the sum of all pixels ni with i representing the gray level of a pixels ni. The probability of intensity level i can be represented as:

The iterative process first selects a threshold k which defines the threshold boundary for two pixel classes C0 and C1, background and object. The pixels in class C0 have intensity levels ranging between 1 and k. The pixels in class C1 have intensity levels ranging between k+1 and L. The probabilities of class occurrence, mean class intensity levels, and class intensity variances are defined by Otsu as shown below [7].







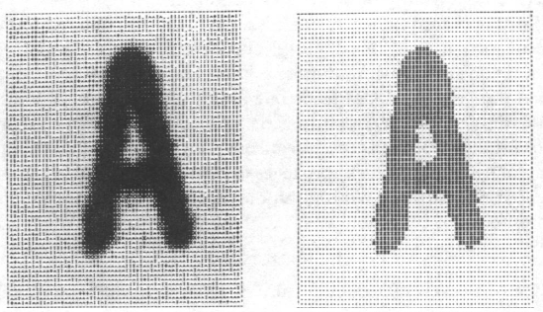
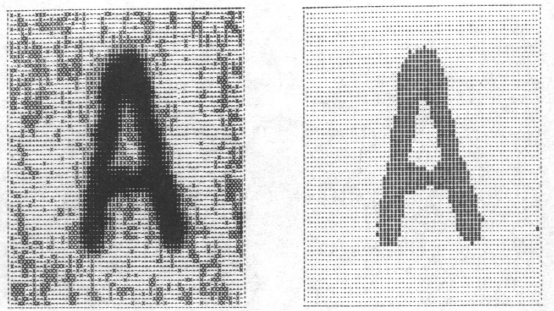


The mean intensity level and variance of intensity of the original image are represented by µT and σT2 respectively. The σw2 within-class and σB2 between-class variance of intensity levels calculations are shown below [7].



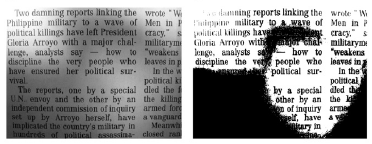
Now having calculated the mean intensities, and various variances for an intensity level k Otsu’s algorithm uses the ratio η defined below to determine the optimal threshold level k.

Essentially the optimum threshold level k is the level which results in the maximum between-class variance σB2 [7]. The images in the figures below are two results from Otsu’s original paper, the originals on the left and thresholded on the right. The 2 images represent the letter typed on a type writer with different type-writer ribbons [7].

Results on new ribbon Results on old ribbon

Otsu’s technique has a crucial strength in that it considers all possible threshold level effect before selecting one. One of its weaknesses is in its computational heaviness. Its other weakness is that in the end it is still applying the threshold to the entire image. By applying the threshold to the entire image there may exist some parts of the image which will remain degraded. This can be seen in the figure analyzing Otsu’s technique from the [10] paper. The image on the left is the original and on the right is the image after Otsu’s technique was applied.



Analysis taken from [10]

More recently developed thresholding techniques have tackled resolving the weaknesses mentioned. Niblack’s techniques, described in the following section, deal with the issue of applying a single threshold to the entire image. The technique by [10] described in section III.d confronts the computational complexity.

* 1. ***Bernsen***

One of the earlier forms of local adaptive thresholding was developed by Bernsen[1],[3]. Because of its early origin, quality of output and simplicity, this method is often used for comparison to other thresholding methods. This method uses a local neighborhood technique. The image is scanned one pixel at a time. The threshold level for each pixel is determined from the intensity level of its neighbors. The size of the neighborhood can be changed to suit the image and processing speed. Like many thresholding techniques, Bernsen’s method employs a statistical analysis to determine the threshold level for each pixel.

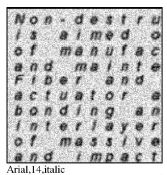
T(i,j) = .5(max[I(i+m,j+n)+min([I(i+m,j+n)])

where (I,j) is the location of center pixel and

m and n are the offsets for the neighborhood pixels

computed for each pixel in the neighborhood

As can be seen from the proceeding formula, Bernsen’s method first determines the minimum and maximum intensity level for the pixels in surrounding neighborhood. The average of these two values is then computed. This intensity level is then set as the threshold level. This threshold level is compared to the intensity level of the pixel at the center of the neighborhood. If the center pixel has a higher intensity level than the threshold, it’s set to the maximum intensity level of the image (the foreground) or if it’s intensity level is less than threshold level it’s set to the minimum intensity level of the image ( the background). In this method a binary image is produced. The setting of the center pixel’s intensity level may be reversed depending on whether the foreground or background has a higher intensity level.

* 1. ***Niblack***

Wayne Niblack’c thresholding technique was introduced in his 1986 text *An Introduction to Digital Image Processing.* His technique has been referenced countless times by researchers. His technique is revolutionary in that a threshold level is not calculated for the entire image. Instead his technique involves a scanning box, smaller than the image. As the box scans through the image the thresholding level is calculated for the image segment in the box, producing a final thresholded image once scanning is complete [8, 10].

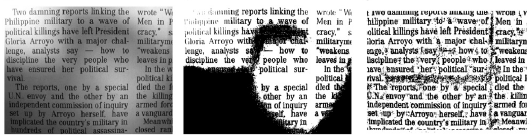
Niblack’s technique is explained in the recent paper [10]. To explain this process let’s consider a scanning box with local area b x b, local threshold level Tni, local mean threshold µni, and local threshold variance σni2.





The constant kni value in the Threshold calculation is a predefined constant ranging on the interval (0, 1). It is selected by the user to make corrections in case there is a very large variance across likely segments of the image.

The image in the following figure was taken from the [10] paper to show the improvement of Niblack’s localized technique over Otsu’s global technique. The image in the left is the original, the middle image has had Otsu’s technique applied to it, and the image on the right has had Niblack’s technique.



Analysis taken from [10] comparing Otsu’s technique to Niblack’s technique.

1. ***Modern Techniques*** 
   1. ***Sauvola***

Sauvola and PietikaKinen [2-4] have developed a method of local adaptive thresholding that uses a hybrid technique. This method is commonly referred to as the Sauvola method. This method was primarily aimed at documents, but it can be used for other types of images. The main thought behind this method is to separate the text of the document from any graphics that may also be in the document. After separating the text and the graphics from each other, different types of thresholding can be applied to each.

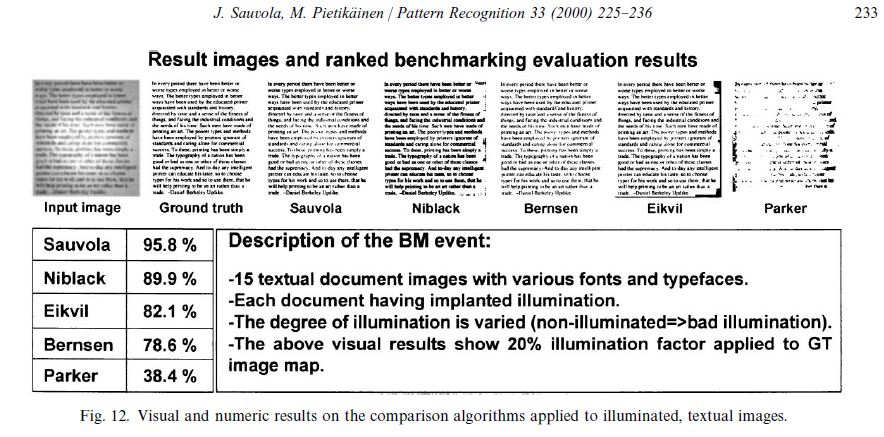
The method of separating the graphics from the text involves dividing the image into smaller rectangular blocks. Each of these blocks is then analyzed using two features. The first feature is simply the average gray value of a window. The second feature, ‘transient difference' measures local changes in contrast. These numbers are then compared to 4 predetermined values. The comparisons are then used to determine whether that window of the image is to be considered as a graphic or as text. This part of the image analyzes is meant to be simple so that it can be done quickly.

The graphic portion of the image is thresholded using the following technique. These regions are analyzed using ‘weighted bound’ and ‘threshold difference’ to determine local characteristics of the image. The local characteristic are then used to determine the local threshold for that region of the image.

The text portion of the image of is thresholded using a variation of Niblack’s method. Niblack’s method is described in more detail elsewhere in this paper. In Sauvalo’s method the standard deviation in the equation is given a variable weighting.

T(i,j) = m(i,j)+[1+k(σ(i,j)/R-1)]

Typical values are k=0.5, R=128

The reasoning behind this is to allow for compensation of the threshold if the background is dirty or stained paper. 

* 1. ***Gatos***

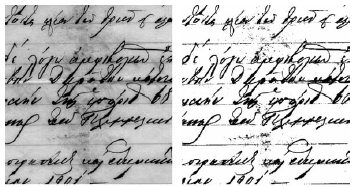
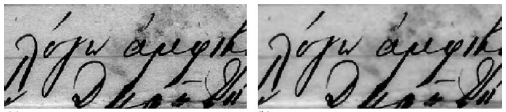
Basilios Gatos and colleagues from the National Center for Scientific Research “Demokritos” in Athens, Greece recently published their new adaptive thresholding technique. In a similar fashion they have analyzed previously developed techniques, with a goal of improvement in mind. Gatos’ technique utilizes the combined effects of a weiner filter and Suavola’s thresholding technique which is described in the previous section. The Gatos paper is written like most other papers comparing thresholding techniques using degraded images of text. The technique is documented in six steps [9]. For this report the six steps will be summarized as 3 macro-steps.

The first step aims to eliminate noisy areas a creating a better contrast boundary between the background and object regions. They have utilized an adaptive Wiener filter. The Wiener filter is described in the Gatos paper as being “based on statistics estimated from a local neighborhood around each pixel” [Gatos]. The equation taken from the paper generates a Wiener filtered image *I* from the source image *Is.* The Gatos paper mentioned that the Wiener filter is applied at in 3 x 3 pixel neighborhoods. In the equation below µ is the local mean, σ2 the variance at the local neighborhood, v2 is the average variance for each pixel in the neighborhood [9].



Equation for adaptive Weiner filter taken from [9]

The Wiener filtered image is a “smoothed” image and in Gatos technique the filtered image is processed using the Sauvola technique creating an estimated foreground image. The result of this macro step is represented in the figures below taken from the Gatos paper.

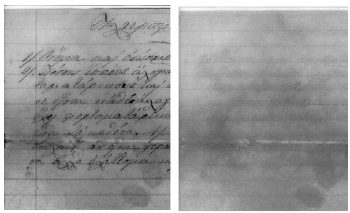


Original Wiener filtered Sauvola applied

The second macro step involves calculating the background image using the Wiener filtered and Sauvola image. The background image, referred to as *B* in the paper, is generated by comparing the Sauvola image pixels, *S(x,y)* to the Weiner filtered image *I(x,y).* The pixels considered as background in *S(x,y)* are set to the corresponding pixel value *I(x,y)* in background image *B(x,y)*. The remaining pixels, which correspond to text in *S(x,y),* are calculated using a neighborhood interpolation, represented by the equation taken from the Gatos paper below. The differential window size *dx* x *dy* is determined such that is covers a minimum of two characters. The resulting background image after interpolation is shown next to the Wiener filtered image below following the equation [9].



Equation for background image interpolation



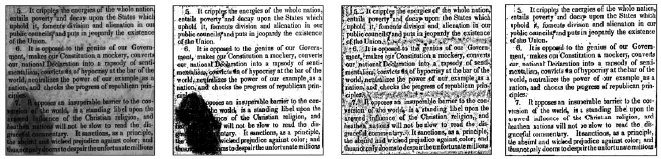
Background image with Wiener image on the left

In the second macro step a threshold image is calculated based on a threshold *d* selected based on the gray scale value of the background image. The threshold image *T(x,y)* is defined by the equation below [Gatos].



The last macro step in the Gatos technique starts by first improving the quality of the binary image. This is achieved by performing the thresholding interpolation in the previous equation but to a larger neighborhood. This results in a larger threshold image. In Gatos paper the new neighborhood size was an average of 16 pixels. For most cases the resulting image is twice as large as the original [9].

Finalizing the Gatos technique involves post-processing. This post processing utilizes a *shrink* *filter* to remove noise and a *swell* *filter* to eliminate line breaks and gaps in the foreground images. The figures provided below are the results from the Gatos paper comparing the Otsu, Niblack, and Gatos techniques



Original Otsu Niblack Gatos

Gato’s technique far improves the quality of the thresholded image, but as it will be seen in the next paper summary there is a cost involving computation.

* 1. ***A new adaptive thresholding involving fusion and computational improvement***

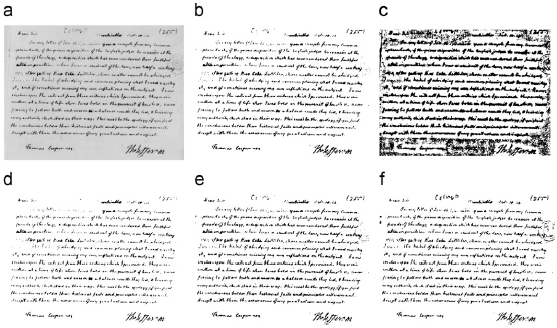
A very recent paper [Y.T Pai] on Adaptive thresholding mentions that over the years adaptive thresholding techniques have improved on differentiation between background and object levels. The states the obvious that local thresholding is an improvement over global thresholding. The new paper tackles the problem associated with the increasing computational complexity of each new technique as well as improving the final thresholded image quality.

The new technique improves upon Niblack’s region thresholding by selecting a variable region size. This is block sizes are calculated by calculating an intensity histogram of the average horizontal intensities by scanning the image vertically. The image is then segmented into horizontal slices with each slice’s width determined by its average intensity compared with the neighboring slice. The ultimate goal is to have slices with minimal variance. Each slice is further segmented into equally sized blocks. The following figure, taken from the paper of interest, shows that their segmentation scheme generally provides segments proportional in size to the characters or objects of interest [10].



Truncated image from [10] demonstrating segmentation scheme

After having determined optimal segments thresholding is applied within the blocks individually. The thresholding technique used is Otsu’s [10]. We mentioned earlier that Otsu’s technique is computationally complex. When applied in this new technique it is much less complex because it is being applied to a significantly smaller region rather than the entire image [10]. The figure below is taken from the paper presenting the new technique; it shows the results of different thresholding techniques for comparison.



1. original image, (b) Otsu’s technique, (c) Niblack’s technique, (d) Sauvola’s technique,

(e) Gatos’s technique, and (f) the proposed technique [10].

It is apparent from the figure above that the new technique provides a higher quality image. The previous techniques seemed to increase computational complexity, but this new technique improves that as well. The table shown below is taken from the [10] paper showing improvements in computation times. The reason for this improvement is mostly attributed to the segmentation scheme employed. Although Otsu’s technique may beat the new technique in computation, an improved thresholded image is apparent from the previous figure.

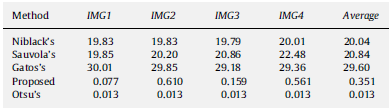
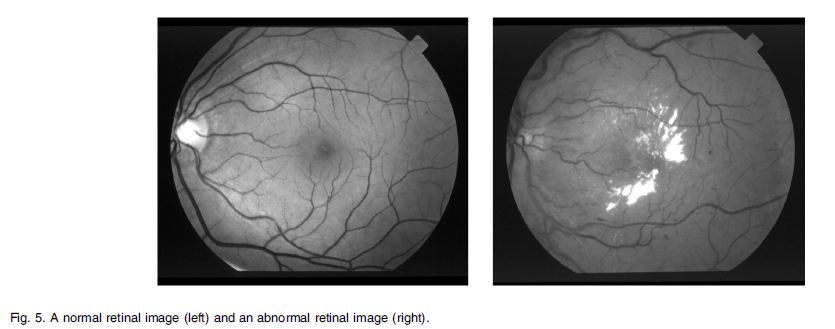
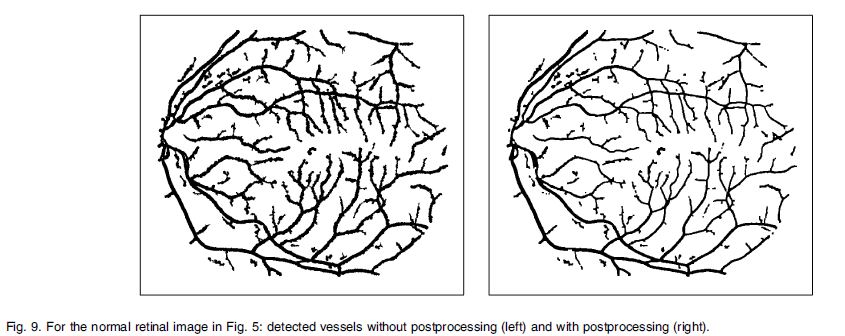


Table [10] reporting computational time in seconds of various techniques

1. ***Applications*** 
   1. ***Medical***

Another area where local thresholding is being applied is in the medical field.





These images of the blood vessels in human eye are from Xiaoyi Jiang’s and Daniel Mojon’s paper[5]. They used adaptive thresholding along with other techniques like filtering and edge detection to achieve these results.

In conclusion it would appear that adaptive local thresholding is ideally suited to images that have a background with many gray levels that need to be separated from many small objects in the foreground. This would include such applications as optical character recognition (OCR), medical imaging (segmenting blood vessels) and separating the cracks in a roadway from the background pavement. It also clear that there is still room for new algorithms to be developed using this basic technique.

* 1. ***Astronomy***

A very recent paper in the Solar Phys journal an automated technique for segmentation and characterizing regions on the surface of the Sun also referred to as solar filaments. Although they are not applying any of the advanced thresholding techniques presented in this paper they are using basic edge and line detection schemes to determine the size of and distance between solar filaments [11].

Effective mapping of solar filaments is important because solar flares could potentially damage orbiting satellites. Mapping Solar filaments improves solar flare forecasting. The technique proposed reported 95% accuracy for the measured filament number, and a 99% accuracy of measured filament area[11].

1. ***Conclusion***

The applications of improved adaptive thresholding go far beyond just character recognition. Over the years many have researched ways to improve the quality of images produced and reduce computational complexity. The trend has generally been to fuse older techniques and form a new technique which performs better. These new techniques have helped improve medical and astronomical imaging.

1. ***References***

[1] J. Brensen "Dynamic Thresholding of Gray Level Images," ICPR'86 Proc. Intl. Patt. Recog., pp 1251-1255 (1986)

[2] J. Sauvola and M. Pitikainen "Adaptive Document Image Binarization," Pattern Recog., 33 pp.225 - 236 (2000)

[3] M. Sezgin and B. Sankur “Survey over image thresholding techniques and quantitative performance evaluation”, Journal of Electronic Imaging 13(1), 146–165 (January 2004).

[4] F. Shafait and D. Keysers and T. Breuel “Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images” Shafait-efficient-binarization-SPIE08.pdf

[5] Xiaoyi Jiang and Daniel Mojon “Adaptive Local Thresholding by Verification-Based Multithreshold Probing with Application to Vessel Detection in Retinal Images” IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 1, JANUARY 2003

[6] Gonzalez, R., Woods, R.: 2007, Digital Image Processing, Prentice Hall, Upper Saddle River.

[7] N. Otsu, *A threshold selection method from gray-level histograms,* IEEE

Transactions onSystems, Man, and Cybernetics SMC9(1) (1979) 62–66.

[8] W. Niblack, *An Introduction to Digital Image Processing,* Pretice-Hall,

Englewood Cliffs, NJ, 1986.

[9] B. Gatos, I. Pratikakis, S.J. Perantonis, *Adaptive degraded document image*

*binarization*, Pattern Recognition 39 (2006) 317–327.

[10] Y.T. Pai, Y.F. Chang, S.J. Ruan, *Adaptive thresholding algorithm: Efficient computation technique based on intelligent block detection for degraded document images*, Pattern Recognition 43 (2010) 3177–3187.

[11] Y. Yuan, F.Y. Shih, J. Jing, H. Wang, J. Chae, *Automatic Solar Filament Segmentation*

*and Characterization*, Solar Phys (2011) 272:101–117, Springer Science + Business Media B.V. 2011

1. ***Appendix*** 
   1. ***Global Thresholding***

%MATLAB code taken from Edris Amin's assignment #1

%modified for term paper

clear all

load lenna

colormap(gray(256))

subplot(1, 3, 1); image(lenna)

x=1:256;

y=zeros(1,256);

for i = 1:256;

for j = 1:256;

y(lenna(i,j))=y(lenna(i,j))+1;

end

end

P = zeros(1,256);%P is the probabilities of intensities 1:256

C = 0; %sum of intensity Porb.

ival = 0; %variable for intensity value of 20%

for i=1:256;

P(i)=100\*y(i)/(256\*256);

C = P(i)+C;

if C>=20

ival = i;

break

end

end

%creating saturated image pixels where intensity <= ival will be set to 0

zero20 = zeros(256,256); %matrix for saturated image

for i= 1:256

for j= 1:256

if (lenna(i,j) <= ival)

zero20(i,j) = 0;

else

zero20(i,j) = 255;

end

end

end

subplot(1, 3, 2);

image(zero20);