Analysis of

J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen's

Research on

Adaptive Document Binarization, 1997

John Leung

CS555 - Digital Image Processing

Cal Poly Pomona

Fall, 2002

Abstract. The authors of the paper, J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen proposed a new method for adaptive document image binarization. This method is based on Niblack's work [2], which the authors believed to be the best of adaptive document image binarization methods at the time. The basic idea is to find a local threshold using the local standard deviation. Binarization will be based on this threshold; pixels will be set to white if their values are higher than the threshold, black otherwise.

1. Introduction

Even as we are going into a more paperless society, paper still remains as the number one medium for visual information. People still prefer to read newspapers, magazines, and books printed on paper even if there are online versions of it. In order to digitally recapture these printed-only texts, they first need to be either captured by scanners or digital cameras. Document image processing is a necessary step involved before these captured text images can be made of any use. Binarization of text images is one of the important techniques involved in this document image processing.

2. Background Information

Document image processing usually involve the follow steps:

Binarization → Image Enhancements → OCR

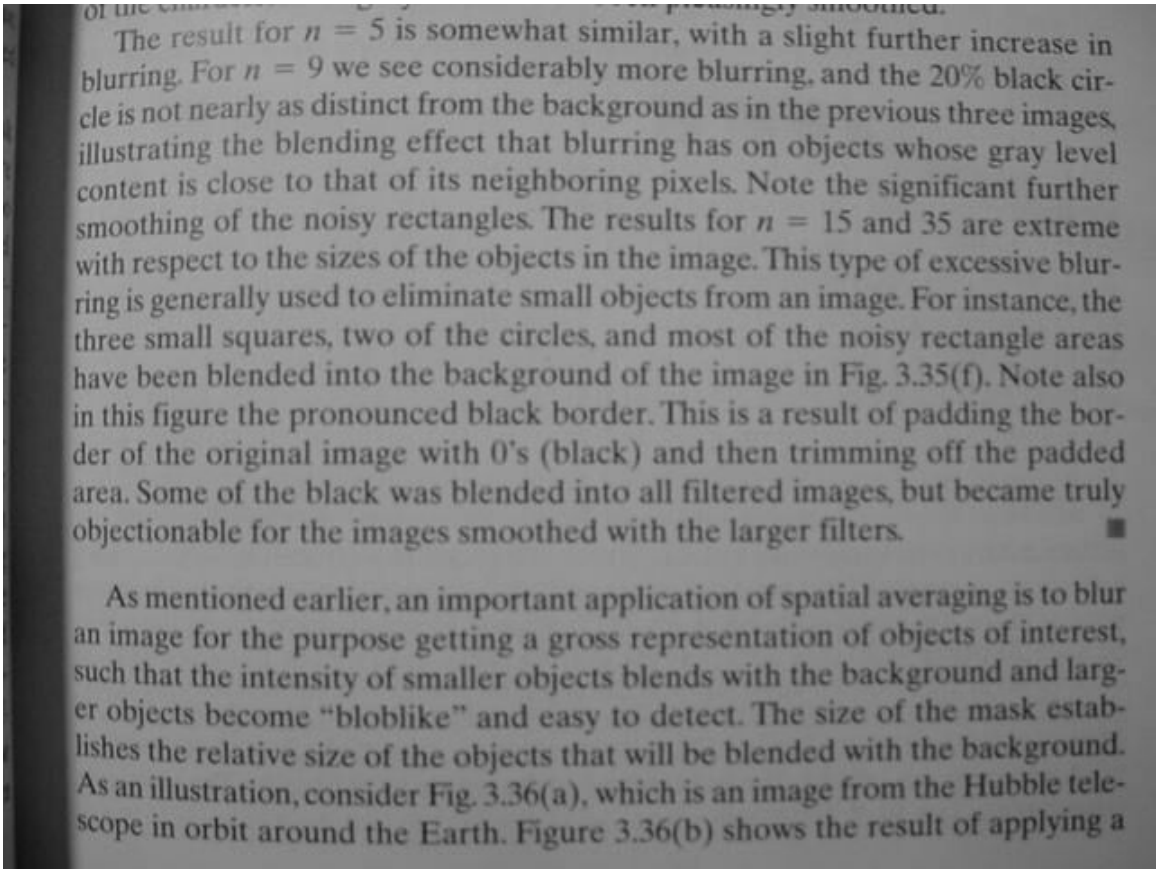
OR

Image Enhancements → Binarization → OCR

(Image enhancement and binarization are usually considered the pre-processing step in document image processing.)

In order for most OCR (Optical Character Recognition) programs to recognize text images, they need to be binarized first. The order of in which binarization comes in depends on the type of image enhancement techniques involved. Some of them require the input image to be in gray scale or color form, while others require the image to be binarized first. The number of image enhancement techniques used for text images are vast, however, they basically are broken down to the following: interpolation, reconnection of broken lines in text, lighting corrections, and noise filtering. Some addresses specifically to one of these issues, and some are more general purposed. Listed in the reference page are some of the researches done by others on this. However, the focus of this paper is the binarization step, to be discussed next.

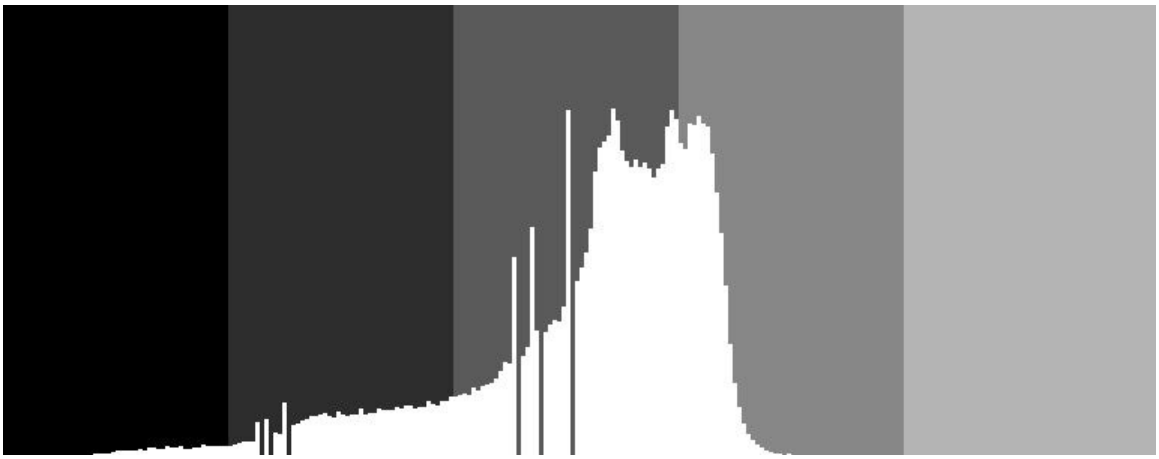
There are many binarization methods also. The simplest method of all is global thresholding. What it does is it simply takes a value that lies between the maximum and minimum grayscale value of the image, and for each pixel in the image, if it lies below this value, it will be turned to black, white otherwise. For text images, the threshold is best determined by taking the histogram of the image and look at where exactly is the best point to divide the background and the foreground (the actually text). In images where the original document's text is dark enough and printed on a near white paper, the histogram will have a nice bimodal shape – that is the histogram will have two distinct population curves, one for foreground text and one for the white background. However, if the original document's text is not dark enough, or printed on grayish newsprints, or even worst, when lighting comes into play with digital cameras, we will not be so lucky to get such nice histograms. The following sample images best describe the problem:



The result for $n = 5$ is somewhat similar, with a slight further increase in blurring. For $n = 9$ we see considerably more blurring, and the 20% black circle is not nearly as distinct from the background as in the previous three images, illustrating the blending effect that blurring has on objects whose gray level content is close to that of its neighboring pixels. Note the significant further smoothing of the noisy rectangles. The results for $n = 15$ and 35 are extreme with respect to the sizes of the objects in the image. This type of excessive blurring is generally used to eliminate small objects from an image. For instance, the three small squares, two of the circles, and most of the noisy rectangle areas have been blended into the background of the image in Fig. 3.35(f). Note also in this figure the pronounced black border. This is a result of padding the border of the original image with 0's (black) and then trimming off the padded area. Some of the black was blended into all filtered images, but became truly objectionable for the images smoothed with the larger filters. ■

As mentioned earlier, an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become "bloblike" and easy to detect. The size of the mask establishes the relative size of the objects that will be blended with the background. As an illustration, consider Fig. 3.36(a), which is an image from the Hubble telescope in orbit around the Earth. Figure 3.36(b) shows the result of applying a

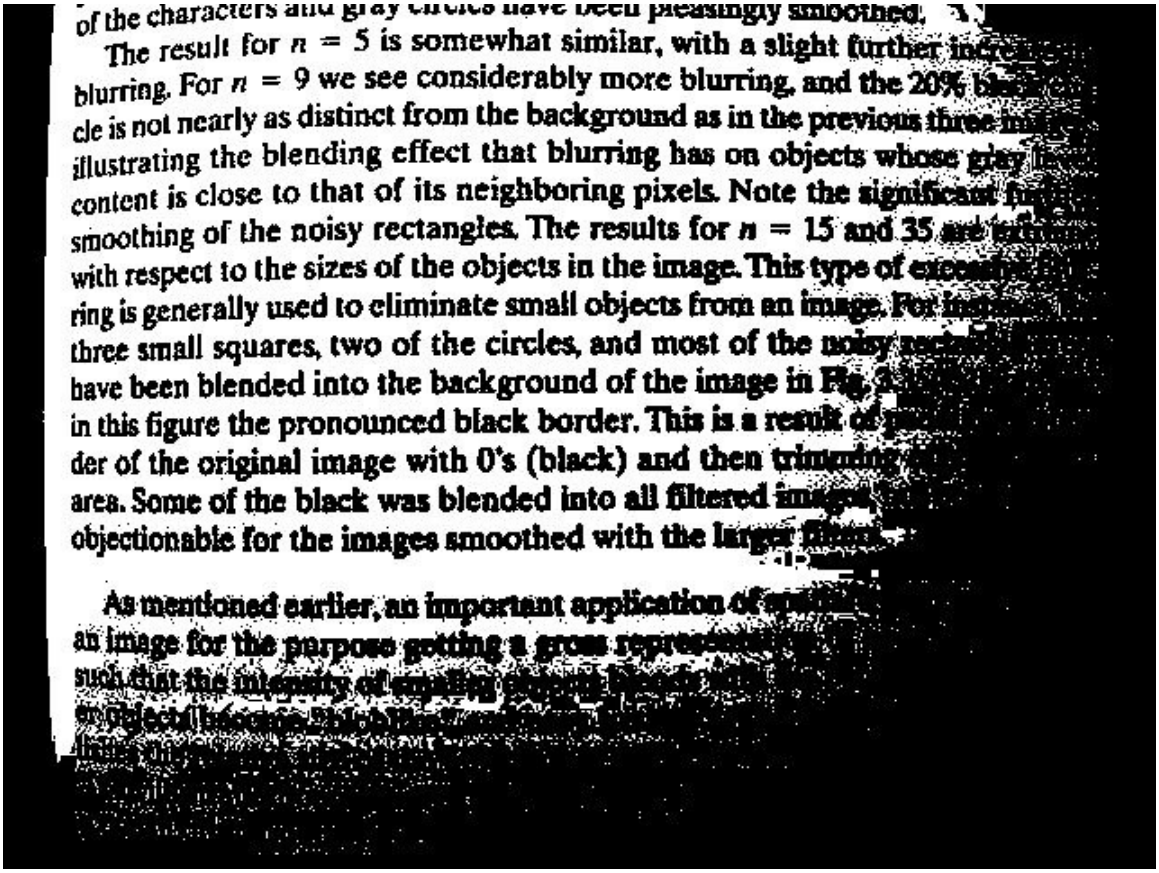
Fig.1. A sample text image taken with a digital camera at 640x480 resolution.



0 50 100 150 200 255

Fig.2. Histogram of Fig.1

As seen in Fig.2, histograms do not always come in a nice bimodal shape. In this histogram we can be fooled to believe that the foreground text population is around pixel value 130 and background to be around pixel value 150. This is not the case; the following is the global binarized image thresholding at the midpoint of 140:



of the characters and gray circles have been pleasingly smoothed.

The result for $n = 5$ is somewhat similar, with a slight further increase in blurring. For $n = 9$ we see considerably more blurring, and the 20% black circle is not nearly as distinct from the background as in the previous three images, illustrating the blending effect that blurring has on objects whose gray level content is close to that of its neighboring pixels. Note the significant further smoothing of the noisy rectangles. The results for $n = 15$ and 35 are [illegible] with respect to the sizes of the objects in the image. This type of excessive blurring is generally used to eliminate small objects from an image. For instance, the three small squares, two of the circles, and most of the noisy rectangles have been blended into the background of the image in Fig. 3. [illegible] in this figure the pronounced black border. This is a result of padding the border of the original image with 0's (black) and then trimming the [illegible] area. Some of the black was blended into all filtered images, [illegible] objectionable for the images smoothed with the larger filters.

As mentioned earlier, an important application of [illegible] an image for the purpose getting a gross representation [illegible] such that the intensity of smaller objects blend [illegible] or objects become [illegible]

Fig.3. Global binarization thresholding at 140.

This is clearly an unacceptable binarization of the original. Other than the high population at 130, there is no other distinct values where there is high enough population to be considered the foreground text. Even if we try all possible values, the best binarization we can get out of this image is thresholding at around 110:

of the characters and gray circles have been pleasingly smoothed.

The result for $n = 5$ is somewhat similar, with a slight further increase in blurring. For $n = 9$ we see considerably more blurring, and the 20% black circle is not nearly as distinct from the background as in the previous three images, illustrating the blending effect that blurring has on objects whose gray level content is close to that of its neighboring pixels. Note the significant further smoothing of the noisy rectangles. The results for $n = 15$ and 35 are extreme with respect to the sizes of the objects in the image. This type of excessive blurring is generally used to eliminate small objects from an image. For instance, the three small squares, two of the circles, and most of the noisy rectangle areas have been blended into the background of the image in Fig. 3.35(f). Note also in this figure the pronounced black border. This is a result of padding the border of the original image with 0's (black) and then trimming off the padded area. Some of the black was blended into all filtered images, but became truly objectionable for the images smoothed with the larger filters. ■

As mentioned earlier, an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become "bloblike" and easy to detect. The size of the mask establishes the relative size of the objects that will be blended with the background. As an illustration, consider Fig. 3.36(a), which is an image from the Hubble telescope in orbit around the Earth. Figure 3.36(b) shows the result of applying a

Fig.4. Global binarization thresholding at 110.

Even with this 'best' global binarization, there are still very dark and blurred letters in the lower right corner, over lighten letters in the upper left corner, and much lost of textual information in the middle such as the letter 's'. Unfortunately global binarization is the **only** binarization method offered by most image processing software. Even the popular Adobe Photoshop includes only this method. Thus, we need to come up with more powerful binarization techniques.

3. The Adaptive Local Binarization

By now, we should see that a more *localized* binarization is needed. Local binarization also comes in many favors. The simplest one is to just extend the global binarization method locally. Instead of applying a single threshold across the whole image, we can find the local threshold by looking at the local histogram information. The problem with this is that areas where the local window does not cover any texts will try to produce some 'text'; thus creating unwanted black blobs in the background. We can fix this a little by saying if the majority of the local area is in one mode, then the current pixel is part of the background. The problem with this is how to define what is considered 'one mode', and what is not. As mentioned before, lighting can come into effect for images taken with digital cameras. Thus, the background does not contain just one nice uniformed grayscale value. So how do we determine if the darker pixels are part of some text or caused by lighting effects? Even for scanned images, as the local window that first only covers the background moves closer to some text, it is very difficult to tell when exactly is there enough text information included in the local window and when there is not. Niblack and the authors answer this by using the standard deviation.

The authors have looked at many binarization methods and found Niblack's method to be the best. However, it still has room for improvement, so they refined it more. Niblack's binarization method involves taking the standard deviation, the local mean, and a user defined parameter to determine the threshold. The thresholding formula is as follows:

$T = m + k * s$

Where:
m is the local mean,
k is the user input(gets negative values)
s is the local standard deviation, calculated by:

$s = [E_l(I^2) - E_l(I)^2]^{1/2}$

where:
$E_l(I^2)$ is local sum of (pixel Intensity^2)/#pixs
$E_l(I)^2$ is local sum of (pixIntensity /#pixs)^2

This Niblack's thresholding method produces some artifacts in the background and does not work too well for images with dark backgrounds (shown in results section). The authors improved on this by coming up with a new formula based on Niblack's work. The author's new thresholding formula is:

$T = m *[1+ k *(s/R -1)]$

where
m and s are the same as above,
k is still the user input, but gets positive values this time, and
R is the dynamic range of the standard deviation
(The authors found by experiment that the best R and k values are 128 and 0.5 respectively.)

For color images, Thouin and Chang [3] suggest converting to grayscale using the following equation:

Grayscale = 0.299Red + 0.587Green + 0.114Blue

4. Implementation

The authors also suggested an algorithm for finding non-text image blocks in the document and turning them to just white background. Since most OCR programs today can recognize non-textual blocks, I did not bother to implement that algorithm. For faster execution, the authors also suggested not doing the adaptive threshold directly on every pixel. Instead, it is done only on some base pixels that are evenly spread. The binarizations of pixels that lie between the base pixels are interpolated by a fast table lookup method based on the base pixels. Even though the authors did not say that this fast algorithm produces inferior output, the fact is that binarization by interpolation are really just 'smart guessing' these non-base pixels. Thus, I do not believe this can always produce as good of an output as directly determining the threshold of every pixel. The authors mentioned that this approach is **suitable** for **most** images seem to imply this as well. Since I am more interested in finding a good binarization method rather than a fast one that does not produce as good of an output, I did not intend to implement that unless the running time really proved it to be necessary. In my experiments, local mask of sizes of 7x7 and 9x9 produces the best output. Mask sizes of 13x13 and above seem to do more harm than help by producing more artifacts surrounding the text. Runs of mask sizes 7x7 and 9x9 ran fairly fast, around 1-2 seconds on my 1.6 GHz PC. This translates to at most a few seconds even on older PCs. Thus, I believe the need for the fast algorithm is necessary only for batch conversions.

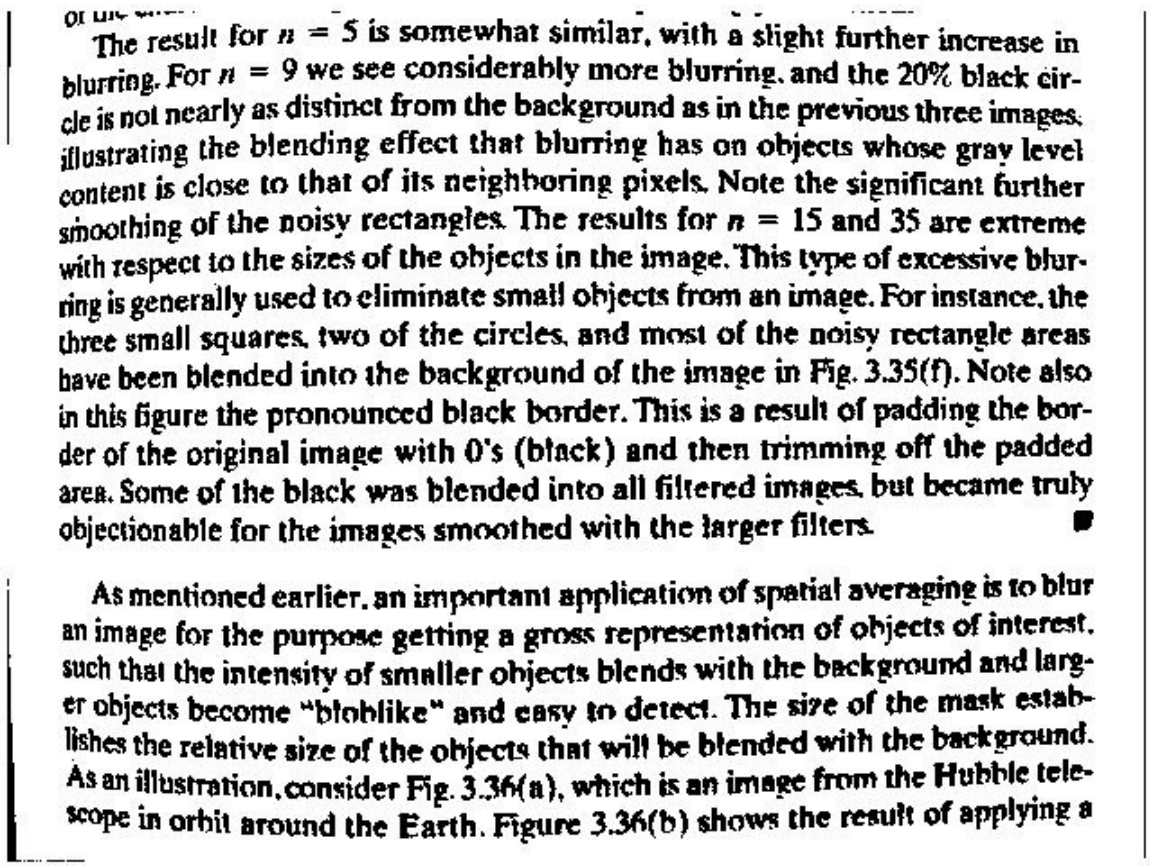
As mentioned before, the authors experimentally found the best R and k values to be 128 and 0.5 respectively. In my experiments, I found the best R and k values to be 26 and 0.15 respectively.

5. Results

The following is a sample result of the authors did on comparing their method against Niblack's method:

Fig. 9. Examples of binarization results for textual and background images.

As for my experiments, most input images are taken with a digital camera since I believe global binarization is actually good enough for many nice scanned images. The following are results of my implementation of the authors' adaptive binarization method:



The result for $n = 5$ is somewhat similar, with a slight further increase in blurring. For $n = 9$ we see considerably more blurring, and the 20% black circle is not nearly as distinct from the background as in the previous three images, illustrating the blending effect that blurring has on objects whose gray level content is close to that of its neighboring pixels. Note the significant further smoothing of the noisy rectangles. The results for $n = 15$ and 35 are extreme with respect to the sizes of the objects in the image. This type of excessive blurring is generally used to eliminate small objects from an image. For instance, the three small squares, two of the circles, and most of the noisy rectangle areas have been blended into the background of the image in Fig. 3.35(f). Note also in this figure the pronounced black border. This is a result of padding the border of the original image with 0's (black) and then trimming off the padded area. Some of the black was blended into all filtered images, but became truly objectionable for the images smoothed with the larger filters. ■

As mentioned earlier, an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become "bloblike" and easy to detect. The size of the mask establishes the relative size of the objects that will be blended with the background. As an illustration, consider Fig. 3.36(a), which is an image from the Hubble telescope in orbit around the Earth. Figure 3.36(b) shows the result of applying a

Fig. 5. Adaptive binarized image using mask size 9x9, k of 0.15, and R of 26

The result for $n = 5$ is somewhat similar, with a slight further increase in blurring. For $n = 9$ we see considerably more blurring, and the 20% black circle is not nearly as distinct from the background as in the previous three images, illustrating the blending effect that blurring has on objects whose gray level content is close to that of its neighboring pixels. Note the significant further smoothing of the noisy rectangles. The results for $n = 15$ and 35 are extreme with respect to the sizes of the objects in the image. This type of excessive blurring is generally used to eliminate small objects from an image. For instance, the three small squares, two of the circles, and most of the noisy rectangle areas have been blended into the background of the image in Fig. 3.35(f). Note also in this figure the pronounced black border. This is a result of padding the border of the original image with 0's (black) and then trimming off the padded area. Some of the black was blended into all filtered images, but became truly objectionable for the images smoothed with the larger filters. ■

As mentioned earlier, an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become "bloblike" and easy to detect. The size of the mask establishes the relative size of the objects that will be blended with the background. As an illustration, consider Fig. 3.36(a), which is an image from the Hubble telescope in orbit around the Earth. Figure 3.36(b) shows the result of applying a

Fig. 6. Adaptive binarized image using mask size 13x13, k of 0.15, and R of 26

If inspected carefully, one can see that the output of mask size 13x13 produces artifacts surrounding the texts. For example, in line 3, it seem to produced unwanted artifacts in the words "distinct" and "background", making them appear as if they are underlined. Other similar artifacts can be found elsewhere. Other outputs samples with different k and R values are not included since there are too many different combinations of them. These k and R values and basically the best values for most of my images.

The results are fairly good, considering the fuzzy quality of the image captured by the digital camera. However, they are still not good enough to be fed to OCR programs

yet. The results of these images can be much better if they are enhanced before and/or after binarization. One example of such enhancement is the BSA method proposed by Thouin and Chang [3]. Other methods I have found are included in the reference page; [3] , [4], and [5] are enhancements on grayscale images and [6] and [7] are enhancements on binary images. A very good text pre-processing system can be realized if it works together with one or more of these enhancement techniques.

6. Conclusion

Text image processing is a very important in this information age. The applications and need for this are endless; some examples are for archival, database entries, information retrieval, machine vision, and text to speech for the blind. Binarization is an imperative step within text image processing. Other image enhancement techniques will be a waste of good effort if the binarization step is not done correctly. But at the same time, binarization cannot be the only step involved also. Together with some good image enhancement and OCR techniques, a very powerful document processing system can be achieved.

References:

[1] Sauvola, J., Seppanen, T., Haapakoski, S., and Pietikainen, M. 1997. *Adaptive Document Binarization,* Proceedings of the Fourth International Conference on Document Analysis and Recognition, Volume 1, 1997.

[2] Niblack, W. 1986. An introduction to image processing. Englewood Cliffs, N.J.: Prentice Hall, pp. 115-116.

[3] Thouin, P. and Chang, C. 2000. *Automated system for restoration of low-resolution document and text images,* Journal of Electronic Imaging, April 2001.

[4] Taylor, M. and Dance, C. 1998. *Enhancement of Document Images from Cameras,* Document Recognition 5, SPIE Vol. 3305

[5] Palenichka, R. and Zinterhof, P. 2001 *Structure-adaptive filtering based on polynomial regression modeling of image intensity,* Journal of Electronic Imaging, April 2001

[6] Hobby, J. and Ho, T. 1997. *Enhancing Degraded Document Images via Bitmap Clustering and Averaging,* Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1999 Vol. 1

[7] Kamat, V., Dougherty, E., and Barrera, J. 2000. *Multiresolution Bayesian design of binary filters*, Journal of Electronic Imaging, July 2000.