Binarization and Localization of Text Images Captured on a Mobile Phone Camera

Bhavna Antony [1], Peeta Basa Pati [2], A G Ramakrishnan [3]
1 *Electrical Engineering Department, Indian Institute of Science Bangalore, India, bhavna@ragashri.ee.iisc.ernet.in*
2 *Electrical Engineering Department, Indian Institute of Science Bangalore, India, pati@ragashri.ee.iisc.ernet.in*
3 *Electrical Engineering Department, Indian Institute of Science Bangalore, India, ramkiag@ragashri.ee.iisc.ernet.in*

Abstract

This paper proposes and compares four methods of binarizing text images captured using a camera mounted on a cell phone. The advantages and disadvantages(image clarity and computational complexity) of each method over the others are demonstrated through binarized results. The images are of VGA or lower resolution.

Keywords: *Global binarization, adaptive thresholding.*

1. INTRODUCTION

Advances in communication technology has enabled easy and rapid sharing of ideas and information, be it in the form of speech or images. Mobile phones, due to their portability, have been largely responsible for this rapid progress. Multimedia Messaging Services (MMS) has made the exchange of pictures possible. The improvements made in image acquisition technology available on mobile phones has made this mode of communication popular.

Presently, mobile phones are equipped with cameras capable of acquiring images with resolutions over a megapixel. Although a scanned image provides better image clarity, cameras mounted on cell-phones offer mobility, instant accessibility and fast data transfer rates. This enables the instantaneous communication of written information with others.

The cameras mounted on cell phones capture images with more than 1-bit depth of information. For text images, each pixel is employed for representing either the text or the background. In the case of a 24-bit color image, the binarized image would be smaller in size by a factor of 24, thus, reducing the transmission costs. Besides, in many situations, we could remove the unwanted portions of the image before its transmission, leading to further reduction in the number of transmitted bits. Furthermore, processing of these images generates outputs which are visually more appealing. Thus, there is a need to convert these captured images to binary images.

While capturing text images using a camera mounted on a cellphone, there is little control on the lighting and environmental noise. Many a time, the plane of the paper/surface is not parallel to the sensor plane, leading to perspective distortion of the text elements. Due to unequal distances of the various points on the surface from the optical center of the lens, there is variation in the intensity level of the captured image from one region to another. This happens for images with homogeneous backgrounds and uniformly illuminated surfaces as well. Fig.1(a) shows an example of such an image.

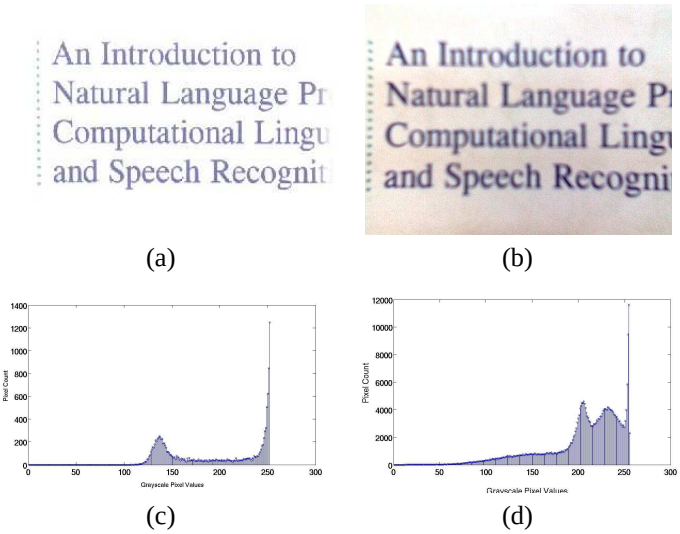

Fig. 1: (a) Image scanned, resolution = 150dpi (b) Image captured on a mobile phone camera, resolution = $480X640$ (c) Histogram of Fig. (a) (d) Histogram of Fig. (b)

Binarization of these camera captured images can be quite tricky. The general assumption of the text images having *bimodal* histogram does not hold true, as is amply demonstrated in Fig.1. Here, we show the same document being imaged through two different imaging mediums. The image in Fig.1(a) is acquired using a scanner and the the one shown in Fig.1(b) is capured using a camera. Figures.1(c) and 1(d) show the histograms of the input images (a) and (b) respectively. Under these circumstances, binarization of the images can hardly be achieved using a global threshold, as proposed by Otsu [1]. Liu *et. al.* [2] proposed an Object Attribute Thresholding Algorithm. In this algorithm, the text is assumed to occupy a

separable range in the gray-scale histogram. This assumption holds true, mostly, for scanned documents images. However, it can observed from Fig. 1(d) that such an assumption is ill-posed for camera captured images.

Burian *et. al.* [3] proposed an adaptive scheme to find the threshold for each pixel in the image. They use (i) the moving average, (ii) the maximum and (iii) the minimum of the currect moving average to determine the threshold at each pixel. The image is treated as a continuous stream of pixels and the threshold, $T_{(i+1)}$, at the $(i+1)^{th}$ pixel is calculated as:

$$T_{(i+1)} = \frac{M_i}{n} \frac{Max_i - Min_i + p_{(i+1)}}{4Max\{Max_i, p_{(i+1)}\}}$$

where, M_i is the current moving average of the last n pixels. Max_i and Min_i are the maximum and minimum, respectively, of the last n encountered pixel values. $p_{(i+1)}$ is the $(i+1)^{th}$ pixel value. While evaluating the $T_{(i+1)}$, the threshold for the pixel present in the previous row and the same column, is also taken into consideration. This accounts for the global properties of the image. However, we observed that the images which are acquired at a lower resolution, VGA resolution or less, contain a rapid transition in the intensity values of the neighboring pixels. Therefore, a threshold calculated on the basis of the previous n pixels do not efficiently binarize such images.

In the present work, we assume the following properties to be present in the images, acquired using a camera mounted on a mobile phone.

- The camera captured images could be non-bimodal in nature.
- The camera captured images could be of VGA resolution or less.
- The camera captured images have a large variation in intensities of pixel values, in a neighborhood, coming from an otherwise homogeneous surface.
- The images captured have the text portions of the image in focus of the lens of the camera.
- the surface, representing the background, is not glazed and is of uniform color.

With these assumptions, we now proceed to propose four different binarization techniques, described in section 3.

2. DATA DESCRIPTION

A camera mounted on a Nokia-6235 mobile phone was used for acquiring about 400 images. These images are stored as 24-bit depth color images in JPEG format. Most of these images contain text, both handwritten and printed, of variable font size and style. So, depending on the size of the font, the text content of the image varies. The images have been captured within a distance of one meter from the surface. For the indoor environment, the room is adequately lighted to simulate the study room condition.

3. METHODS OF BINARIZATION

We have already demonstrated that the images captured on mobile phone cameras may not be bimodal (see Fig.1). Hence, binarization of such images using a global thresholds yields unsatisfactory results with undesirable binary blobs. In this paper, we present a comparative study of four different techniques on the binarization of the above mentioned images.

Since the captured images are stored in color format, we first merge the RGB planes of the image to generate the intensity image and discard the color information. The conversion of the 24 bit color images to 8-bit gray is achieved by averaging the red, blue and green color components for each pixel. The grayscale value of a pixel p_{gray} is evaluated from the three red, green and blue color components.

$$p_{gray} = (p_r + p_b + p_g)/3$$

where p_r, p_b, p_g are the red,blue and green components of a pixel.

A. Global Threshold

The variation in the intensity levels of pixel values of the otherwise homogenous background makes the use of a global threshold for the binarization of the image almost impossible. However, if this variation is compensated for, a global threshold would be sufficient to binarize the image with a minimal loss of significant information. Here, we propose a technique to compensate for the variation in the background pixel values.

The image, $\mathcal{I}$, is down-sampled by a factor x to generate the image $\mathcal{I}_{\mathcal{D}}$. The value of x is chosen to be little greater than the width of the character limbs. This ensures the removal of most of the text elements from the down-sampled image. $\mathcal{I}_{\mathcal{D}}$ is up-sampled and, subsequently, low-pass filtered using a mask of size $(2y+1) \times (2y+1)$ to generate $\hat{\mathcal{I}}$ which is our estimate of the background. the value of y is appropriately chosen to lie between x and $2x$. We now subtract the $\hat{\mathcal{I}}$ from the original image, $\mathcal{I}$. This eliminates the variation in intensity of the background. The values of x and y have a significant impact on the final image quality and vary with the resolution of the image. For our experiments, we have taken $x = 4$ and $y = 6$. Now, the image can be binarized using a global threshold. This threshold is set at 95% of the mean value of the subtracted image.

Binarized results obtained using this algorithm can be seen in Fig. 4. The computational complexity is as shown in table 1 where it is compared with the other three techniques.

B. Local Thresholds Determined using Pixel Neighbourhoods

The rapid variation in intensity seen in these images makes the use of local thresholds an ideal choice. The mask is chosen carefully to take into account only the local properties of a pixel in the image. We begin by finding the average pixel value Ap, of the image.

$$Ap = (max + min)/2$$

where max and min are the maximum and minimum occuring values, respectively, in the image. A threshold is found for each pixel in the image by considering the histogram of the 5×5 neighbourhood around the pixel.

TABLE 1: THE COMPUTATIONAL COMPLEXITY OF THE 4 METHODS

Method	Computations
Global Threshold	4MN+ MN(2y+1)^2
Local Threshold using Pixel Neighbourhoods	294MN
Local Thresolds using Blocks	3MN
Local Thresholds using Bilinear Interpolation	11MN

$$threshold = (index1 + index2)/2$$

where $index1$ and $index2$ are the indices of the pixel values with maximum frequency of occurance in the intervals [0, Ap] and [Ap+1, 255], respectively.

This scheme requires more computation than the previous method. The time required to binarize an image of resolution $480X640$ can be up to 4 times longer than the Global Threshold method. The binarized results are shown in Fig. 3(g), Fig 3(h) and Fig. 3(i). A systematic evaluation of the computational complexities is performed and the results are displayed in table 1. The computational requirement at various stages in the algorithm is described in the Appendix.

C. Local Thresholds Determined by Dividing the Image into Blocks

Niblack [4] proposed a scheme where the threshold adapts to the characteristics of the image at a pixel location. The threshold $T_{(i,j)}$ at pixel location (i, j), is evaluated from the mean (μ) and standard deviation (σ) of a windowed region around this pixel.

$$T(i,j) \;=\; \mu(i,j) \;+\; k\;\sigma(i,j); \qquad k = 0.2$$

Rais *et. al.* [5] proposed a modification to the above method by making the k-factor dependent on the image characteristics, both global and local. The value of k at any pixel location (i, j) is calculated as:

$$k \;=\; -0.3\frac{m_g \times \sigma_g \;-\; m_i(i,j) \times \sigma_i(i,j)}{max\;[m_g \times \sigma_g,\; m_i(i,j) \times \sigma_i(i,j)]}$$

where m_g and σ_g are the global mean and standard deviation of the image, respectively. $m_i(i, j)$ and $\sigma_i(i, j)$ are the mean and standard deviation of the windowed portion of the image, respectively. The calculation of each threshold for a window of size NXN is of the order $O(N^2)$.

The adaptive threshold selection procedure is a computationally heavy process. Since, we assume a homogenous, iso-color background, we can consider the statistical properties of the image to remain stationary within a small region. This way, we still exploit the local variations but at a relatively smaller cost. So, we divide the image into non-overlapping blocks. The number of blocks and the size of the blocks is not fixed and is derived from the image size. First, the image is divided into 10 horizontal-blocks. The number of vertical blocks is decided from the aspect ratio of the image.

$$ver = \frac{10 * height}{width}$$

where, ver is the number of blocks in the vertical direction. Therefore, $\frac{height}{ver}$ and $\frac{width}{hor}$ will be the block height and width respectively. The mean, μ, and standard deviation, σ, is found for each block. While the standard deviation is a measure of the diversity of pixel values found in a given block, μ represents the intensity of the pixel values. Higher standard deviations are therefore, indicative of the presence of text in a block.

For lower values of the standard deviation, $\sigma <= 15$, the block is assumed to contain only background or foreground pixels. Under this circumstance, whenever, $\mu < 130$, we assume that the block contains only black pixels and, hence, each pixel in the block is substituted by a zero. For μ taking values higher than 130, we substitute all pixels by 255. When $\sigma > 15$, the block is binarized using a threshold, calculated as shown below.

$$threshold = 0.875 * \mu$$

The results obtained using this method can be seen in Fig 3(j), Fig 3(k) and Fig 3(l). The computational complexity is described in the Appendix.

D. Bilinear Interpolation to Determine Local Thresholds

The use of a single threshold for an entire block, leads to noise along the edges of the blocks and the binarized image will contain blocky artifacts. This noise occurs due to the sudden change in the threshold from one block to another. Although this rarely leads to a complete loss of information, the clarity of the image is affected. Any kind of variation in the choice of the size of these blocks does not reduce the occurrence of these noisy artifacts. However, it increases the computational complexity. Bilinear interpolation provides a smooth change in the threshold values, as we move from one block to another. Thus, reducing the development of blocky artifacts.

In the Block Analysis method, the image is divided into blocks and the mean of each block is evaluated. 90% of the mean value is set to be the thereshold for binarization at the center of each block. For all other pixels, we calculate the threshold, using a bilinear interpolation technique, from the four near block-centers.

The threshold at pixel p is,

$$threshold = \frac{t1 * A4 + t2 * A3 + t4 * A2 + t5 * A1}{A1 + A2 + A3 + A4}$$

where, $A1$, $A2$, $A3$ and $A4$ are the areas of the sub-blocks, as shown in the figure.

The binarized images obtained using this method are shown in Figs. 3(m), 3(n) and 3(o). The computational complexity is as described in the Appendix.

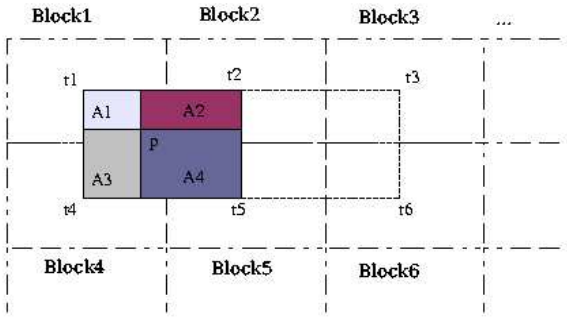

Fig. 2: Bilinear Interpolation is used to find threshold at pixel p, given thresholds $t1$, $t2$, $t4$ and $t5$ which are the centers of the Block1, Block2, Block4 and Block5 respectively.

4. LOCALIZATION OF TEXT

Since the background details are of no significance, localization of the text offers savings in size and costs incurred in tranmission. This is accomplished using information obtained from the horizontal project profile (HPP) and the vertical projection profile (VPP). The HPP and VPP are normalized with respect to the width and height respectively. Values of over 0.97 in the HPP indicate rows that do not contain any information and these rows can be deleted from the image. Similarly, the empty columns are also removed.

In images of resolution $480X640$ or higher, the use of a threshold of 0.97, efficiently removes empty rows and columns. But in images of lowers resolutions, this could result in the loss of regions of text. In such cases, the projection profiles are not normalized and the threshold is set at $(width - x)$ and $(height - x)$ for the removal of rows and columns respectively. x is chosen such that it is less than the average width of a character limb.

5. EXPERIMENTAL RESULTS

The effectiveness of the proposed methods was tested on about 400 images of varying resolution. The 400 images included images of handwritten and printed text on light and dark backgrounds. The algorithms were implemented in C++. The results of the four presented schemes are shown in Fig 3. Figures 3(a) and 3(b) are captured images with resolutions of $480X640$, while Fig. 3(c) is a captured image of resolution $176X144$.

Figures 3(d), 3(e) and 3(f) show the results of the Global Threshold method for the input images shown in Figures 3(a), 3(b) and 3(c) respectively. At first glance these images appear noisy, but this method binarizes images with minimal loss of information. All small fonts and blurred areas of text appear clear and legible in the binarized images obtained using this method. However, the speckling makes the localization of the text difficult.

Figures 3(g), 3(h) and 3(i) show the results of the method using Pixel Neighborhoods for the images shown in Figures 3(a), 3(b) and 3(c) respectively. In Fig 3(b), an image captured from a newspaper, we see details of the text on the reverse side of the page as well, which is due to the paper being extremely thin. The darkening is seen in 3(d) is in fact, the binarization of the fine details of the image.

Figures 3(j), (k) and (l) show the results of the Block Analysis technique for the images shown in Figures 3(a), 3(b) and 3(c) respectively. There is no loss of information in any of the images, but Fig 3(j) shows some blocky artifacts produced by the constant thresholds used in the blocks. This type of artifact is common in the binarized images obtained using this method. However, this scheme also produces clear images with minimal noise, as can be seen in 3(k) and 3(l). Even the slightly blurred sections of text in 3(b) is legible in the binarized image.

Figures 3(m), 3(n) and 3(o) show the results of the method utilizing Bilinear Interpolation of the block thresholds. This algorithm produces clear results of images captured in all conditions. In Fig. 3(o), the light handwritten text from the top of the Fig. 3(c) has also been reproduced. It is not very computationally intensive and is ideal for an application meant to run on mobile phones. However, fixing the central thresholds of the blocks along the edges can be difficult and this can sometimes lead to the loss of some information along the edges of the image.

TABLE 2: QUANTITATIVE PERFORMANCE EVALUATION OF THE 3 ADAPTIVE THRESHOLDING METHODS

Total Number of Characters in 15 Images = 356

	Pixel Neighbourhood	Block Analysis	Bilinear Interpolation
Detected Elements	376	383	368
Correct Elements	346	351	355
Precision Rate	92.02	91.64	96.46
Recall Rate	97.19	98.59	99.72

As the images contain only text, the performance can be evaluated using the number of characters in the image. It is determined on the basis of 2 factors: (i) the ability of the algorithm to binarize the image without any loss of information (ii) the occurrence of extraneous binary blobs (noise) in the image, which reduces the visual appeal of the image. Two parameters, defined below, are considered to evalute the performance of the algorithms:

$$Precision\ Rate = \frac{No.\ of\ correctly\ detected\ elements}{No.\ of\ detected\ elements}$$

$$Recall\ Rate = \frac{No.\ of\ correctly\ detected\ elements}{Expected\ number\ of\ elements}$$

The speckling effect seen in the images obtained using Global Binarization makes this type of evaluation impossible. Although this method produces visually appealing images, the extraction of the characters from these images is difficult. Hence, we have not quantitatively evaluated the performance of this method. Table 2 shows the quantitative performance

of the 3 methods on 15 images containing a total of 356 characters. The Precision rate is a measure of the occurrence of false positives while the Recall Rate is a measure of the loss of information. As shown in Table 2, the Pixel Neighborhood and Block Analysis techniques show the occasional loss of information, while the method that uses Bilinear Interpolation minimizes the loss of information.

6. Conclusion & Discussion

The images under consideration predominantly contain text, handwritten and/or printed, and is intended for transmission over MMS. As the output images are for human consumption, small amounts of noise and breaking and joining of the characters are tolerated.

The algorithms we have proposed are independent of the imaging resolution and lighting variation. They are also independent of the nature of the surface, such as cloth and glass, on which the text is present. Even the images of newspaper clippings, where both the quality of paper and print are relatively poor, produce clear results. Though we assumed that the images are well focused, we included some slightly out of focus but, legible images. Such images also produce outputs which are qualitatively at par with well focused images.

The quantitative comparison of the three methods that use local adaptive thresholds has been described in the previous section. We saw that the method that uses the bilinear interpolation technique (described in section 3.D) delivers better quality outputs with minimal loss of information. It is computationally efficient as well. The quality factor of the images produced by all four techniques was also evaluated based on the visual assessment of 5 people.

The use of HPP and VPP to localize the presence of text locations in the input image has helped in reduction in the number of bits required to represent the image. However, this works best with images where the text alignment is either along the horizontal or vertical direction. Moreover, we have assumed that the text is darker than the background. So, in cases where the text is brighter than the background (such as writings on a blackboard), the image needs to be inverted.

Since, the algorithm is meant to be implemented in a device such as a mobile phone, with very low computational resources, we aimed to minimize the dynamic memory requirement and the computational complexity of the algorithms. In the present day scenario, the processors mounted on most mobile phones have a clock speed of about 250 MHz and have about 100 KBytes of RAM. Our intention was to develop an algorithm which would deliver quality results in less than 2 seconds. Moreover, we also intended to minimize the storage size of the output image so that the cost of tranmitting the data is minimum. We met these specifications using the binarization technique described in section 3.D, followed by the size reduction scheme presented in section 4.

7. Acknowledgement

The authors would like to express their heart-felt gratitude to Sasken Communications Ltd. for funding the research.

8. Appendix: Computational Complexity of Binarization Schemes

For an image of size MXN, the computations required at various stages of the four presented schemes, for binarization, is described below.

Global Threshold

Down-sampling followed by up-sampling	MN
Filtering with mask of size $(2y+1)*(2y+1)$	$MN(2y+1)^2$
Image subtraction	MN
Finding mean of the image	MN
Binarization of image	MN
Total	$4MN + MN(2y+1)^2$
If y=6	$\Rightarrow 173MN$

Local Threshold Using Pixel Neighborhoods

To find Ap	MN
To find histogram of each pixel neighbourhood	36
To find threshold for each pixel	256
Binarization of image	MN
Total	$294MN$

Local Threshold Determined by Dividing the Image into Blocks

To find mean of each block	MN
To find standard deviation of each block	MN
Binarize each block	MN
Total	$3MN$

Local Thresholds Determined Using Bilinear Interpolation

Find thresholds for blocks	MN
Find a threshold for each pixel	9
Find thresholds for image	$9MN$
Binarize image	MN
Total	$11MN$

References

[1] N. Otsu, "A Threshold Selection Method from Gray Level Histogram", *IEEE Trans. SMC-9, pp.62-66,1979.*

[2] Ying Liu, Richard Fenrich, and Sargur N Srihari "An Object Attribute Thresholding Algorithm for Document Image Binarization"*Proc. 2^{nd} Intl. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan, pp. 278-281 October, 1993.*

[3] Adrian Burian, Markku Vehniläinen, Mejdi Trimeche, and Jukka Saarinen "Document Image Binarizatzon Using Camera Device in Mobile Phones", *Proc. Intl. Conf. on Image Processing, 2005, (ICIP 2005), Volume 2, pp:546-548.*

[4] W. Niblack, "An Introduction to Digital Image Processing ", pgs: 115-116, Englewood Cliffs, NJ, Prentice Hall, 1986.

[5] Naveed Bin Rais, M.Shehzad Hanif and Imtiaz A.Taj "Adaptive Thresholding Technique for Document Image Analysis ", *Proc. Intl. Multitopic Conference, 2004, (INMIC 2004), pp:61-66.*

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Fig. 3: (a) and (b) are captured images with resolution $480X640$, (c) is a captured image with reslution $176X144$; (d) ,(e) and (f) are binarized images using Global Threshold Method; (g), (h) and (i) are binarized images using Pixel Neighborhood Method; (j), (k) and (l) are binarized images using Block Analysis; (m), (n) and (o) are binarized images using Bilinear Interpolation Method