# A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras

Daniel Marques Oliveira and Rafael Dueire Lins
*Universidade Federal de Pernambuco – Recife – PE – Brazil*
*{daniel.moliveira, rdl}@ufpe.br*

## Abstract

*Photo documents, documents digitized with portable digital cameras, often are affected by non-uniform shading. This paper proposes a new method to remove the shade of document images captured with digital cameras followed by a new binarization algorithm. This method is able to automatically work with images of different resolutions and lighting patterns without any parameter adjustment. The proposed method was tested with 300 color camera documents, 20 synthetic images with ground truth lighting pattern and grayscale images of the dataset of CBDAR 2007 dewarping contest. The results show that the new algorithm proposed here works in a wide variety of scenarios and keeps the paper texture of the original document.*

## 1. Introduction

Use of portable digital cameras for the digitalization of documents is a new research area. Such images, called *photo documents*, often have non-uniform shading, perspective distortion and blur. Fixing these distortions improves image readability and increases OCR accuracy rates. A new method for shading removal and binarization of photo documents is proposed in this paper.

For shade removal, Tan et al [12] proposes a scheme for scanned document images when the shading appears as a result of documents not being perfectly positioned on the scanner flatbed, due to book binding for instance. The variation in the illumination pattern can be modeled as a light source, which is found by a Hough transform. Reference [13] identifies document boundaries and assumes that the document has a rectangular shape; this *a priori* knowledge is used to remove image warp and shading.

Several papers in the literature attempt to remove shading by using 3D model obtained with special setup such as in [1], these alternatives are not portable.

Non-uniform shading requires adaptative binarization, first approaches use a sliding window around every image pixel to measure pixel threshold. Niblack's approach [8] uses equation (1) to calculate the pixel threshold, where $m$ and $s$ are the mean and the standard deviation of pixels in the window around the current pixel, and $k$ is a constant value in [-1,0). It was improved by Sauvola [10] using equation (2) with same $m$ and $s$; constant $k$ set in (0,1] which controls the standard deviation contribution to the threshold; constant $R\ is$ set according to image contrast.

$$T = m + k \times s \tag{1}$$

$$T = m \times \left[1 + k\left(\frac{s}{R} - 1\right)\right] \tag{2}$$

Reference [4] estimates the document background by a polynomial surface approximation of grayscale images. The surface is used to remove shading followed by binarization using a global threshold. This idea is also used in method proposed herein, with the advantage of shading removal of true color images.

## 2. The proposed method

For document images, two assumptions can be made [12]: the book surface has lambertian reflection (i.e. the specular index is too low) and it has non-uniform albedo distribution. With these assumptions, the image of the paper background has a constant value of intensity ($I_{us}$), independent of the location of the viewer if illuminated by the same amount of light. As the image is the result of reflected light ($I_o$) at arbitrary levels, one can express light variance by ratio (3):

$$\frac{I_o(i,x,y)}{I_{us}(i,x,y)} = L(i,x,y);\ \ i\ \in \{r,g,b\} \tag{3}$$

The goal of the proposed method is to calculate the value of $I_{us}(i,x,y)$ for all pixels in an image by assuming that the document background (paper) has one predominant color which is not uniform in the image due to illumination variance.

### 2.1. Narrow Gaussian Blocks

When observing a small area of the document one may notice that when this area belongs to the document background the histogram on all components have a "Narrow Gaussian shape" as depicted in the Figure 1.a. As other elements of the document are present, the histogram starts to be spread as can be seen on Figure

1.b. The relative area of Gaussians distributions in the interval $[-\Delta\sigma + \mu, +\Delta\sigma + \mu]$ is shown in Figure 2.

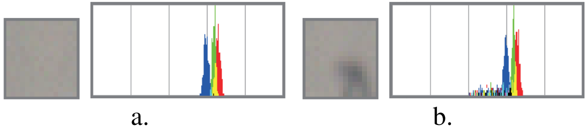


**Figure 1.** Small blocks and their histograms: Background block (a); Block with “A” letter (b)

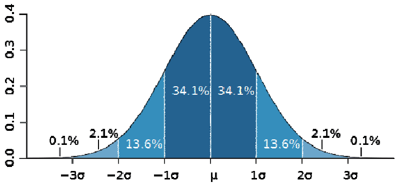
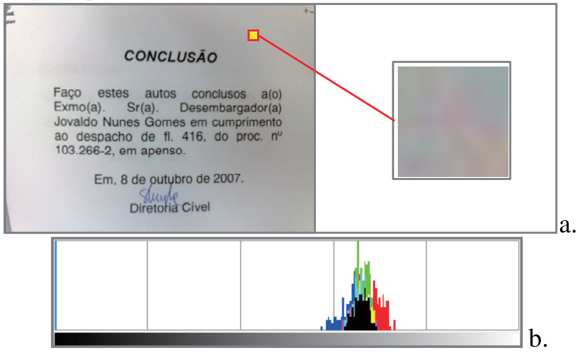


**Figure 2.** Gaussian distribution area relative to σ [16]

The aim here is to identify which areas belong to the document background by verifying if it has a histogram close to a Narrow Gaussian form. The choice of a block size must consider:

- There should be enough pixels for a statistical representation.
- If it is too large, will have no uniform color
- Its size should be less than the spacing between lines

A block area of 15x15 pixels was proven to be enough for all the test images used. In order to estimate the parameters for a Narrow Gaussian block (NGB), an analysis was done with Nokia N95 image (Figure 3.a), which was considered as the worst case due to the presence of blur and other noises (salt-and-pepper, Bayern pattern smoothness, etc).



CONCLUSÃO

Faço estes autos conclusos a(o) Exmo(a). Sr(a). Desembargador(a) Jovaldo Nunes Gomes em cumprimento ao despacho de fl. 416, do proc. nº 103.266-2, em apenso.

Em, 8 de outubro de 2007.

Diretoria Cível

**Figure 3.** Worst case scenario: whole image with highlighted block (a); histogram of the block (b)

The use of the mode rather than the mean for the Gaussian center is more straightforward as it is less affected by noisy values [5], thus modes of Figures 1.a and 1.b are the same.

Observing the worst case image, it was empirically found that the uniform areas have at least 75% of pixels within the area defined by ±6 pixels around the component mode (with component values in [0,255]). Then a process to identify a NGB block is outlined as follows.

1. Compute histogram of each component value
2. Identify the histogram modes of each component: $mode_r$, $mode_g$ and $mode_b$.
3. Count the number of pixels in the interval (mode-6, mode+6). If this count in all components is more than 75% of the number of pixel in the block, then the area is identified as NGB.
4. The background value is set to be equal to the value of the pixel with the smallest value using eq. (4), which is an approximation to the Euclidean distance in the RGB space.

$$D(c) \cong |c_r - mode_r| + |c_g - mode_g| + |c_b - mode_b| \quad (4)$$

## 2.2. Defining block area

Ideally, every image pixel should be evaluated to identify if its surrounding pixels have a uniform color. This can be done by a sliding window around every pixel and using the procedure in section 2.1, although this is not feasible. On the other hand, the image could be divided into non-overlapping blocks of sizes 15x15 pixels. The main drawback of this approach is that the text spacing is between 15 and 25 pixels for high resolution documents, and blocks may not located, necessarily, in the middle of text lines as illustrated on Figure 4.a.

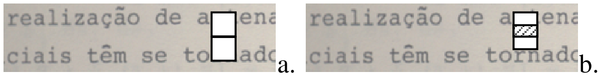


**Figure 4.** Example of location of blocks

A third approach is proposed by dividing the image into 5x5 blocks. An “expanded” 15x15 area centered on each 5x5 block is used for histogram calculation as depicted in Figure 5. This increases the probability of the expanded block to be located between text characters as shown on Figure 4.b. The computational complexity is$9 \times W \times H$, as each pixel is read $(3 \times 3)$ times due to every pixel to participate in 9 blocks.

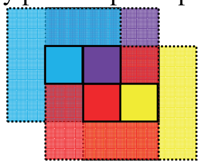


**Figure 5.** Four 5x5 pixels center boxes with a window of size 15x15 pixels

Then, to find the NGBs the following steps are executed:

1. Split the image into small blocks of 5x5 pixels.
2. Identify which blocks have expanded version with Narrow Gaussian histograms (as described in 2.1).

## 2.3. Finding background blocks

Now, the gathered information is a set of NGBs. The aim is to split into NGBs regions with little color variation; one of these will be the document background as will be described in this section.

The main criterion is to put together pairs of NGBs neighbors with background component-wise difference of its color less than a threshold (*Tneighbor*) into the same region. If this threshold is set too low it can yield a false-negative classification of similar neighbors. Otherwise, the threshold is set if too high it blurred text areas can be set as part of document background. This work assumed *Tneighbor* set to 5. Considering the component values in [0,255], the flood-fill algorithm with the whole procedure is showed bellow:

```
B: Set of all NGBs
Q: Queue used by to flood-fill
N: Closest NGBs used
Tneighbor: Threshold for BDmax value
BDmax(b1,b2): max(|b1.red-b2.red|,|b1.green-
   b2.green|,|b1.blue-b2.blue|)
for every element b of B do
   If element b was not visited then
        Q <= {b};
        while Q is not empty
           q = first(Q);
           q visited state is set true;
           Q = Q - {q};
           N <= NGBs of every 8-direction
              closest to q
           Remove visited NGBs from N
           For every element n of N
              If (BDmax(n,q) < Tneighbor) then
                Q = Q + {n};
              end if
           end for
        end while
  end if
end for
```

Once the regions are identified, it is needed to select the one that represents the document background. As no assumptions are made about the image structure, two criteria are used to select the region as the document background: the percentage of NGBs of the region; quantity of NGBs in the center of the image, which is defined as a rectangle with dimensions $\left(\frac{img\_width}{3},\frac{img\_height}{3}\right)$ centered on the intersection of the diagonals of the image.

If there exists a region with more than 15% of all NGBs and most blocks in the rectangle of image center, this region is set as document background seed (DBS). Otherwise, the region with most NGBs is set as the DBS.

The computational complexity for the first part was found to be proportional to the number of blocks as each NGB are visited exactly one time. Considering memory usage, in the worst case the queue contains all the blocks in the image. A more efficient flood-fill and component labeling procedures could also be used but they are more complex to implement, an example can be found in [7]. Region statistics for the second part can be computed while executing pseudo-code presented, which has a computational complexity proportional to the number of blocks.

## 2.4 Interpolation of non-background blocks

Once the DBS is found, the color values of the non-DBS blocks must be estimated. These NGBs are arbitrarily located, so a classical interpolation method (bilinear, bicubic etc) cannot be used.

A new approach is proposed similar to iterative dilation; where at every iteration the background regions are expanded, the color of the blocks are computed by the weighted average of its neighboring blocks. This process stops when all blocks have their color defined. The pseudo-code with this process is presented below; Figure 6 illustrates an example of it.

```
BGs: set of background with color set
B: set of currently expanding blocks
Q: set of expanding blocks of next iteration
BGs <= all blocks in DBS
//initial fill of B set
for every element n of BGs do
   for every m 8-neighboor of n and not in BGs
        B <= B + {m};
   end for
end for
//algorithm iteration
do
   Q <= empty set;
   for every element b of B do
        Set color value of b to weighted sum
          of BGs 8-neighboors, where the weight
          equals neighbor distance inverse;
        For every n that are 8-neighboor of b
            and is not in Q or BGs
              Q <= Q + {n};
        end for
  end for
  BGs = BGs + B; B <= Q;
until B is empty
```

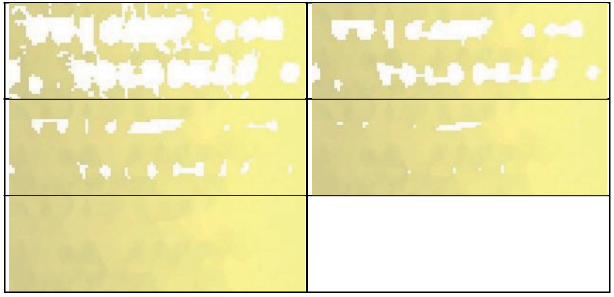


**Figure 6.** Interpolation iterations process

Observe that each non-DBS block is filled once, thus the computational complexity is proportional to the number of blocks.

Figure 7.a presents a synthetic image, Figure 7.b its lighting pattern ground truth. The result of the processing described in section 2.3 is presented in Figure 7.c, with pure white blocks as non-DBS blocks; Figure 7.d shows the predicted background, note that the background is estimated for the whole image.

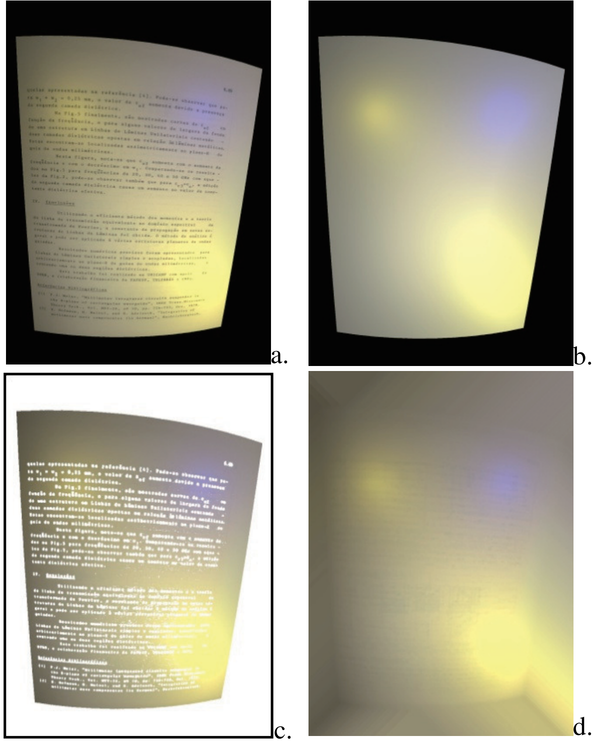


**Figure 7.** Background estimation: synthetic image (a); ground truth (b); DBS (c); estimated (d)

## 3. Shading Removal

Observing equation (3), $I_{us}(i,x,y)$ values for the document background color should be constant for every RGB component, hence $I_{us}(i,x,y) = I_{us}(i)$. It is estimated by calculating the component-wise mean of the DBS blocks and locating the closest DBS block to this mean.

The only unknown value is $I_{us}(i,x,y)$ for every pixel in the image. It is calculated using equation (5) or (6), where $BG_o(i,x,y)$ and $BG_{us}(i)$, denotes $I_o(i,x,y)$ and $I_{us}(i,x,y)$ of estimated document background, respectively; $I_{us}$, $I_o$, $BG_{us}$ and $BG_o$ are in [0,255].

Eq. (5) is only applied for the case when $I_o(i,x,y)/BG_o(i,x,y)$ is less than 1. Whenever greater, the ratio is difficult to represent as it is in (1,∞], negatives are used instead, so $\overline{I_o(\imath,x,y)}/\overline{BG_o(\imath,x,y)}$ is in [0,1). When $I_o(i,x,y) = BG_o(i,x,y)$, both eqs. (5) and (6) yields $BG_{us}(i)$, thus it is used in this case. No floating point is required as numerator may be computed first followed by an integer division.

$$I_{us}(i,x,y) = \frac{BG_{us}(i)}{BG_o(i,x,y)} I_o(i,x,y) \qquad (5)$$

$$I_{us}(i,x,y) = \overline{\left(\frac{\overline{BG_{us}(\imath)}}{\overline{BG_o(\imath,x,y)}}\overline{I_o(\imath,x,y)}\right)} \qquad (6)$$

$$\overline{C(\imath,x,y)} = 255 - C(i,x,y) \qquad (7)$$

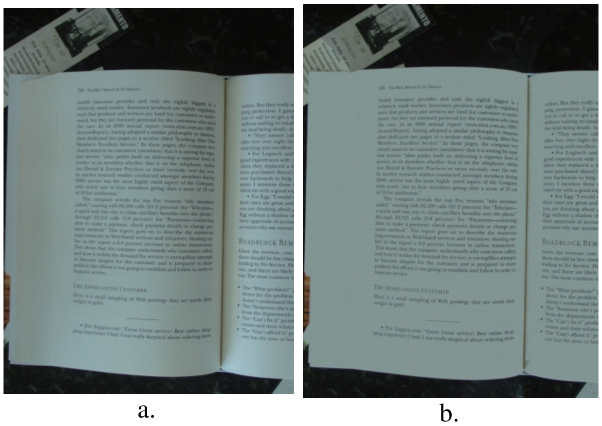


**Figure 8.** Shade removal: original image (a); result (b)

## 4. Document Binarization

Once shade is removed, binarization can be done by a global threshold approach [4]. Although, by applying it to the grayscale enhanced image purely yields poor results as camera documents may contain undesired objects around document disrupting the histogram, which is input for global threshold algorithms.

In Figure 7(c) the non-DBS blocks (in white) can be separated in two types: text blocks (TB) and image bordering blocks (IBB). Regarding these categories one may see that most TBs are surrounded by DBS blocks gathered on section 2.3. A process to identify IBBs is described in the pseudo-code below, and it can be summarized by a search starting at every block in the image border towards the opposite direction looking for a DBS block, until it is not found the intermediary blocks are set to be IBBs. Observe that the computational complexity in the worst case is proportional to the number blocks, but in the cases where the document borders are close to the image borders the complexity approaches the sum of the dimensions of the image.

```
I: Number of block columns in the image
J: Number of block lines in the image
B: bidimensional array of image blocks. B(0,0)
   means upper left block; B(I,J) lower right
Initially all blocks are NOT IBBs
for every i varying from 1 to I
   execute findBorderBlocks(i,0,right);
   execute findBorderBlocks(i,J,left);
end for
for every j varying from 1 to J
   execute findBorderBlocks(0,j,down);
   execute findBorderBlocks(I,j,up);
end for
procedure findBorderBlocks(i, j, dir)
   //while stops when first DBS block is found
   while ((B(i,j) is not DBS block)
      B(i,j) is set as IBB
      //move i or j to given direction
      if dir = left then j = j - 1;
      if dir = right then j = j + 1;
      if dir = up then i = i - 1;
      if dir = down then i = i + 1;
   end while
end procedure
```

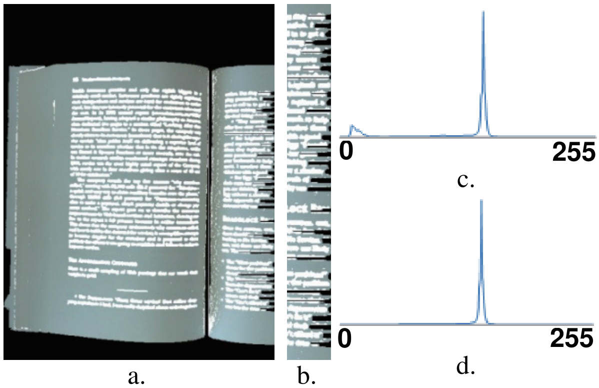
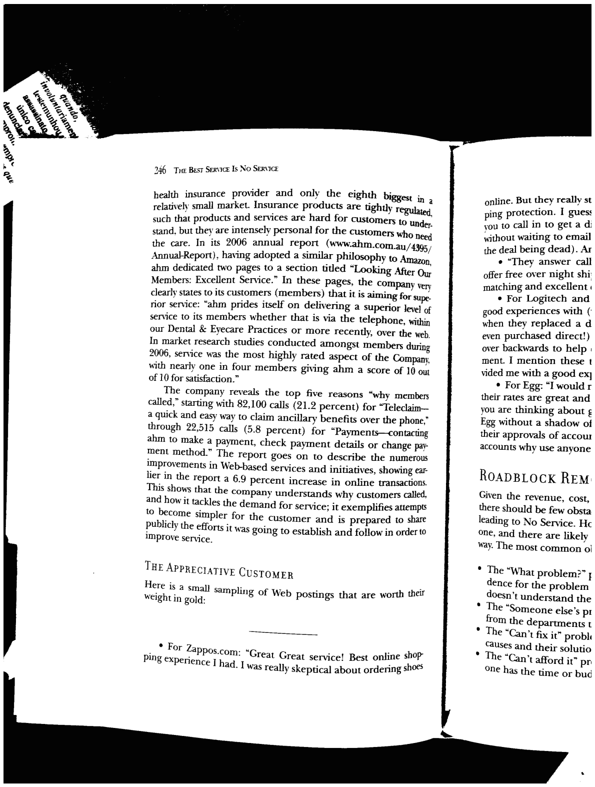


**Figure 9.** Rough boundary definition: block classification (a); some misclassification (b); enhanced image histogram (c); modified histogram (d)



246 The Best Service Is No Service

health insurance provider and only the eighth biggest in a relatively small market. Insurance products are tightly regulated, such that products and services are hard for customers to understand, but they are intensely personal for the customers who need the care. In its 2006 annual report (www.ahm.com.au/4395/Annual-Report), having adopted a similar philosophy to Amazon, ahm dedicated two pages to a section titled "Looking After Our Members: Excellent Service." In these pages, the company very clearly states to its customers (members) that it is aiming for superior service: "ahm prides itself on delivering a superior level of service to its members whether that is via the telephone, within our Dental & Eyecare Practices or more recently, over the web. In market research studies conducted amongst members during 2006, service was the most highly rated aspect of the Company, with nearly one in four members giving ahm a score of 10 out of 10 for satisfaction."

The company reveals the top five reasons "why members called," starting with 82,100 calls (21.2 percent) for "Teleclaim—a quick and easy way to claim ancillary benefits over the phone," through 22,515 calls (5.8 percent) for "Payments—contacting ahm to make a payment, check payment details or change payment method." The report goes on to describe the numerous improvements in Web-based services and initiatives, showing earlier in the report a 6.9 percent increase in online transactions. This shows that the company understands why customers called, and how it tackles the demand for service; it exemplifies attempts to become simpler for the customer and is prepared to share publicly the efforts it was going to establish and follow in order to improve service.

The Appreciative Customer

Here is a small sampling of Web postings that are worth their weight in gold:

- For Zappos.com: "Great Great service! Best online shopping experience I had. I was really skeptical about ordering shoes

online. But they really st
ping protection. I guess
you to call in to get a d
without waiting to email
the deal being dead). Ar
• "They answer call
offer free over night shi
matching and excellent
• For Logitech and
good experiences with (
when they replaced a d
even purchased direct!)
over backwards to help
ment. I mention these t
vided me with a good ex
• For Egg: "I would r
their rates are great and
you are thinking about
Egg without a shadow of
their approvals of accou
accounts why use anyone

Roadblock Rem

Given the revenue, cost,
there should be few obsta
leading to No Service. Ho
one, and there are likely
way. The most common o

- The "What problem?"
dence for the problem
doesn't understand the
- The "Someone else's p
from the departments
- The "Can't fix it" probl
causes and their solutio
- The "Can't afford it" pr
one has the time or bud

**Figure 10.** Proposed binarization using Otsu's

Figure 9.a shows the rough estimation with right border details on Figure 9.b, where IBBs are in black, TBs are in white and DBS blocks with its corresponding color. It is observable that the boundary estimation can be set to some text blocks as image border, but it does not affect the global threshold performance as this fact is not statistically significant. Figure 9.c shows the histogram of the grayscale version of the whole enhanced image showed in Figure 8.b, where Figure 9.d shows the histogram without IBBs. Figure 10 shows Figure 8.a's binarized version.

## 5. Results

To compare the method proposed here, Savoula [10] approach was implemented with [2] optimizations. The memory cost is measured by the second moment of the whole image with two 64-bit arrays of size W*H are allocated at once. Parameters were adjusted for the test images of this work yielding to a 21x21 window, k = 0.2 and 0.5 and R set to 128.

The method presented in reference [4] could not be implemented due to its large computational time of 5 Mpixel images polynomial surface calculation, with 5 million points interpolation.

The complexity of the proposed work was outlined in all sections to be proportional to the image dimensions or the number of blocks. The latter depends on the block area and its expansion factor. Needed memory for the execution is at least $(3 \times W \times H + k \times W \times H/block\_area)$ bytes: one 24-bit array for the enhanced image; an array of size $W \times H/block\_area$ with block information and queue lists with at most $W \times H/block\_area$ elements (where k block information size plus queue element, our implementation required k equal to 30).

Table 2 shows the processing time statistics using Java 1.5, DELL D531; Turion TL56 1.80GHz; 3Gb RAM on Windows Vista Business. The first column presents mean and standard deviation of images sizes in Mpixels followed by the total processing time average in ms. Other columns show the percentages of the execution times for each part; Section 3 part and the modified histogram computation were implemented together. Times do not include file loading and screen refresh. Floating points were only used on global thresholds, as section 2.4 neighbor distances can only be 1 or $\sqrt{2}$, an integer approximation was used.

**Table 2 – Processing time statistics**

| | Size | Time (ms) | Sec. 2.2 | Sec. 2.3 | Sec. 2.4 | Sec. 3 + modified histo. | Otsu |
|---|---|---|---|---|---|---|---|
| Mean | 6.69 | 4325 | 69.3% | 3.9% | 2.6% | 23.4% | 0.8% |
| Std. | 1.82 | 1096 | 1.7% | 1.1% | 0.7% | 1.4% | 0.1% |

A visual evaluation was performed with 300 photo documents. Figure 11 shows that the results with both binarizations (b and c) had similar outcome. Figure 11.c shows shading completely removed. While Figure

12.a Savoula's algorithm does not binarizes properly where in 12.c new method does. Removed shade version is illustrated in Figure 12.c.

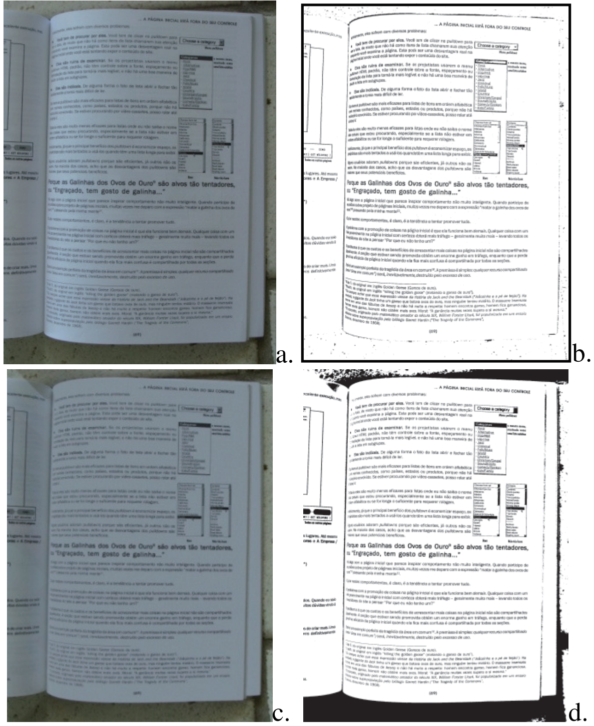


**Figure 11.** Comparison: original (a); Savoula (b); new shade removal (c); new binarization with Otsu (d)

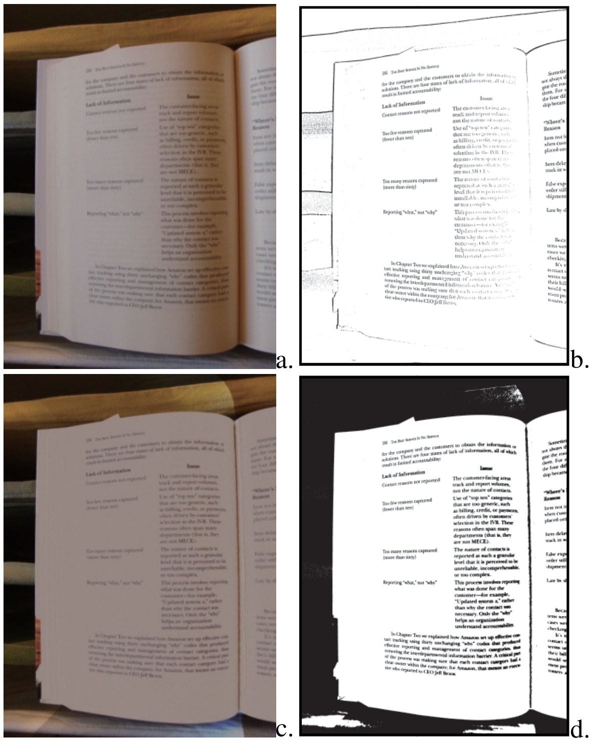


**Figure 12.** Processing results: original (a); Savoula (b); new shade removal (c); new binarization with Otsu (d)

A quantitative comparison of the estimated lighting pattern with its ground truth was done with 20 synthetic images covering different scenarios generated by Adobe After Effects CS4 [14]. The mean error was about 0.0566 and its standard deviation of 0.0024, with component values scaled in the interval [0,1]. Examples are shown in Figures 7 and 13.

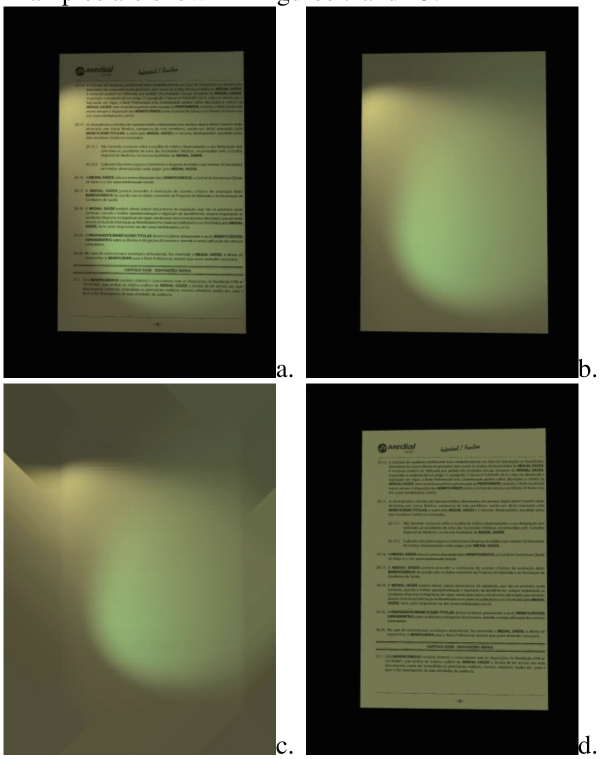


**Figure 13.** Background estimation: synthetic image (a); ground truth (b); predicted (c); shading removed (d)

The binarization was compared with CBDAR 2007 dewarping contest [3], as they contain a 102 grayscale images and the corresponding binary image which it was used as a ground truth, this comparison is more straightforward than OCR as no dewarping is done for the binary image. Three global threshold algorithms were used: Otsu's [9], Mello-Lins [6] and Silva et al [11]. All those algorithms were tested with the original and enhanced image with the modified histogram computation described in section 4. Savoula's [10] adaptative approach was also compared. Four different metrics were calculated, the same metrics were used in DIBCO 2009 contest [15]:

- Error (E) = (FP + FN)/TOTAL
- Recall (RC) = TP/(FN+TP)
- Precision (PR) = TP/(FP+TP)
- F-measure (FM) = 2*RC*PR/(RC+PR)

Where FP denotes false positives, FN false negatives, TP true positives, TN true negatives and TOTAL the number of pixels in the analyzed area.

Notice that for all presented metrics, greater values means better performance, except for error rate.

It was found that in CBDAR2007 dewarping dataset, 3 binary images (img_1179, img_1203, img_1235) has a region with uniform black color classified as white pixels, one example is shown in Figure 14.

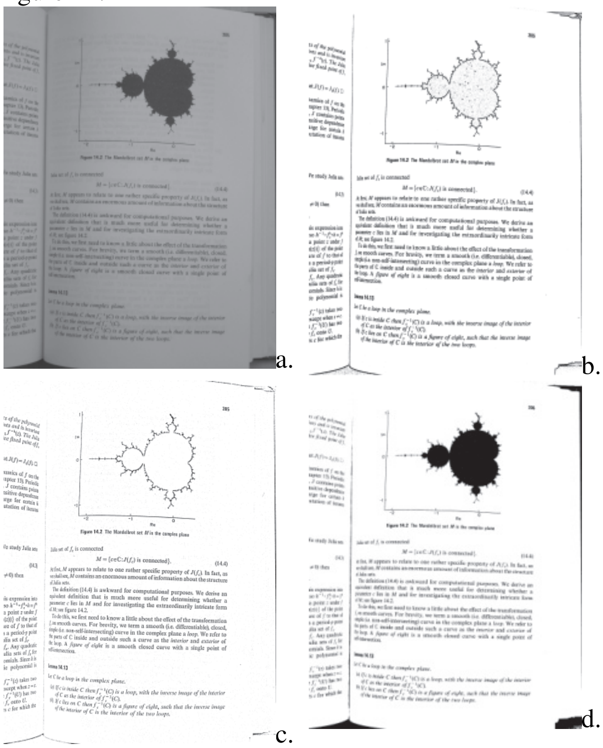


**Figure 14.** CBDAR2007 image img_1179: original (a); binarized (b); Savoula (c); proposed binarization with Otsu (d)

Another issue regards document boundary, both ground truth and Savoula's approach set to white pixels external to the document. These pixels are marked as "don't care" pixels, as is not important whether they is classified as white or black due to they not belonging to text area in the document. For the algorithm comparison, the documents were cropped to encompass only the text information.

Table 3 shows the statistical metrics for the global threshold algorithms for the original image and the proposed modification of the histogram calculation. The metric shows a performance improvement when the proposed approach is used. Among all global threshold methods, Otsu's showed to provide more consistent good results than the others.

Table 4 shows the metrics with Savoula's approach using original image. The performance of the proposed method yielded better results than Savoula's and handle a wider range of "scenarios" as the one showed in Figure 12. Another result can be seen in Figure 15.

**Table 3 – Comparison between binarization approaches using the original images and the proposed approach**

| | | Min | Max | Mean | Std. |
|---|---|---|---|---|---|
| Otsu (original) | E | 0.5% | 35.8% | 3.9% | 5.1% |
| | RC | 55.7% | 100.0% | 82.5% | 9.4% |
| | PR | 2.7% | 100.0% | 80.1% | 25.1% |
| | FM | 5.3% | 94.1% | 77.6% | 18.3% |
| Otsu (proposed) | E | 0.2% | 5.4% | 1.4% | 0.8% |
| | RC | 69.5% | 99.7% | 83.1% | 5.5% |
| | PR | 72.8% | 100.0% | 95.9% | 6.2% |
| | FM | 77.2% | 94.7% | 88.8% | 3.4% |
| Silva-et-al (original) | E | 0.6% | 9.0% | 3.2% | 1.6% |
| | RC | 28.7% | 100.0% | 69.4% | 16.0% |
| | PR | 11.6% | 100.0% | 86.1% | 19.9% |
| | FM | 20.7% | 91.4% | 73.5% | 13.1% |
| Silva-et-al (proposed) | E | 0.3% | 10.5% | 2.4% | 1.5% |
| | RC | 37.0% | 100.0% | 77.5% | 16.2% |
| | PR | 23.7% | 100.0% | 89.9% | 16.5% |
| | FM | 38.3% | 97.1% | 80.4% | 10.8% |
| Mello-lins (original) | E | 0.9% | 98.5% | 61.9% | 35.1% |
| | RC | 52.8% | 100.0% | 98.8% | 5.1% |
| | PR | 1.5% | 99.7% | 18.5% | 22.4% |
| | FM | 2.9% | 94.1% | 26.2% | 22.7% |
| Mello-lins (proposed) | E | 0.1% | 93.6% | 19.5% | 29.3% |
| | RC | 66.7% | 100.0% | 98.2% | 5.1% |
| | PR | 5.7% | 99.9% | 56.8% | 31.5% |
| | FM | 10.7% | 97.6% | 65.8% | 28.3% |

**Table 4 – Savoula's approach Metrics**

| | | Min | Max | Mean | Std. |
|---|---|---|---|---|---|
| Savoula k=0.5 | E | 17.2% | 47.4% | 23.2% | 5.0% |
| | RC | 38.6% | 96.3% | 70.8% | 13.6% |
| | PR | 2.5% | 34.6% | 18.9% | 7.0% |
| | FM | 4.9% | 48.9% | 29.2% | 9.4% |
| Savoula k=0.2 | E | 0.4% | 4.4% | 1.7% | 0.7% |
| | RC | 60.8% | 91.2% | 78.5% | 6.8% |
| | PR | 75.5% | 99.1% | 94.9% | 4.0% |
| | FM | 71.4% | 94.9% | 85.7% | 4.7% |

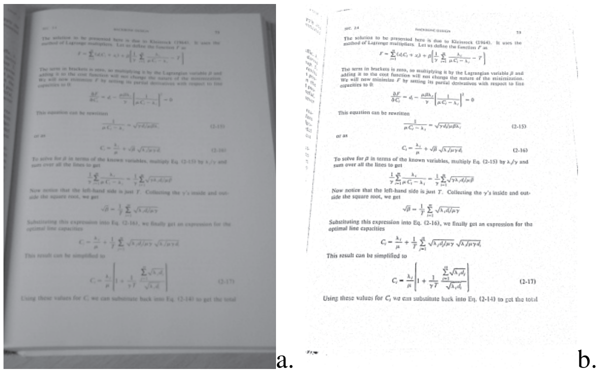


**Figure 15.** dsc00626 Binarization: original (a); Sauvola (b)

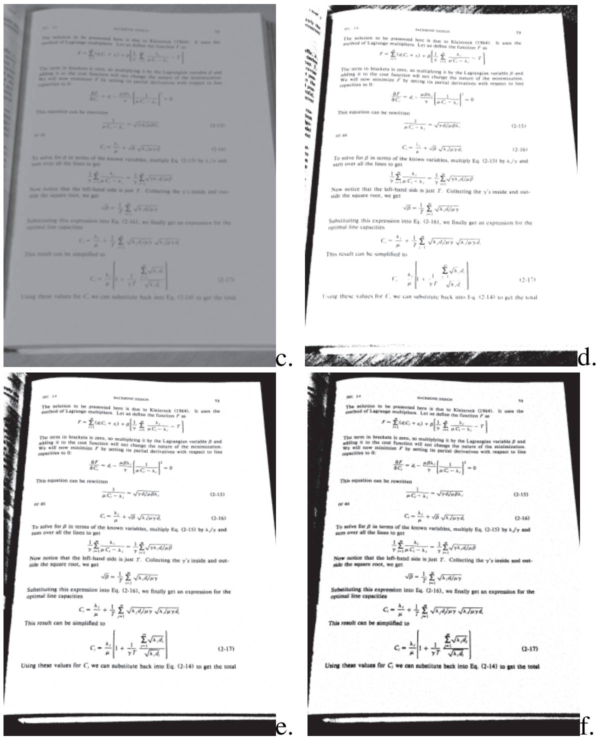


**Figure 15. (cont.)** without shading (c); new binarization with Otsu (d); new binarization with Mello-Lins (e); new binarization with Silva et al (f)

## 6. Conclusions and Lines for Further Work

This paper showed new schemes for color shading removal and binarization of documents captured with portable digital cameras. The performance of the shading removal algorithm was compared the lighting pattern with ground truth and with more than 300 images, which provided good and fast (does not require floating point operations) results for a wide variety of scenarios using only the captured image. The output of the binarization algorithm introduced was compared with one of the most used local algorithm with the same 300 images and CBDAR2007 dewarping dataset, it proves to cover more scenarios than Sauvola's method.

Some of the images in CBDAR 2007 dewarping dataset exhibit a light back-to-front interference [6][11] (bleeding). The binarization of photo documents with strong back-to-front interference is left a line for further work.

## 7. Acknowledgements

Research reported herein was partly sponsored by CNPq – Conselho Nacional de Pesquisas e Desenvolvimento Tecnológico, Brazil. We want also to thank Rafael Lellys for providing synthetic model.

## 8. References

[1] M. S. Brown et al. "Restoring 2D Content from Distorted Documents". IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. pp 1904-1916.

[2] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," in Document Recognition and Retrieval XV, San Jose, USA, Jan. 2008.

[3] F. Shafait and T. M. Breuel. "Document Image Dewarping Contest", 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, Brazil. Sep. 2007. pp 181-188.

[4] S. Lu and C. L. Tan. Binarization of Badly Illuminated Document Images through Shading Estimation and Compensation. ICDAR 2007, Volume 1, 23-26 Sept. 2007, Brazil, Curitiba, 2007, pp 312 – 316.

[5] R. A Maronna, D. R. Martin and V. J. Yohai. Robust Statistics: Theory and Methods. John Wiley & Sons, Ltd, England, 2006. ISBN: 0-470-01092-4.

[6] C. A. B. Mello and R. D. Lins. "Image segmentation of historical documents". Visual 2000, Mexico City, Mexico. 2000.

[7] W. Kesheng, O. Ekow, and S. Arie, "Optimizing connected components labeling algorithms," in SPIE Int. Symposium on Medical Imaging, San Diego, CA, USA, Feb. 2005.

[8] W. Niblack. An Introduction to Digital Image Processing. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[9] N. Otsu. A threshold selection method from graylevel histogram. IEEE Transactions on System, Man, Cybernetics, 19(1):62–66, January 1978.

[10] J. Sauvola and M. Pietikainen. Adaptive document image binarization. Pattern Recognition, 33(2):225–236, January 2000.

[11] J. M. M. da Silva, R. D. Lins and V. C. da Rocha Jr., "Binarizing and Filtering Historical Documents with Back-to-Front Interference". Proceedings of SAC 2006. New York : ACM Press, 2006. p. 853-858.

[12] C.L. Tan, L. Zhang, Z. Zhang and T. Xia. Roberts, "Restoring Warped Document Images through 3D Shape Modeling", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, New York, Vol. 28, No. 2, Feb. 2006, pp. 195-208.

[13] Y.C. Tsoi and M.S. Brown. "Geometric and Shading Correction for Images of Printed Materials: A Unified Approach Using Boundary". Proc. IEEE CVPR 2004, vol. 1, pp. 240-246, 2004.

[14] Adobe. Adobe After Effects CS4. http://www.adobe.com/products/aftereffects/.

[15] B. Gatos. DIBCO 2009 – Evaluation. http://www.iit.demokritos.gr/~bgat/DIBCO2009/Evaluation.html, accessed on 1$^{st}$ may of 2009.

[16] J. Kemp. File:Standard deviation diagram.svg. http://en.wikipedia.org/wiki/File:Standard_deviation_diagram.svg

# COCOCLUST: Contour-based Color Clustering for Robust Binarization of Colored Text

Thotreingam Kasar and AG Ramakrishnan
Medical Intelligence and Language Engineering Laboratory
Indian Institute of Science, Bangalore, INDIA - 560 012
{tkasar, ramkiag}@ee.iisc.ernet.in

## Abstract

*This paper proposes COCOCLUST, a contour-based color clustering method which robustly segments and binarizes colored text from complex images. Rather than operating on the entire image, a 'small' representative set of color pixels is first identified using the contour information. The method involves the following steps: (i) Identification of prototype colors (ii) A one-pass algorithm to identify color clusters that serve as seeds for the refining step using k-means clustering (iii) Assignment of pixels in the original image to the nearest color cluster (iv) Identification of potential candidate text regions in individual color layer and (v) Adaptive binarization. We propose a robust binarization technique to threshold the identified text regions, taking into account the presence of inverse texts, such that the output image always has black text on a white background. Experiments on several complex images having large variations in font, size, color, orientation and script illustrate the robustness of the method.*

## 1. Introduction

The use of digital cameras for image acquisition has enabled human interaction with any type of document in any environment. In addition to imaging hard copy documents, digital cameras are now used to acquire text information present in 3-D real world objects such as buildings, vehicles, road signs, billboards and T-shirts rendering the images more difficult for any recognition task. Such camera-captured documents are generally characterized by varying illumination, blur, perspective distortion and deformations. Moreover, large variations in font style, size, color, orientation and layout pose a big challenge to document analysis. Conventional optical character recognition engines meant for document images obtained using flat-bed scanners fail on images acquired by this promising mode. Specialized techniques are required to deal with these problems. Thus, research on camera-based document image analysis is growing [4].

In most document processing systems, a binarization process precedes the analysis and recognition procedures. It is critical to achieve robust binarization since any error introduced in this stage will affect the subsequent processing steps. The simplest and earliest method is the global thresholding technique that uses a single threshold to classify image pixels into foreground or background classes. Global thresholding techniques are generally based on histogram analysis [6, 9]. It is simple, fast and works well for scanned images that have well-separated foreground and background intensities. Camera-captured images often exhibit non-uniform brightness because it is difficult to control the imaging environment unlike the case of the scanner. The histograms of such images are generally not bi-modal and a single threshold can never yield an accurate binary image. As such, global binarization methods are not suitable for camera images.

Local binarization techniques use a dynamic threshold across the image according to the local image statistics and offer more robustness to non-uniform illumination and background noise. These approaches are generally window-based and the local threshold for a pixel is computed from the gray values of the pixels within a window centred at that particular pixel. In Niblack's method [8], the sample mean $\mu(x, y)$ and the standard deviation $\sigma(x, y)$ within a window W centred at the pixel location $(x, y)$ are used to compute the threshold $T(x, y)$ as follows:

$$T(x, y) = \mu(x, y) - k \times \sigma(x, y), \tag{1}$$

The constant parameter $k$ is set to 0.2. In [11], Trier and Jain evaluated the performance of 11 popular local thresholding methods on scanned documents and reported that Niblack's method performs the best for optical character recognition. However, Niblack's method produces a noisy output in smooth regions since the expected sample variance becomes the background noise variance. Sauvola and

Pietikainen [10] address this drawback by introducing a hypothesis that the gray values of the text are close to 0 (Black) while the background pixels are close to 255 (White). The threshold is computed with the dynamic range of standard deviation (R) which has the effect of amplifying the contribution of standard deviation in an adaptive manner.

$$T(x, y) = \mu(x, y)\,[1 + k\,(\frac{\sigma(x, y)}{R} - 1)] \tag{2}$$

where the parameters $R$ and $k$ are set to 128 and 0.5 respectively. This method overcomes the effect of background noise and is more suitable for document images. However, as pointed out by Wolf and Jolion in [13], the Sauvola's method fails for images where the assumed hypothesis is not met and accordingly, the former proposed an improved threshold estimate by taking the local contrast measure into account.

$$T(x, y) = [1 - a]\mu(x, y) \,+\, aM + a\frac{\sigma(x, y)}{S_{max}}[\mu(x, y) - M] \tag{3}$$

where $M$ is the minimum value of the grey levels of the whole image, $S_{max}$ is the maximum value of the standard deviations of all windows of the image and $a$ is a parameter fixed at 0.5. This method combines Savoula's robustness with respect to background textures and the segmentation quality of Niblack's method. The Wolf's method, however, requires two passes since the parameter $S_{max}$ is obtained only after the first pass of the algorithm.

Local methods offer more robustness to the background complexity, though at a cost of higher computational complexity. The performance of these methods depend on the size of the window used to compute the image statistics. They work well if the window encloses at least 1 character. For large fonts, where the text stroke is wider than the window, undesirable voids appear within the text stroke. This puts a constraint on the maximum font size and limits their application only to known document types. In addition, all these methods require a priori knowledge of the polarity of the foreground-background intensities and hence cannot handle documents that have inverse text. Kasar *et al.* [7] address these issues by employing an edge-based approach that derives an adaptive threshold for each connected component (CC). Though it can deal with arbitrary font size and the presence of inverse text, its performance significantly degrades, like most CC-based methods do, in the presence of complex backgrounds that interfere in the accurate identification of CCs. It also uses script-specific characteristics to filter out non-text components before binarization and works well only for Roman script.

Most approaches [5, 12, 14] for the analysis of color documents involve clustering on the 3D color histogram followed by identification of text regions in each color layer using some properties of text.

Badekas *et al.* [2] estimate dominant colors in the image and CCs are identified in each color plane. Text blocks are identified by CC filtering and grouping based on a set of heuristics. Each text block is applied to a Kohonen SOM neural network to output only two dominant colors. Based on the run-length histograms, the foreground and the background are identified to yield a binary image having black text in white background. The performance of these methods rely on the accuracy of color reduction and text grouping, which are not trivial tasks for a camera-captured complex document image. The method does not consider isolated characters for binarization. Zhu *et al.* [15] proposed a robust text detection method that uses a non-linear Niblack thresholding scheme. Each CC is described by a set of low level features and text components are classified using a cascade of classifiers trained with Adaboost algorithm. An accurate identification of CCs is the key to the success of these algorithms. Complex backgrounds and touching characters can significantly degrade their performance.

In this paper, we introduce a novel color clustering approach that robustly segments the foreground text from the background. Text-like regions are identified and individually binarized such that the foreground text is assigned black and the background white regardless of its color in the original input image.

## 2 COCOCLUST for color segmentation

We propose a novel contour-based color clustering technique that obviates the need to specify the number of colors present in the image and to initialize. Rather than operating on the entire image, a representative set of color pixels is first identified using the contour information. This significantly reduces the computational load of the algorithm since their number is much smaller than the total number of pixels in the image. A single-pass clustering is then performed on the reduced color prototypes to identify color clusters that serve as seeds for a subsequent clustering step using the k-means algorithm. CCs are accurately identified since text and background objects fall into separate color layers. Based on the assumption that every character is of a uniform color, we analyze each color layer individually and identify potential text regions for binarization. Figure 1 shows the schematic block diagram of the proposed method.

### 2.1 Determination of color prototypes

The segmentation process starts with color edge detection to obtain the boundaries of homogeneous color regions. Canny edge detection [3] is performed individually on $R$, $G$ and $B$ channel and the overall edge map $\mathcal{E}$ is obtained as follows:

$$\mathcal{E} = \mathcal{E}_R \cup \mathcal{E}_G \cup \mathcal{E}_B \tag{4}$$

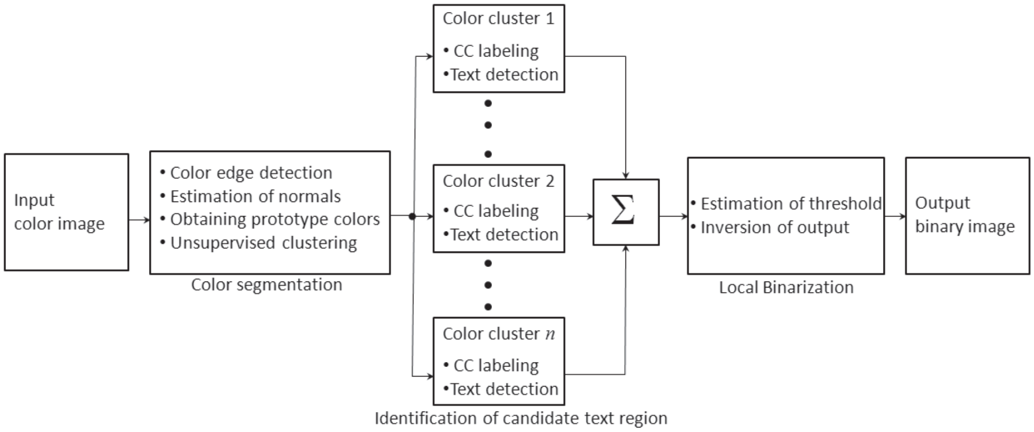


**Figure 1. Block diagram of the proposed binarization method.**

where $\mathcal{E}_R$, $\mathcal{E}_G$ and $\mathcal{E}_B$ are the edge images corresponding to the three color channels and $\cup$ denotes the union operation. The resulting edge image gives the boundaries of all the homogeneous color regions present in the image.

An 8-connected component labeling is performed on the edge image to obtain a set of $M$ disjoint components

$$\{\mathcal{CC}^j\} \quad j = 1,\, 2\,, \cdots,\, M \;\text{ such that } \bigcup_{j=1}^{M} \mathcal{CC}^j = \mathcal{E}$$

The boundary pixels are identified and represented as follows:

$$\mathbf{X}_i^j = \{x_i, y_i\} \quad i = 1,\, 2,\, \cdots,\, n_j \tag{5}$$

where $n_j$ is the number of pixels that constitutes the boundary and the index $j$ refers to the connected component $\mathcal{CC}^j$. Our method employs a few vectors normal to the edge contour for every CC. To estimate the normal vector, the edge contour is smoothed locally as follows:

$$\bar{\mathbf{X}}_i^j = \left\{ \frac{1}{s} \sum_{i-\frac{s-1}{2}}^{i+\frac{s-1}{2}} x_i,\; \frac{1}{s} \sum_{i-\frac{s-1}{2}}^{i+\frac{s-1}{2}} y_i \right\} \tag{6}$$

where $s$ defines the span of pixels over which smoothing is performed and is set to 5 in this work. Here, the index $i$ takes a circular convention to maintain continuity of the contour. The normal vectors are then computed from the smoothed contour using the following relation.

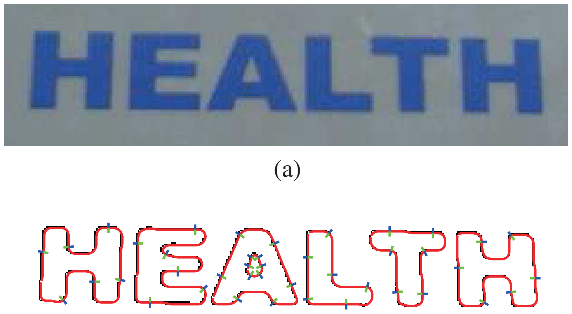
$$\begin{aligned} \mathbf{n}_i^j \quad &= \quad \begin{bmatrix} cos(\frac{\pi}{2}) & -sin(\frac{\pi}{2}) \\ sin(\frac{\pi}{2}) & cos(\frac{\pi}{2}) \end{bmatrix} \times \\ & \quad\; \frac{1}{2} \left( \frac{\bar{\mathbf{X}}_i^j - \bar{\mathbf{X}}_{i-1}^j}{\|\bar{\mathbf{X}}_i^j - \bar{\mathbf{X}}_{i-1}^j\|} + \frac{\bar{\mathbf{X}}_{i+1}^j - \bar{\mathbf{X}}_i^j}{\|\bar{\mathbf{X}}_{i+1}^j - \bar{\mathbf{X}}_i^j\|} \right) \end{aligned} \tag{7}$$

Here, the subscript $i$ denotes the position of the boundary pixel at which the normal vector is computed and $\| \cdot \|$ denotes $\text{L}_2$ norm.

Since the edge image gives the boundaries of homogeneous color regions, the color values of a few pixels that lie normal to the contour are 'good' representatives of the colors present in the image. Figure 2(a) shows a sample color image and Figure 2(b) illustrates the selection of color prototypes from the pixels that lie normal to the edge contour. The median color values of the pixels in the normal direction that lie 'inside' ($\mathbf{n}_-^j$) and 'outside' ($\mathbf{n}_+^j$) the boundary are computed based on 3 pixels each to obtain 2 color prototypes from each normal. The color difference between two points with the same Euclidean distance in the $RGB$ color space does not reflect the same change in the perceived color. So, we use a uniform color space, namely CIE $L^*a^*b^*$, in which similar changes in color distance also correspond to similar recognizable changes in the perceived color. The color prototypes ($CP$), thus obtained, are stacked column-wise as follows.

$$CP = \left\{ \begin{array}{ccccc} L_{1-}^* & L_{1+}^* & \cdots & L_{N-}^* & L_{N+}^* \\ a_{1-}^* & a_{1+}^* & \cdots & a_{N-}^* & a_{N+}^* \\ b_{1-}^* & b_{1+}^* & \cdots & b_{N-}^* & b_{N+}^* \end{array} \right\}$$

where $N$ is the number of normals along which the color values are sampled. In this work, we sample the color values from 6 regularly spaced points along the boundary of each CC yielding a total of 12 $M$ colors. This set of color values, though much smaller in number than the total number of pixels, captures all the colors present in the image. This offers a significant advantage in terms of cheaper computation and provides an effective initialization of k-means algorithm regardless of the complexity of image content.



(a)

(b)

**Figure 2. (a) A sample color image (b) Its edge contours and the computed normals that guide the selection of color prototypes. From each normal, one color value each is obtained from the pixels that lie 'inside' (Green segment) and 'outside' (Blue segment) the contour.**

### 2.2 Unsupervised color clustering

A single-pass clustering is performed on color prototypes, as obtained above, to group them into clusters. The pseudo-code for the clustering algorithm is given below.

```
Input: Color prototypes, CP={$C_1$, $C_2$, ..., $C_{2N}$}
       Color similarity threshold, $T_s$
Output: Color clusters, CL
Assign CL[1]=$C_1$ and Count =1
For i = 2 to 2N, do
  For j = 1 to Count, do
    If Dist(CL[j],$C_i$) $\leq$ $T_s$
      CL[j] = Update Mean(CL[j])
      Next i
    Else
      Count = Count + 1
      CL[Count] = $C_i$
   EndIf
   EndFor
 EndFor
```

where Dist($C_1, C_2$) denotes the distance between the colors $C_1 = (L_1^*, a_1^*, b_1^*)^T$ and $C_2 = (L_2^*, a_2^*, b_2^*)^T$ and is computed as follows:

$$\text{Dist}(C_1, C_2) = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} \tag{8}$$

The threshold parameter $T_s$ decides the similarity between two colors and hence the number of clusters. Antonacopoulos and Karatzas [1] perform grouping of color pixels based on the criterion that only colors that cannot be differentiated by humans should be grouped together. The threshold below which two colors are considered similar was experimentally determined and set to 20. We use a slightly higher threshold to account for the small color variations that may appear within the text strokes. In our implementation, the threshold parameter $T_s$ is empirically fixed at 45, after trial and error.

The color clusters, thus obtained, are used as seeds for a subsequent clustering step using the k-means algorithm. Each resulting cluster is then examined for 'compactness' by computing distances of all the pixels in that cluster from its mean color. The maximum intra-cluster distance from its mean color is ensured to be less than 75 % of $T_s$, if required, by recursively splitting non-compact clusters into two using k-means algorithm initialized with the mean color and the one that is furthest from it. The color clusters obtained at the end of this splitting process are then used as the seed colors for a final pass of k-means clustering. Note that the whole clustering process is performed only on the selected prototypes. Finally, each pixel in the original image is assigned to the closest color cluster.

## 3 Adaptive binarization

Each color layer is then individually analyzed and text-like components are identified. We filter out the obvious non-text elements by making some sensible assumptions about the document. The aspect ratio is constrained to lie between 0.1 and 10 to remove highly elongated components. Components larger than 0.6 times the image dimensions are removed. Furthermore, small and spurious components with areas less than 8 pixels are not considered for subsequent processing.

Text components have well-defined boundaries and hence have a high degree of overlap with the edge image as compared to non-text components. The boundary $\mathbf{X}^j$ of a component $\mathcal{CC}^j$ and its corresponding edge image region $\mathcal{E}^j$ are first dilated and their intersection is computed as a measure of stability of the boundary ($BS$).

$$BS^j = \frac{Area((\mathcal{E}^j \oplus S_3) \cap (\mathbf{X}^j \oplus S_3))}{Area(\mathbf{X}^j \oplus S_3)} \tag{9}$$

where $S_3$ is a 3×3 structuring element. Components that yield a $BS$ measure of more than 0.5 are selected for binarization from all the color layers. Overlapping CCs are resolved by retaining only the dominant component.

### 3.1 Estimation of threshold

The binarization technique proposed in [7] automatically computes the threshold value from the image data without the need for any user-defined parameter. We use a similar approach to binarize each CC by estimating its foreground and background intensities. The foreground intensity of a component $\mathcal{CC}^j$ is computed as the mean gray level of its boundary pixels.

$$\mathcal{FG}^j = \frac{1}{n_j} \sum_{(x,y)\in\mathbf{X}^j} \mathcal{I}(x,y) \tag{10}$$

where $\mathcal{I}(x,y)$ denotes the intensity value at the pixel position $(x,y)$ and $n_j$ is the number of pixels that constitute the boundary $\mathbf{X}^j$.

Rather than using the bounding box to obtain an estimate of the background intensity [7], we use the available contour information that yields a more reliable decision for inversion in the presence of inverse text. Bounding boxes can have a significant overlap for inclined text and touching text lines that can result in incorrect inversion of the binary output. The contour is traced in a clock-wise direction and the normals are estimated. The background intensity is then computed as the median intensity value of the pixels along the normal direction 'outside' the boundary of the CC.

$$\mathcal{BG}^j = \text{Median}(\mathcal{I}(x,y)) \quad (x,y) \in \mathbf{n}_+^j \tag{11}$$

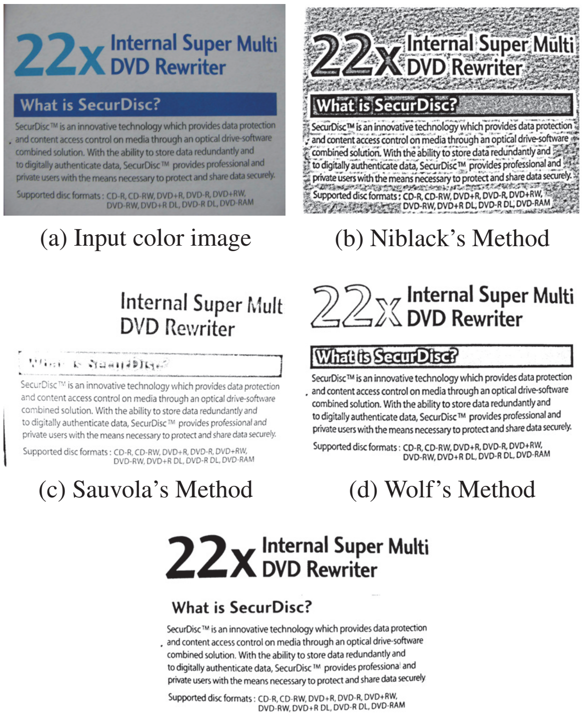
Note that the boundary of the CC is always 'closed' unlike during the prototype color identification stage where we may have broken as well as bifurcating edge contours. The CC is binarized using the estimated foreground intensity as the threshold value.

$$\mathcal{O}_j(x,y) = \begin{cases} 1 & \text{if } \mathcal{CC}^j(x,y) \geq \mathcal{FG}^j \\ 0 & \text{if } \mathcal{CC}^j(x,y) < \mathcal{FG}^j \end{cases} \tag{12}$$

The estimated values of the foreground and background intensities indicate their relative polarity. Whenever the estimated foreground intensity is higher that that of the background, the binary output is inverted to ensure that text is always represented by black pixels.

## 4 Experiments and results

The test images used in our experiments include physical documents such as books and charts as well as non-paper documents like text on 3-D real world objects. These images are characterized by complex backgrounds, irregular text orientation and layout, overlapping text, variable fonts, size, color, multiple scripts and presence of inverse text.


22x Internal Super Multi DVD Rewriter
What is SecurDisc?
SecurDisc™ is an innovative technology which provides data protection and content access control on media through an optical drive-software combined solution. With the ability to store data redundantly and to digitally authenticate data, SecurDisc™ provides professional and private users with the means necessary to protect and share data securely.
Supported disc formats : CD-R, CD-RW, DVD+R, DVD-R, DVD+RW, DVD-RW, DVD+R DL, DVD-R DL, DVD-RAM

(a) Input color image (b) Niblack's Method

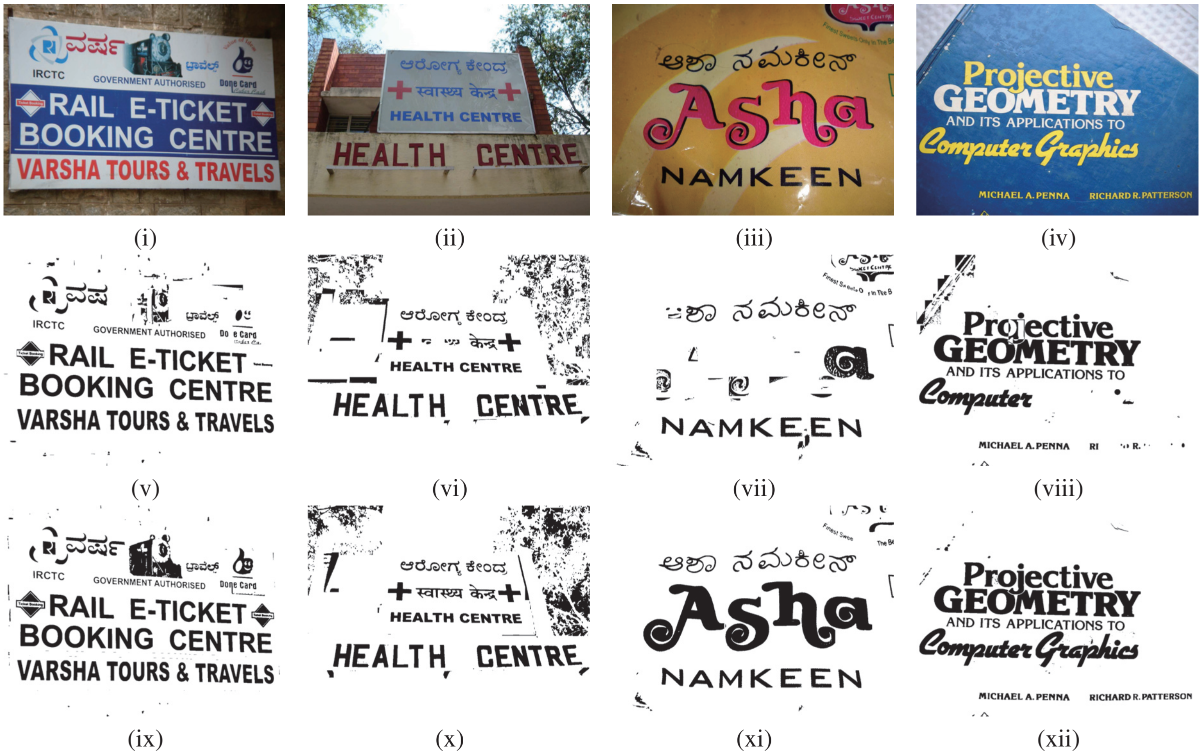
(c) Sauvola's Method (d) Wolf's Method

(e) Proposed method

**Figure 3. Comparison of some popular local binarization methods for a document image having multiple text color and size. While the proposed method is able to handle characters of any size, all other methods fail to binarize properly the components larger than the size of the window and require a priori knowledge of the polarity of foreground-background intensities as well.**

Figure 3 compares the results of our method with some popular local binarization techniques, namely, Niblack's method, Sauvola's method and Wolf's method on a document image having multi-colored text and large variations in sizes with the smallest and the largest components being 4×3 to 174×245 respectively. Clearly, these local binarization methods fail when the size of the window is smaller than stroke width. The size of the window used here is 33 × 33. While small text regions are properly binarized, large characters are broken up into several components and undesirable voids occur within the character stroke. It requires a priori knowledge of the polarity of foreground-background intensities as well. On the other hand, our method automatically derives the threshold from the image without any user-defined parameter. It can deal with characters of any font size and color.

Figure 4 shows the result of the proposed method on images having inverse text, multiple scripts, background objects touching the text and cursive letters. The method proposed in [7] is sensitive to the background and several instances of text get filtered out since it uses an edge-based segmentation. Moreover, it uses script-dependent characteristics and works well only for isolated Roman letters.



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

(ix) (x) (xi) (xii)

**Figure 4. (i - iv): Input images having graphic objects, inverse text, multiple scripts, complex backgrounds and cursive letters. (v - viii): The corresponding binary outputs obtained using the method proposed in [7]. Clearly, the method is sensitive to background objects since it relies on the edge information to locate CCs. It also invokes script-dependent characteristics and works well only for isolated Roman letters. (ix - xii): Binarized output images obtained with the proposed method. Color segmentation provides robustness to background complexity as well as independence to script.**

In Figure 4(v) and (vii), the background and graphic objects touch some text regions and they get filtered out. In Figure 4(vi-viii), the script-dependency of the method is clearly observed. In addition to the merged characters and cursive text being eliminated, Figure 4(viii) also shows an instance of incorrect inversion of the binary output due to overlapping text lines. In contrast, the new method shows a marked improvement thanks to the color clustering algorithm that enables an accurate identification of CCs. The color decomposition effectively disambiguates background objects interfering with text as they are separated into different color layers. Each CC is individually binarized based on a threshold derived from its foreground and background intensity estimates. The background intensity estimate obtained using the contour normals provides a reliable decision to invert the binary output in the the presence of inverse text. As desired, all the text components are represented by black on white background regardless of their colors in the original image. The method is tested on several images and is found to have good adaptability. More results of the proposed method on various input images that have inverse text, arbitrary text orientation and layout are shown in Figure 5.

## 5 Conclusions

This paper describes an important preprocessing step for the analysis of color document images. The use of the contour information makes the method robust to the complexity of the input image. This is a desirable feature for processing camera-based images that are generally characterized by arbitrary content and layout. It does not require a priori knowledge of the number of colors present or their initialization. The contour information is successfully exploited both in color segmentation that enables accurate identification of CCs and in the inversion of the binary output to deal with inverse text. Preliminary results on camera-captured images with variable fonts, size, color, orientation, script

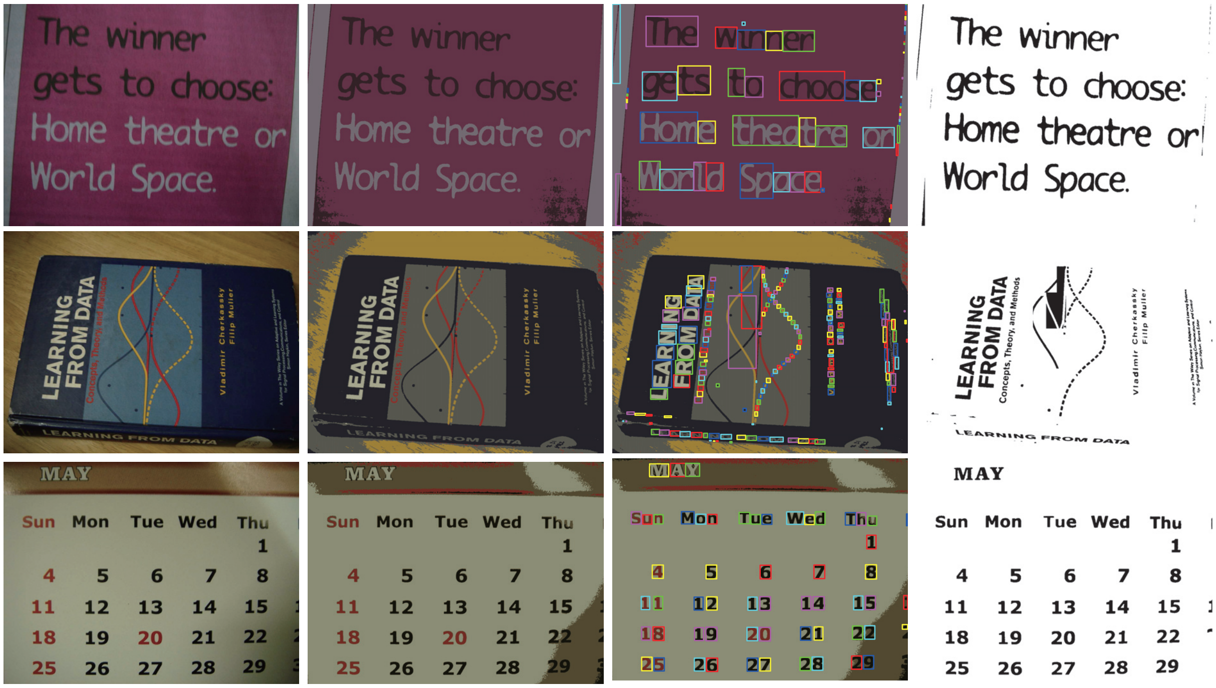


**Figure 5. Input images and the corresponding color clusters, identified text regions and binarized outputs shown column-wise.**

and the presence of inverse text are encouraging.

Our future work is to augment the method with a trained classifier for robust extraction of only the text regions.

## References

[1] A. Antonacopoulos and D. Karatzas. Fuzzy segmentation of characters in web images based on human colour perception. *Proc. Workshop Doc. Anal. Systems*, pages 295–306, 2002.

[2] E. Badekas, N. Nikolaou, and N. Papamarkos. Text binarization in color documents. *Intl. Jl. Imaging. Syst. Technol.*, 16:262–274, 2007.

[3] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986.

[4] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. *Proc. Intl. Conf. Doc. Analy. Recog.*, 1:606–616, 2003.

[5] A. Jain and B. Yu. Automatic text location in images and video frames. *Pattern Recog.*, 3(12):2055–2076, 1998.

[6] J. N. Kapur, P. K. Sahoo, and A. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Comp. Vision Graphics Image Process.*, 29:273–285, 1985.

[7] T. Kasar, J. Kumar, and A. G. Ramakrishnan. Font and background color independent text binarization. *Proc. Intl. Workshop Camera Based Doc. Anal. Recog.*, pages 3–9, 2007.

[8] W. Niblack. An introduction to digital image processing. *Prentice Hall*, pages 115–116, 1986.

[9] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems Man Cybernetics*, 9(1):62–66, 1979.

[10] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recog.*, 33:225–236, 2000.

[11] O. D. Trier and A. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. PAMI*, 17(12):1191–1201, 1995.

[12] B. Wang, X. F. Li, F. Liu, and F. Q. Hu. Color text image binarization based on binary texture analysis. *Proc. IEEE Intl. Conf. Acoustics, Speech Sig. Proc.*, 3:585 – 588, 2004.

[13] C. Wolf and J. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Anal. Applic.*, 6:309–326, 2003.

[14] Y. Zhong, K. Karu, and A. Jain. Locating text in complex color images. *Pattern Recog.*, 28(10):1523 – 1535, 1995.

[15] K. Zhu, F. Qi, R.Jiang, L. Xu, M. Kimachi, Y. Wu, and T. Aizawa. Using adaboost to detect and segment characters from natural scenes. *Proc. Intl. Workshop Camera Based Doc. Anal. Recog.*, pages 52–59, 2005.

# Foreground-Background Regions Guided Binarization of Camera-Captured Document Images

Syed Saqib Bukhari[1], Faisal Shafait[2], Thomas M. Breuel[1,2]
[1]Technical University of Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
bukhari@informatik.uni-kl.de, faisal@iupr.dfki.de, tmb@informatik.uni-kl.de

## Abstract

*Binarization is an important preprocessing step in several document image processing tasks. Nowadays handheld camera devices are in widespread use, that allow fast and flexible document image capturing. But, they may produce degraded grayscale image, especially due to bad shading or non-uniform illumination. State-of-the-art binarization techniques, which are designed for scanned images, do not perform well on camera-captured documents. Furthermore, local adaptive binarization methods, like Niblack [1], Sauvola [2], etc, are sensitive to free parameter values, which are fixed for whole image. In this paper, we describe a novel binarization technique using ridges-guided local binarization method, in which appropriate free parameter value(s) is(are) selected for each pixel depending on the presence or absence of ridge(s) in the local neighborhood of a pixel. Our method gives a novel way of automatically selecting parameter values for local binarization method, this improves binarization results for both scanned and camera-captured document images relative to previous methods. Experimental results on a subset of CBDAR 2007 document image dewarping contest dataset show a decrease in OCR error rate using reported method with respect to other stat-of-the-art bianrization methods.*

## 1 Introduction

Most of the state-of-the-art document analysis systems have been designed to work on binary images [3]. Therefore, document image binarization is an important initial step in most of the document image processing tasks, like page segmentation [4], layout analysis [5, 6] or recognition. Performance of these tasks heavily depends on the results of binarization. The main objective of document image binarization is to divide a grayscale or color document into two groups, that are foreground text/images and clear background.

On one hand, cameras offer fast, easy and non-contact document imaging as compared to scanners and are in more common use nowadays. But on the other hand, the quality of camera-captured documents is worse as compared to scanned documents because of the degradations which are not very common in scanned images, like non-uniform shading, image blurring and lighting variations. Due to this, binarization of camera-captured documents is more challenging than scanned documents.

From decades, many different approaches for the binarization of grayscale [7, 2, 8, 9, 1, 10, 11, 12, 13, 14] and color [15, 16, 17] documents have been proposed in the literature. Additionally, grayscale binarization techniques can be applied by first converting the color documents into grayscale. Grayscale binarization approaches can be classified into two main groups: i) global binarization methods and ii) local binarization methods.

Global binarization methods, like Otsu [7], try to estimate a single threshold value for the binarization of whole document. Then based on the intensity values, each pixel is assigned either to foreground or background. Global binarization methods are computationally inexpensive and perform better for typical scanned document images. However, they produce marginal noise artifacts if grayscale document contains non-uniform illumination, which is usually present in case of scanned thick book, came-captured document or historical document.

Local binarization methods, like Sauvola [2], try to overcome these problems by calculating threshold values for each pixel differently using local neighborhood information. They perform better on degraded document images but are computationally slow and sensitive to the selection of window size and free parameter values [18]. Some special techniques [11, 12, 13] based on local binarization have been proposed recently for improving the binarization results of degraded camera-captured and historical documents. In general these methods produce good bianrization results under non-uniform illumination as compared to other types of local binarization methods, but are still sensitive to free parameter values.

In this paper, we deal with the binarization of degraded grayscale camera-captured document images. Here, we describe a local binarization method based on Sauvola's binarization method, which is less sensitive to free parameter values. Unlike Sauvola's method, instead of using the same free parameter values for all pixels, we select different values for foreground and background pixels. We use ridges detection technique for finding foreground regions information.

The rest of the paper is organized as follows: Section 2 explains the technical and implementation details of our binarization method. Section 3 deals with experimental results and section 4 describes conclusion.

## 2 Foreground-Background Guided Binarization

Researchers [19, 20] have evaluated different state-of-the-art global and local binarization methods and reported that Sauvola's binarization method [2] is better than other types of local binarization methods for degraded document images. But the performance of local binarization methods is sensitive to free parameter values [18]. Our binarization method, presented here, is an extension of Sauvola's method. In section 2.1 we discuss about the Sauvola's binarization method and how to improve its performance by selecting different free parameter values for foreground and background pixels. In section 2.2 we describe the method for detecting foreground regions using ridges. In section 2.3 we describe the guided Sauvola's binarization method with respect to foreground/background regions information. .

### 2.1 Local Binarization using Sauvola's method

Grayscale document images contain intensity values in between 0 to 255. Unlike global binarization, local binarization methods calculate a threshold $t(x, y)$ for each pixel such that

$$b(x,y) = \begin{cases} 0 & \text{if } g(x,y) \leq t(x,y) \\ 255 & \text{otherwise} \end{cases} \tag{1}$$

The threshold $t(x, y)$ is computed using the mean $\mu(x, y)$ and standard deviation $\sigma(x, y)$ of the pixel intensities in a $w \times w$ window centered around the pixel $(x, y)$ in Sauvola's binarization method:

$$t(x,y) = \mu(x,y)\left[1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right] \tag{2}$$

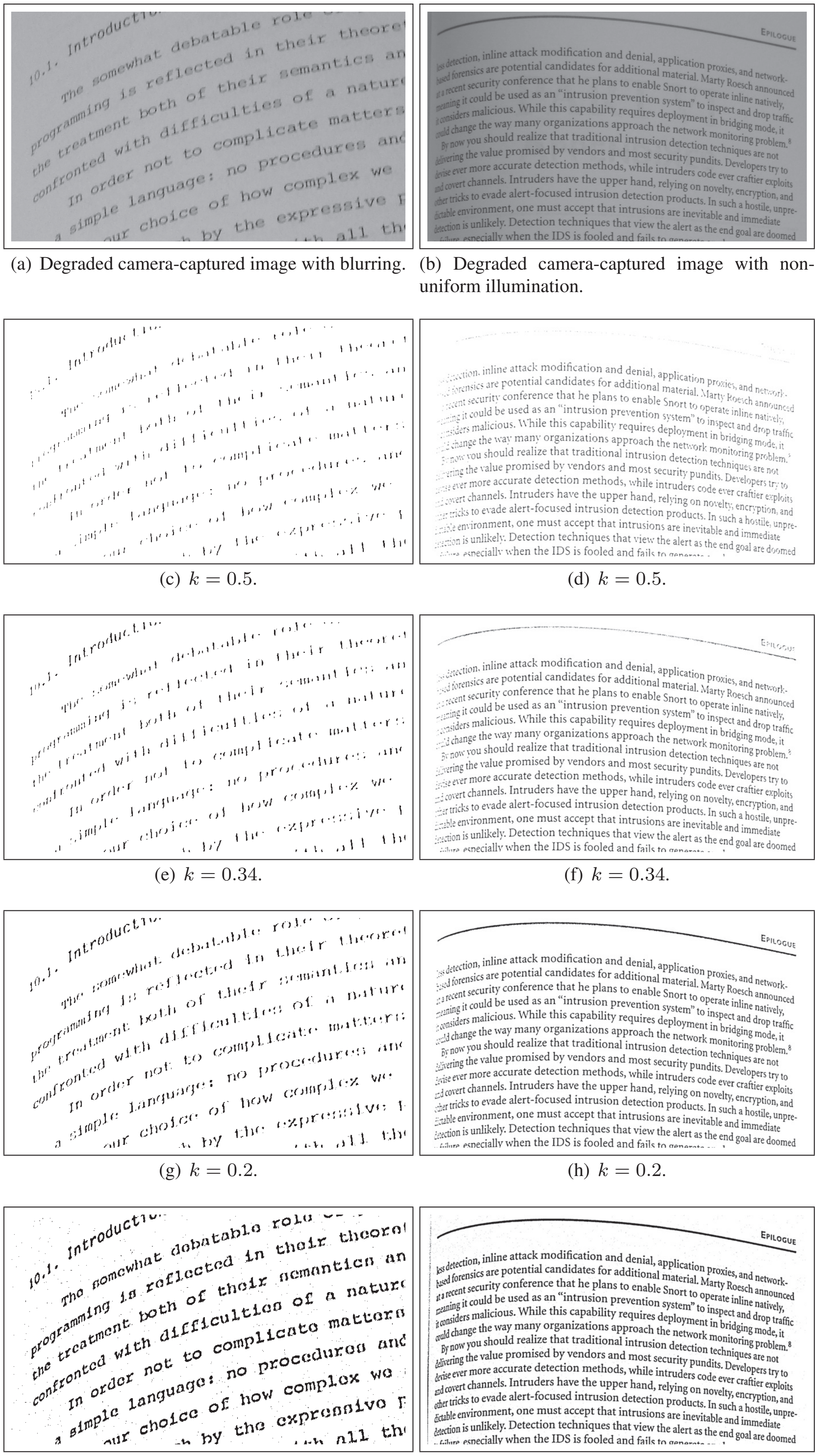
where $R$ is the maximum value of the standard deviation ($R = 128$ for a grayscale document), and $k$ is a parameter which takes positive values. The formula (Equation 2) has been designed in such a way that, the value of the threshold is adapted according to the contrast in the local neighborhood of the pixel using the local mean $\mu(x, y)$ and local standard deviation $\sigma(x, y)$. Because of this, it tries to estimate appropriate threshold $t(x, y)$ for each pixel under both possible conditions: high and low contrast. In case of local high contrast region ($\sigma(x, y) \approx R$), the threshold $t(x, y)$ is nearly equal to $\mu(x, y)$. Under quite low contrast region ($\sigma << R$), the threshold goes below the mean value thereby successfully removing the relatively dark regions of the background. The parameter $k$ controls the value of the threshold in the local window such that the higher the value of $k$, the lower the threshold from the local mean $m(x, y)$.

The statistical constraint in Equation 2 gives acceptable results even for degraded documents. But, there is a contradiction regarding the appropriate value of $k$ in research community. Badekas et al. [20] experimented with different values and found that $k = 0.34$ gives the best results, but Sauvola[2] and Sezgin[19] used $k = 0.5$.

We have analyzed Sauvola's binarization method with different values of $k$ for degraded camera-captured document images. Some of the experimental results are shown in the Figure 1. These results clearly show the sensitivity of Sauvola's binarization on the value of $k$. Additionally, already reported values of $k$,i.e $k = 0.5$ [2, 19] and $k = 0.34$ [20], do not give acceptable result under blurring or non-uniform illuminations, as shown in Figure 1.

However, we have noticed that, $k = 0.2$ gives low noise in the background but produces broken characters, shown in Figures1(g) and 1(h). On the other hand, $k = 0.05$ gives good results for foreground text/images pixels with unbroken characters but with some noise in the background, as shown in Figures 1(i) and 1(j). These experiments allows us to claim that, Sauvola's method can perform better on degraded documents, if we use different value of $k$ for each pixel depending upon its association with foreground or background region.

In next section we describe the method of estimating foreground region using ridges detection and in section 2.3 we describe the adaptation of Sauvola's method using foreground/background region information.



(a) Degraded camera-captured image with blurring.

(b) Degraded camera-captured image with non-uniform illumination.

(c) $k = 0.5$.

(d) $k = 0.5$.

(e) $k = 0.34$.

(f) $k = 0.34$.

(g) $k = 0.2$.

(h) $k = 0.2$.

(i) $k = 0.05$.

(j) $k = 0.05$.

**Figure 1. Sauvola's binarization results for different values of $k$. $k = 0.5$ is reported by Sauvola[2] and Sezgin[19]. $k = 0.34$ is used by Badekas et al. [20]. We have selected $k = 0.2$ and $k = 0.05$. With $k = 0.2$, results have cleaned-background and broken-foreground-characters. And with $k = 0.05$ results have uncleaned-background and unbroken-foreground-characters.**

## 2.2 Foreground Regions detection using Ridges

We have already described textline detection techniques for handwritten and camera-captured documents using ridges in [21, 22]. In this paper, we use this technique for finding foreground regions, that are central lines structure of textlines and drawings. Detection of foreground regions using ridges is divided into two sub steps: (i) image smoothing and (ii) ridges detection. Following sections discuss these steps in detail.

### 2.2.1 Image Smoothing

Camera-captured document images contain variety of curled textlines and drawings structure with respect to size and orientation angle. Match filter bank approach has been used for enhancing the structure of multi-oriented blood vessels [23] and finger prints [24]. In [21, 22] , we have described multi-oriented multi-scale anisotropic Gaussian smoothing, based on matched filter bank approach, for enhancing textlines structure. In this paper, we use multi-oriented multi-scale anisotropic Gaussian smoothing for enhancing curled textlines and drawings structure. A single range is selected for both $\sigma_x$ and $\sigma_y$, which is the function of the height of the document image ($H$), that is $aH$ to $bH$ with $a < b$. The suitable range for $\theta$ is from -45 to 45 degrees. From these ranges, a set of filters is generated for different combinations of $\sigma_x$, $\sigma_y$ and $\theta$. This set of filters is applied to each pixel of grayscale image and the maximum resulting value is selected. Figures 2(a) and 2(b) show the input and smoothed images respectively.

### 2.2.2 Ridges Detection

Multi-oriented multi-scale anisotropic Gaussian smoothing enhances the foreground structure well, which is clearly visible in Figure 2(b). Now the task is to find the foreground regions information. Since decades, ridges detection has been popularly used for producing rich description of significant features from smoothed grayscale images [25] and speech-energy representation in time-frequency domain [26]. Ridges detection over smoothed image can produce unbroken central lines structure of foreground textlines/drawings. In this paper, Horn-Riley [25, 26] based ridges detection approach is used. This approach is based on the informations of local direction of gradient and second derivatives as the measure of curvature. From these informations, which are calculated by Hessian matrix, ridges are detected by finding the zero-crossing of the appropriate directional derivatives of smoothed image. Detected Ridges over the smoothed image of Figure 2(b) are shown in Figure 2(c) and Figure 2(d). It is clearly visible in the Figure 2(c) that ridges are present where the foreground data are present and each ridge covers the complete central line structure of a foreground object.

## 2.3 Foreground-Background Guided Sauvola's Binarization

We have already discussed in section 2.1 that no single value of parameter $k$ in Sauvola's method is suitable for different types of degraded camera-captured documents. But $k = 0.05$ gives good results for foreground textlines/drawings with some background noise and $k = 0.2$ gives noise free background with broken characters, as shown in Figure 1. Ridges have been detected in section 2.2, that give information about foreground data. Therefore, instead of using fixed value of $k$ for all pixels, we use different values of $k$ for foreground and background pixels to improve the binarization result. We redefine Sauvola's binarization method, such that:

$$t(x,y) = \mu(x,y)\left[1 + \mathbf{k(x,y)}\left(\frac{\sigma(x,y)}{R} - 1\right)\right] \quad (3)$$

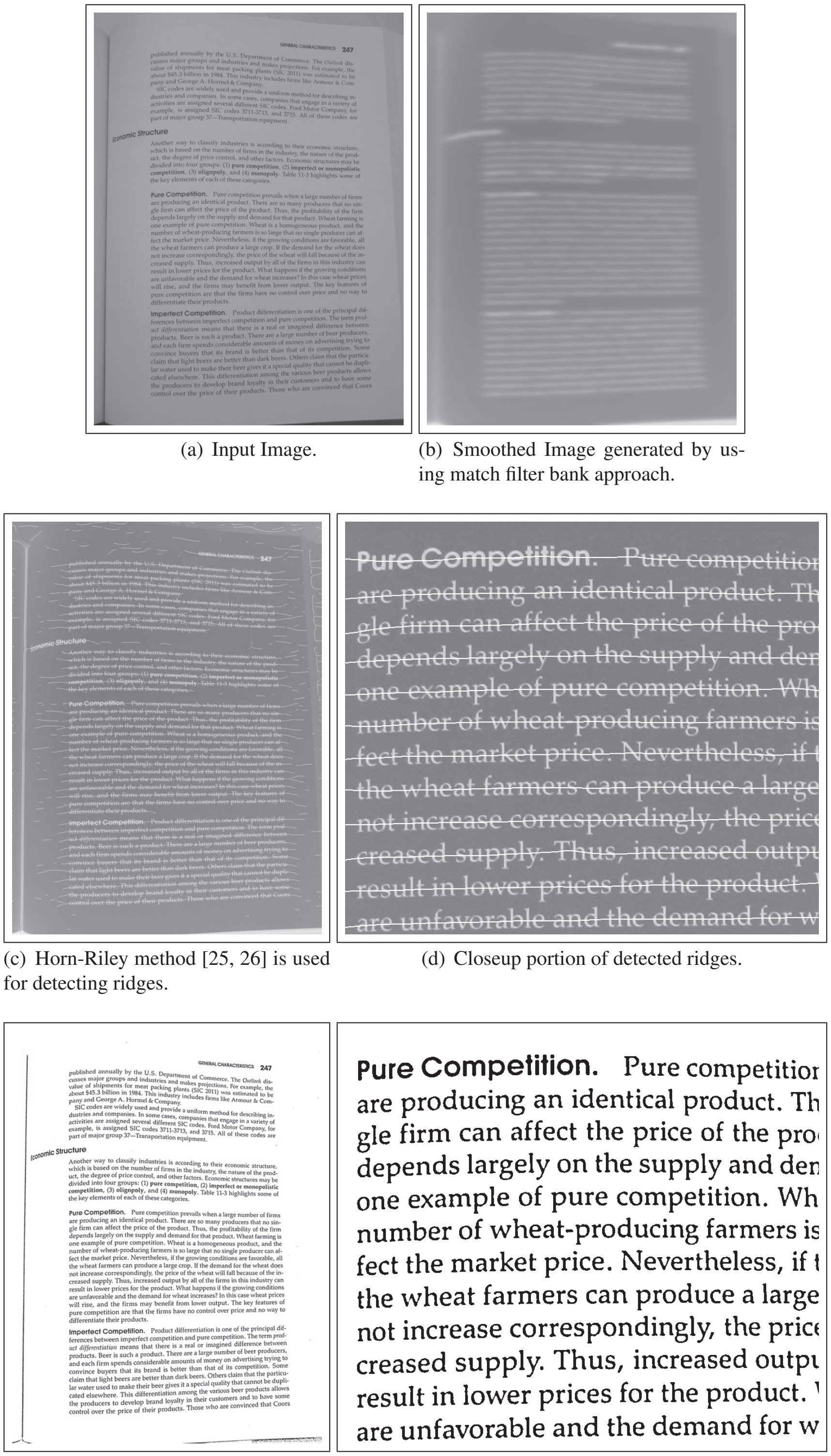
where **k(x,y)** is equal to $0.05$ if ridge(s) is(are) present in the local neighborhood window $w \times w$ window centered around the pixel $(x, y)$, otherwise equal to $0.2$. After thresholding, median filter is applied to remove the salt and pepper noise. Binarization results based on foreground/background guided Sauvola's method are shown in Figures 2(e) and 2(f).

# 3 Experiments and Results

We evaluate our binarization approach on the hand-held camera-captured document images dataset used in CBDAR 2007 for document image dewarping contest [27]. For this purpose, we have selected 10 degraded documents from the dataset. State-of-the-art Otsu's [7] and Sauvola's [2] binarization methods are used for comparative evaluation. The results of Otsu's, Sauvola's and foreground-background guided Sauvola's binarization methods on some example documents are shown in Figure 3.

We compare the OCR error rate of all three binarization methods for 10 selected documents. These documents have non-planar shape, therefore we apply dewarping algorithm[1] on the results of all three binarization methods. Then dewarped documents of all methods are processed through a commercial OCR system **ABBYY Fine Reader 9.0**. After

[1] We have described dewarping method using ridges based coupled-snakes model, which is currently in review phase of CBDAR 2009.



(a) Input Image.

(b) Smoothed Image generated by using match filter bank approach.

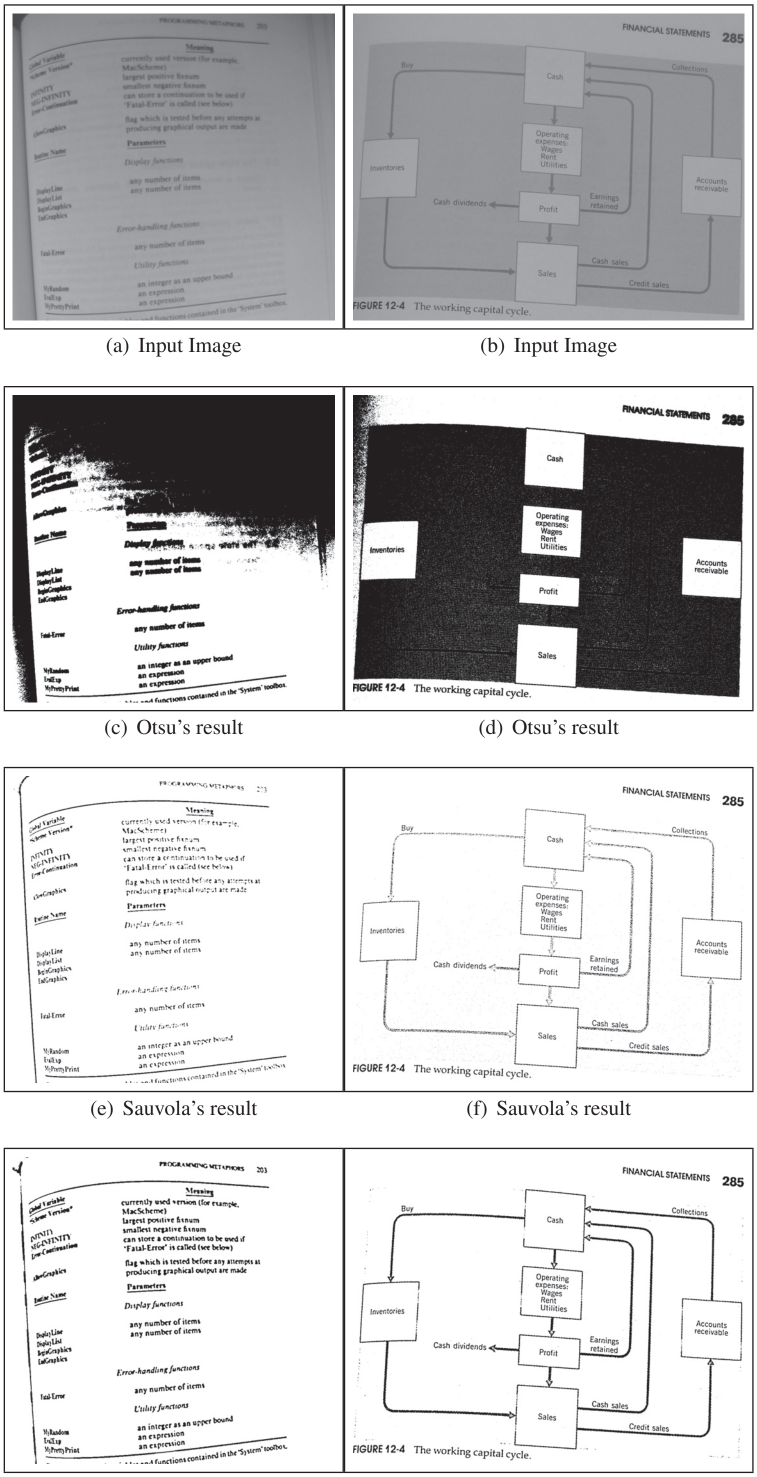
(c) Horn-Riley method [25, 26] is used for detecting ridges.

(d) Closeup portion of detected ridges.

(e) Result of foreground/background guided Sauvola's binarization.

(f) Closeup portion of binarized result.

**Figure 2. Binarization algorithm snapshots.**



(a) Input Image (b) Input Image

(c) Otsu’s result (d) Otsu’s result

(e) Sauvola’s result (f) Sauvola’s result

(g) Guided-Binarization’s result (h) Guided-Binarization’s result

**Figure 3. Binarization results of Otsu [7], Sauvola [2] and our Guided-Binarization. Note that Otsu’s results have large amount of noise. For Sauvola’s binarization we have manually selected the appropriate paramerter values $w = 15$ and $k = 0.15$ for given dataset. Sauvola’s results ($w = 15$,$k = 0.15$) have broken-characters for blured images. Our proposed guided binarization method shows better results for both text and drawing regions, even in the presence of bluring.**

obtaining text from the OCR software, the block edit distance[2] with the ASCII ground-truth has been used as the error measure. Table 1 shows the comparative results of all methods with respect to mean edit distance, median edit distance and the number of documents for each algorithm on which it has the lowest edit distance (in case of tie, all algorithms having the lowest edit distance are scored for that document).

## 4 Conclusion

In this paper we presented a novel way of automatically selecting free parameter values for locally adaptive binarization methods. Local binarization methods, like Niblack's [1] and Sauvola's [2] binarization, use constant values of free parameter for all pixels in the image and are sensitive to these values. We overcome this sensitivity by not using constant values of free parameters for all pixels. We used different free parameter values in Sauvola's methods for foreground and background pixels and achieved promising results for degraded camera-captured documents having blurring and non-uniform illumination. We have also described the simple and efficient way of finding foreground regions of document image using ridges detection.

Comparative results in Figure 3 and Table 1 show that, our guided Sauvola's method outperforms other state-of-the-art global and local binarization methods for degraded documents. Furthermore, our method of selecting free parameter values can also be used with other types of local binarization techniques.

## References

[1] W. Niblack. *An Introduction to Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[2] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[3] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical report, IRST, Trento, Italy, 1998.

[4] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941–954, Jun 2008.

[5] F. Shafait, J. V. Beusekom, D. Keysers, and T. M. Breuel. Structural mixtures for statistical layout analysis. In *Proceedings 8th International Workshop on Document Analysis Systems*, pages 415–422, Nara, Japan, 2008.

[6] F. Shafait, J. V. Beusekom, D. Keysers, and T. M. Breuel. Background variability modeling for statistical layout analysis. In *Proc. 19th International Conference on Pattern Recognition (ICPR)*, 2008. Accepted for publication.

[7] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[8] J. M. White and G. D. Rohrer. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM Journal of Research and Development*, 27(4):400–411, July 1983.

[9] J. Bernsen. Dynamic thresholding of gray level images. In *Proceedings 8th International Conference on Pattern Recognition*, pages 1251–1255, 1986.

[10] L. O'Gorman. Binarization and multithresholding of document images using connectivity. *Graphical Model and Image Processing*, 56(6):494–506, Nov. 1994.

[11] In-Jung Kim. Multi-window binarization of camera image for document recognition. In *Proceedings 9th International Workshop on Frontiers in Handwriting Recognition*, pages 323–327, Washington, DC, USA, 2004.

[12] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3):317–327, 2006.

[13] S. Lu and C. L. Tan. Thresholding of badly illuminated document images through photometric correction. In *Proceedings 2007 ACM symposium on Document engineering*, pages 3–8, Winnipeg, Manitoba, Canada, 2007.

[14] F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proceedings 15th International Conference onDocument Recognition and Retrieval*, volume 6815, page 81510, San Jose, CA, USA, 2008.

[2] http://sites.google.com/site/ocropus/release-notes

**Table 1. OCR error rates of different binarization algorithms on subset of dataset of CBDAR 2007 Document Image Dewarping Contest using ABBYY Fine Reader 9.0.**

| Algorithm | Mean Edit Distance % | Number of documents[a] |
|---|---|---|
| Otsu's Binarization | 6.96 | 2 |
| Sauvola's Binarization[b] | 4.92 | 3 |
| Guided-Binarization | **4.62** | **5** |

[a] Number of documents for each algorithm on which it has the lowest edit distance.

[b] manually selected: ($w = 15$, $k = 0.15$); tested different values for $k$ in between 0.1 to 0.5 and found 0.15 is the best for the given dataset.

[15] K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke. Text extraction from colored book and journal covers. *International Journal on Document Analysis and Recognition*, 2(4):163–176, June 2000.

[16] C.M. Tsai and H.J. Lee. Binarization of color document images via luminance and saturation color features. *IEEE Transactions on Image Processing*, 11(4):434–451, April 2002.

[17] E. Badekas, N. Nikolaou, and N. Papamarkos. Text binarization in color documents. *International Journal of Imaging Systems and Technology*, 16(6):262–274, 2006.

[18] Y. Rangoni, F. Shafait, and T. M. Breuel. Ocr based thresholding. In *Proceedings IAPR Conference on Machin Vision Applications*, Yokohama, Japan, 2009.

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

[20] E. Badekas and N. Papamarkos. Automatic evaluation of document binarization results. In *Proceedings 10th Iberoamerican Congress on Pattern Recognition*, pages 1005–1014, Havana, Cuba, 2005.

[21] S. S. Bukhari, F. Shafait, and T. M. Breuel. Script-independent handwritten textlines segmentation using active contours. In *Proceedings 10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, 2009.

[22] S. S. Bukhari, F. Shafait, and T. M. Breuel. Ridges based curled textline region detection from grayscale camera-captured document images. In *Proc. The 13th International Conference on Computer Analysis of Images and Patterns*, Mnster, Germany, 2009.

[23] S. Chaudhuri, S. Chatterjee, N Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transaction on Medical Imaging*, 8(3):263–269, 1989.

[24] L. O. Gorman. Matched filter design for fingerprint image enhancement. In *Proceedings International Conference on Acoustics, Speech, and Signal Processing*, pages 916–919, New York, NY, USA, 1988.

[25] B. K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. *PhD Thesis, MIT*, 1970.

[26] M. D. Riley. Time-frequency representation for speech signals. *PhD Thesis, MIT*, 1987.

[27] F. Shafait and T. M. Breuel. Document image dewarping contest. In *Proceedings 2nd International Workshop on Camera Based Document Analysis and Recognition*, pages 181–188, Curitiba, Brazil, 2007.