Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 389–397
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

# OCR binarization and image pre-processing for searching historical documents

Maya R. Gupta*, Nathaniel P. Jacobson, Eric K. Garcia

*Electrical Engineering, University of Washington, Seattle, Washington 98195, United States*

Received 28 October 2005; received in revised form 27 February 2006; accepted 28 April 2006

**Abstract**

We consider the problem of document binarization as a pre-processing step for optical character recognition (OCR) for the purpose of keyword search of historical printed documents. A number of promising techniques from the literature for binarization, pre-filtering, and post-binarization denoising were implemented along with newly developed methods for binarization: an error diffusion binarization, a multiresolutional version of Otsu's binarization, and denoising by despeckling. The OCR in the ABBYY FineReader 7.1 SDK is used as a black box metric to compare methods. Results for 12 pages from six newspapers of differing quality show that performance varies widely by image, but that the classic Otsu method and Otsu-based methods perform best on average.


*Keywords:* OCR; Binarization; Multiresolutional; Halftoning

## 1. Introduction

Historical printed documents, such as old books and old newspapers, are being digitized and made available through software interfaces such as web-based libraries. Scanned images of the original documents are usually displayed in grayscale or color for the benefit of human readers, but optical character recognition (OCR) is used to enable keyword searches, document categorization, and other referencing tasks. These documents are challenging for OCR because they use non-standard fonts and suffer from printing noise, artifacts due to aging, varying kerning (space between letters), varying leading (space between lines), frequent line-break hyphenation, and other image problems due to the conversion from print-to-microfiche-to-digital. Commercially, competitive OCR algorithms are designed to interpret bi-level (black and white) images. We consider the problem of denoising and binarizing scanned historical printed documents as a pre-processing step for OCR to enable keyword search. In this work, state-of-the-art commercial OCR is treated as a black box. Comprehensive experiments compare the effectiveness of a number of methods proposed in the literature and some newly developed methods to binarize images for keyword extraction for searching and indexing historical documents. The contributions of this paper are in the large-scale experimental comparisons on real data, computationally simpler versions of proposed methods, analysis of different OCR approaches, and a multiresolutional version of the Otsu method which achieves slightly improved performance.

First, in Section 2, we describe the methods used, some of which have been modified slightly to be significantly more computationally feasible. De-noising is discussed in Section 3. Experiments are detailed in Section 4 with results in Section 5. A discussion about the future of this problem is given in Section 6.

## 2. Binarization methods

Here, we describe the methods compared; two new approaches are detailed, and we explain some modified implementations of previously proposed methods from the

* Corresponding author. Tel.: +1 206 245 0050.
*E-mail address:* gupta@ee.washington.edu (M.R. Gupta).

0031-3203/$30.00 
doi:10.1016/j.patcog.2006.04.043

literature that enable processing large newspaper documents. The binarization methods compared are either recently proposed and promising experimentally, or standard methods that are esteemed by practitioners. The compared methods cover a number of different approaches to the problem, from a fixed global threshold to Markov modeling. A recent review of many other binarization methods for OCR can be found in Ref. [1].

The input grayscale pixels are considered in raster scan order and denoted $x_i \in [0, 1]$. The corresponding output binarization pixels are denoted $b_i \in \{0, 1\}$, where 0 refers to "black" and 1 refers to "white".

### 2.1. Global fixed threshold

The simplest binarization technique is to use a global fixed threshold such that $b_i = 1$ if $x_i \geqslant 0.5$ and $b_i = 0$ if $x_i < 0.5$.

### 2.2. Otsu threshold

Otsu's global threshold method [2] finds the global threshold $t$ that minimizes the intraclass variance of the resulting black and white pixels. This is a standard binarization technique, and was implemented using the built-in Matlab function "graythresh" [3]. Then the binarization is formed by setting $b_i = 1$ if $x_i \geqslant t$ and $b_i = 0$ if $x_i < t$.

### 2.3. Multiresolution Otsu (MROtsu) (new approach)

This is a local version of the Otsu method which considers blocks of pixels at several resolution levels when determining the threshold. The goal is to adapt to changing backgrounds and differing font sizes. The smallest block size is adaptively chosen to be twice the dominant line height $h$ (see Appendix for an algorithm to calculate the dominant line height automatically). This fundamental block size $2h \times 2h$ was designed so that it is large enough to contain intact letters (when located in a text region), but small enough to adapt to background changes. Several larger block sizes are also used $4h \times 4h$, $8h \times 8h$, $16h \times 16h$, $32h \times 32h$, $64h \times 64h$, and the entire image. For each block size, the image is considered to be tiled with adjacent non-overlapping blocks. This means that a $2h \times 2h$ block is not centered in its containing $4h \times 4h$ block, but this speeds up the algorithm significantly over using centered multiresolutional blocks.

Starting at the fundamental (smallest) scale, each binarized block is tested against a hypothesized white/black ratio. Based on preliminary experiments with digitally created documents, a 2:1 ratio was selected. If the local binarized block is too dark, it is assumed to be too small and a larger block size will be needed. This usually occurs inside headline text, pictures, or blank areas.

The binarization consists of the following steps (note that Steps (2)–(5) can be implemented in parallel over the $2h \times 2h$ blocks, and Step (7) is also a parallel operation).

*Step* (1): The image is completely divided up into non-overlapping adjacent blocks of size $2h \times 2h$.

*Step* (2): For each block, an Otsu threshold is calculated based on the pixels in that block.

*Step* (3): The pixels in each block are binarized.

*Step* (4): If the ratio of binarized white pixels to binarized black pixels is less than two, then Steps (2)–(4) are repeated for the next larger block which contains the given block.

*Step* (5): Each $2h \times 2h$ block is assigned the last Otsu threshold calculated for that block.

*Step* (6): Thresholds $t_i$ for each pixel are formed by bilinear interpolation of the thresholds in Step (5).

*Step* (7): Each pixel $x_i$ of the original image is compared to the corresponding threshold $t_i$ to form the binarized pixel $b_i$.

### 2.4. Chang's method

Chang's method was developed for OCR of Chinese characters and makes an adaptive decision between thresholds calculated at different spatial scales [4]. First, a global threshold $t$ is determined using Otsu's method. When the gray value is "far away" from the global threshold, it is classified using the global threshold. The term "far away" was not quantified in Chang's paper. Our preliminary results showed that $0.25\sigma$, where $\sigma$ is the standard deviation of the full image, works well. Thus, we first binarize pixels far from the global threshold:

$$\begin{aligned} b_i &= 1 \quad \text{if } x_i \geqslant t + 0.25\sigma, \\ b_i &= 0 \quad \text{if } x_i < t - 0.25\sigma. \end{aligned} \tag{1}$$

When a pixel gray value is close to the global threshold $t$ a local decision is made instead, as per the following details. First, one determines the scale of local features. This is implemented by convolving the image with a set of "yardstick" vectors $Y_n$ of length $2n$, where $n = \{2, 4, 6, 8\}$. These vectors consist of $n/2$ "1"s followed by $n$ "−1"s followed by $n/2$ "1"s. For example, $Y_4 = \{1, 1, -1, -1, -1, -1, 1, 1\}$.

A set of auxiliary images is produced by convolving the grayscale image with the horizontal (or vertical) yardstick vector at each scale "$n$". The horizontal (or vertical) "reading" of a yardstick at a pixel $x_i$ is then the corresponding pixel in the appropriate auxiliary image.

For the $i$th pixel, the maximum is taken over the horizontal and vertical directions to create $Y_{ni} = \max(\textit{vertical reading}, 0) + \max(\textit{horizontal reading}, 0)$. The scale $n$ of the yardstick that yields the highest overall reading $Y_{ni}$ is defined as the scale for pixel $x_i$. The square window size for $x_i$ is then set to $s + 1$, where $s$ is the chosen scale for that pixel. In Chang's paper, window size is equal to the average scale for all pixels in each connected component (as calculated by the global threshold). In our implementation the optimal scale is described as above for each pixel in turn, which significantly reduces the complexity.

For each pixel $x_i$, one then considers an $(s+1) \times (s+1)$ local window centered on the pixel. Let $\max(\textit{window})$ be the maximum grayscale value of any pixel in the window,

and let min(*window*) be the minimum grayscale value of any pixel in the window. Define the local threshold as $\ell_i = 0.5\ \max(window) + 0.5\ \min(window)$. Then the binarization is $b_i = 1$ if $x_i \geqslant \ell_i$ or $b_i = 0$ if $x_i < \ell_i$.

Chang reported that worst-case performance rose from 26% to 60% accuracy compared to simply using a global Otsu threshold based on two commercial OCR systems and Chinese character documents from 1969.

### 2.5. Sauvola–Niblack method

Sauvola recently presented promising results [1] using a variation of Niblack's binarization [5, p. 115–116]. Niblack's method performed best in an extensive review of OCR of handwritten numbers by Trier and Jain [6]. Niblack proposed that a threshold for each pixel be calculated based on the local mean and local standard deviation. Sauvola's variant of Niblack's method is implemented by dividing the grayscale image into $N \times N$ adjacent and non-overlapping blocks and processing on each block separately. Sauvola explicitly considers a document to be a collection of subcomponents of text, background, and pictures. Only Sauvola's *text binarization method* was applied to these historical documents due to the overwhelming text content. Since binarization of pictures is not tested here, it is assumed that this simplification will not reduce performance. A block size of $64 \times 64$ was used based on preliminary experiments.

For a pixel $x_i$, let $\mu_i$ and $\sigma_i$ be the mean and standard deviation of the grayscale block to which $x_i$ belongs. Then the Sauvola threshold for pixel $x_i$ is given as $\ell_i = \mu_i(1 + k((\sigma_i/R) - 1))$, where $k = 0.5$ and $R = \frac{128}{255}$, as per Sauvola's paper. To reduce the computational load, we calculate $\ell_i$ for the center pixel of each block, and then use bilinear interpolation to determine the threshold $\ell_i$ for non-center pixels. The grayscale image is compared to the threshold image to produce the final binarized image such that $b_i = 1$ if $x_i \geqslant \ell_i$ or $b_i = 0$ if $x_i < \ell_i$.

### 2.6. Margin error diffusion (MarginED) (new approach)

Error diffusion is a binarization process commonly used for halftoning [7]. Halftoning is used in printing to transform a grayscale image into a 1-bit image, where each bit tells the printer whether to print a dot of ink at each location on a page. Error diffusion leads to high-quality halftones. It locally preserves the image mean, and noise added by error diffusion binarization is mostly high-frequency noise. Error diffusion automatically performs some edge sharpening as it binarizes. In signal processing, error diffusion is termed "sigma–delta modulation."

It was hypothesized that error diffusion could be a good approach to OCR binarization because when a pixel is rounded the error (between the rounded pixel and the original image pixel) is passed forward. Passing the error forward allows subsequently binarized pixels to compensate for past binarization errors. In this way, the binarization decisions occur jointly (in a causal manner) between neighboring pixels.

Error diffusion is usually implemented as a raster scan process, so one considers an image to be a long vector where the rows are concatenated and read into the vector from left to right. Error diffusion performs the following operations with respect to some threshold $\ell$ (usually $\ell = 0.5$):

$$\begin{aligned} b_i &= 1 \quad \text{if } x_i + e_i \geqslant \ell, \\ b_i &= 0 \quad \text{if } x_i + e_i < \ell. \end{aligned} \tag{2}$$

The vector $e_i$ is an error vector. It is the sum of weighted inherited errors received from previously quantized pixels:

$$e_i = \sum_{h=1}^{H} (x_{i-h} - b_i) f_h. \tag{3}$$

The error filter $f$ specifies how to weight and pass on the error between a pixel and the binarized version. The Floyd–Steinberg error filter is the most common. The Floyd–Steinberg filter passes on error in the following spatial pattern:

| | | |
|---|---|---|
| | $x_i$ | 7/16 |
| 3/16 | 5/16 | 1/16. |

Note that the filter taps sum to one; this constraint is standard for error diffusion filter design and ensures stability. It does not work to directly run error diffusion in order to do OCR binarization because the image mean is preserved, which translates into very noisy backgrounds if the background does not have zero mean. Instead, similar to Chang's method, the proposed MarginED binarization only error diffuses marginal or uncertain pixels; more certain pixels are quantized outright. The algorithm has four steps as per below and uses an auxiliary image $z$ whose $i$th component is $z_i \in [0, 1]$.

*Step* (1): Initialize $z_i = x_i$ for all $i$. Let $\delta = std(x_i)/2$, $t = OtsuThreshold(x)$.

*Step* (2): $z_i = 0$ for all $x_i < (t - \delta)$.

*Step* (3): $z_i = 1$ for all $x_i > (t + \delta)$.

*Step* (4): $z_i = (z_i - (t - \delta))/2\delta$ for all $(t - \delta) \leqslant x_i \leqslant (t + \delta)$.

*Step* (5): Set $b_i$ to be the $i$th output of the error diffusion of the image $z$, using threshold 0.5 and Floyd–Steinberg filter.

Step (2) clips those pixels which are significantly darker than the Otsu threshold to black. Step (3) clips those pixels which are significantly lighter than the Otsu threshold to white. Step (4) re-normalizes the remaining values to the range between zero and one. Since the extreme pixels have been effectively binarized, the only unbinarized pixels in the auxiliary image are those that were originally close to the Otsu threshold, and these pixels can be considered somewhat uncertain. Error diffusion is then performed on the marginal pixels. The error diffusion could flip a pixel of auxiliary image $z$ that is already binarized, but it is unlikely.

Different error diffusion filters were experimented with in preliminary trials without significantly better results

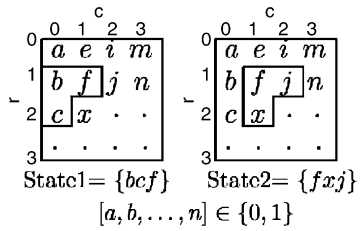



Fig. 1. One state transition.

than the Floyd–Steinberg filter, including traditional error diffusion filters [7], green noise filters [8], and filters specially designed to re-create vertical, horizontal, and diagonal features or letters.

### 2.7. Markov model for OCR binarization

Text occurs in discrete units of letters. Considering the structure of each letter to be independent of its surroundings leads to a model for binarization that decides whether a pixel is black or white based only on the local spatial neighborhood on the scale of a letter. Such a hypothesis can be implemented as a Markov model of the probability that each pixel is text, given its neighboring pixels. Wolf and Doermann recently explored using a non-causal Markov model for this problem [9] that required solving a global optimization problem with multiple minima to learn an optimal model. Their experimental results on artificially degraded images were not, as they noted in their conclusion section, a significant improvement over previous methods. However, Markov models can be a powerful tool for modeling. Thus, to explore the possibilities, we implemented a similar Markov model, but made it causal in order to reduce computational complexity.

Since there is no inherent notion of causality in an image, the processing is imposed in a raster scan. The state space is then defined to include only pixels in the "past" (above or to the left of the current pixel). A causal Markov model implies that, given the entire past of an image $z$, the distribution of pixel $x$ is given by $Pr(x|z) = Pr(x|N)$, where $N$ is a pre-defined neighborhood around $x$. $N$ is chosen here as the group of 3 pixels to the upper left of $x$ as in Fig. 1.

Our research showed that the Markov model could in fact model text curves and serifs well, but since similar curves and serifs and other text features can occur in a variety of places in a font, this successful modeling exhibited extreme behavior: in some places a letter was correctly reconstructed based on very little (Fig. 2, sample 1); in other places, extra serifs appeared where they should not have existed (Fig. 2, sample 2).

These problems seem to be inherent in Markov modeling of a font. Due to its poor performance in preliminary experiments, our lack of confidence that the model could work in a computationally feasible format for this problem, and the necessity of labeled data (or expert information) needed to train the model for a new document, further experiments were not run for Markov model approaches.

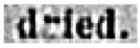
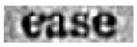
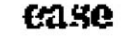
| Method Name | Sample 1 | Sample 2 |
|---|---|---|
| Original Image |  |  |
| Markov Model |  |  |
| Global Fixed Threshold |  |  |

Fig. 2. Markov model binarized image samples.

## 3. Post-processing and pre-filtering

In addition to these binarization methods, some post-processing and pre-filtering methods were explored.

### 3.1. Post-binarization despeckling

The type of noise and artifacts seen in the historical documents is varied but tends to be characterized by splotches, specks, and streaks. A post-binarization denoising step was used to eliminate such non-text objects from the image.

The line height (see Appendix for algorithm) is used to derive the minimum number of pixels that a *black blob* must have to be considered text and not noise. A *black blob* is defined as a group of eight-connected black pixels. Blobs with smaller than the *minimum area* will be classified as noise and removed, where the *minimum area* is calculated as

$$\text{minimum area} = \frac{1}{k}(\text{line height})^2.$$

Setting $k = 172$ will erase any blobs smaller than a punctuation mark in an image of "Times New Roman" font text. Similarly, setting $k = 37$ will erase blobs smaller than a full letter, such as the dot of an "i", but the main stem would remain. Setting a value as extreme as $k = 20$ was found to work well in preliminary tests. In poor quality images, the benefits of removing more noise compensates for the loss of some letters.

The results of using these parameter values are shown in Table 2.

### 3.2. Wavelet denoising

Wavelet denoising by shrinking wavelet coefficients is provably optimal for additive white Gaussian noise where the noise variance is known [10]. It is clear that the noise and artifacts present in scanned historical documents are not white Gaussian, and thus do not statistically fit the wavelet denoising theory. However, it was hypothesized that the capability of wavelet representations to preserve edge

information would improve the quality of the grayscale image and possibly be a useful pre-processing step. Wavelet denoising was applied to denoise the grayscale images prior to binarization. Preliminary experiments showed that this approach was in fact unpromising, and this method was not explored further.

## 4. Experiments

We compared the binarization methods on a varied set of real historical printed documents. The test set consisted of 12 images scanned from archival newspapers by independent entities using different equipments; some scans were from original prints and others from microfiche. The newspapers used were the San Juan Record, Barrington Review, Millard Progress, Progress Review, Salt Lake Times, and East Side Journal. Two full newspaper pages were selected from each publication. To create a ground truth (correct transcription), these pages were transcribed by human typists and then checked for errors by independent reviewers. Examples of the historical documents are shown in Fig. 3. The number of words in each document is listed in Table 2; there were 57,305 words total.

Most images were either the original 8-bit scans or 4-bit quantized versions of 8-bit scans in TIFF format. These images were labeled with the original scan resolution. The remaining images, from the San Juan Record and Barrington Review, were obtained as 8-bit JPEG images at a reduced resolution. We suspect that they were originally scanned at 300 or 400 dpi, then downsampled to decrease file size. The San Juan Record images were estimated to be 165 dpi, while the Barrington Review images were estimated to be 95 dpi. These values were inferred from the known physical dimensions of that newspaper and the pixel dimensions of the images. The original documents were no longer available, and the challenge was to implement OCR on these low resolution digital images. The archiving process can easily result in such sub-optimal versions of documents being the only option if the original documents have been lost, discarded, or misfiled.

The OCR software used for the comparisons was the ABBYY Finereader 7.1 SDK Engine [11].

### 4.1. Metrics

The primary goal for document indexing is to build an index of all keywords in a document, along with the locations of each keyword. This differs from the general application of OCR, which has the goal of transcribing digital text documents for direct replacement of the original paper documents.

The accuracy metric used to rank the performance of the different methods is *recognition rate*, similar to that defined by O'Gorman [12,13]. O'Gorman defined recognition rate to be the percentage of words in the document with all characters correctly recognized.

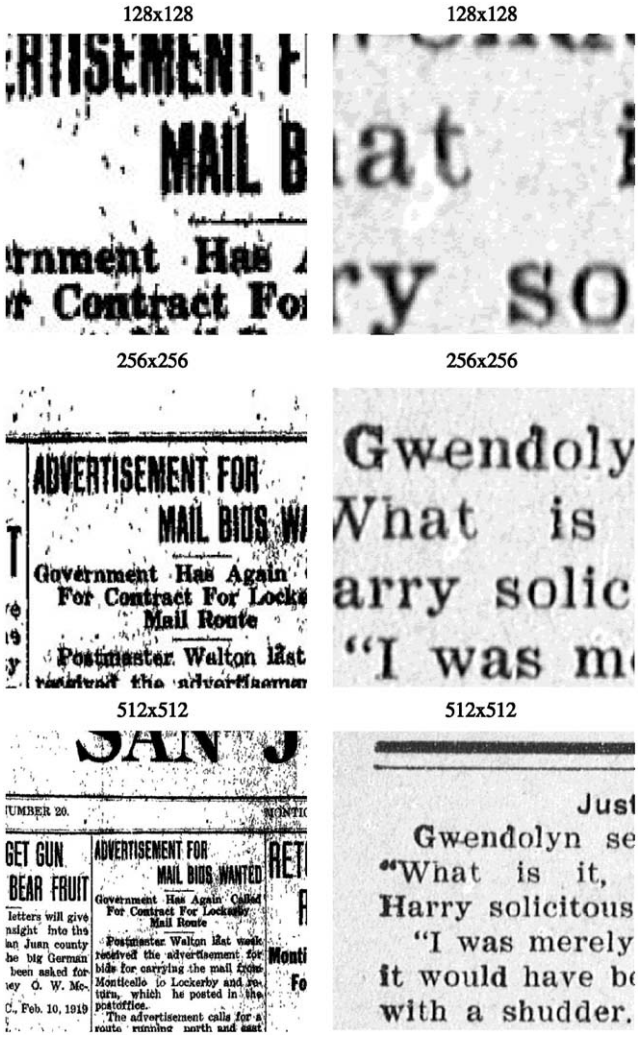


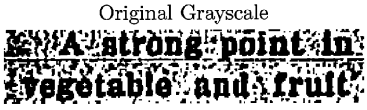

Fig. 3. Left: sample crops from an original grayscale San Juan Record document, a low image-quality newspaper from the test set. Some JPEG compression artifacts are visible. Right: sample crops from an original grayscale Progress Review document, a medium image-quality newspaper from the test set. This image had been reduced to 4-bit grayscale.

We found that some short, frequently used words appear more times in the OCR transcript than in the ground truth. This happens when words are split across lines or difficult to recognize. For example, the word *the* may be over-recognized if the OCR recognizes partially obscured words like *their* or *father* as the word *the*. In order to minimize counting these incorrect matches as correct, the recognition rate is calculated:

$$\frac{\sum_{j=1}^{J} \min(groundtruth(j), OCR(j))}{\sum_{j=1}^{J} groundtruth(j)}, \tag{4}$$

where $j = 1, \ldots, J$ indexes the unique words in the document, $groundtruth(j)$ is the number of times the $j$th unique word is found in the ground truth, and $OCR(j)$ is the number of times the $j$th unique word is found in the OCR transcript.

Original Grayscale



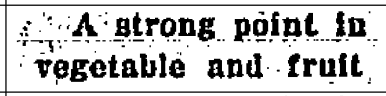
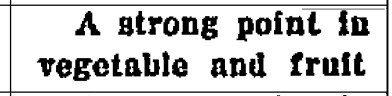
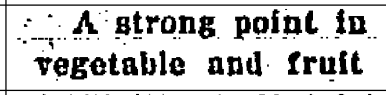
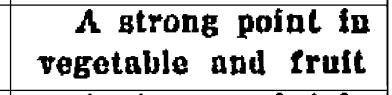
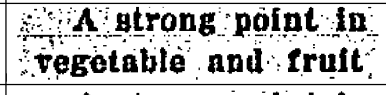
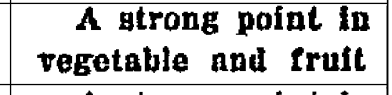
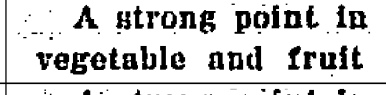
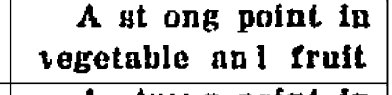
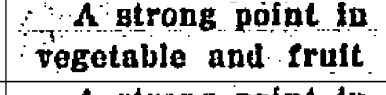
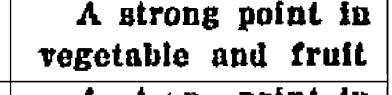
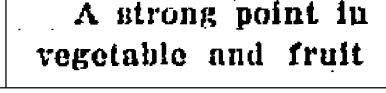
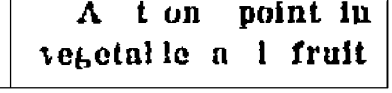

| Method Name | No Despeckle | k=20 Despeckle |
|---|---|---|
| Chang |  |  |
| Global Fixed |  |  |
| MarginED |  |  |
| MROtsu |  |  |
| Otsu |  |  |
| Sauvola Niblack |  |  |

Fig. 4. Binarized image samples.

The above formula is not perfect; for example, a short word could be unrecognized five times, and incorrectly recognized five times, and these errors would counteract in the above recognition rate formula. However, we found this formula to preserve the truth more accurately than using the sum of correct words in the OCR transcript as the numerator of the recognition rate. The remaining errors are few, and do not appear to be biased in favor of any particular method. To fully correct these errors a text-registration method would be needed to determine when OCR is recognizing a word that is not the correct word (such as "their" as "the"). A major obstacle to such a calculation is that the OCR application does not always recognize column breaks, and thus the OCR transcript may proceed across columns. Since the focus of this paper is on OCR for keyword searches and not on accurate paragraph reconstruction, missing column breaks do not affect the OCR's ability to identify keywords.

### 4.2. Examples

Fig. 4 shows examples of the different binarization methods without despeckling and after despeckling with parameter setting $k = 20$.

The different binarization algorithms result in different amounts of speckle noise. Much of this noise can be removed by our despeckling technique, for example, the MarginED technique produces a noisy image but the noise is effectively removed by the despeckling. Another obvious problem is the Sauvola–Niblack method, which has chosen fewer pixels to blacken. This causes some letters in the original to be thin, and cross-marks on letters like "t" and "e" are reduced or missing. Despeckling is not well matched to this method, as it identifies some disconnected letter parts as speckle noise and removes them. The MROtsu also has some broken letters, but the effects of despeckling are not as severe.

Items to compare include the cross-line on the capital "A", some algorithms miss this, some capture it. The "t"s should have visible cross-marks as well. The "n"s should be open on the bottom. Dots on "i"'s should be well separated. The letter "e" should have a clear horizontal line and be open in the bottom right. In many fonts there is a cross-line in the lower-case "a" which is helpful to distinguish it from an "o".

## 5. Results

The results show that the performance of a particular method can widely vary but some useful trends can be distilled. Table 1 ranks the top five methods by median and

Table 1
Ranking of methods based on median and mean word recognition rate, for all images

| Median | Mean |
|---|---|
| 0.514 MROtsu $k = 172$ | 0.557 MROtsu |
| 0.504 Otsu | 0.531 Otsu |
| 0.503 MarginED $k = 172$ | 0.528 MarginED |
| 0.493 Chang | 0.525 Chang |
| 0.436 Sauvola–Niblack | 0.466 Sauvola–Niblack |
| 0.369 Global fixed $k = 172$ | 0.454 Global fixed |

For brevity, only the best performing despeckle setting is shown for each method.

Table 2
Word recognition rates

| Binarization | Despeckling $k$ value | (165 dpi, 2591 words) sjr-2 | (165 dpi, 3749 words) sjr-3086 | (95 dpi, 4664 words) barrington-130 | (95 dpi, 3331 words) barrington-191 | (400 dpi, 3132 words) millard 0003 | (400 dpi, 7475 words) millard 0004 | (400 dpi, 1967 words) progress 0312b | (400 dpi, 1287 words) progress 0313a | (300 dpi, 10179 words) salt lake times 5 | (300 dpi, 9377 words) salt lake times 6 | (300 dpi, 1 4860 words) east side journal 1 | (300 dpi, 4693 words) east side side journal 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Chang | — | 0.543 | 0.280 | 0.339 | 0.443 | 0.210 | 0.586 | 0.885 | 0.858 | 0.244 | 0.286 | 0.864 | 0.769 |
| | 172 | 0.541 | 0.281 | 0.339 | 0.443 | 0.162 | 0.593 | 0.872 | 0.855 | 0.246 | 0.264 | 0.859 | 0.760 |
| | 37 | 0.543 | 0.287 | 0.371 | 0.466 | 0.076 | 0.497 | 0.707 | 0.663 | 0.220 | 0.259 | 0.795 | 0.732 |
| | 20 | 0.496 | 0.287 | 0.401 | 0.476 | 0.061 | 0.449 | 0.516 | 0.412 | 0.192 | 0.239 | 0.646 | 0.675 |
| Global fixed (0.5) | — | 0.550 | 0.271 | 0.323 | 0.414 | 0.035 | 0.193 | 0.871 | 0.831 | 0.160 | 0.192 | 0.852 | 0.756 |
| | 172 | 0.547 | 0.278 | 0.323 | 0.414 | 0.021 | 0.191 | 0.853 | 0.791 | 0.158 | 0.196 | 0.850 | 0.767 |
| | 37 | 0.524 | 0.280 | 0.346 | 0.424 | 0.022 | 0.212 | 0.613 | 0.364 | 0.144 | 0.177 | 0.790 | 0.739 |
| | 20 | 0.499 | 0.303 | 0.366 | 0.442 | 0.022 | 0.213 | 0.356 | 0.236 | 0.118 | 0.162 | 0.636 | 0.700 |
| MarginED | — | 0.547 | 0.261 | 0.325 | 0.460 | 0.184 | 0.611 | 0.888 | 0.871 | 0.270 | 0.284 | 0.881 | 0.759 |
| | 172 | 0.502 | 0.270 | 0.387 | 0.489 | 0.034 | 0.432 | 0.534 | 0.425 | 0.199 | 0.242 | 0.690 | 0.686 |
| | 37 | 0.542 | 0.259 | 0.360 | 0.456 | 0.080 | 0.484 | 0.718 | 0.687 | 0.225 | 0.267 | 0.817 | 0.746 |
| | 20 | 0.547 | 0.261 | 0.325 | 0.460 | 0.170 | 0.566 | 0.875 | 0.849 | 0.243 | 0.287 | 0.876 | 0.779 |
| MROtsu | — | 0.568 | 0.277 | 0.328 | 0.461 | 0.296 | 0.637 | 0.888 | 0.877 | 0.328 | 0.356 | 0.883 | 0.788 |
| | 172 | 0.521 | 0.281 | 0.389 | 0.507 | 0.171 | 0.544 | 0.646 | 0.419 | 0.277 | 0.295 | 0.726 | 0.732 |
| | 37 | 0.525 | 0.309 | 0.374 | 0.499 | 0.203 | 0.587 | 0.772 | 0.676 | 0.301 | 0.317 | 0.822 | 0.76 |
| | 20 | 0.568 | 0.277 | 0.328 | 0.461 | 0.227 | 0.640 | 0.885 | 0.868 | 0.325 | 0.34 | 0.880 | 0.782 |
| Otsu | — | 0.554 | 0.297 | 0.349 | 0.454 | 0.210 | 0.586 | 0.885 | 0.858 | 0.244 | 0.286 | 0.878 | 0.769 |
| | 172 | 0.551 | 0.294 | 0.349 | 0.454 | 0.162 | 0.593 | 0.872 | 0.855 | 0.246 | 0.264 | 0.876 | 0.768 |
| | 37 | 0.522 | 0.278 | 0.394 | 0.494 | 0.076 | 0.497 | 0.707 | 0.663 | 0.220 | 0.259 | 0.810 | 0.727 |
| | 20 | 0.481 | 0.288 | 0.408 | 0.481 | 0.061 | 0.449 | 0.516 | 0.412 | 0.192 | 0.239 | 0.667 | 0.679 |
| Sauvola–Niblack | — | 0.558 | 0.301 | 0.307 | 0.379 | 0.102 | 0.494 | 0.865 | 0.800 | 0.105 | 0.109 | 0.834 | 0.738 |
| | 172 | 0.564 | 0.308 | 0.307 | 0.379 | 0.073 | 0.483 | 0.842 | 0.756 | 0.112 | 0.111 | 0.846 | 0.719 |
| | 37 | 0.509 | 0.278 | 0.333 | 0.382 | 0.016 | 0.210 | 0.567 | 0.325 | 0.095 | 0.117 | 0.718 | 0.647 |
| | 20 | 0.475 | 0.288 | 0.334 | 0.375 | 0.006 | 0.154 | 0.310 | 0.196 | 0.074 | 0.092 | 0.508 | 0.513 |

mean recognition rate performance based on all images. Averaged over all images, little or no denoising performs best, with Otsu-based methods leading, including the proposed MROtsu method, the original global Otsu threshold, and the Otsu-based MarginED method. Performing roughly 13% worse than the Otsu method, the Sauvola–Niblack is ranked fifth by both median and mean. The global fixed threshold gives a baseline for the comparisons.

In several cases, OCR is extremely poor. One particular source image, "millard3", had less than 5% recognition rate for some methods, such as the global fixed threshold and Sauvola–Niblack. The despeckling algorithm always decreased mean recognition rate, but sometimes improved median recognition rate.

Table 2 gives all the raw data on the recognition rate for each combination of source image, binarization method, and post-binarization denoising.

## 6. Discussion

The performance of each method is an indication of the usefulness of the method's underlying assumptions. For example, the global fixed method assumes that the scanned gray levels are very well normalized. The Sauvola–Niblack method uses parameters $k$ and $R$ that may not be optimal for the statistical properties of historical documents. The Chang method assumes that the ideal threshold is exactly midway between the lightest and darkest pixels in a block.

The methods based on the Otsu algorithm do best. The main assumption behind Otsu approaches is that there are two classes of pixels, which is always true globally, leaving only problems with local illumination. The multiresolution variant (MROtsu) adapts to illumination changes by making local decisions, using the assumption of a 2:1 white/black ratio to establish the smallest valid scale where two classes are present. This assumption is valid for this data set, and should hold for most printed text. The error diffusion method was competitive, but its error passing did not improve upon the global Otsu threshold upon which it was based.

The better performance of the Otsu method compared to the Sauvola–Niblack method is surprising, given the contrary results in Refs. [1,6]. There are several possible contributions to this:

- Purely local algorithms cause undesired results in background areas, affecting text-detection capabilities of the OCR black box.
- The simplification of using blocks instead of a sliding window may reduce the ability to adapt to sharp illuminance changes (this also affects the MROtsu method and the implementation of the Chang method).
- Many images have poor resolution, quantization, or are noisy.
- In general, the Sauvola–Niblack images were too light, causing more broken characters.

In summary, the underlying assumption behind the Otsu method appears to best model the truth behind historical printed documents, based on results over a broad range of historical newspapers. The proposed multiresolutional version has been shown to consistently improve performance. More complex approaches to the binarization and image preprocessing were not shown to be as useful as might theoretically be expected.

More broadly, this research leads us to hypothesize that the additional gains available in binarization with a black box OCR are limited. Larger gains are likely with a system that uses a feedback loop with the OCR determining confidence, or a system with human input for training, or using multiple local binarizations and a natural language processing module after the OCR to make final decisions.

## Acknowledgments

This work was supported in part by DiMeMa Inc. and in part by an Intel GEM Fellowship. The authors would like to thank Greg Zick for helpful discussions.

## Appendix. Dominant line height detection

The line height of a text image is the number of pixels between the bottom of one line of text and the bottom of the next line of text. We assume that the smallest text in historical documents, such as newspapers, covers most of the page. The periodic spacing of this majority text is used to determine the dominant line height, which is calculated from the vertical frequency spectrum of the image. For computational efficiency, 512 evenly spaced columns of pixels from the image are selected. The average vertical frequency spectrum is computed by averaging the magnitude of the discrete Fourier transform of each column.

The vertical frequency spectrum in Fig. 5 is characteristic of documents considered in this paper, and should hold for any image with a significant amount of text with uniform line height. The largest magnitude peak is centered at 0 cycles per pixel, which is caused by large-scale features of the document, such as transitions between photographs, text blocks, and white space.

The second largest magnitude peak is also the peak with the next lowest frequency. This fundamental vertical frequency is the inverse of the line height in all document images examined. This is reasonable since the presence of text at a regular spacing is the main periodic feature of a document image.

Higher frequency peaks (harmonics) occur at integer multiples of the "line height frequency", forming the fine detail and sharp edges of individual letters.

For this study, the lowest frequency peak was assumed to correspond to the greatest magnitude of the vertical

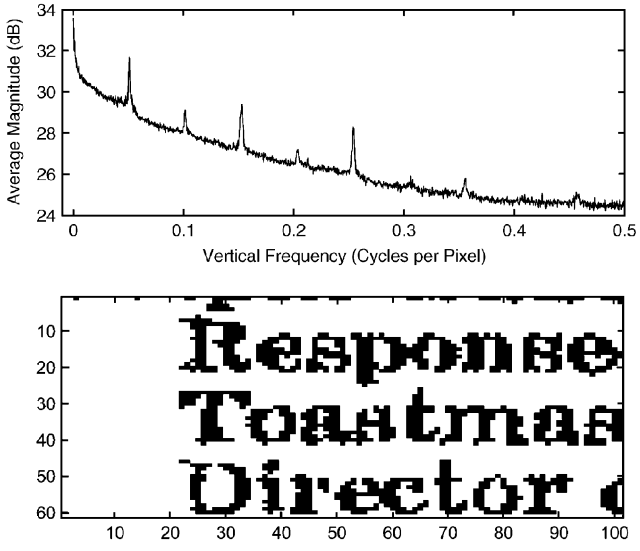



Fig. 5. *Top*: Vertical frequency spectrum for document "sjr-2" after Otsu threshold. For this document, the fundamental vertical frequency peak is at 0.0508 cycles per pixel. Its inverse, 19.7 pixels per cycle, is the detected dominant line height. *Bottom*: Small subsection of document "sjr-2" after Otsu threshold, showing a line spacing of approximately 20 pixels.

frequency spectrum at 0.0166 cycles per pixel or greater (60 pixel line height or less). This was found to effectively mask the larger 0 cycles per pixel peak. For higher resolution scans, a lower minimum frequency would be appropriate. If scan resolution is unknown, a different peak finding method would be needed.

## References

[1] J. Sauvola, M. Pietikainen, Adaptive document image binarization, Pattern Recognition 33 (2) (2000) 225–236.
[2] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[3] Matlab version 7.0 by Mathworks, 2005 Available at ⟨http://www.matlab.com⟩.
[4] F. Chang, Retrieving information from document images: problems and solutions, Int. J. Doc. Anal. Recognition 4 (2001) 46–55.
[5] W. Niblack, An Introduction to Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[6] A.J.O. Trier, Goal-directed evaluation of binarization methods, IEEE Trans. Pattern Anal. Mach. Intell. 17 (12) (1995) 1191–1201.
[7] H.R. Kang, Digital Color Halftoning, SPIE Press, Bellingham, 1999.
[8] D.L. Lau, G.R. Arce, N.C. Gallagher, Green-noise digital halftoning, Proc. of the IEEE 86 (1998) 2424–2444.
[9] C. Wolf, D. Doermann, Binarization of low quality text using a Markov random field model, in: Proceedings of the 16th International Conference on Pattern Recognition, vol. 3, 2002, pp. 160–163.
[10] D.L. Donoho, I.M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, J. Am. Stat. Assoc. 90 (1995) 1200–1224.
[11] ABBYY Finereader 7.1 SDK, ⟨http://www.abbyy.com⟩.
[12] L. O'Gorman, Experimental comparisons of binarization and multithresholding methods on document images, in: Proceedings of the IAPR International Conference on Pattern Recognition, vol. 2, IEEE, 1994, pp. 395–398.
[13] L. O'Gorman, Binarization and Multithresholding of document image using connectivity, in: CVGIP: Graphical Models and Image Processing, vol. 56, No. 6, 1994, pp. 494–506.