1-1-2010

Effect of Pre-Processing on Binarization

Elisa Barney Smith
Boise State University

Laurence Likforman-Sulem
Telecom ParisTech

Jérôme Darbon
University of California - Los Angeles

Effect of Pre-processing on Binarization

Elisa H. Barney Smith[a] and Laurence Likforman-Sulem[b] and Jérôme Darbon[c]

[a]Boise State University, Boise, Idaho, USA;
[b]Telecom ParisTech, Paris, France
[c]UCLA, Los Angeles, California, USA

ABSTRACT

The effects of different image pre-processing methods for document image binarization are explored. They are compared on five different binarization methods on images with bleed through and stains as well as on images with uniform background speckle. The binarization method is significant in the binarization accuracy, but the pre-processing also plays a significant role. The Total Variation method of pre-processing shows the best performance over a variety of pre-processing methods.

Keywords: Image Binarization, Image Filtering, Image Regularization, Total Variation, Wiener Filters

1. INTRODUCTION

Image binarization is an important process for document image analysis. The inherently bilevel nature of text documents has led to many of the document analysis algorithms being designed for use on bilevel images. If the image binarization is improperly done, then the follow-on steps cannot proceed appropriately.

Many studies of binarization have been completed[1].[2] These surveys focus on the binarization algorithm. The algorithms have evolved from global thresholding to local adaptive thresholding to allow for variations in the image background and today range from relatively simple algorithms, to some that are rather complex. Some algorithms are a hybrid of local and global by using background shading estimation.[3] The choice of 'best' binarization algorithm usually depends on the other constraints of the problem. The images can have stains or bleed through, or a uniformly noisy background. The text can suffer additional degradations. Algorithms that work best on one image may not be best for another.

The methods used to evaluate binarization algorithms have varied. Evaluation based on OCR accuracy[1] gives information on what quality of binarization is needed for use, but the tolerance of the follow-on system to noise or errors means some of the binarization errors are not evaluated. To directly compare the binarization with the true image it is common to use synthetic data.[4] This makes degraded and ground truth data easy to acquire. Here the concern is that the degradations may not reflect the images that are encountered and for which binarization algorithms are called to work. Real gray-scale images that present a fair amount of "challenge" to the binarization system can be used,[5] but they are difficult to ground truth and are not available in large quantities.

In the study of binarization algorithms, the possibility of varying the pre-processing of the image is usually overlooked. This step is particularly important for the functionality of the algorithms. This paper compares several different possibilities for image pre-processing. This pre-processing step is applied to a variety of binarization algorithms. In total 17 pre-processing algorithms or algorithm variations are evaluated on 16 images with ground truth and 5 binarization methods.

In Section 2 the filters that are evaluated are introduced. The binarization methods that are used in the evaluation are described in Section 3. The data is described in Section 4. Section 5 describes the tests that were run and their results. The paper concludes in Section 6.

Further author information: EBS: ebarneysmith@boisestate.edu, LLS: likforman@telecom-paristech.fr, JD: jerome@math.ucla.edu

2. PRE-PROCESSING FILTERS

Five categories of pre-processing filters were evaluated in this study: mean filter, median filter, Wiener filter, the Total Variation filter and the Non-local Means filter. Most of these categories had a selection of variations in implementation that were also evaluated. The use of a pre-processing filter was compared against the possibility of no pre-processing as a control. The filters that were evaluated are described next.

Mean Filter. A uniform convolutional averaging filter was implemented. It was implemented with both a 3x3 window and a 5x5 window, as the window size strongly affects the resulting noise reduction and the associated text contrast.

Median Filter. Two-dimensional median filters with both 3x3 and 5x5 windows were evaluated.

Wiener Filter. The Wiener filter implemented in the spatial domain was also evaluated. The filtered image is computed through

$$I_{filt}(x,y) = \mu + \frac{(\sigma^2 - v^2)(I_{orig}(x,y) - \mu)}{\sigma^2}, \tag{1}$$

where μ and σ^2 are the local mean and variance respectively and v^2 is the estimate of the noise variance.

Here in addition to varying the window within which the local statistics are calculated from 3x3 to 5x5, the method to estimate the noise variance was also varied. A Wiener filter is used by Gatos et al.[3] in their binarization technique. This targets the images with stains or other spatial varying noise. The paper states that v^2 is "the average of all estimated variances for each pixel in the neighborhood." The Gatos binarization technique was implemented for the Gamera binarization toolbox as well.[6] There the implementation of the Wiener filter used a median of all the standard deviations calculated for the whole image. In this paper both the Gatos implementation and the Gamera implementation are evaluated. They were both implemented with both 3x3 and 5x5 windows.

Total Variation. The use of Total Variation (TV) as a pre-processing filter was found to improve OCR results on an OCR system that applied an Otsu threshold to all incoming images.[7] Thus its comparison with other pre-processing filters for image binarization was initiated. The Total Variational formulation of the image restoration problem consists of minimizing a weighted combination of the data fidelity and the regularization (also called prior). The data fidelity D measures how far the current solution I_{filt} is from the observed image I_{orig} based on the nature of noise that corrupts the image. As for the Wiener filter, it is assumed here that the noise is Gaussian additive and independently and identically distributed and thus yields a quadratic data fidelity term. The data fidelity is defined as

$$D(I_{filt}|I_{orig}) = \frac{1}{2}\sum_{(x,y)} (I_{filt}(x,y) - I_{orig}(x,y))^2 \ . \tag{2}$$

The main characteristics of the TV prior is that the solution lives in the space of functions of Bounded Variation that allows for sharp edges and discontinuities. Total Variation[8] is defined as the weighted l^1-norm of a discrete gradient,

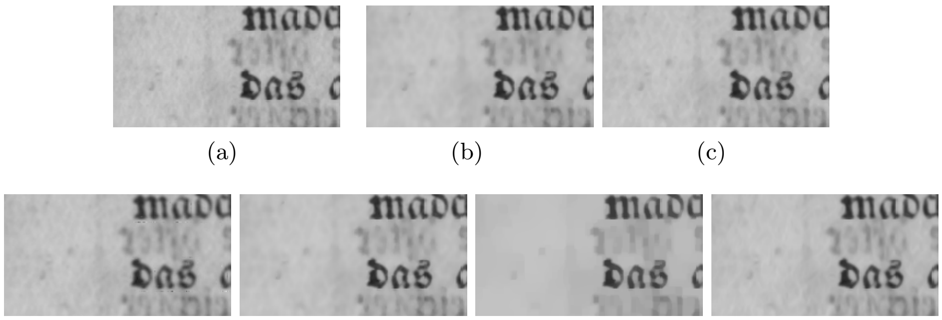
$$TV(I_{filt}) = \sum_{(x_s,y_s)} \sum_{(x_t,y_t)\in N(x_s,y_s)} w_{st} \ \left| I_{filt}(x_s,y_s) - I_{filt}(x_t,y_t) \right| \ , \tag{3}$$

where w_{st} are some non-negative coefficients and $N(x_s, y_s)$ denotes the set of pixels that are the 4-nearest neighbors to the site (x_s, y_s) and all the weights w_{st} are set to 1.

The restored image $\widehat{I_{filt}}$ is the minimizer of the energy $E(I_{filt}|I_{orig})$ that is a weighted combination of the data fidelity and total variation terms from Equations 2 and 3,

$$E(I_{filt}|I_{orig}) = D(I_{filt}|I_{orig}) + \beta \ TV(I_{filt}), \tag{4}$$

where the parameter β is a non-negative coefficient that governs the balance between the data fidelity and the regularization. A large value for β will produce an image with few details, often removing small features that can be text and reducing the contrast, while a tiny one will yield an image that leaves I_{orig} almost unchanged. Values of $\beta = 2$, 5, 10, 20 were used in this study.



(a) (b) (c)

(d) (e) (f) (g)

Figure 1. Samples of the effect of the pre-processing algorithms (a) The original image, (b) 3x3 mean filter, (c) 3x3 median filter, (d) 3x3 Wiener filter with Gatos implementation, (e) 5x5 Wiener filter with Gamera implementation, (f) Total variation filter with $\beta = 10$, (g) Non-Local Means filter.

Non-local Means (NLmeans). The NLmeans method averages neighboring parts of the central pixel, similar to a two-dimensional convolution, but instead of a spatially invariant weight based on the location relative to the target point, the averaging weights depend on the similarities between a small patch around the pixel and the neighboring patches within a search window.[9] This capitalizes on the redundancy present in many images, and this redundancy is particularly present in document images. The NLmeans filter considers the similarity of a block of neighboring pixels to the block centered on the pixel under evaluation. Such a block is a squared patch whose side is $(2P+1)$, denoted by Δ. The similarity measure $w(s,t)$ for the two sites $s = (x_s, y_s)$ and $t = (x_t, y_t)$ is defined as

$$w(s,t) = g_h\left(\sum_{\delta \in \Delta} \left(I_{orig}(s+\delta) - I_{orig}(t+\delta)\right)^2\right) \quad , \tag{5}$$

where $g_h(x) = \frac{1}{h}$. The parameter h is used to control the amount of filtering. The filtered image is then produced by

$$I_{filt}(x_s, y_s) = \frac{1}{Z(s)} \sum_{t \in N(s)} w(s,t) I_{orig}(x_t, y_t), \tag{6}$$

where $Z(s)$ is a normalizing factor chosen so that $w(s,t)$ has area of 1.

Examples of how these filters process some document images are shown in Figure 1. The amount of background smoothing depends on the implementation parameters. Mean filters are very prone to smoothing the character edges as well as the background regions. The median filter is less likely to do so, but still often will. The Gatos implementation of the Wiener filter also produces a fair amount of edge blurring. The Gamera implementation of the Wiener filter and the Total Variation method are less likely to, but this depends on the choice of parameters used.

3. BINARIZATION

In the attempt to separate the effect of the pre-processing from the binarization algorithm, five different binarization algorithms were implemented. While these by no means produce an exhaustive survey, the selection was designed to be diverse and to provide a good basis on which to draw preliminary conclusions.

The binarization algorithms that were evaluated include Otsu, Niblack, Sauvola, Gatos and a background estimation and subtraction algorithm the authors of this paper submitted to the DIBCO 2009 contest.[5] Descriptions of these binarization algorithms follow.

Otsu is an often used global thresholding method.[10] It is based on treating the gray level intensities present in the image as values to be clustered into two sets, one foreground (black) and one background (white). To accomplish this the algorithm minimizes the weighted sum of within-class variances of the foreground and background pixels to establish an optimum threshold. This is equivalent to maximizing the between-class scatter. From this a scalar number, K, is returned. This is then used to binarize the image through

$$I_{bin}(x,y) = \begin{cases} 1, & \text{if } I_{gray}(x,y) \leq K \\ 0, & \text{if } I_{gray}(x,y) > K \end{cases}. \tag{7}$$

Niblack is a local adaptive thresholding algorithm. The threshold for each pixel is determined by examining the average of the pixels in a neighborhood, $m(x,y)$, and the standard deviation, $\sigma(x,y)$, in that same neighborhood. The threshold for Niblack is then chosen as

$$T(x,y) = m(x,y) + k \cdot \sigma(x,y). \tag{8}$$

Instead of a global threshold K as in Equation 7 each pixel is subjected to the threshold process separately. The most common value for the constant k is -0.2, which is what was used in this paper. While Niblack is one of the more widely cited local adaptive binarization algorithms, in low contrast regions it is prone to producing ghosting speckle. Variations on it are therefore often implemented, or a post-processing filter is applied. The implementation in the Gamera toolkit looks at the absolute intensity of the pixel in question and has an upper and lower bound beyond which the adaptive evaluation is not used. This prevents much of the ghosting speckle. That variation was implemented for these tests with bounds at 20 and 150.

Sauvola is a similar local adaptive thresholding algorithm. It has fewer of the side effects associated with Niblack, and thus was also considered in this study. The threshold for Sauvola is also determined by combining the local average and standard deviation of the pixels,

$$T(x,y) = m(x,y) + \left\{ 1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right\}. \tag{9}$$

Values of $k = 0.5$ and $R = 128$ are used in this study.

Gatos et al.[3] developed a binarization algorithm that is particularly designed to work on documents with uneven backgrounds resulting from bleed through and stains. The algorithm has four main parts. The first is application of a Wiener filter. This was described in Section 2. The second step is to apply the Sauvola threshold to get a rough estimate of foreground and background pixels. Next an estimate of the background is made for the pixels determined in step 2 to belong to the foreground $S(x,y)$ by

$$B(x,y) = \frac{\sum_{i_x=x-dx}^{x+dx} \sum_{i_y=y-dy}^{y+dy} (I(i_x,i_y)(1 - S(i_x,i_y)))}{\sum_{i_x=x-dx}^{x+dx} \sum_{i_y=y-dy}^{y+dy} (1 - S(i_x,i_y))}. \tag{10}$$

The final thresholding is accomplished by comparing the difference from the background image, $B(x,y)$, and the preprocessed gray level image, $I_{filt}(x,y)$, through

$$B(x,y) - I_{filt}(x,y) > d(B(x,y)). \tag{11}$$

The threshold d is a function of the image background and takes on variable values that are smaller in darker regions. This is achieved through the function

$$d(B(x,y)) = q\delta\left(\frac{(1-p_2)}{1 + exp(\frac{-4B(x,y)}{b(1-p_1)} + \frac{2(1+p_1)}{(1-p_1)})} + p_2\right). \tag{12}$$

The parameter b is the average background surface value $B(x,y)$ over the text areas. Parameter of $q = 0.6$, $p_1 = 0.5$, and $p_2 = 0.8$ are suggested. In this paper the preprocessing is not restricted to use of the Wiener filter, but is applied to all the filter choices discussed in Section 2.

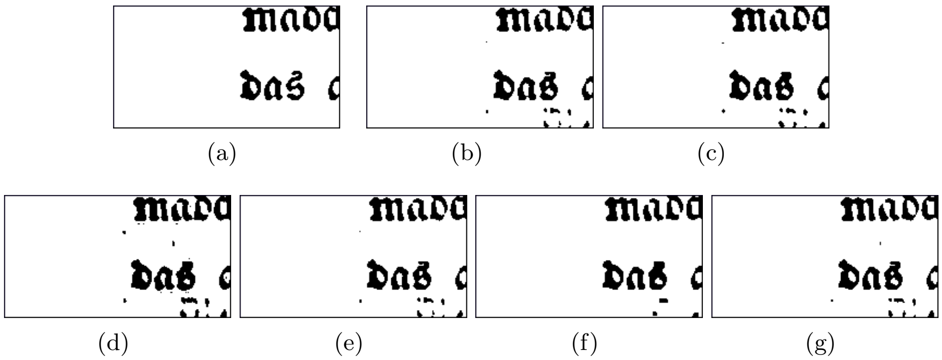


Figure 2. Samples of the effect of the pre-processing algorithms on Otsu binarization. (a) The ground truth image, (b) 3x3 mean filter, (c) 3x3 median filter, (d) Wiener filter with Gatos implementation, (e) Wiener filter with Gamera implementation, (f) Total variation filter with $\beta = 10$, (g) Non-Local Means filter.

Background Estimation and Subtraction: A simpler variation on Gatos' algorithm was also implemented. The background of the filtered image is calculated through a multiple scale based estimator.[11] A pair of envelopes for the background are estimated over larger and larger windows by taking the maximum, or the minimum, of the center pixel and the average value of pairs of pixels on opposite vertical, horizontal and diagonal edges of a square of sides $2 \cdot d_i + 1$ centered around the target pixel

$$S_{max}(x,y;i) = max\left\{\frac{I(x-d_i,y-d_i)+I(x+d_i,y+d_i)}{2}, \frac{I(x+d_i,y-d_i)+I(x-d_i,y+d_i)}{2},\right.$$
$$\left.\frac{I(x-d_i,y)+I(x+d_i,y)}{2}, \frac{I(x,y-d_i)+I(x,y+d_i)}{2}, I(x,y)\right\}, \tag{13}$$

and similarly for $S_{min}(x,y;i)$. This is repeated until the smaller sum of the magnitude of the gradient across the image from one iteration to the next changes by less than 1%. The choice between S_{min} and S_{max} is determined by which has the smallest magnitude gradient. The background is then subtracted from the image and a global threshold determined by Otsu is applied. This algorithm with the Total Variation pre-processing was submitted to the DIBCO 2009[5] binarization contest and performed in the upper quartile of entries.

4. DATA

Any evaluation of a binarization method needs a method to determine the quality of the results. A quantitative method is considered all the better, but is often hard to achieve. Quantitative methods have been achieved by looking at the OCR performance that results.[1] Another option is to measure the resulting image against a ground truth. This has the added challenge of getting a solid ground truth image. The ground truth issue is often achieved by synthetically creating the images. This does provide a good quantity of data with minimum effort, but it must be assured that the noise that is introduced is realistic. This is not always achieved. Sezgin[2] degraded the images and then added noise to them. The noise was spatially uniform in structure. The text was also blurred using the Baird degradation model,[12] but the method of getting the ground truth separating the effects of blurring and additive noise not described. In Stathis[4] a real degraded document background was combined with generated digitized text. The DIBCO 2009[5] contest provided a small data set of real images with a semi-automatically generated ground truth. Thumbnails of these are shown in Figure 3. The images were color images converted to gray level for this study. They ranged in size from 1605x525 pixels to 824x201 pixels. The data set required significant effort to create, and has relatively good correspondence to subjective choices for ground truth.

The images from the DIBCO 2009 training and testing data sets were used for these tests. There were 7 images of machine print and 7 images of handwriting in this data set. Not all images that researchers and practitioners wish to binarize will have bleed through or page border noise. In addition, two synthetic images

(a)

Figure 3. Samples of the DIBCO 2009 images.

generated with two different levels of uniform additive noise (50 & 100) were also used. Since the additive noise is commonly used in the literature, this provides a data point of comparison with those studies.

5. TESTS AND RESULTS

The 16 images were each pre-filtered by all the filtering algorithms described in Section 2. Each filter output plus the unfiltered original was then binarized by each of the binarization algorithms described in Section 3. The binarized image, I_{bin}, was then compared to the ground truth image, I_{GT}. Two metrics were used to do this comparison.

The first looks at the problem as an information retrieval problem. The true positives are those pixels that were black in the ground truth image and are still black in the binarized image. The false positives were white in the ground truth image and black in the binarized image. The false negatives are black in the ground truth image, but white in the binarized image. From these counts the statistics of

$$Recall = \frac{TruePositive}{FalseNegatives + TruePositives} \tag{14}$$

and

$$Precision = \frac{TruePositive}{FalsePositives + TruePositives} \tag{15}$$

are calculated. These are combined into a single F-measure through the geometric mean of the precision and recall

$$F - measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}. \tag{16}$$

The second metric measures cross correlation.

$$\rho = \frac{\sum_i \sum_j (I_{GT}(i,j) - \overline{I_{GT}})(I_{out}(i,j) - \overline{I_{out}})}{\sqrt{\left(\sum_i \sum_j (I_{GT}(i,j) - \overline{I_{GT}})^2\right)\left(\sum_i \sum_j (I_{out}(i,j) - \overline{I_{out}})^2\right)}}. \tag{17}$$

These scores were calculated for every image/filter/binarization combination. The average score by binarization algorithm are shown in Tables 1 and 2. When there were multiple filtering options, only the best performing option was chosen for display in the table. Images for Otsu are shown in Figure 2. Otsu shows the filter effects more clearly as it operates directly on the filtered image.

In most cases the best pre-processing filter was the Total Variation filter with $\beta = 10$. The Gamera implementation of the Wiener filter also performed well. For the mean, median and Gatos implementation of the Wiener filter, the smaller (3x3) filter size was better than the larger (5x5) filter size. In these three cases, the 5x5 filters actually resulted in worse performance than not doing any pre-processing. The Gamera definition of the Wiener filter performed better than the Gatos definition of the Wiener filter as it produces crisper text edges. For the Gamera definition of the Wiener filter, the 5x5 outperformed the 3x3.

The largest variation in performance can be attributed to the different image quality within the data set. The variation due to binarization algorithm was half that. The pre-processing method contributed less, but still a non-negligible amount. The envelope background estimation and subtraction and Otsu binarization methods were most affected by the pre-processing filter choice, followed by Gatos, Niblack and Sauvola. The same conclusions can be drawn from both the F-measure as well as the cross correlation metrics.

Table 1. Performance measured by F-Measure of binarization algorithms under a variety of image pre-processing methods.

Binarization algorithm	no pre-processing	mean 3	median 3	Wiener Gatos 3	Wiener Gamera 5	TV $\beta = 10$	NLMeans
Otsu	0.781	0.798	0.812	0.795	0.815	0.814	0.785
Niblack	0.819	0.828	0.840	0.826	0.840	0.852	0.827
Sauvola	0.821	0.826	0.841	0.825	0.841	0.849	0.828
Gatos	0.873	0.880	0.891	0.878	0.893	0.897	0.881
Background-Est	0.796	0.818	0.825	0.808	0.863	0.861	0.834

Table 2. Performance measured by normalized cross correlation of binarization algorithms under a variety of image pre-processing methods.

Binarization algorithm	no pre-processing	mean 3	median 3	Wiener Gatos 3	Wiener Gamera 5	TV $\beta = 10$	NLMeans
Otsu	0.789	0.806	0.819	0.803	0.822	0.821	0.793
Niblack	0.812	0.821	0.833	0.820	0.832	0.845	0.820
Sauvola	0.817	0.823	0.836	0.821	0.837	0.845	0.824
Gatos	0.867	0.875	0.885	0.872	0.887	0.891	0.875
Background-Est	0.785	0.802	0.811	0.792	0.852	0.851	0.825

6. CONCLUSION

No one binarization algorithm is uniformly best over all possible images, and neither is one pre-processing algorithm. The Total Variation pre-processing improved the binarization more than the other pre-processing algorithms that were tested, with the computationally simpler Gamera interpretation of the Wiener filter close behind. The binarization algorithms that were more complex were less sensitive to the choice of pre-processing algorithms, as were the images where foreground and background were better separated.

The effect that these differences in binarization performance would have on a follow-on algorithm has not been evaluated in this study. Also the type of binarization errors were not evaluated. It would be good in follow-on work to separate the analysis between background pixels, edge pixels that bridge between characters or strokes and other edge pixels, as errors in these places would have different types and significance of effects on follow-on processes.

REFERENCES

[1] Trier, O. D. and Jain, A. K., "Goal-directed evaluation of binarization methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(12), 1191–1201 (1995).

[2] Sezgin, M. and Sankur, B., "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging* **13**(1), 146–168 (2004).

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

[4] Stathis, P., Kavallieratou, E., and Papamarkos, N., "An evaluation technique for binarization algorithms," *Journal of Universal Computer Science* **14**(18), 3011–3030 (2008).

[5] Gatos, B., Ntirogiannis, K., and Pratikakis, I., "ICDAR 2009 document image binarization contest (DIBCO 2009)," in [*Proc. of ICDAR'09*], 1375–1382 (2009).

[6] Droettboom, M. Gamera Toolkit.

[7] Likforman-Sulem, L., Darbon, J., and Barney Smith, E. H., "Pre-processing of degraded printed documents by non-local means and total variation," in [*Proc. of ICDAR'09*], 758–762 (2009).

[8] Darbon, J. and Sigelle, M., "Image restoration with discrete constrained total variation, Part I: fast and exact optimization," *JMIV* **26**, 261–276 (2006).

[9] Buades, A., Coll, B., and Morel, J., "A review of image denoising algorithms, with a new one," *SIAM-Multiscale Modeling and Simulation* **4**, 490–530 (2005).

[10] Otsu, N., "A threshold selection method from gray level histograms," *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979).

[11] Reyes-Aldasoro, C. C., "A retrospective shading correction algorithm based on signal envelope estimation," *Electronics Letters* **45**(9), 454–456 (2009).

[12] Baird, H. S., "Document image defect models and their uses," in [*Proceedings of the Second International Conference on Document Analysis and Recognition, 1993.*], (1993).