Pattern Recognition Letters 34 (2013) 1299–1306

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

An optimization for binarization methods by removing binary artifacts

Marte A. Ramírez-Ortegón [a,c,*], Volker Märgner [a], Erik Cuevas [c], Raúl Rojas [b,1]

[a] Institut für Nachrichtentechnik, Technische Universität Braunschweig, Schleinitzstrae 22, 38106 Braunschweig, Germany
[b] Institut für Informatik, Freie Universität Berlin, Takustr. 7, 14195 Berlin, Germany
[c] Departamento de Ciencias Computacionales, Universidad de Guadalajara, Av. Revolución 1500, Guadalajara, Jalisco, Mexico

ARTICLE INFO

Article history:
Received 1 October 2012
Available online 23 April 2013

Communicated by M. Couprie

Keywords:
Historical documents
Threshold
Denoising
Binarization
Minimum error rate
Bayes theory

ABSTRACT

In this article, we introduce a novel technique to remove binary artifacts. Given a gray-intensity image and its corresponding binary image, our method detects and remove connected components that are more likely to be background pixels. With this aim, our method constructs an auxiliary image by the minimum-error-rate threshold and, then, computes the ratio of intersection between the connected components of the original binary image and the connected components of the auxiliary image. Connected components with high ratio are considered true connected components while the rest are removed from the output. We tested our method in binarization methods for historical documents (handwritten and printed). Our results are favorable and indicate that our method can improve the outputs from diverse binarization methods. In particular, a high improvement was observed for printed documents. Our method is easy to implement, has a moderate computational cost, and has two parameters whose model interpretation allows an easy empirical selection.

1. Introduction

In the context of document analysis, the process of distinguishing pixels that constitute ink strokes (foreground pixels) from the rest (background pixels) is known as *binarization*.

Binarization is a crucial task for document analysis methods since such methods rely on features extracted from foreground pixels. Hence, an inaccurate binarization tends to systematically propagate noise through the whole system. Some methods where binarization is crucial are methods for line detection (Louloudis et al., 2008), optical character recognition (Lázaro et al., 2010), text segmentation (Nikolaou et al., 2010), thinning (Bag and Harit, 2011), type of text classification (Peng et al., 2012), and writer identification (Brink et al., 2012).

During the past three decades, binarization methods for documents have been intensively researched; see surveys of binarization methods in Sahoo et al. (1988), Trier and Jain (1995), Sezgin and Sankur (2004) and Stathis et al. (2008). This is motivated in part because digital documents make easier the accessing and searching of document contents; and in part because the definition of binarization depends on the specific application. Actually, a great deal of research groups continue developing binarization methods specialized in historical documents (Gatos et al., 2011; Pratikakis et al., 2010, 2011) due to the importance of preserving cultural heritage and the complexity of historical documents.

For historical documents, binarization is a challenging task because such documents frequently have non-standard printing styles, like diverse fonts, ornamental strokes, background printing patterns, and irregular stroke widths. In addition, historical documents may have several types and degrees of degradation due to aging and mistreat, such as bleed-through, ink stains, smudged characters, and outlines of paper folds.

Because of the complexity of historical documents, some binarization methods specialized on historical documents like in Moghaddam and Cheriet (2012), Ben Messaoud et al. (2011), Lu et al. (2010), Ntirogiannis et al. (2009), Gupta et al. (2007) and Gatos et al., 2006 compute a preliminary binary image and, subsequently, the pixels are toggled from one class to the other in order to minimize the misclassification rate. For the purposes of this article, we refer as *binarization core* to the initial process of classifying the pixels, and as *binary restoration* to the process of toggling pixels.

Unlike the methods of binarization core that transform images from gray/color intensities to binary values, methods of binary restoration have a binary image as an input and as an output. Hence, any binarization method can be followed by any method of binary restoration. However, the election of a suitable binary restoration mainly depends on three factors: the type of noise in which the binarization core fails, the expected noise generated by the improper parameter selection for the binarization core, and the a priori

* Corresponding author at: Institut für Nachrichtentechnik, Technische Universität Braunschweig, Schleinitzstrae 22, 38106 Braunschweig, Germany. Tel.: +49 (0) 531 3912483; fax: +49 (0) 531 3918218.

E-mail addresses: mars.sasha@gmail.com (M.A. Ramírez-Ortegón), maergner@ifn.ing.tu-bs.de (V. Märgner), erik.cuevas@cucei.udg.mx (E. Cuevas), rojas@inf.fu-berlin.de (R. Rojas).

[1] Tel.: +49 (0) 30 83875130.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.04.007

knowledge of the objects of interest (ink strokes). While the former two factors are related with false binary strokes (binary artifacts) and diminished binary strokes, the latter factor is related to the data adequacy for further process.

We define a *binary artifact* as a connected component such that all of its pixels have been wrongly classified. Under this definition, we have two types of binary artifacts: foreground binary artifacts that are constituted by background pixels, and background binary artifacts that are constituted by foreground pixels.

Removing either fore- or background binary artifacts are dual problems. So, we detail our method only for foreground binary artifacts and, from now on, we refer to foreground binary artifacts as binary artifacts.

Methods to remove binary artifacts are important for binarization methods. In particular, all the binarization methods in Ramírez-Ortegón et al. (2010a,b), Ramírez-Ortegón and Rojas (2010), Moghaddam and Cheriet (2012), Ben Messaoud et al. (2011), Lu et al. (2010), Ntirogiannis et al. (2009), Gupta et al. (2007) and Gatos et al., 2006 compute one or more techniques to remove binary artifacts caused by splotches, paper cracks and folds, specks, bleed-through, and background printing patterns that frequently appear in historical documents.

Small binary artifacts tend to be caused by splotches and specks. To deal with this kind of noise, authors like Lu et al. (2010) and Gupta et al. (2007) remove connected components with n or less pixels. Lu et al. assume $n = 3$, but Gupta et al. compute n from an equation that involves the space (in pixels) between lines. Other approaches to eliminate small binary artifacts are binary templates (Ramírez-Ortegón et al., 2010a,b) and shrink filters (Gatos et al., 2006, 2008).

Counting only the number of pixels is not enough to distinguish between strokes and binary artifacts when the binary artifacts are large. Unfortunately, large binary artifacts are common in historical documents due to bleed-through images, paper cracks, outline folds, and bi-level background. To overcome these problems, Lu's method calculates the average of certain feature for each connected component (Lu et al., 2010), if such an average is higher than a threshold, then the evaluated connected component is considered as binary artifacts.

Instead computing only the average of some feature as in Lu's method, Moghaddam's method compute a vector of features for each connected component of both fore- and background (Moghaddam and Cheriet, 2012), then the vectors are clustered and classified to determine which connected components are binary artifacts.

Although both Lu's and Moghaddam's restoration methods have been applied with positive results. Both methods have the disadvantage of being complex in their implementation; both methods estimate a background surface that considerably increases the computational load and increase the number of parameters to be set. Furthermore, their parameter selection is not simple and, as a consequence, the incorporation of such methods in binarization techniques is difficult.

We propose a method to remove binary artifacts (Section 2) whose implementation is simple, whose computational cost is linear to the number of pixels, and whose number of parameters is two. Moreover, both parameters have model interpretation so that they can be empirically adapted for diverse applications.

Our method computes an auxiliary image (Section 2.2) from the initial binary input and, subsequently, it determines whether or not a connected component should be removed based on the intersection between the input binary image and the auxiliary image.

We evaluated the performance of our method based on DIBCO 2011 benchmark in order to standardize our evaluation.

2. Our method

Our method is a technique that strongly depends of the accuracy of binary input. Then, it may introduce more noise if the binary input is considerably inaccurate. However, we will show that our method has satisfactory results in general if the initial binarization is good.

Strictly speaking, our method has a single parameter α, but it calculates an auxiliary binary image. How to compute the auxiliary image is a parameter in itself and it may involves more parameters. To compute the auxiliary image, we suggest the minimum-error-rate threshold which only has a parameter r which controls the radius of the pixel neighborhood. In this manner, our implementation has a total of two parameters.

2.1. Notation

For the purpose of this paper, pixels are denoted in bold, and the gray intensity of a pixel $\boldsymbol{p}$ is denoted as $I(\boldsymbol{p}) \in \mathbb{N}$, where black is set to zero, and the color white is set to g. The image of gray intensities is denoted by I.

We denote the fore- and background sets by $\mathcal{F}$ and $\mathcal{B}$, respectively, such that $\mathcal{P} = \mathcal{F} \cup \mathcal{B}$. The neighborhood of a pixel $\boldsymbol{p}$, denoted by $\mathcal{P}_r(\boldsymbol{p})$, are those pixels within a square centered at the pixel $\boldsymbol{p}$ of sides with length $2r + 1$. Moreover, given a set of pixels $\mathcal{A}$, we will write $\mathcal{A}_r(\boldsymbol{p})$ as shorthand for $\mathcal{A} \cap \mathcal{P}_r(\boldsymbol{p})$. For instance, $\mathcal{F}_r(\boldsymbol{p}) = \mathcal{F} \cap \mathcal{P}_r(\boldsymbol{p})$ and $\mathcal{B}_r(\boldsymbol{p}) = \mathcal{B} \cap \mathcal{P}_r(\boldsymbol{p})$. In addition, the cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

In the following sections we adopt the thresholding approach as:

$$B(p) = \begin{cases} 1 \text{ (foreground)} & \text{if } I(\boldsymbol{p}) \leqslant T(\boldsymbol{p}), \\ 0 \text{ (background)} & \text{otherwise}, \end{cases} \tag{1}$$

where B denotes the binary image of I, and $T(\boldsymbol{p})$ is the threshold calculated for $\boldsymbol{p}$.

2.2. Auxiliary image

We compute an auxiliary image based on Bayes theory: the minimum-error-rate threshold. We elected this threshold because it is more robust to gray-intensity outliers than thresholds based on mean and variances of gray intensities.

2.2.1. Minimum-error-rate threshold

In the context of binarization for historical documents, we assume that ink strokes are darker than the background. This empirical idea is then exploited by assuming that the gray-intensity distribution of foreground pixels is left shifted from the gray-intensity distribution of background pixels and that the overlapping between these two distributions is small.

According to Bayesian decision theory, the probability of a pixel misclassification is minimized by Bayes decision rule: Classify $\boldsymbol{p}$ as foreground if

$$\Pr(\boldsymbol{p} \in \mathcal{F}_r(\boldsymbol{p}) | I(\boldsymbol{p}) = i) \geqslant \Pr(\boldsymbol{p} \in \mathcal{B}_r(\boldsymbol{p}) | I(\boldsymbol{p}) = i). \tag{2}$$

Otherwise, classify $\boldsymbol{p}$ as background.

Following Bayes criteria, let $\hat{\mathcal{F}}$ and $\hat{\mathcal{B}}$ be the estimated fore- and background, respectively, of our input binary image. Then, the minimum-error-rate threshold for a pixel $\boldsymbol{p}$ within the neighborhood of radius r is defined by

$$T(\boldsymbol{p}) = \underset{t=0,1,\ldots,g}{\arg\min} \left\{ \sum_{i=0}^{t} h_b(i) + \sum_{i=t+1}^{g} h_f(i) \right\}, s \tag{3}$$

where $h_f(i) = |\{\boldsymbol{p} \in \hat{\mathcal{F}}_r(\boldsymbol{p}) \,|\, I(\boldsymbol{p}) = i\}|$ and $h_b(i) = |\{\boldsymbol{p} \in \hat{\mathcal{B}}_r(\boldsymbol{p}) \,|\, I(\boldsymbol{p}) = i\}|$. Note that no any particular distribution is assumed in this equation.

The value r should be large enough to contain the connected components (or a large proportion) and, at the same time, it should be small enough to preserve the gray-intensity distribution of the pixel neighborhood. In particular, small values in r may lead to a great deal of false positives in the auxiliary image so that no binary artifact may be removed. On the other hand, large values of r may lead to a great deal of both false positives and false negatives in the auxiliary image so that true strokes may be removed while binary artifacts may not be removed. For our experiments, we found $r = 60$ a suitable value.

2.2.2. Computational cost of the minimum-error-rate threshold

The computational cost of $T(\boldsymbol{p})$ in (3) in terms of Big-O is $O(g+1) = O(g)$, where $g+1$ is the number of gray intensities. Therefore, computing $T(\boldsymbol{p})$ for all pixels has a complexity $O(g \cdot n)$, where n is the number of pixels in the image.

In addition to $T(\boldsymbol{p})$, our method computes the gray-intensity histogram for each pixel, which has a complexity $O(r)$; see an explicit implementation in Ramírez-Ortegón et al. (2010a). Therefore, computing the histogram for all pixels has a complexity of $O(r \cdot n)$. Then, the computational cost to compute the threshold for all pixels is $O(g \cdot n + r \cdot n) = O(n(g+r))$, which is linear to the number of pixels assuming that g and r are constants.

Notice that the gray intensities are commonly set to 8-bits precision ($g = 255$), and that r could be upper bounded by a constant since large r's are unsuitable to capture the local gray-intensity features of the pixel of interest. Therefore, our method has a linear complexity to the number of pixels but it has a large constant factor.

2.3. Removing binary artifacts

Our method is based on set intersection (voting). A flow diagram of our method is shown in Fig. 1. The detailed steps are as follows:

1. Denote the foreground of the input binary image by $\hat{\mathcal{F}}$.
2. Compute an auxiliary image by the minimum-error-rate threshold as in Section 2.2, and denote the foreground of the auxiliary image by $\hat{\mathcal{F}}'$.
3. Compute all the connected components of $\hat{\mathcal{F}}$.
4. For each connected component $\mathcal{C} \subset \hat{\mathcal{F}}$, compute the ratio
$$IR(\mathcal{C}) = \frac{|\hat{\mathcal{F}}' \cap \mathcal{C}|}{|\mathcal{C}|}. \tag{4}$$
5. The foreground of the output is
$$\hat{\mathcal{F}}^* = \hat{\mathcal{F}} \setminus \{\mathcal{C} \subset \hat{\mathcal{F}} \,|\, IR(\mathcal{C}) < \alpha\}. \tag{5}$$

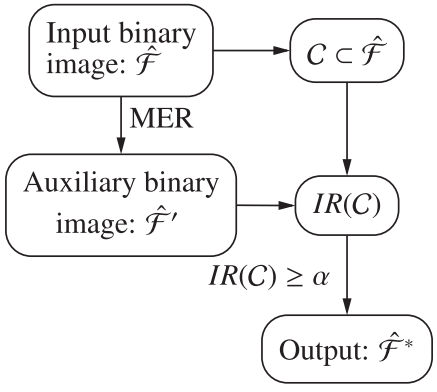

Fig. 1. Flow diagram of our method.

An example of our method is shown in Fig. 2 where several binary artifacts have been removed without removing true strokes.

Observe that $IR(\mathcal{C})$ is the percent of pixels of $\mathcal{C}$ that appear in both $\hat{\mathcal{F}}$ and $\hat{\mathcal{F}}'$. Then, we expect that IR's values of true strokes are large while IR's values of binary artifacts are small. In the following section, we discuss how α affects the performance of our method.

2.4. Breakdown interval

In this section, we first study the relation between α and the performance of our method when the input binary image matches with the groundtruth ($\hat{\mathcal{F}} = \mathcal{F}$ and $\hat{\mathcal{B}} = \mathcal{B}$) and, subsequently, we study the case when the binary input is different to the groundtruth.

2.4.1. Case: $\hat{\mathcal{F}} = \mathcal{F}$

Fig. 3(a) shows the graph's performance of our method by varying α. These graphs plot the recall measure that is defined as the number of foreground pixels that remain in the restored image:
$$recall(\hat{\mathcal{F}}^*, \mathcal{F}) = \frac{|\hat{\mathcal{F}}^* \cap \mathcal{F}|}{|\mathcal{F}|}. \tag{6}$$

As we expected, small α's values holds all or nearly all foreground pixels, and large α's values discard all foreground pixels. Note, however, that there is a breakdown interval where the transition from high performance (one) to low performance (zero) occurs rapidly. We prove in the following paragraphs that this breakdown is determined by the probability of classifying foreground pixels correctly.

To simplify our analysis, let us assume that $u = \Pr(\boldsymbol{p} \in \mathcal{F}')$ for all foreground pixels in the image. Hence, we model the number of pixels of a connected component $\mathcal{C}$ that remains in $\mathcal{F}'$ as a random variable that obeys a binomial distribution with parameters $n = |\mathcal{C}|$ and u. Then, our method keeps $\mathcal{C}$ with a probability given by:
$$\Pr(IR(\mathcal{C}) \geqslant \alpha) = 1 - \Psi(\lceil n \cdot \alpha \rceil; u, n) = \Psi^c(\lceil n \cdot \alpha \rceil; u, n), \tag{7}$$
where $\lceil n \cdot \alpha \rceil$ is the smallest integer greater than or equal to $n \cdot \alpha$, and $\Psi(x; u, n)$ denotes the cumulative density distribution of a binomial distribution with parameters u and n given by
$$\Psi(x; u, n) = \frac{n!}{x!(n-x)!} u^x (1-u)^{n-x}, \tag{8}$$
such that $\cdot!$ denotes the factorial function.

For simplicity, let us assume that $n \geqslant 100$ in the average and that
$$u \approx recall(\hat{\mathcal{F}}', \mathcal{F}) = \frac{|\hat{\mathcal{F}}' \cap \mathcal{F}|}{|\mathcal{F}|}. \tag{9}$$
Then, the recall graph is fitted by (7); see Fig. 3(b).

Based on our model, the rapidly transition from good to poor performance in the recall graphs is explained by the shape of the probability density function of the binomial distribution, which is a unimodal graph that clusters around its mean; see Fig. 3.

2.4.2. Case: $\hat{\mathcal{F}} \neq \mathcal{F}$

When $\hat{\mathcal{F}} \neq \mathcal{F}$, the recall graph can be estimated in similar manner as in Section 2.4.1. However, we have another assumption:
$$u' = \Pr(\boldsymbol{p} \in \hat{\mathcal{F}}' \cap \hat{\mathcal{F}} \,|\, \boldsymbol{p} \in \hat{\mathcal{F}}) \approx recall(\hat{\mathcal{F}}', \hat{\mathcal{F}}) = \frac{|\hat{\mathcal{F}}' \cap \hat{\mathcal{F}}|}{|\hat{\mathcal{F}}|}. \tag{10}$$
Then,
$$recall(\hat{\mathcal{F}}^*_\alpha, \mathcal{F}) \approx u \cdot \Psi^c(\lceil n \cdot \alpha \rceil; u', n), \tag{11}$$
where u is computed as in (9), and $\hat{\mathcal{F}}^*_\alpha$ is the estimated foreground of the restored image subjected to α.

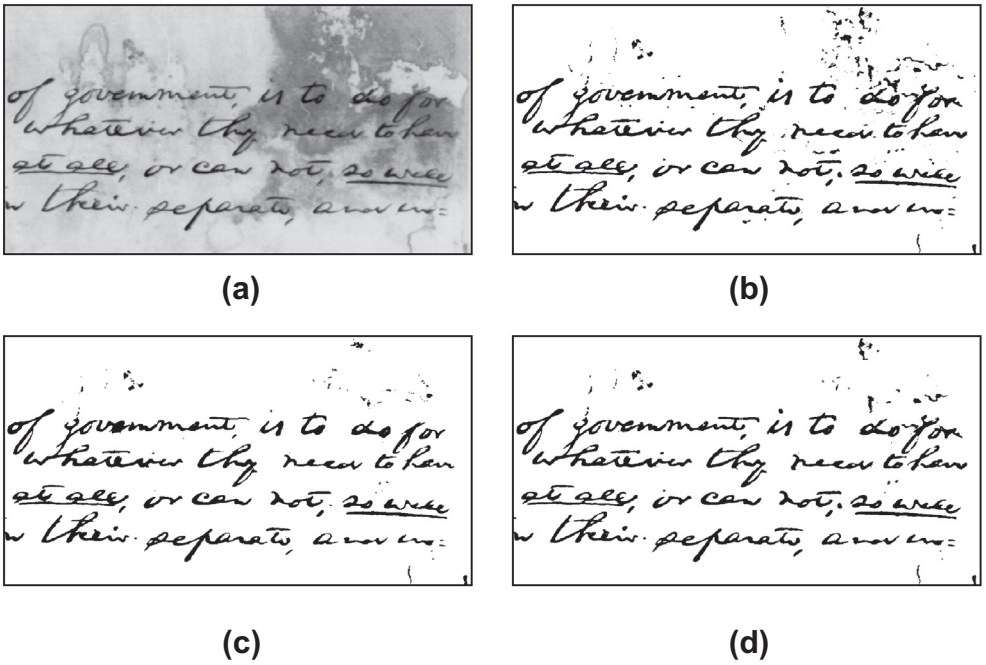

Fig. 2. Example of our method. (a) Original image of gray intensities. (b) Binarization by Sauvola's threshold ($k = 0.2$ and $r = 15$), (c) Binarization by the minimum-error-rate threshold ($r = 60$) assuming (b) as groundtruth. (d) Restored binarization with $\alpha = 0.15$.

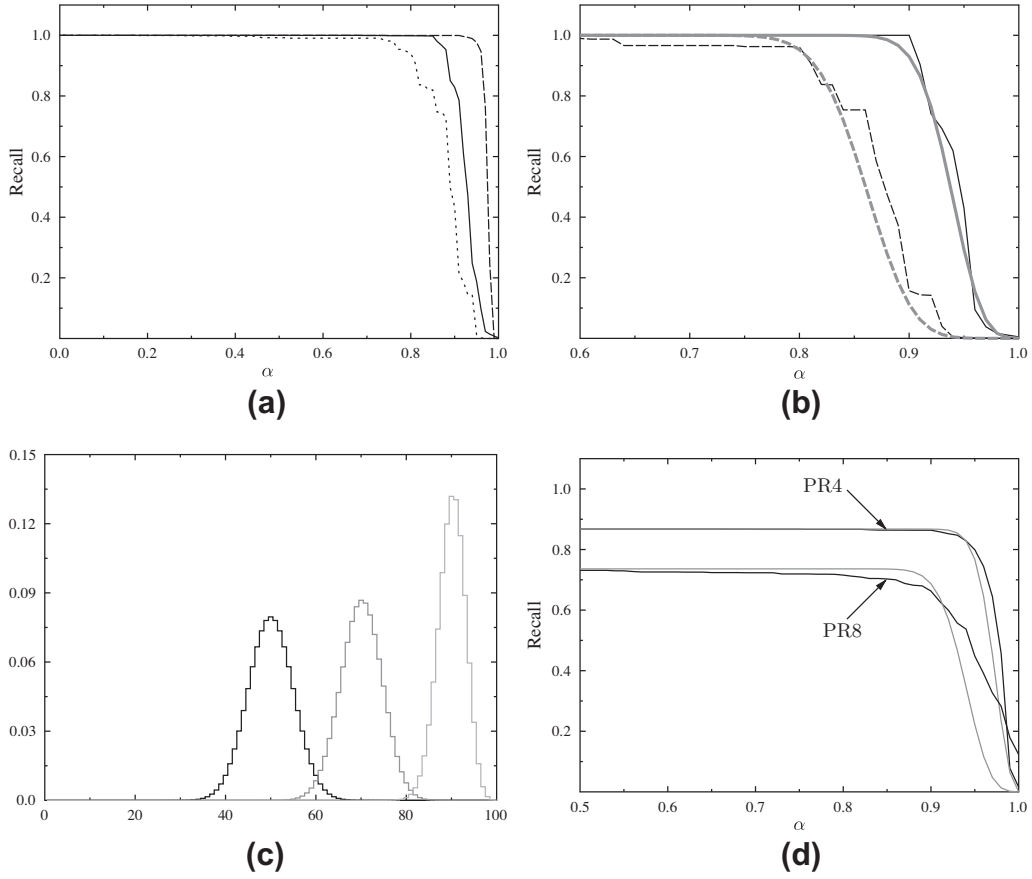

Fig. 3. (a) Recall measurements of our method for three historical documents. (b) Approximation of recall graphs by the complementary cumulative density function of a binomial distribution. (c) Probability density functions of binomial distributions for $n = 100$ and $u = 0.5, 0.7, 0.9$. (d) Approximation of recall graphs when $\hat{\mathcal{F}} \neq \mathcal{F}$. In black, recall graphs; in gray, recall estimation by binomial distribution.

For example, in Fig. 3(d), the recall graphs are fitting by (11). These graphs correspond to the images PR4 and PR8 of DIBCO 2011 with the method 6 (Pratikakis et al., 2011).

The equations in (10) and (11) provide us a clue in which cases our method fails. If a connected component $\mathcal{C}$ is considerably over-extended, like in Fig. 4 is shown, then the value u' for $\mathcal{C}$ is likely to

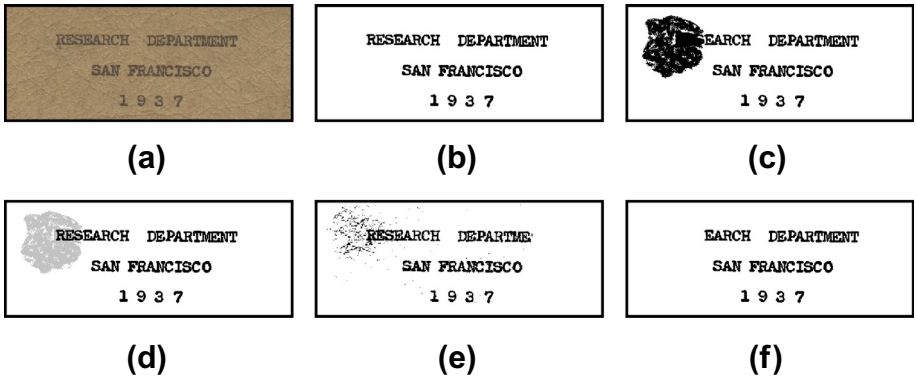

Fig. 4. (a) Original image. (b) Groundtruth. (c) Noisy binary image manually created from the groundtruth. (d) Binary image where the false alarms are displayed in gray. (e) Binarization by the minimum-error-rate threshold assuming (c) as groundtruth. (f) Restored image by our method with $r = 70$ and $\alpha = 0.2$.

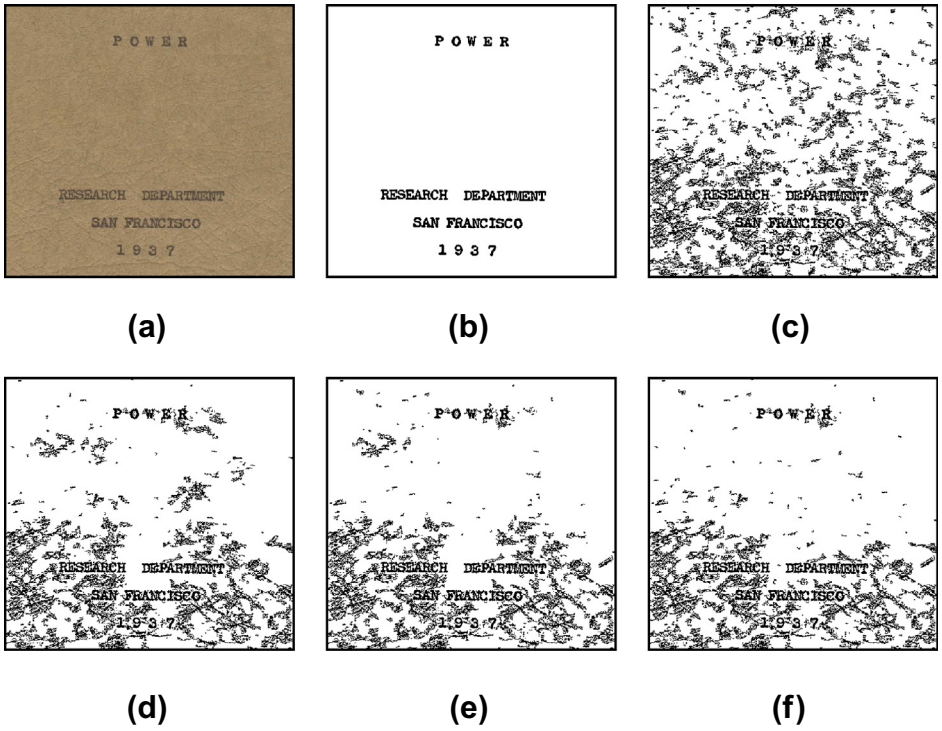

Fig. 5. (a) Original image. (b) Groundtruth. (c) Output of method 8 from DIBCO 2011. (d) The output of our method ($r = 30$ and $\alpha = 0.1$). (e) The output of our method ($r = 60$ and $\alpha = 0.1$). (f) The output of our method ($r = 120$ and $\alpha = 0.1$).

be small because the minimum-error-rate threshold identifies a great deal of the *false alarms* in $\mathcal{C}$; see Fig. 4. Therefore, $IR(\mathcal{C})$ is low and, consequently, $\mathcal{C}$ is removed along the letters which are contained in $\mathcal{C}$; see Fig. 4.

Although our method cannot handle considerably overextended connected components, it is capable of handling heavily noisy images by binary artifacts such as in Fig. 5, where the noise is mainly caused by binary artifacts. In such an example, our method removes a great deal of binary artifacts without losing recall; see Fig. 5(e). This example also shows the effects for small and large values of r; see Fig. 5(d)–(f).

2.5. Performance evaluation

To evaluate the goodness of our method, we compare the trade-off between removing true strokes and removing artifacts. With this aim, we computed our method for the top fourth ranked methods in DIBCO 2011 competition (Pratikakis et al., 2011):

- First place, method 10 (M10) by Lelore and Bouchara (2013).
- Second place, method 8 (M08) by Su, Lu and Tan (2010).
- Third place, method 11 (M11) by Howe (2012).
- Fourth place, method 6 (M06) by Ben Messaoud, Amiri, El Abed Haikal, Märgner (2011).

The original outputs of these methods are available in DIBCO 2011 website.[2]

We computed the F-measure for the binary image before and after our method. The F-measure for the input binary image is defined as

$$FM(\hat{\mathcal{F}}, \mathcal{F}) = \frac{2 \times recall(\hat{\mathcal{F}}, \mathcal{F}) \times precision(\hat{\mathcal{F}}, \mathcal{F})}{recall(\hat{\mathcal{F}}, \mathcal{F}) + precision(\hat{\mathcal{F}}, \mathcal{F})}, \tag{12}$$

[2] <http://utopia.duth.gr/~pratika/DIBCO2011/dibco2011results.htm>.

Table 1
Average of FM's measurements of the four top-ranked methods in DIBCO 2011 before and after our method. Restored images by Gatos's, Lu's, and our binary restoration method are denoted with the suffixes +Gatos, +Lu, and +BR, respectively. The best values are printed in bold.

Method	Handwritten		Printed		Average
	α	FM	α	FM	
M06		83.7490		83.4341	83.5915
M06 + Gatos		83.7490		83.4341	83.5915
M06 + Lu		83.9370		83.4781	83.7076
M06 + BR	0.17	**84.0108**	0.50	**89.5558**	**86.7833**
M08		88.7376		81.6597	85.1987
M08 + Gatos		88.7376		81.6597	85.1987
M08 + Lu		88.7577		81.6712	85.2145
M08 + BR	0.12	**88.8338**	0.60	**87.7214**	**88.2776**
M10		92.3776		69.3519	80.8647
M10 + Gatos		92.3776		69.3519	80.8647
M10 + Lu		92.3873		**77.3264**	84.8569
M10 + BR	0.12	**92.7753**	0.70	77.0520	**84.9137**
M11		89.5853		87.8738	88.7295
M11 + Gatos		89.5853		87.8738	88.7295
M11 + Lu		89.9765		88.1618	89.0691
M11 + BR	0.15	**90.8460**	0.13	**90.5892**	**90.7176**

where

$$precision(\hat{\mathcal{F}},\mathcal{F}) = \frac{|\hat{\mathcal{F}} \cap \mathcal{F}|}{\hat{\mathcal{F}}}. \quad (13)$$

In similar manner, $FM(\hat{\mathcal{F}}^*,\mathcal{F})$ denotes the F-measure of the restored images.

We also compare our method with two binary restoration methods that remove binary artifacts:

(1) Gatos's binary restoration method (Gatos et al., 2006), based on the shrink filter.
(2) Lu's binary restoration method (Lu et al., 2010) based on background surface estimation and connected component analysis.

Both methods were implemented according their original publications.

Table 1 reports the original FM's measurements for all four tested binarization methods and their corresponding values after being processed with Gatos's, Lu's, and our binary restoration method. According this table, our method has the best overall performance. Gatos's binary restoration method is unable to improve the tested binary images because the shrink filter is designed to remove small connected components, like salt and pepper noise, but the tested binary images do not exhibit such a kind of noise. On the other hand, the binary images are improved by Lu's binary restoration method. However, such an improvement is marginal in most of the tested images.

Fig. 6 shows the graphs of the sum of FM's measurements by varying α and by type of image (handwritten and printed); the parameter r was set to 60. Graphs on the left are the total sum of the differences between $FM(\hat{\mathcal{F}}^*,\mathcal{F})$ and $FM(\hat{\mathcal{F}},\mathcal{F})$. Positive values correspond to an overall performance increase of the outputs, while negative values correspond to an overall performance decrease of the outputs. Graphs on the right shows in detail how positive and negative FM's measurements accumulate by varying α.

The graphs in Fig. 6 also show that our method combined is more robust for printed images than for handwritten images. Observe, for instance, that M11 improves for $\alpha \in [0, 0.7]$ but, at the same time, some images have negative *FM*'s values for $\alpha > 0.20$.

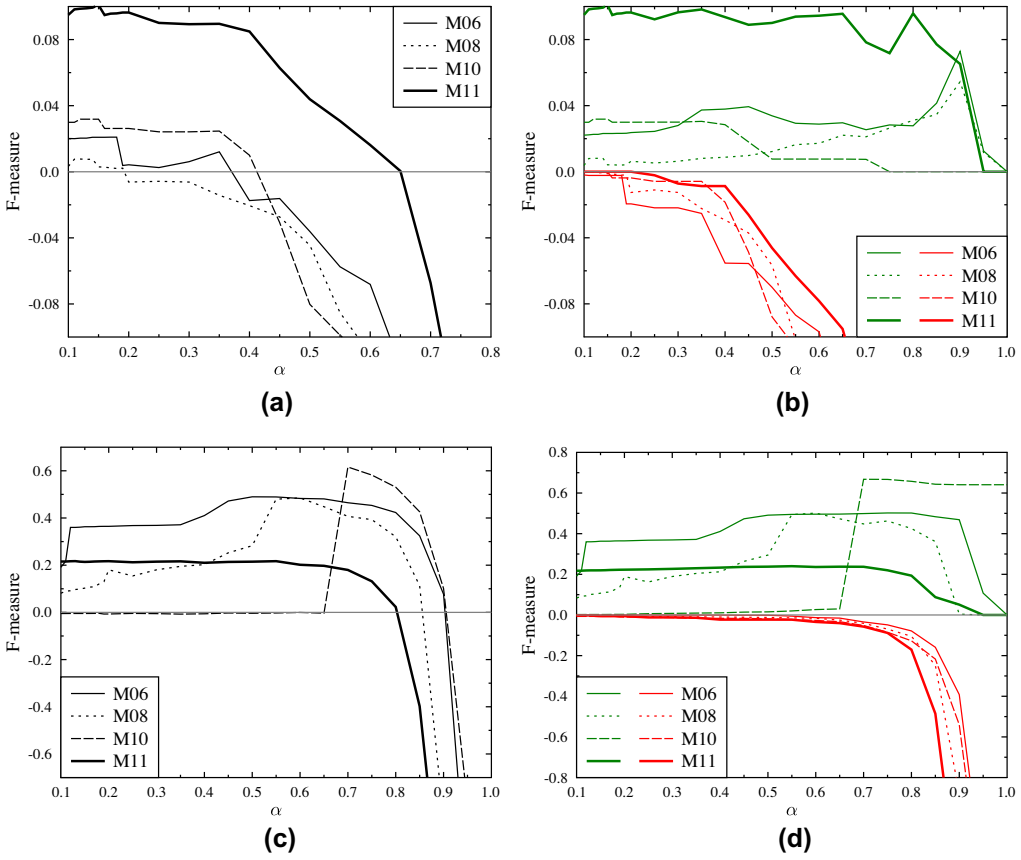

Fig. 6. (a) Sum of F-measures of handwritten images (DIBCO 2011). (b) Sum of positive and negative F-measures of handwritten images (DIBCO 2011). (c) Sum of F-measures of printed images (DIBCO 2011). (d) Sum of positive and negative F-measures of printed images (DIBCO 2011).

Table 2
Average of the F-measure for those images where $FM(\hat{\mathcal{F}},\mathcal{F}) \neq FM(\hat{\mathcal{F}}^*,\mathcal{F})$. HW and PR stand for handwritten and printed. All values are given in percent, and the best values are shown in bold.

α	M06			M08			M10			M11		
	HW	PR	Total	HW	PR	Total	HW	PR	Total	HW	PR	Total
0.10	0.2877	2.3874	1.4075	**0.0466**	1.0153	0.5633	0.5987	−0.0956	0.2515	1.1870	3.0801	2.0704
0.15	**0.2982**	4.5375	2.5592	0.0423	1.2861	0.7057	**0.6363**	−0.0922	0.2720	**1.2607**	3.0555	**2.0982**
0.20	0.0604	4.5576	2.4589	−0.0778	2.2554	1.0888	0.5243	−0.1372	0.1935	1.2046	**3.0997**	2.0890
0.30	0.0873	4.6084	2.4985	−0.0778	2.2554	1.0888	0.4834	−0.0885	0.1714	1.1163	3.0714	2.0287
0.40	−0.2179	5.1286	2.4554	−0.2560	2.5300	1.1370	0.1666	−0.1001	0.0332	1.0612	3.0022	1.9670
0.50	−0.4517	**6.1218**	**2.8351**	−0.5557	3.5319	1.4881	−1.3373	−0.0631	−0.7002	0.5487	3.0696	1.7251
0.60	−0.8518	6.0360	2.5921	−1.3808	**6.0617**	**2.3404**	−1.8877	−0.0045	−0.9461	0.2021	2.5223	1.3622
0.70	−1.9540	5.8077	1.9268	−2.2500	5.0865	1.4182	−3.4645	**8.8002**	**3.1396**	−0.8414	2.2476	0.7031
0.80	−4.1680	5.2897	0.5609	−4.3388	4.0070	−0.1659	−7.1380	7.5801	0.2210	−2.7005	0.2871	−1.2067

The negative values are canceled for some others high positive FM's; see Fig. 6(b).

Table 2 reports the average of *FM*'s measurements for images where $FM(\hat{\mathcal{F}},\mathcal{F}) \neq FM(\hat{\mathcal{F}}^*,\mathcal{F})$. Based on this table, we conclude that cautious values for α are between 0 and 0.2 for handwritten images, and between 0 and 0.4 for printed.

3. Conclusions

We have proposed a novel method to remove binary artifacts based on the minimum-error-rate threshold and the recall measure. Although our method can be applied for both handwritten and printed documents, our experiments suggest that our method performs better for printed document. In fact, we experimentally found that $\alpha = 0.15$ is a good value for handwritten documents while $\alpha = 0.3$ is a good value for printed documents assuming $r = 60$.

We mathematically studied the relation between α with the recall measure. This study describe accurately the effects of α over our method's performance. It also explain why the method's performance quickly decay from good to bad at certain values of α. Based on this study, techniques for an automatic selection of α may be developed.

We tested our method in four independent binarization algorithms and showed that the outputs of such methods are improved after being processed by our method.

Finally, we would like to remark that our method is easy to implement, has a moderate computational cost (linear to the number of pixels but with a large constant factor), has only two parameters r (neighborhood radius) and α. Furthermore, α is empirically easy to adjust and has favorable results for $0.1 \leqslant \alpha \leqslant 0.2$ almost for certain.

Acknowledgments

We thank the anonymous referees for helpful and relevant comments. This research was partially supported by The National Council on Science and Technology (CONACYT) of Mexico (Grant No.: C00/587/11).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.patrec.2013.04.007.

References

Bag, S., Harit, G., 2011. An improved contour-based thinning method for character images. Pattern Recognition Letters 32 (14), 1836–1842.

Ben Messaoud, I., El Abed, H., Amiri, H., Märgner, V., 2011. New method for the selection of binarization parameters based on noise features of historical documents. In: Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data. ACM, New York, NY, USA, pp. 1:1–1:8.

Ben Messaoud, I., Amiri, H., El Abed, H., Märgner, V., 2011. New binarization approach based on text block extraction. In: ICDAR, pp. 1205–1209.

Brink, A., Smit, J., Bulacu, M., Schomaker, L., 2012. Writer identification using directional ink-trace width measurements. Pattern Recognition 45 (1), 162–171.

Gatos, B., Pratikakis, I., Perantonis, S., 2006. Adaptive degraded document image binarization. Pattern Recognition 39 (3), 317–327.

Gatos, B., Pratikakis, I., Perantonis, S.J., 2008. Improved document image binarization by using a combination of multiple binarization techniques and adapted edge information. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4.

Gatos, B., Ntirogiannis, K., Pratikakis, I., 2011. DIBCO 2009: document image binarization contest. International Journal on Document Analysis and Recognition 14, 35–44.

Gupta, M.R., Jacobson, N.P., Garcia, E.K., 2007. OCR binarization and image preprocessing for searching historical documents. Pattern Recognition 40, 389–397.

Howe, N.R., 2012. Document binarization with automatic parameter tuning. International Journal on Document Analysis and Recognition, 1–12.

Lázaro, J., Martín, J.L., Arias, J., Astarloa, A., Cuadrado, C., 2010. Neuro semantic thresholding using OCR software for high precision OCR applications. Image and Vision Computing 28, 571–578.

Lelore, T., Bouchara, F., 2013. Fair: a fast algorithm for document image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99 p. 1, PrePrints.

Louloudis, G.E., Gatos, B.G., Pratikakis, I., Halatsis, C., 2008. Text line detection in handwritten documents. Pattern Recognition 41, 3758–3772.

Lu, S., Su, B., Tan, C.L., 2010. Document image binarization using background estimation and stroke edges. International Journal on Document Analysis and Recognition 13, 303–314.

Moghaddam, R.F., Cheriet, M., 2012. AdOtsu: an adaptive and parameterless generalization of Otsu's method for document image binarization. Pattern Recognition 46 (6), 2419–2431.

Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., Papamarkos, N., 2010. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image and Vision Computing 28, 590–604.

Ntirogiannis, K., Gatos, B.G., Pratikakis, I., 2009. A modified adaptive logical level binarization technique for historical document images. In: 10th International Conference on Document Analysis and Recognition, pp. 1171–1175.

Peng, X., Setlur, S., Govindaraju, V., Ramachandrula, S., 2012. Using a boosted tree classifier for text segmentation in hand-annotated documents. Pattern Recognition Letters 33 (7), 943–950 (Special issue on awards from ICPR 2010).

Pratikakis, I., Gatos, B., Ntirogiannis, K., 2010. H-DIBCO 2010 - handwritten document image binarization competition. In: International Conference on Frontiers in Handwriting Recognition. IEEE Computer Society, Los Alamitos, CA, USA, pp. 727–732.

Pratikakis, I., Gatos, B., Ntirogiannis, K., 2011. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). In: 2011 International Conference on Document Analysis and Recognition. IEEE, pp. 1506–1510.

Ramírez-Ortegón, M.A., Rojas, R., 2010. Transition thresholds for binarization of historical documents. In: 20th International Conference on Pattern Recognition. IEEE Computer Society, pp. 2362–2365.

Ramírez-Ortegón, M.A., Tapia, E., Ramírez-Ramírez, L.L., Rojas, R., Cuevas, E., 2010a. Transition pixel: a concept for binarization based on edge detection and gray-intensity histograms. Pattern Recognition 43, 1233–1243.

Ramírez-Ortegón, M.A., Tapia, E., Rojas, R., Cuevas, E., 2010b. Transition thresholds and transition operators for binarization and edge detection. Pattern Recognition 43 (10), 3243–3254.

Sahoo, P.K., Soltani, S., Wong, A.K., Chen, Y.C., 1988. A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing 41 (2), 233–260.

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

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

Trier, Ø.D., Jain, A.K., 1995. Goal-directed evaluation of binarization methods. Transactions on Pattern Analysis and Machine Intelligence 17 (12), 1191–1201.