Journal of Universal Computer Science, vol. 14, no. 2 (2008), 266-283
submitted: 15/6/06, accepted: 17/11/07, appeared: 28/1/08 © J.UCS

Detailing a Quantitative Method for Assessing Algorithms to Remove Back-to-Front Interference in Documents

Rafael Dueire Lins
(Universidade Federal de Pernambuco, Recife, Brazil
rdl@ufpe.br)

João Marcelo Monte da Silva
(Universidade Federal de Pernambuco, Recife, Brazil
joaommsilva@gmail.com)

Fernando Mário Junqueira Martins
(Universidade do Minho, Braga, Portugal
fmm@di.uminho.pt)

Abstract: Documents written on both sides on translucent paper make visible the ink from one side on the other. This artefact is called “back-to-front interference”, “bleeding” or “show-through”. The direct binarization of documents with such interference yields unreadable documents. The literature presents several algorithms for suitably removing such artefact. This paper presents a quantitative method to assess algorithms to remove back-to-front interference.

Keywords: Document engineering, Back-to-front interference, Show through, Bleeding
Categories: H.3.3

1 Introduction

Whenever a document is typed or written on both sides and the opacity of the paper is such as to allow the back printing to be visualized on the front side, yielding different hues of paper and printing whenever compared to documents written on a single side a sheet of the same paper with the same ink. This phenomenon, first addressed in the literature by [Lins, 95], was called “back-to-front interference”. If the document is scanned either in true-color (Figure 1) or gray-scale (Figure 2) the human eye is able to filter out that sort of noise keeping document readability. The direct binarization of such document overlaps the written or printed part of both sides producing an unreadable document for the human reader and drastically degrading the performance of automatic tools such as OCRs. Thus, it is important to find better segmentation techniques to suitably solve that problem.

Binarized images (black and white images) claim less storage space, allow for faster network transmission, and are suitable to be processed by most commercial OCR tools. Image processing environments (such as Jasc Paint Shop Pro™ [Adobe, 07]) offer a great variety of binarization filters. However, the use of such softwares requires a specialized operator and that is not feasible to handle large quantities of documents. Besides that, the palette reduction algorithms provided by standard commercial tools whenever applied to documents with back-to-front interference yield unreadable images, even for humans.

Figure 1 provides an example of a document with back-to-front interference and Figure 2 is the gray-scale version of the same document. The binarized version of this document generated by the direct application of the binarization algorithm by using Jasc Paint Shop Pro™ version 8 (Palette component: Grey values, Reduction component: nearest color, Palette weight: non-weighted) is completely unreadable, as one may observe in Figure 3.

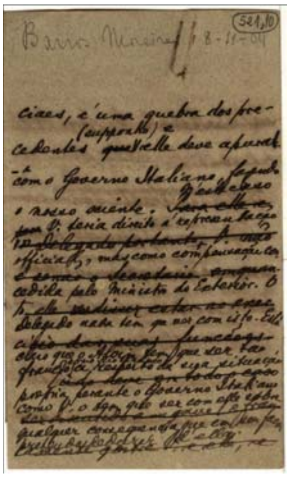


Figure 1: Historical Document with back-to-front interference.

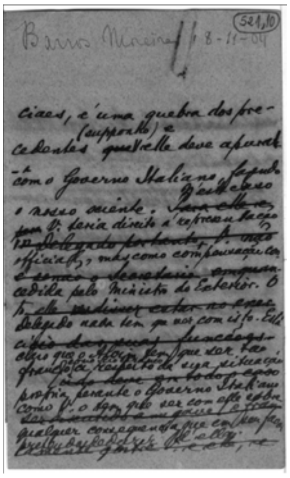


Figure 2: Gray-scale version of Figure 1.

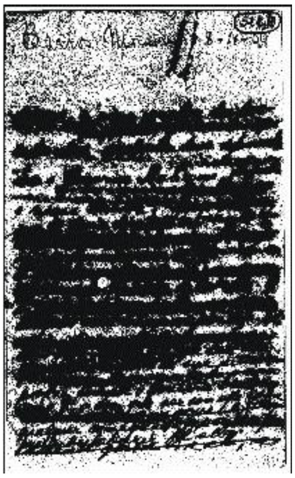


Figure 3: Binarized document of Figure 1.

In a document such as the one presented in Figure 1, one expects to find three color clusters corresponding to the ink in the foreground, the paper background and the trespassed ink (the back-to-front interference). Unfortunately, despite the efforts of several researchers in over a decade of work, no image representation provided such clustering to allow the easy filtering out of the back-to-front interference.

Several papers in the literature addressed the back-to-front interference problem. Some authors use waterflow models [Oha, 05], other researchers have used wavelet filtering [Tan, 02], but the technique of most widespread used is thresholding [Kavallieratou, 05], [Leedham, 02] and [Wong, 01]. The most successful techniques for filtering out back-to-front interference are based on the entropy [Abramson, 63] of the grey-scale document [Mello, 00] and [Mello, 02]. Although recent advances were made in finding efficient algorithms that yield good quality images [Silva, 06], a final solution to the filtering of back-to-front interference is still sought off.

Visual inspection of the filtered images provides a weak quantitative assessment of the performance of the algorithms under comparison. Analyzing the quality of images produced by filtering algorithms is far from being a trivial task. Subjectivity must be avoided by every means. Thus, a quantitative method to measure the quality of algorithms for binarizing documents with back-to-front interference is introduced here. The method presented herein generalizes and provides better comparison grounds than the one presented in [Lins-b, 06], detailing further the results presented in reference [Lins, 07].

This article is organized as follow. Section 1 presents this introduction. Section 2 presents eight threshold-based algorithms used to demonstrate the method described in section 3. The results obtained are discussed and analysed in section 4. Finally, section 5 presents the conclusions of the method introduced herein and draws lines for further work.

2 Threshold Techniques

The survey paper by Sankur and Sezgin [Sankur, 04] presents a comprehensive overview and comparison of thresholding algorithms, clustering them according to their nature. That survey does not address the back-to-front interference, however. From the almost forty algorithms presented six schemes have shown suitable to work in such documents: Pun [Pun, 81], Kapur-Sahoo-Wong [Kapur, 85], Johannsen-Bille [Johannsen, 82], Yen-Chang-Chang [Yen, 95], Wu-Songde-Hanquing [Wu, 98], and Otsu [Otsu, 79]. The first five algorithms are based on the entropy of the image, whereas the last one makes discriminator analysis. Those six algorithms were not designed to filter back-to-front interference. Besides those algorithms, two algorithms based on entropy that were created in the scope of the Nabuco Project [FUNDAJ, 07] to filter that interference are presented: the Mello and Lins's algorithm [Mello, 00][Mello, 02] and the Silva, Lins and Rocha's algorithm [Silva, 06].

The use of gray-scale images with 256 levels as an intermediate step towards image binarization has shown to be a valuable simplification. Thus, the first processing step is generating grey-scale documents from the true-color ones by using the standard equation to calculate the gray level value of the new pixel:

$$grey = 0.299r + 0.587g + 0.114b \tag{1}$$

where *r*, *g*, and *b* are the red, green and blue values of the original pixel.

In general, entropy-based algorithms take the image histogram and normalize each of its entries by the total number of pixels in the image, yielding a distribution of probabilities provided by relative frequencies. Thus,

$$p_i = \frac{n_i}{N},\ 0 \le i \le 255 \tag{2}$$

$$P_t = \sum_{i=0}^{t} p_i \tag{3}$$

where n_i is the number of pixels with grey level *i* (0 to 255), *N* is the total number of pixels in the image, $\{p_0, p_1, \ldots, p_{255}\}$ is the probability distribution of the pixel gray-levels taking into account their relative frequencies, and P_t is the adding of all probabilities up to entry *t*.

All of the algorithms presented here were implemented in standard C using the dev-C++ v4.9.8.0 program.

2.1 Pun's Algorithm

In the algorithm proposed by Pun [Pun, 81], the gray levels are considered like statistically independent 256-symbol source. Pun considers the ratio of the *a posteriori* entropy

$$H'(t) = -P_t \log(P_t) - (1 - P_t)\log(1 - P_t) \tag{4}$$

with the source entropy

$$H(t) = H_b(t) + H_w(t) \tag{5}$$

where H_b and H_w are:

$$H_b(t) = -\sum_{i=0}^{t} p(i)\log(p(i)) \tag{6}$$

$$H_w(t) = -\sum_{i=t+1}^{255} p(i)\log(p(i)) \tag{7}$$

and $p(i) = p_i$ given by equation (2).

Pun shows that

$$\frac{H'(t)}{H} \geq Fe(\alpha) = \alpha \frac{\log P(t)}{\log[\max(p_0, ..., p_t)]} + (1-\alpha)\frac{\log[1 - P(t)]}{\log[\max(p_{t+1}, ..., p_{255})]} \tag{8}$$

where

$$H_b(t) = \alpha H \tag{9}$$

The threshold is obtained by the value t that satisfies the equation (9), where α is the argument that maximizes $Fe(\alpha)$.

2.2 Kapur, Sahoo and Wong

The algorithm by Kapur, Sahoo and Wong [Kapur, 85] considers the foreground and background images as two distinct sources, such that whenever the addition of the two entropies reach a maximum, its argument t reaches the optimal value.

The distribution of the object A and the distribution of the background B are given by:

$$A:\ p(i) = \frac{p_i}{P_t},\quad 0 \leq i \leq t \tag{10}$$

$$B:\ p(i) = \frac{p_i}{1 - P_t},\quad t + 1 \leq i \leq 255 \tag{11}$$

The values of the entropies H_w and H_b are calculated through equations (6) and (7), with $p(i)$ given by equations (10) and (11).

2.3 Johannsen and Bille

The algorithm proposed by Johannsen and Bille [Johannsen, 82] aims at minimizing the function:

$$S(t)=S_b(t)+S_w(t)=\log(P_t)+\frac{1}{P_t}\left[E(p_t)+E(P_{t-1})\right]+ \\ +\log(1-P_{t-1})+\frac{1}{(1-P_{t-1})}\left[E(p_t)+E(1-P_t)\right] \tag{12}$$

where $E(p)$=–p.log(p), and p_i and P_t are provided by equations (2) and (3), respectively.

The value of t that minimizes $S(t)$ is its “optimal” value.

2.4 Yen, Chang and Chang

The algorithm by Yen, Chang and Chang [Yen, 95] follows the same idea as the one by Kapur and his colleagues in respect to the foreground and background distributions. An *entropic correlation* is defined as

$$TC(t)=C_b(t)+C_w(t)=-\log\left\{\sum_{i=0}^{t}\left[\frac{p_i}{P_t}\right]^2\right\}-\log\left\{\sum_{i=t+1}^{255}\left[\frac{p_i}{1-P_t}\right]^2\right\} \tag{13}$$

and the threshold is the argument that maximizes that expression. The functions $C_b(t)$ and $C_w(t)$ are known as Ranyi entropy [MathWorld, 07], with ρ=2.

2.5 Wu, Songde and Hanquing

This algorithm calculates the same entropies evaluated by the Kapur, Sahoo and Wong’s algorithm. But, instead of maximizing the addition of theses, Wu, Songde and Hanquing [Wu, 98] minimize the difference given by:

$$F(t)=\left|H_b(t)-H_w(t)\right|. \tag{14}$$

2.6 Otsu’s Algorithm

The algorithm by Otsu [Otsu, 79] does not belong to the class of algorithms based on entropy. It is included here because it is one of the most often used algorithms in image segmentation. Otsu’s algorithm makes discriminator analysis for defining if a grey level t will be mapped into object or background information. This algorithm works to maximize the between-class variance $\sigma_B^2(t)$ given by:

$$\sigma_B^2(t)=P_t\left(\mu_b(t)-\mu_T\right)^2+\left(1-P_t\right)\left(\mu_w(t)-\mu_T\right)^2 \tag{15}$$

where

$$\mu_b(t)=\sum_{i=0}^{t} i\frac{p_i}{P_t},\ \mu_w(t)=\sum_{i=t+1}^{255} i\frac{p_i}{1-P_t},\ \mu_T=\sum_{i=0}^{255} i.p_i\,. \tag{16}$$

The threshold is the argument t that maximizes the between-class variance $\sigma_B^2(t)$.

2.7 Mello and Lins

The algorithm by Mello and Lins [Mello, 00][Mello, 02] looks for the most frequent gray level of the image and takes it like initial threshold to evaluate the values H_b, H_w and H by equations (6), (7) and (5), respectively, but the entropies must be calculated with the logarithm to the base N. The entropy H determines tow weights m_b and m_w:

- If $H \leq 0.25$, then $m_w = 2$ e $m_b = 3$.
- If $0.25 < H < 0.30$, then $m_w = 1$ e $m_b = 2.6$.
- If $H \geq 0,30$, then $m_w = 1$ e $m_b = 1$.

(17)

And the threshold is directly calculated by

$$t^* = 256(m_b H_b + m_w H_w)\,. \tag{18}$$

2.8 Silva, Lins and Rocha

The main idea behind this algorithm is to consider the histogram distribution as the 256-symbol source (*a priori source*) distribution. One may assume the hypothesis, as in Pun [Pun, 81], that all symbols are statistically independent. In the case of real images one knows that this hypothesis does not hold. However, this largely simplifies the algorithm and yields good results. Thus, the entropy of the *a priori source* is given by:

$$H = -\sum_{i=0}^{255} p_i \log_2(p_i) \tag{19}$$

where p_i is provided by equation (2). As the resulting image is binarized, the distribution of its histogram may be seen as a distribution of a binary source (*a posteriori source*). The entropy of the *a posteriori source* is given by:

$$H'(t) = h(P_t) \tag{20}$$

where $h(p)$=–p.log$_2$(p)–(1–p).log$_2$(1–p) is the entropy function [Abramson, 63] and P_t is provided by equation (3).

One makes an extension of a binary source to represent without losses all the 256 symbols of the *a priori source*. This new binary source is called *a priori binary source*. The value of the entropy of this new source is given by:

$$H_{\substack{a\,priori \\ binary\ source}} = \frac{H}{\log_2(256)} = \frac{H}{8} \tag{21}$$

One looks for a value of t such that the entropy of the *a posteriori source* were as close as possible to the value of the entropy of the *a priori binary source*, that is, one looks for the following equality:

$$H'(t) = H_{\substack{a\,priori \\ binary\ source}} \tag{22}$$

This argument maps the distribution of the *a posteriori source* onto the distribution of the *a priori binary source*.

Applying equations (20) and (21) to (22), one obtains

$$h(P_t) = \frac{H}{8} \tag{23}$$

One should consider the behavior of the entropy function. For that purpose one must take into account that the target images are of documents, with a much higher frequency of background (paper) pixels than object (print or writing) ones. Thus, it is reasonable to work with the argument of P_t within interval [0, 0.5]. In this interval, the entropy function is injective, thus there is only one value of P_t that satisfies the equation, unless if p_i is zero. In such case it would not matter if the calculated limit were i or $i - 1$.

The target of the proposed algorithm is to filter out the back-to-front interference in binarization. Due to its features, the interference raises the value of the *a priori source*'s entropy. A *loss factor* $\alpha(H_{a\ priori\ binary\ source})$ (24), experimentally determined, is introduced to reduce the presence of the interference.

$$\alpha(H_{\substack{a\,priori\\ b.s.}}) = \begin{cases} -\frac{3}{7} H_{\substack{a\,priori\\ b.s.}} + 0.8 & \text{if } H_{\substack{a\,priori\\ b.s.}} < 0.7 \\ H_{\substack{a\,priori\\ b.s.}} - 0.2 & \text{if } H_{\substack{a\,priori\\ b.s.}} \geq 0.7 \end{cases} \tag{24}$$

Thus, the following relation holds:

$$H'(t) = \alpha(H_{\substack{a\,priori\\ b.s.}}) \cdot H_{\substack{a\,priori\\ b.s.}} \tag{25}$$

Once the bases of the algorithm are presented, its steps are now detailed:

1. one calculates H, the entropy of the image histogram.
2. one scans the t levels, calculating of each of them the distributions $\{P_t, 1-P_t\}$, while $P_t \leq 0.5$, and the entropy associated with that distribution $H'(t) = h(P_t)$;
3. one determines the "optimal" limit that minimizes $|e(t)|$ given as:

$$|e(t)| = \left| \frac{H'(t)}{H/8} - \alpha(H/8) \right| . \tag{26}$$

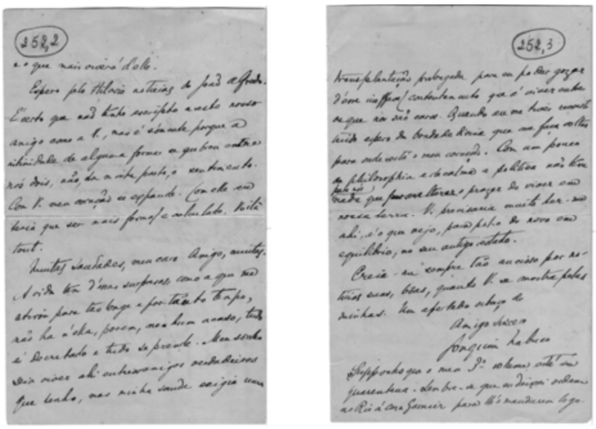
3 Assessment Method

In this section a new method to assess algorithms used to binarize document images with back-to-front interference is presented. The proposed method is divided into two steps:

- *Synthesis of image with interference:* based on two images without back-to-front interference.
- *Calculating the quality factors:* three quality factors that together inform the quality of binarized document are proposed.

3.1 Synthesis of Images with Interference

This section presents how test images with back-to-front interference are generated. The basic idea is to introduce such interference in a well controlled way, thus one is able to really know which pixels ought to be removed and which should not be removed in the filtered image. The mismatching pixels from the reference and filtered images will be used to calculate three quality factors of the algorithm, allowing a fair comparison between the results obtained. As in [Lins-b, 06], the image generation process is detailed.



Front (S) Back (I)

Figure 4: Documents without back-to-front interference from Nabuco's bequest.

1) The first step is take two 256-grayscale images without back-to-font interference, such as the ones presented in Figure 4.

   - **S** – the image that plays the role of the front of the document (signal image); and
   - **I** – the image that plays the role of back (back-to-front interfering image).

2) The second step is to synthesize a third image, called $\mathbf{G}_{fade}$, by overlapping the **S** image with a faded version of the **I** image, as follows:

   - One fades the **I** image, producing the $\mathbf{I}_{fade}$ image given by

$$l = i(m,n) + fade,$$

$$i_{fade}(m,n) = \begin{cases} l & \text{, if } l \leq 255 \\ 255, & \text{if } l > 255 \end{cases}, \tag{27}$$

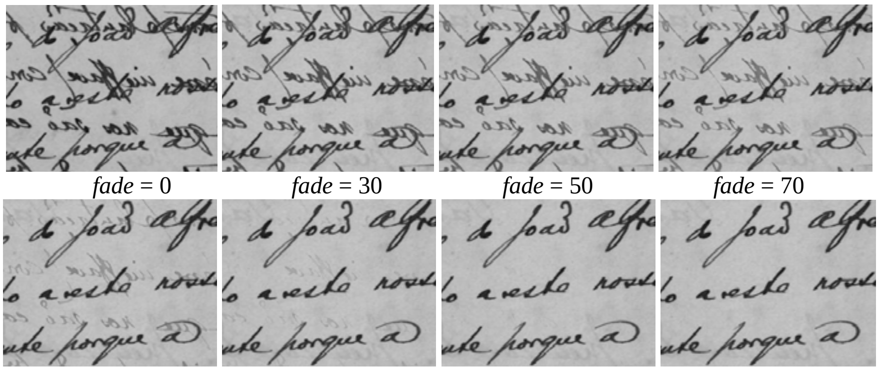
   where the $i(m,n)$ and $i_{fade}(m,n)$ are the intensity values in the pixel (m,n) from the **I** and $\mathbf{I}_{fade}$ images, respectively, and *fade* is the brightness offset applied. Notice that the maximum value of the sum is 255.

   - Finally, the overlapping process merges the **S** and $\mathbf{I}_{fade}$ images, selecting the darker pixel between $s(m,n)$ and $i_{fade}(m,n)$, then,

$$g_{fade}(m,n)=\begin{cases} s(m,n) & \text{, if } \; s(m,n)\le i_{fade}(m,n) \\ i_{fade}(m,n), & \text{if } \; s(m,n)>i_{fade}(m,n) \end{cases}, \tag{28}$$

where $s(m,n)$, $i_{fade}(m,n)$ and $g_{fade}(m,n)$ are the intensity values in the pixel (m,n) from the **S**, $\mathbf{I}_{fade}$ and $\mathbf{G}_{fade}$ images, respectively.

To assess the filtering capability of algorithms *fade* assumes values from 0 to 255. The effect of *fade* variation on the final synthesized document generated from the documents presented in Figure 4 with **I** mirror-reflected is presented in Figure 5.



fade = 0 *fade* = 30 *fade* = 50 *fade* = 70

fade = 100 *fade* = 130 *fade* = 150 *fade* = 170

Figure 5: Pieces of synthesized images for different fade values.

3.2 Calculating The Quality Factors

First one takes as reference the $\mathbf{S}^{(manual)}$ image, which is obtained from **S** by manually searching a threshold that yields a good quality binary image (vide Figure 6). After, one binarizes the $\mathbf{G}_{fade}$ images by the application of the algorithm *k*, generating the $\mathbf{G}_{fade}^{(k)}$ images. Than, one calculates the three quality factors for each $\mathbf{G}_{fade}^{(k)}$ image, in other words, each algorithm will have three quality factor values for each *fade* value.

Before the definitions of the quality factors one needs to define "text area" and "non text area":

- The **Text Area** is the area formed by text in the reference image, in other words, it is the black pixels in the $\mathbf{S}^{(manual)}$ image. The number of pixels in this area is defined as $N_{\text{"text area"}}$.
- The **Non Text Area** is the area formed by the part of the reference image that has no text, in other words, it is the white pixels in the $\mathbf{S}^{(manual)}$ image.

Now one defines the three quality factors:

a) **Quality Factor 1: *Text Error***

The *Text Error* of the $\mathbf{G}_{fade}^{(k)}$ image is defined as

$$q_{fade,k}^{(\text{Text Error})} \triangleq \frac{n_{\text{w,"text area"}}}{N_{\text{"text area"}}}, \tag{29}$$

where $n_{\text{w,"text area"}}$ is the number of white pixels in the $\mathbf{G}_{fade}^{(k)}$ image that are presented in the “text area”, defined by reference image.

b) Quality Factor 2: *Paper Error*

The *Paper Error* of the $\mathbf{G}_{fade}^{(k)}$ image is defined as

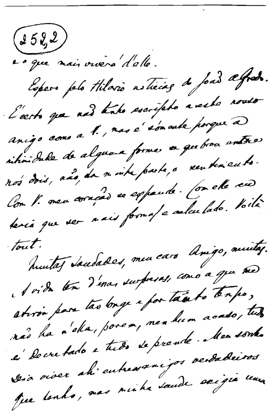


Figure 6: Reference Image S Threshold value chosen by the operator.

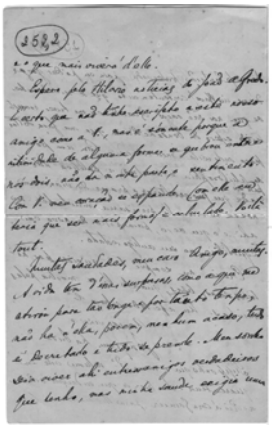


Figure 7: Synthesized Image with fade=80.

$$q_{fade,k}^{(\text{Paper Error})} \triangleq \frac{n_{\text{b,"non text area",paper}}}{N_{\text{"text area"}}}, \tag{30}$$

where $n_{\text{b,"non text area",paper}}$ is the number of black pixels in the $\mathbf{G}_{fade}^{(k)}$ image that are presented in the “non text area”, defined by the reference image, supplied by the paper.

c) Quality Factor 3: *Interference Error*

The *Interference Error* of the $\mathbf{G}_{fade}^{(k)}$ image is defined as

$$q_{fade,k}^{(\text{Interf. Error})} \triangleq \frac{n_{\text{b,"non text area",interf.}}}{N_{\text{"text area"}}}, \tag{31}$$

where $n_{\text{b,"non text area",interf.}}$ is the number of black pixels in the $\mathbf{G}_{fade}^{(k)}$ image that are presented in the “non text area”, defined by the reference image, supplied by the back-to-front interference.

The second and third quality factors defined abouve adopted the same normalization factor as the first, because it is of interest to know the amount of “dirt”, brought by the paper and interference, in relation to the size of the “text area”.

The three quality factors examined together provide information of the filtered image:

- Text Error shows how much text has been erased;
- Paper Error measures the quantity of “dirt” there is in the image due the paper pixels; and
- Interference Error states how much of the original interference is presented in the resulting image.

For a synthesized image using the pair of images in Figure 4, a binary image of acceptable quality, provides quality factors values less than 40%, 50% and 10% for the Text Error, Paper Error and Interference Error, respectively. The first limit depends on the width and gradient of the foreground document, the second depends on the distribution of the paper pixels and the third depends on the interference location in the document image.

4 Results and Analysis

The proposed assessment method was applied to the eight algorithms presented here. For the eight set of 256 binarized images, one for each algorithm, the three quality factors introduced were measured and their graphics were plotted. This experiment was made for twenty pairs of images from the Nabuco’s bequest. The results of applying the proposed method using the pair of images shown in the Figure 4 are presented in this paper. Figure 8 presents eight graphs, one for each algorithm, that contain three curves generated by the three quality factors. Notice that the range of the quality factor axis is different for each algorithm. Figure 9 presents three graphs; each of them brings the same quality factor for all of the algorithms. Notice that the range of the quality factor axis is from 0 to 100 in the three graphs.

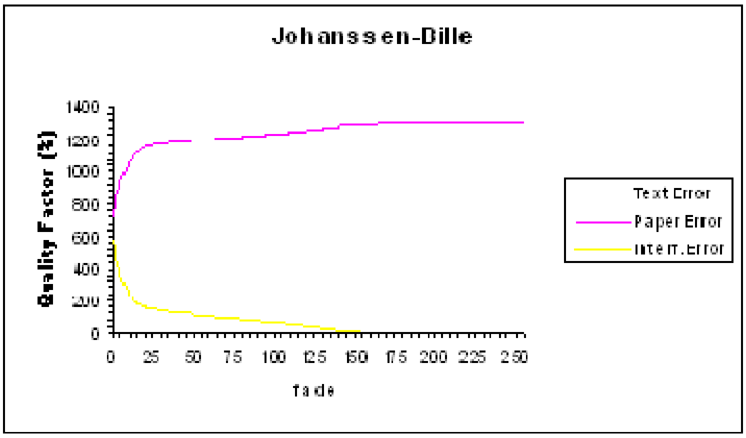
The analysis of the graphs in Figure 8 and 9 allows one to observe that the Johanssen-Bille algorithm always produced the highest values of the Paper Error and Interference Error factors. The Text Error factor always was zero, but this can not help it. One can see in Figure 10a the result of applying Johanssen-Bille algorithm on the image shown in Figure 7, that was synthesized with *fade*=80, one of the most frequent values of back-to-front interference in the Nabuco’s bequest.

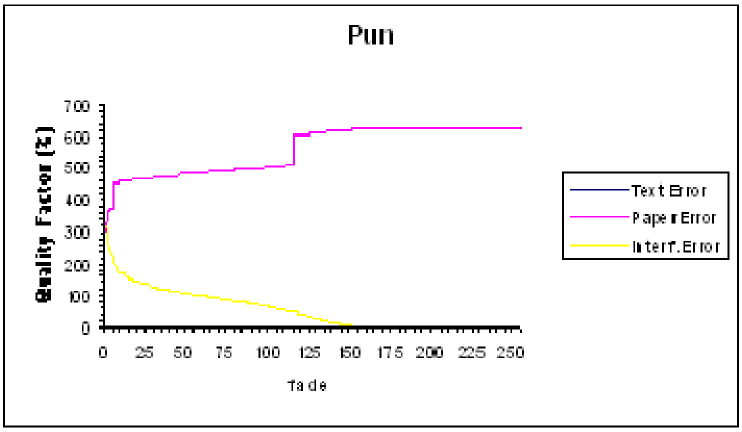
Algorithm	**Text Error (%)**	**Paper Error (%)**	**Interf. Error (%)**
Johanssen-Bille	0	1,215.66	86.51
Pun	0	492.09	83.90
Yen-Chang-Chang	0	41.38	66.78
Kapur-Sahoo-Wong	0	28.48	57.23
Otsu	0	6.40	26.31
Mello-Lins	0	9.67	32.44
Wu-Songde-Hanqing	50.22	0	0
Silva-Lins-Rocha	6.13	0	6.07

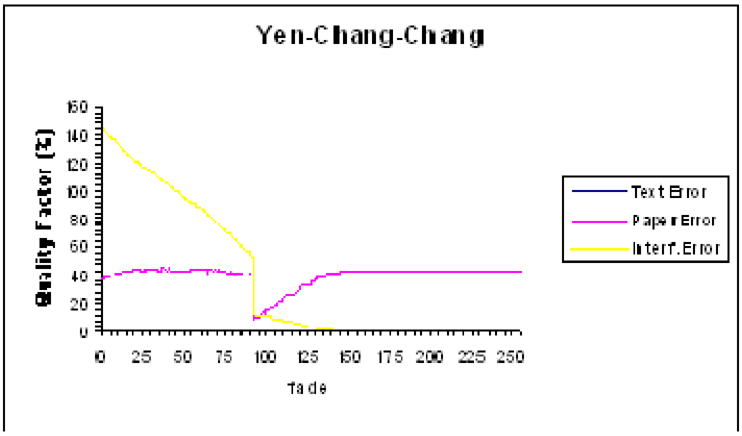
Table 1: Values of the three quality factors (fade=80).

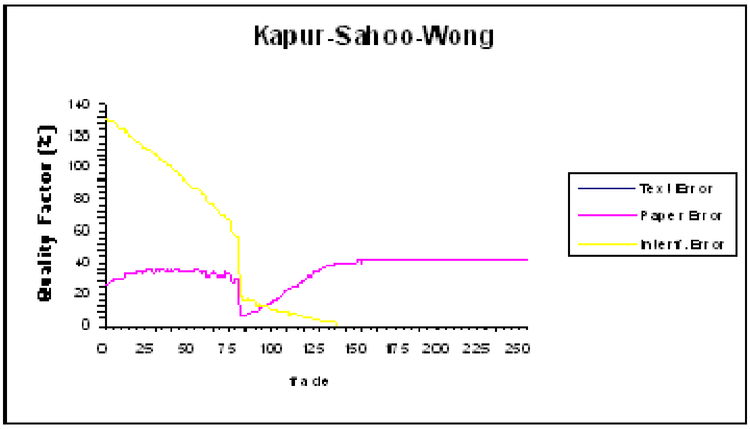
Table 1 shows the values of the quality factors for all binarized image ($\mathbf{G}^{(k)}_{fade=80}$), that is the result of applying algorithm *k* to the $\mathbf{G}_{fade=80}$ image presented in Figure 7.

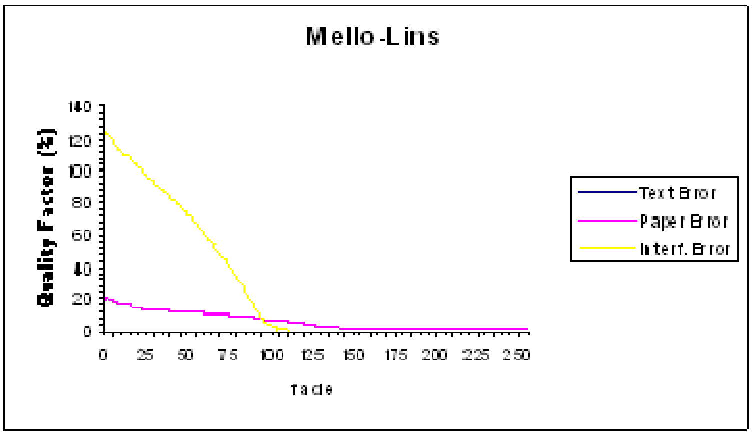
The performance of Pun´s algorithm, although far superior than Johanssen and Bille´s, as may be observed from the plots in Figures 8 and 9, yields unsatisfactory images (see Figure 10b).






Yen-Chang-Chang
Quality Factor (%)
160
140
120
100
80
60
40
20
0
0 25 50 75 100 125 150 175 200 225 250
fade
Text Error
Paper Error
Interf. Error


Kapur-Sahoo-Wong
Quality Factor (%)
140
120
100
80
60
40
20
0
0 25 50 75 100 125 150 175 200 225 250
fade
Text Error
Paper Error
Interf. Error


Mello-Lins
Quality Factor (%)
140
120
100
80
60
40
20
0
0 25 50 75 100 125 150 175 200 225 250
fade
Text Error
Paper Error
Interf. Error

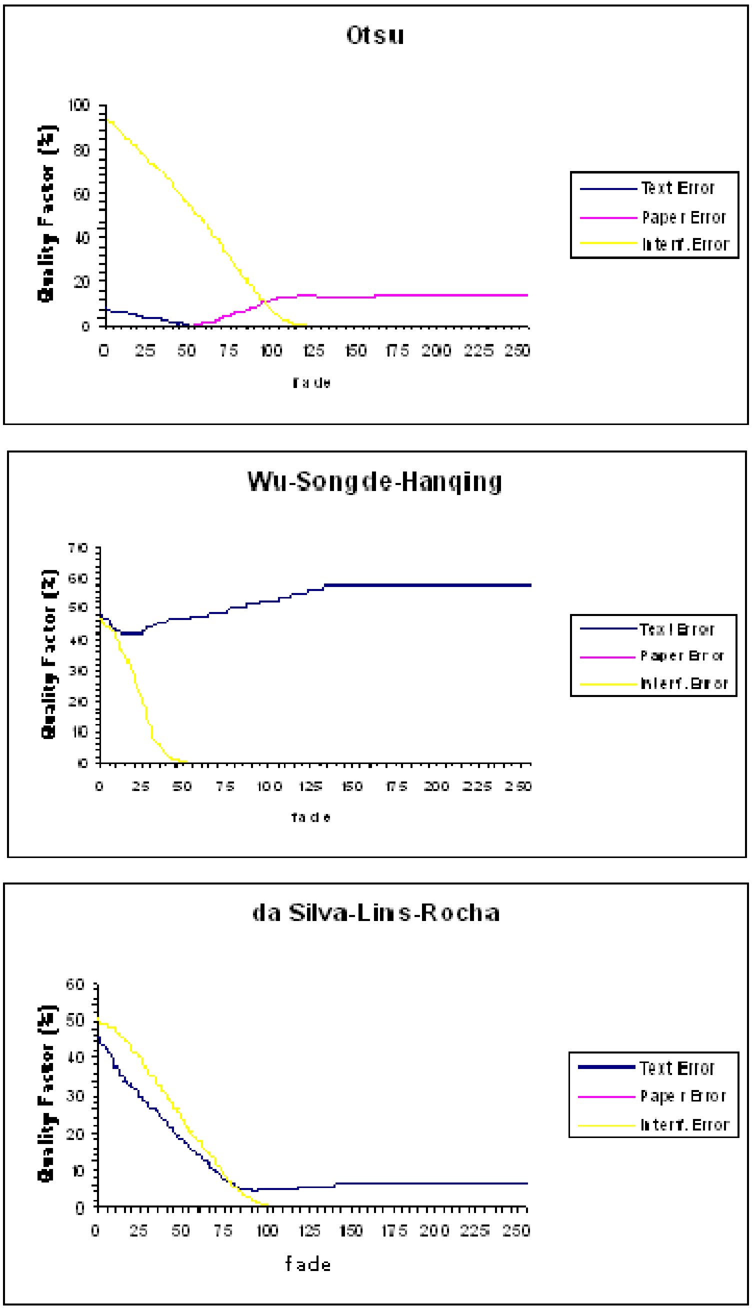
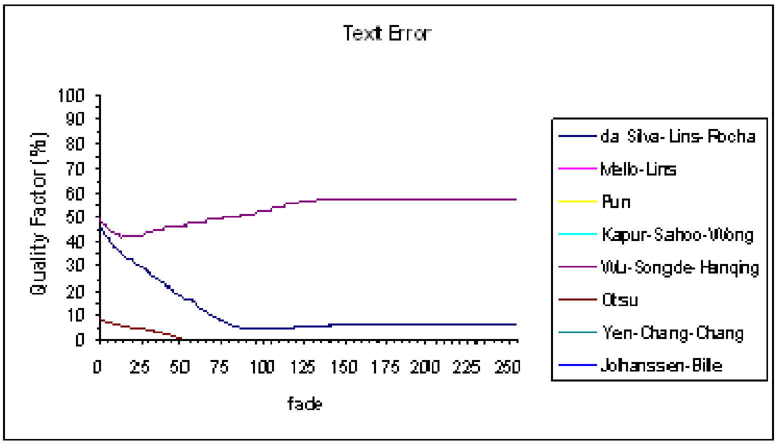
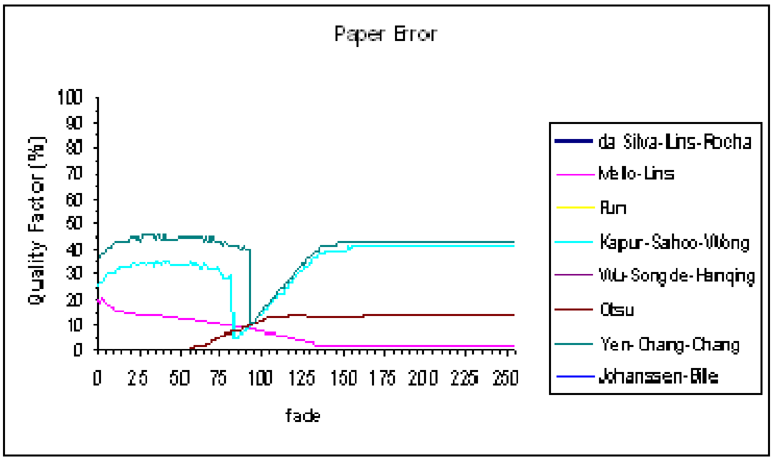


Figure 8: Graphs of the three quality factors for each algorithm.

According to the assessment method proposed herein six algorithm of the eight algorithms analyzed are suitable to remove the bleeding noise in documents. Their performances vary according to the strength of back-to-front interference.

The graphs in Figure 8 and 9 show that the algorithms by Yen-Chang-Chang and Kapur-Sahoo-Wong only produce reasonable filtering for images with medium-to-weak back-to-front interference (*fade*≥110). In the most frequent noise region (*fade*≈80) these algorithms are unable to filter out significant amount of the back-to-front interference, as shown in Figure 10c and 10d. Analyzing the graphs shown, Silva-Lins-Rocha, Mello-Lins and Otsu algorithms are able to filter images with *fade* greater than 70, 90 and 100, respectively, enhancing their performances as the noise weakens. The resulting images from the Otsu and Mello-Lins algorithms are shown in Figure 10e and 10f. For images with strong noise (30≤*fade*≤60), the algorithm proposed by Wu-Songde-Hanqing has good chances of performing well in back-to-front noise removal, however it tends to be greedy and remove part of the foreground information as one can evidence by its Text Error values in the Figures 8 and 9 and Table 1. Figure 10g presents the result of applying that algorithm with *fade* = 80.





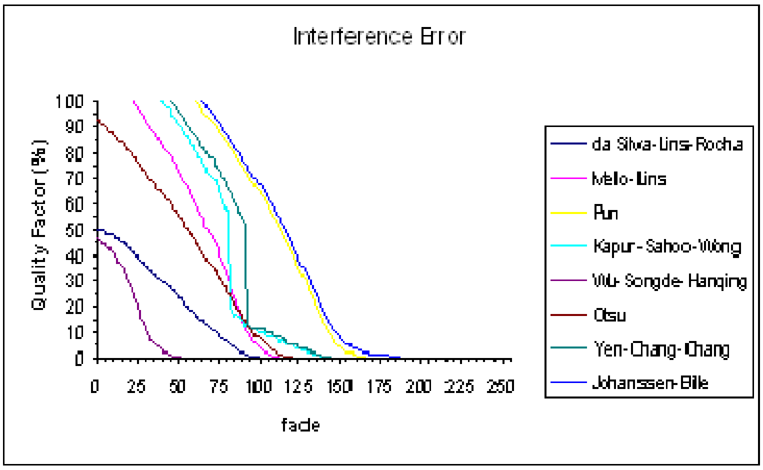
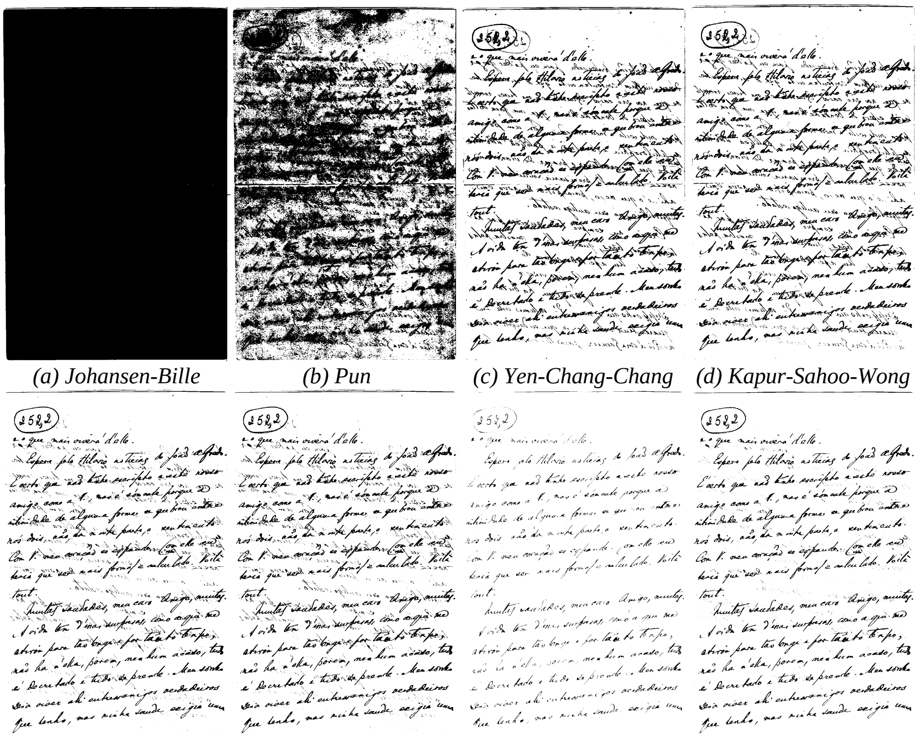


Figure 9: Graphs of the three quality factors for all algorithms.



(a) Johansen-Bille (b) Pun (c) Yen-Chang-Chang (d) Kapur-Sahoo-Wong

(e) Otsu (f) Mello-Lins (g) Wu-Songde-Hanqing (h) Silva-Lins-Rocha

Figure 10: Filtered images by the algorithms presented in section 2 (fade=80).

The steadiest good performance in filtering out the bleeding noise is provided by Silva-Lins-Rocha algorithm, whose may be seen in Figure 10h. As one can see in Figures 8 and 9, the Interference Error curve of the Silva-Lins-Rocha stands below the curve of Wu-Songde-Hanquing; however, the Text Error curve of the Silva-Lins-Rocha decreases while the curve of Wu-Songde-Hanquing increases.

Although the six suitable algorithms have worked well in the cases of images with very low back-to-front noise (*fade*>120), it is important to say that in the most of the experiments, Otsu's algorithm worked better then the others.

5 Conclusions and Lines for Further Work

A quantitative method to assess the quality of binarization algorithms for images with back-to-front interference was introduced. The results obtained with this assessment method are consistent with the obtained by visual inspection of filtered documents.

The quality factors introduced herein are able to inform whether the application of an algorithm yields a readable or unreadable binary document. An important point of the proposed method is that it is able to spot which algorithm is more likely to perform better at filtering out the bleeding noise by analyzing the features of the document. This attribute may allow the automatic choice of the best suitable algorithm to filter a specific document, thus permitting to be incorporated into an automatic document processing environment such as BigBatch [Lins-a, 06].

The assessment method proposed here did not take into account the color of the background as a controlled parameter. Work on progress are widening the scope of this work to model aged background, giving complete control of all document parameters.

Acknowledgements

Research reported herein was partly sponsored by CNPq – Conselho Nacional de Pesquisas e Desenvolvimento Tecnológico, Brazilian Government.
The authors are grateful to the Joaquim Nabuco Foundation for granting the permission to use the images from Nabuco´s bequest.

References

[Abramson, 63] N. Abramson, "Information Theory and Coding", McGraw-Hill Book Co, 1963.

[Adobe, 07] Adobe Systems Inc. http://www.adobe.com.

[FUNDAJ, 07] FUNDAJ – Fundação Joaquim Nabuco: http://www.fundaj.gov.br

[Johannsen, 82] G. Johannsen and J. Bille, "A threshold selection method using information measures", ICPR'82, pp. 140–143 (1982).

[Kapur, 85] J. N. Kapur, P. K. Sahoo and A. K. C. Wong, "A New Method for Gray-Level Picture Thresholding using the Entropy of the Histogram", C.Vision, Graph. and Im.Proc., 29(3), 1985.

[Kavallieratou, 05] E. Kavallieratou and H. Antonopoulou, “Cleaning and Enhancing Historical Document Images”, Intelligent Vision Systems, Springer-Verlag 3708, pp. 681-688, 2005.

[Leedham, 02] G. Leedham, et al., “Separating text and background in degraded document images—a comparison of global thresholding techniques for multi-stage thresholding”, Proceedings of the Eighth International Workshop on Frontiers in Handwritten Recognition, pp. 244–249, 2002.

[Lins, 95] R. D. Lins, et al, “An Environment for Processing Images of Historical Documents”, Microproc. & Microprogramming, pp. 111-121, North-Holland, 1995.

[Lins-a, 06] R.D.Lins, B.T.Ávila, and A.A.Formiga, “BigBatch: An Environment for Processing Monochromatic Documents”, ICIAR2006, LNCS 4142, pp.886-896, Springer Verlag 2006.

[Lins-b, 06] R. D. Lins and J. M. M. da Silva, “Assessing Algorithms to Remove Back-to-Front Interference in Documents”, ITS-2006, Fortaleza, Brazil, IEEE Press 2006.

[Lins, 07] R. D. Lins and J. M. M. da Silva, “A Quantitative Method for Assessing Algorithms to Remove Back-to-Front Interference in Documents”, ACM-SAC 2007, Seoul (Korea), pp.610-616, ACM Press, 2007.

[MathWorld, 07] MathWorld: http://www.mathworld.com.

[Mello, 00] C. A. B. Mello and R. D. Lins, “Image segmentation of historical documents”, Visual 2000, Mexico City, Mexico, 2000.

[Mello, 02] C. A. B. Mello and R. D. Lins, “Generation of images of historical documents by composition”. ACM Document Engineering 2002, McLean, VA, USA.

[Oha, 05] Hyun-Hwa Oha, Kil-Taek Limb, Sung-Il Chienc, “An improved binarization algorithm based on a water flowmodel for document image with inhomogeneous backgrounds”. Pattern Recognition 38 (2005) 2612 – 2625, 2005.

[Otus, 79] N. Otsu, “A threshold selection method from gray level histograms”, IEEE Tran. Syst. Man Cybern., 9, 62–66 (1979).

[Pun, 81] T. Pun, “Entropic Thresholding, A New Approach”, C. Graphics and Image Processing, 16(3), 1981.

[Sankur, 04] B. Sankur and M. Sezgin, “A survey over image thresholding techniques and quantitative performance evaluation”, Journal of Electronic Imaging, 13(1), 146-165 (2004).

[Silva, 06] J.M.M. da Silva, R.D.Lins and V.C.da Rocha Jr. “Binarizing and Filtering Historical Documents with Back-to-Front Interference”, In: ACM Symposium on Applied Computing, 2006, Dijon. Proceedings of SAC 2006. New York : ACM Press, 2006. p. 853-858.

[Tan, 02] C. L. Tan, R. Cao, P. Shen, Restoration of archival documents using a wavelet technique, IEEE Trans.Patt. Analysis and M.Intelligence, 24(10), pp. 1399-1404, 2002.

[Wang, 01] Q. Wang, C. L. Tan, Matching of double-sided document images to remove interference, IEEE CVPR2001, Dec 2001.

[Wu, 98] L. U. Wu, M. A. Songde, and L. U. Hanqing, “An effective entropic thresholding for ultrasonic imaging”, ICPR’98: Intl. Conf. Patt. Recog., pp. 1522–1524 (1998).

[Yen, 95] J. C. Yen, F. J. Chang, and S. Chang. “A new criterion for automatic multilevel thresholding”. IEEE Trans. Image Process. IP-4, 370–378 (1995).