Document Image Binarization using Local Features and Gaussian Mixture Modelling

Nikolaos Mitianoudis, Nikolaos Papamarkos

Image Processing and Multimedia Laboratory,
Department of Electrical and Computer Engineering,
Democritus University of Thrace,
67100 Xanthi, Greece

Abstract

In this paper, we address the document image binarization problem with a three-stage procedure. First, possible stains and general document background information are removed from the image through a background removal stage. The remaining misclassified background and character pixels are then separated using a Local Co-occurrence Mapping, local contrast and a two-state Gaussian Mixture Model. Finally, some isolated misclassified components are removed by a morphology operator. The proposed scheme offers robust and fast performance, especially for both handwritten and printed documents, which compares favourably with other binarization methods.

Keywords: Binarization, handwritten documents, historic documents, classification, background estimation

1. Introduction

Document images commonly arise from historical documents, books or printed documents that are digitized using a scanning device. The advancement of imaging devices, such as scanners and digital cameras, has widely facilitated the digitization of paper-printed material, including historical documents and books. Many libraries throughout the world, such as the British Library in London, UK[1], have digitized books, manuscripts and other printed material from their collection, which are available online as images. We can extract the text information from these document images using Optical Character Recognition (OCR) techniques. Nevertheless, to enhance the performance of OCR algorithms, a number of preprocessing steps are systematically applied, including page skew detection, artifact and noise removal, document page layout analysis

Email address: nmitiano@ee.duth.gr (Nikolaos Mitianoudis)

[1] http://www.bl.uk/aboutus/stratpolprog/digi/digitisation/

and document image binarization [1, 2, 3, 4]. In this paper, we address the problem of background removal and document image binarization.

Scanned documents often contain undesired textual noise, such as specks, dots, black borders, lines, and hole-punch marks. *Background estimation* and removal is a preparatory step that enhances the quality of the document images and is beneficial for binarization techniques [5, 6, 7, 8]. For example, historic document images often suffer from different types of degradation that render document image binarization and character recognition very challenging tasks. In summary, the main objective of background removal techniques is to remove all these degradations from a document image and enhance the discrimination of characters from the page background.

After the original document images have been enhanced, the output of most document processing systems is a bi-level image containing characters and background. Image binarization can then be performed either on a global or a local basis. Conventional binarization techniques of gray-scale documents were initially based on global thresholding algorithms (clustering approaches) [9], which have proved to be efficient in converting simple gray-scale images into a binary form but are inappropriate for complex documents, and degraded documents. For this purpose, the local binarization techniques of Niblack [10], Sauvola [11] and Bernsen [12] have been extensively used by the document image processing community. There are numerous specialized binarization techniques for document images (see [13] for a more detailed review). Here, we will outline several important binarization methods that have appeared so far.

In [1], Papamarkos proposed a neuro-fuzzy technique for binarization and gray-level (or color) reduction of mixed-type documents. Badekas and Papamarkos [13] proposed a binarization technique that combines the results of multiple binarization algorithms using a Kohonen Self-Organizing Map (KSOM) neural network. In [14], the binarization results of many independent techniques were initially produced and then combined with a Kohonen Self-Organising Map (KSOM). Badekas et al. [15] also introduced a binarization technique, specialized for color documents, where the resulting "binary" image contains the detected text regions with black characters in white background leaving the remaining original color parts of the document intact. In [16], Makridis and Papamarkos introduced a two-stage approach to image binarization. The first stage included a background removal technique that was based on fixed-size median filtering of the document image. Once the background was removed, the second stage aimed at creating 2D clusters of neighboring pixels of similar intensity, i.e. document characters and background. Binarization was then performed by identifying 2 clusters (text-background) using the multithresholding technique of Reddi et al [17].

Gatos et al. [18] (GPP method) estimated the document background by an adaptive threshold which labels each pixel as either text or background. To estimate the background surface, they used Sauvola's binarization algorithm to roughly extract the text pixels and calculated the background surface from them by interpolation of neighboring background pixels intensities. For the other pixels, background surface is set to the gray level of the original image.

Ntirogiannis et al. [19] proposed a modular system for handwritten document binarization. Background is initially estimated via an inpainting procedure starting from the Niblack binarization output. The background estimate is then normalised to smooth great variations and is used as an input to Otsu's global thresholding which removes most unwanted noise but also some faint characters. Therefore, the local binarization algorithm of Niblack is also used, but initialised using the stroke width information, extracted by skeletonization of Otsu's output, window size and contrast information. The two binarization outputs are combined at connected component level.

In [20], Su et al. demonstrated the use of local contrast image thresholding in estimating the text stroke width more accurately. In [6], Lu et al. performed background estimation using a modified version of 1D iterative polynomial smoothing to compensate for several degradation types. Text-stroke edges are then identified via Otsu's global thresholding on L1-norm horizontal and vertical edge detection. Document text pixels are extracted, since they are surrounded by text stoke edges and feature lower intensity levels.

Hedjam et al [7] used grid-based modelling and impainting techniques to recover text pixels starting from an under-binarization result using Sauvola's technique. The proposed technique featured smooth and continuous strokes, due to its spatially adaptive estimation of the text pixels' statistical features. Moghaddam and Cheriet [8] presented an adaptive form of Otsu's thresholding for binarization. Based on a rough binarization result, they produce an estimated background and a stroke gray level map using a multi-scale framework. This estimated background is further refined using the AdOtsu method, which is an adaptive, parameterless form of Otsu's thresholding, which is generalised to a multiscale setup. Finally, skeleton-based post-processing is employed to remove possible artifacts and sub-strokes.

Valizadeh and Kabir [21] devised a novel feature space consisting of the structural contrast and the intensity value of each pixel. Structural contrast relates text stroke width, pixels' intensities and their relationships with their neighbours at stroke width distances. This results in a 2D image representation where text and background pixels are separable. Clustering is performed by partitioning the feature space into small regions. Then, using the result of another binarization algorithm with at least 50% successful labeling (Niblack), each region is classified either as background or text, according to the prevailing number of text or background pixels in the region. The reverse procedure procedure produces the document binary image.

Howe [2] performed binarization by minimising a global energy functional inspired by Markov Random Fields, where a) the image Laplacian edge map is employed to distinguish between ink and background in the energy data fidelity term and b) ink discontinuities are enforced in the binarization result by incorporating a Canny edge detector into the smoothness term. Howe also introduced a procedure to automate the optimal parameter selection for his algorithm.

Ramirez-Ortegon et al [22] introduced the concept of transition pixel, i.e. calculating intensity differences over a small neighborhood, which can then be employed by common gray-level thresholding algorithms to produce a bina-

rization result (transition method). This was further refined in [23], where an unsupervised thresholding was proposed for unimodal histograms, assuming Gaussian priors for the distribution of character and background neighborhoods. In [4], the method was enriched with a mechanism to remove binary artifacts after binarization. An auxiliary image is calculated via minimum-error-rate thresholding. The connected components of the auxiliary and the original binary image are compared in terms of an intersection ratio to remove possible binarization artifacts. In [24], Ramirez-Ortegon et al explored possible effects of inaccurate estimations of the transition proportion on the estimated thresholds. In [25], Ramirez-Ortegon et al proposed the use of skewed log-normal, instead of symmetrical Gaussian, priors [23] for the background and character clusters.

Lelore and Bouchara [3] introduced the FAIR binarization algorithm, where they ran the S-FAIR (simplified) algorithm for two threshold values: one giving a noiseless binarization output but with important edges missing and another containing all character edges but with some misclassification noise. The S-FAIR algorithm first performs text localization using the Canny algorithm. A Gaussian Mixture Model is then used to classify pixels around edges to belong either to the text or the background image or to a third class where pixels cannot be attributed with certainty to text or background. The FAIR algorithm merges the two outputs with a "max" rule. Finally, a post-filtering process classifies unknown pixels using a variety of rules. The most important feature is an iterative procedure where the text labeled regions grow into the unknown using morphological dilation and the previous EM algorithm is used to define the final class of these regions. Final unknown areas are connected morphologically and labeled according to neighborouring pixels.

In this paper, the authors extend the previous work of Makridis and Papamarkos [16] towards a more automated three-stage document image binarization system. In the first stage, the background removal technique in [16] is enhanced by automating the window size selection for the median filter and improving the threshold selection between the document image and the background estimate. In the second stage, the proposed local neighborhood representation is redesigned to also include local contrast information to enhance the presence of character outlines. Binarization is then performed by separating two clusters of document characters and background artifacts that were not removed in the first stage of background removal. Clustering is performed using Mixtures of Gaussians (MoG). The Gaussian with lowest value mean corresponds to the character cluster. The local neighborhood representation share a similar concept with those introduced by Valizadeh and Kabir [21] and Ramirez-Ortegon et al [22], however, the proposed multidimensional representation is different to the 1D representations discussed in [21, 22]. Contrast information for binarization was also used by Su et al [20], however, in this work contrast is incorporated into a local intensity representation forming a joint, rather than an isolated feature. Similarly, Gaussian modelling for binarization has been employed before by Hedjam et al [7] and Ramirez-Ortegon et al [23], but here it is applied on the novel LCM representation. Moreover, MoG-based clustering is a common clustering technique in pattern recognition, thus it is the application that is

novel here. In the final post-processing stage, small-size 8-connected clusters are removed to eliminate possible binarization noise.

The paper is organised as follows: Section 2 sets the essential notation and outlines the system. Section 3 describes the background removal process in detail; Section 4 describes the binarization stage using GMM clustering; Section 5 explains the post-processing step; Section 6 presents the evaluation results of the proposed methodology and finally Section 7 concludes this paper.

2. System Description

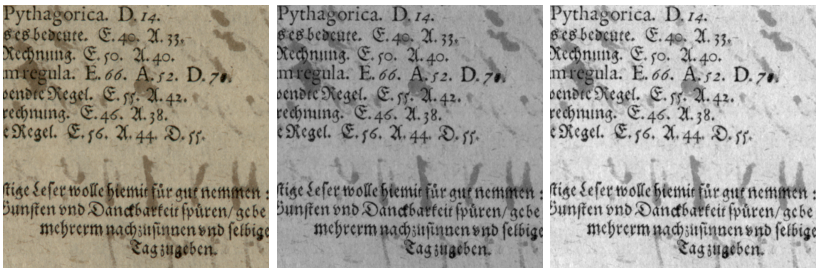
Let $\mathbf{I}(x, y)$ be the initial color document image of size $3 \times M \times N$, where x, y denote integer samples across the horizontal and vertical axes. The desired output of a document image binarization algorithm is a bi-level $M \times N$ image $I_{BN}(x, y)$ that attributes the value 255 (white) to background pixels and the value 0 (black) to character pixels. It consists of three stages: a) the Background Removal stage, b) the Image Binarization stage and c) the post-processing stage, which are then presented in detail.

3. Background Removal Stage

Background removal is a preprocessing stage in a document binarization system that can eliminate the presence of artifacts, including stains, paper cuts, paper coloring and opposite-page ink leaks, prior to binarization.

3.1. Grayscale conversion

The first step is to map the three-channel RGB image to an one-channel intensity image that detains all the useful information from all color channels. One method is to simply average all three channels to create the intensity image, which has been shown not to be effective in our experiments. Another method is to move to another color space, such as the Hue - Saturation - Luminance (HSL) cylindrical color space, where the color information (H S channels) is isolated from the Luminance (L) channel, which is kept for further processing (as implemented by MATLAB's `rgb2gray` function). Several techniques have also been proposed that attempt to produce gray-scale images with visual contrast similar to the color contrast [26, 27]. In [28], a linear transform is proposed that converts a color image to a gray-scale image in such a way that the variance of the transformation is maximized and at the same time, the gray-scale image preserves the brightness of the color image. Also, Kanan and Cottrell [29] proposed new techniques for general color to gray conversion. Recently, Moghaddam and Cheriet [30] developed a new heuristic technique that is based on a dual transformation, color reduction and interpolation. In order to ensure that all useful information from all color channels is conveyed to the grayscale image, we perform Principal Component Analysis [31] on the multichannel image. The principal component image is then retained as the grayscale image. This methodology for grayscale conversion is pursued in our system. In Fig. 1,



(a) Initial Color Document Image (b) Grayscale Image using the `rgb2gray` function (c) Grayscale Image using PCA

Figure 1: Transforming the original color document image to grayscale using PCA seems to improve the output's contrast.

we can see an example of a color document image conversion to grayscale using PCA. The final grayscale image appears to have increased contrast compared to a typical grayscale conversion.

3.2. Background estimation

The proposed background removal algorithm is based on the observation that the aforementioned artifacts can be isolated from the original image by performing low-pass filtering of long window size [16]. This long-window low-pass filtering can essentially filter out the document characters, as they are generally small-size high-pass details, leaving only an image containing artifacts and the document background that needs to be removed. Median filtering is more preferable to ordinary low-pass filtering, since this will not create new intensity values in the document image, but will simply replace the character intensity values with background or artifact intensity values. Nonetheless, the size of the median filter window needs to be defined. In [16], Makridis and Papamarkos used a fixed window size, which was defined by the user. In this study, we propose to automate the procedure, by starting with a small median filter window of size $G = 5$. After median filtering the input image $I(x, y)$, we measure the standard deviation of every possible 3×3 image patch. If the standard deviation of the majority of image patches (e.g. 98%) is greater than a threshold value S_I (e.g. $S_I = 6$), this implies that image still contains character information and the median filter window has to increase by 5 , i.e. $G \leftarrow G + 5$. This procedure is repeated until most 3×3 patches have low standard deviation, i.e. low-order texture, background. The final image $I_{MED}(x, y)$ is an estimate of the document background. The above values values of 98% and $S_I = 6$ have been determined by experimentation on the DIBCO [32, 33, 34, 35, 36] image datasets and remain unchanged. A more detailed study to determine the statistical properties of a background image is presented by Ramirez-Ortegon et al [25], where similar values for S_I are reported. A more extensive investigation of this parameter goes beyond the scope of this paper, since it does not appear

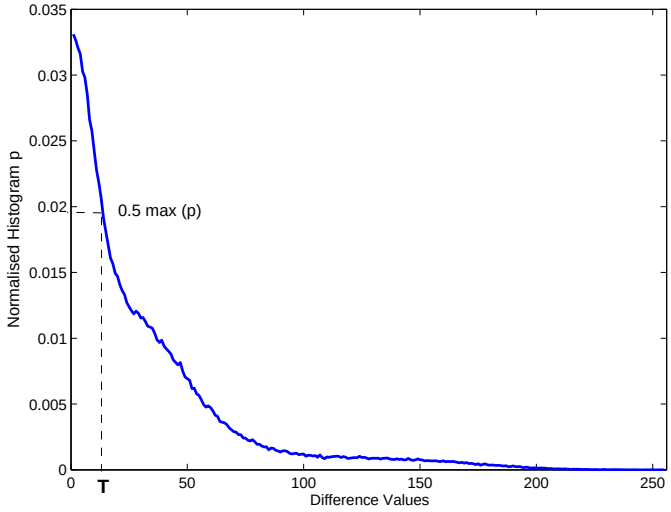


Figure 2: Estimating the threshold for background removal for $q = 0.5$.

to greatly affect performance.

3.3. Background Removal

To remove the document background from the document image and form the "No-Background" $I_{NBG}(x, y)$ image, a simple comparison classifies every pixel (x, y) as background or text. If the absolute difference between the original image intensity $I(x, y)$ and the $I_{MED}(x, y)$ is below a selected threshold value T, then this pixel must be part of the document background and is attributed the value white, i.e. $I_{NBG}(x, y) = 255$. In the opposite case, this pixel is very different from the background image and thus must be a character pixel. Therefore, we set $I_{NBG}(x, y) = I(x, y)$.

In Fig. 3, we depict the various stages of the background removal algorithm. An original document image of size 682×690 is depicted in Fig. 3 (a). The background image estimate for $G = 20$ is shown in Fig. 3 (b) and the final estimate for $G = 30$ in Fig. 3 (c). The document image with the estimated background removed for various values of T are shown in Fig. 3 (d), (e). Selecting larger values of T, more parts of the document image are classified as background and thus are removed (transformed to 255) from the image. Hence, the value of T can define the background removal strength of the algorithm. However, selecting larger values for T may remove character information apart from unwanted noise.

One can make the selection of T more adaptive, by calculating a histogram of $|I(x, y) - I_{med}(x, y)|$. Dividing histogram values by the number of image pixels, we get an approximate probability density estimate p_i of the previous difference.

In most document images we encountered in this study, this density seems to be a decreasing function (see Fig. 2). Thus, an adaptive threshold value of T can be set at the point, where this probability falls below a fraction q of this maximum value, i.e. $q \max(p_i)$ with $q \in [0, 1]$. This provides a more general threshold which is more adaptive for different images than selecting a specific value for T. Lowering q removes more background information, while increasing q leaves more background information unprocessed. In our system, we tend to keep the background removal stage less strict, so as not to accidentally remove character parts or outlines in the bakcground removal stage. In Fig. 3 (f), the background removal result is depicted using a value of $q = 0.5$. Although thresholding is now more adaptive to a variety of document images, the parameter q has to be manually selected. The specification of this parameter remains key to the performance of the binarization stage, as it will be explained in the experimental section.

This image is then presented to the image binarization algorithm, described in the next section. The proposed algorithm is summarized below:

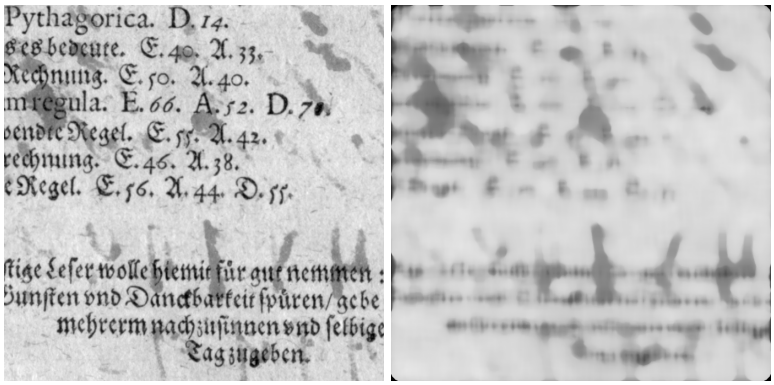
Document Image Background Removal Algorithm

1. Transform the initial $M \times N$ color document image $\mathbf{I}(x, y)$ to an 1-channel image $I(x, y)$, using only the Principal Component.
2. Set a neighborhood size $G \times G$, where $G = 5$.
3. Estimate
$$I_{MED}(x, y) = \text{median}_G \left(I(x, y) \right) \tag{1}$$
4. Calculate the standard deviation $\sigma(x, y)$ of every 3×3 patch in $I_{MED}(x, y)$.
5. If the number of patches that satisfy the condition $\sigma(x, y) < S_I$ is less than $0.98MN$ then set $G \leftarrow G + 5$ and repeat steps 3, 4, 5.
6. Estimate a value for T, where the normalised histogram p_i of $|I(x, y) - I_{med}(x, y)|$ falls below $q \max(p_i)$.
7. The document image without background $I_{NBG}(x, y)$ is given by:
$$I_{NBG}(x, y) = \begin{cases} I(x, y), & \text{if } |I(x, y) - I_{med}(x, y)| > T \\ 255, & \text{if } |I(x, y) - I_{med}(x, y)| \leq T \end{cases} \tag{2}$$

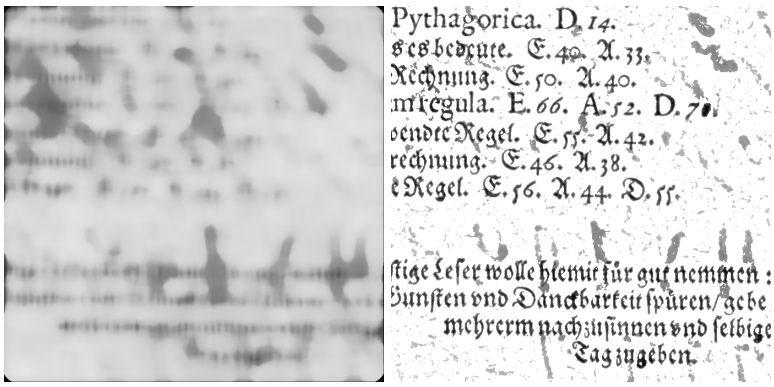
4. Image Binarization

Document Image Binarization is defined as the process where a grayscale document image $I(x, y)$ is transformed into a bi-level image $I_{BN}(x, y)$, where $I_{BN}(x, y) = 0$ for each pixel (x, y) that is attributed to a document character and $I_{BN}(x, y) = 255$ for each pixel (x, y) that is attributed to background. Local thresholding methods seem to offer more stable solutions, exploiting local statistical measurements, including the local mean, standard deviation, entropy and contrast.

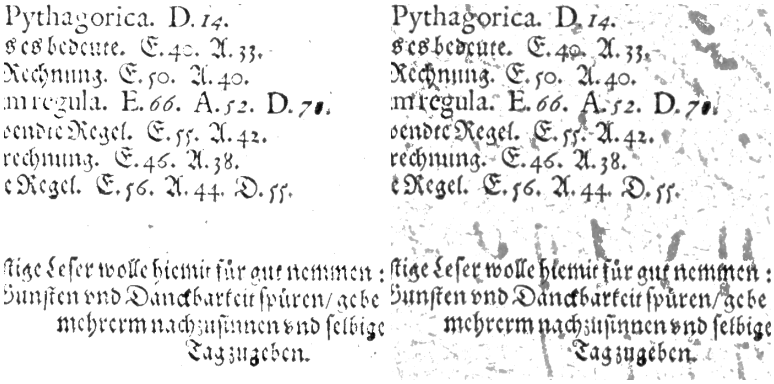
Some other local character properties that can be exploited to perform binarization are the following:



(a) Initial grayscale Document Image (b) Background Image estimate for $G = 20$

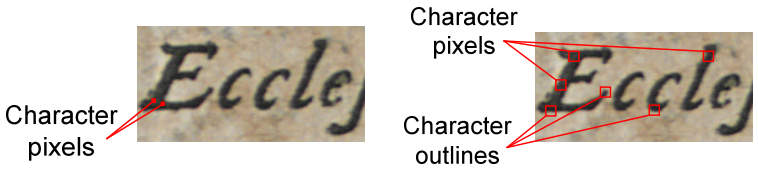


(c) Final Background Image estimate ($G = 30$) (d) Background removal $T = 10$



(e) Background removal $T = 50$ (f) Background removal $q = 0.5$

Figure 3: Background Estimation for various values of G and T and the final document image after background removal



(a) (b)

Figure 4: (a) Pixels belonging to the same character are geometrically close and feature similar intensities. (b) Local areas around character outlines should have increased contrast compared to areas inside the characters.

- Pixels belonging to the same character are geometrically close.
- Pixels belonging to the same character should feature similar intensity values.
- Any local area (neighborhood) that includes the outline of a character should have increased contrast, compared to areas containing only background or only character pixels.

In Fig. 4, we show some examples of the above principles in a document image. These principles were also discussed in a more mathematical manner in [22].

In this section, we will use these properties to create a Local Co-occurrence Mapping (LCM) that will assist us in discriminating between the character and the background pixels. The first two properties were initially discussed in [16], leading to the introduction of a Symmetrical Frequency Map (SFM) that was used to perform binarization. Here, we extend this framework to use these three properties simultaneously and increase binarization performance.

4.1. Improved LCM representation

To emphasize proximity and connectivity between neighboring character pixels, the main concept is to devise a co-occurrence map in the following manner. The image is divided into every distinct $Q \times Q$ patch. This implies that these patches are created with 1-pixel overlap from the original image. Let (x_c^i, y_c^i) be the center pixel of the i-th patch. Each distinct patch is then transformed to the following $(Q^2 - 1)$ points in the 2D space given by:

$$\begin{bmatrix} I_{NBG}(x_c^i, y_c^i) \\ I_{NBG}(x_c^i + dx, y_c^i + dy) \end{bmatrix}, \qquad \forall \ -\lfloor Q/2 \rfloor \leq dx, dy \leq \lfloor Q/2 \rfloor \tag{3}$$

In other words, each pixel in the i-th patch is transformed to a 2D point containing the intensity of the central patch pixel and the pixel's intensity. The whole procedure is visualised in Fig. 5. We note that the combination of the center pixel with itself is not included in the formation of this group of 2D points, since it does not offer any information about connectivity. Thus,

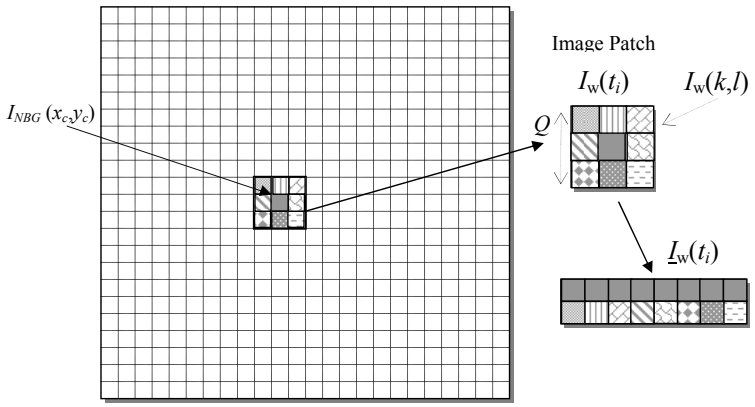
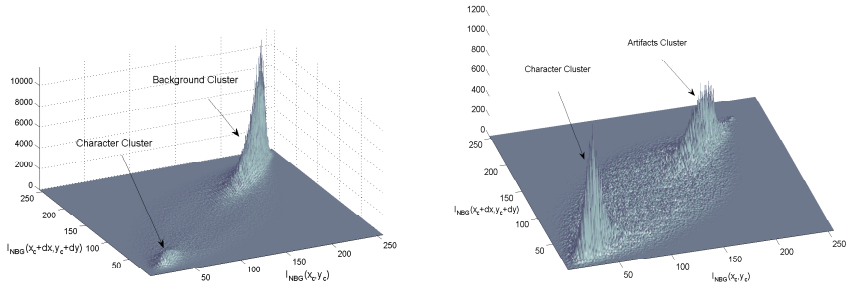


Figure 5: Creating the Local Co-occurrence points for $Q = 3$.

each patch produces a set of $(Q^2 - 1)$ 2D points denoted by $\mathbf{I}_W(t_i)$, where $t_i = 1, \ldots, (M - Q + 1)(N - Q + 1)$ is an index that runs through all possible image patches. Repeating the procedure for all possible $Q \times Q$ patches of the image yields the Local Co-occurrence Mapping (LCM), i.e. the new image representation $\mathbf{I}_W(k)$, where k represents the 2D-point index. The new image representation is of size $2 \times (M - Q + 1)(N - Q + 1)(Q^2 - 1)$. Calculating the 2D histogram of the 2D points $\mathbf{I}_W(k)$, we acquire the Symmetrical Frequency Map (SFM), as proposed by Makridis and Papamarkos [16]. In Fig. 6 (a), a typical SFM histogram is depicted.

One can observe the basic properties of this histogram. First of all, the SFM plot is symmetric over the main diagonal, because in two overlapping patches for i.e. $Q = 3$, one can get the symmetric points $[I_{NBG}(x, y)\ I_{NBG}(x + 1, y + 1)]^T$ $[I_{NBG}(x + 1, y + 1)\ I_{NBG}(x, y)]^T$ and are counted twice. The most important property is that there are two main concentrations of points: one where the center pixel takes higher intensity levels along with its neighboring pixels and one where the center pixels and its neighbors take lower intensity levels. The first point-cluster represents background pixels and the second point-cluster represents character pixels. The same trend appears in most printed or handwritten document images in our experiments using the DIBCO [32] image database. The only difference is there might be more visible clusters, due to paper stains or other artifacts of different intensity (see Fig. 7 (a)). However, these small clusters can be re-grouped in two main clusters: one of lower intensity denoting characters and one of higher intensity denoting background. This can be achieved during the clustering phase and will be discussed in a later section.

Observing the original SFM histograms in many document images, we made the following observations. Firstly, the character cluster is usually shorter and smaller compared to the background cluster, since characters constitute only a small part of the image, compared to background pixels. This will hinder the task of any clustering attempt to estimate the character cluster, since the



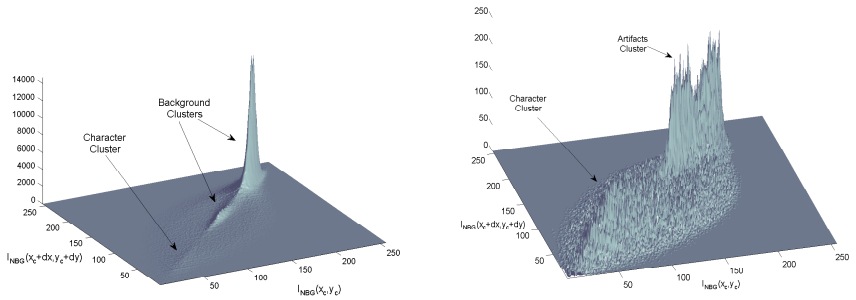
(a) Two visible clusters (b) Improved SFM

Figure 6: A typical SFM histogram from document images. Two prominent clusters are visible: characters and background (a). After removing the background pixels, the SFM now contains two strong clusters: characters and artifacts (b).

background cluster dominates the SFM histogram. In addition, this mapping is usually following the background removal stage, which implies that many pixel values will be set to 255 by the background removal process. This will cause a huge concentration of points around the point $(255, 255)$, which will make the character cluster barely visible and thus really difficult to be identified by a clustering algorithm.

The main proposal here is to remove all 2D-points whose central pixel value equals to 255 from $\mathbf{I}_W(t_i)$. These points have already been classified as background and therefore should not be part of the binarization process. After removing these points, the SFM histograms change considerably. The character clusters are more visible compared to the background cluster. In addition, the actual task that is required to solve here has also changed. After removing the pixels that have been classified as background, this image binarization step aims at discriminating between the character and the misclassified background or artifact pixels. In Fig. 6 (b), 7 (b), the improvement in the two previous SFM histograms is depicted. The new SFM histograms in either case contain two prominent clusters : the character and the artifacts cluster. The character cluster is now much stronger, compared to previous SFM histograms. After removing the background pixels, the proportion of character and artifact pixels is now comparable. This will improve clustering performance, since the character cluster is more clearly separable than previously.

Another improvement in the LCM framework is to remove 2D points far from the main diagonal. Ideally, character pixels should have similar intensity values with the central pixel, allowing for some slight deviation. Thus, pixels far from the main diagonal should be attributed to local noise and should be removed. We measure the *distance* d of each 2D point from the main diagonal and if it exceeds a threshold then it is rejected. The choice of threshold d should be carefully selected, as we will see in the experimental section. A narrow choice



(a) Three visible clusters (b) Improved SFM

Figure 7: A typical SFM histogram from the document image in Fig. 3. Three prominent clusters are visible: characters and stains-background (a). After removing the background pixels, the SFM now contains two strong clusters: characters and artifacts (b).

of d results into character thinning. A rather large choice of d may undermine performance, since it incorporates noise. Optimal values for d will be discussed in the experimental section.

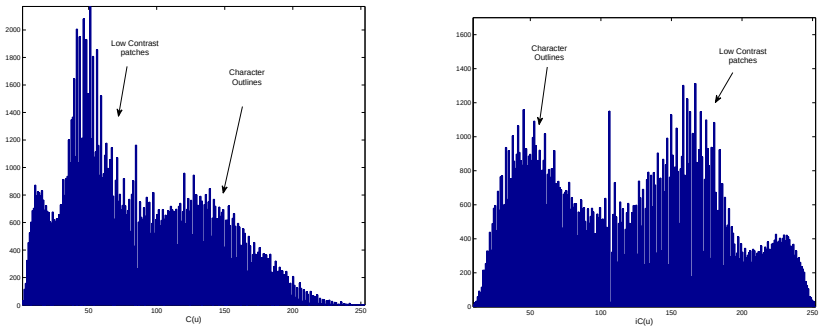
One can also use different neighbourhood patterns around each central pixel, such as cross or diamond neighbourhoods. This produced inferior results in our experiments. Also, the proposed 2D point representation resembles the 2D point representation proposed by Valizadeh and Kabir [21], with the difference being that their points contain structural contrast and local intensity and they look at neighbouring pixels at stroke-width distance.

4.2. LCM representation with local contrast information

So far, we have incorporated the first two of the three previously mentioned local character properties in the LCM representation. The third property emphasizes the existence of strong contrast in the $Q \times Q$ neighborhood, which denotes the existence of character outlines. To include this information in the LCM, we will simply calculate local contrast for each $Q \times Q$ image patch and its value will be incorporated in the LCM representation as a third dimension. The contrast of each patch $C(I_W(t_i))$ is calculated by the following equation:

$$C(I_W(t_i)) = \frac{\max(I_W(t_i)) - \min(I_W(t_i))}{\max(I_W(t_i)) + \min(I_W(t_i))} \tag{4}$$

The above definition of contrast is known as the Michelson contrast [37] and is recommended for patterns, where the amount of bright and dark pixels in the examined area is almost equal. The use of contrast to estimate text stroke width was also discussed in [20], using a similar definition of contrast. We also experimented with other textural measures that can identify character outlines (strong edges), including standard deviation and entropy, but the use of contrast seemed to be more stable in our experiments. The value of constrast is



(a) Histogram of $C(\cdot)$ (b) Histogram of $iC(\cdot)$

Figure 8: Histograms of $C(\cdot)$ and $iC(\cdot)$ for a document image after removing the white background pixels. The nonlinear mapping reverses and equalises the existence of the two desired clusters.

greater for patches containing character outlines (the desired patches), whereas is smaller for background patches. To move the desired cluster towards small values, in a similar manner to the previous 2D LCM representation and in order to suppress its range values, we propose the following nonlinear mapping to the original $C(\cdot)$ values.

$$iC(u) = 255(1 - \tanh(2C(u))) \tag{5}$$

The nonlinear function $\tanh(\cdot)$ serves as a method of increasing separation between the two clusters: character outlines and low-contrast patches. In Fig. 8, we depict the original contrast histogram of a document image and the proposed mapping $iC(\cdot)$, which features improved range and the character outlines cluster mapped to lower values. The new $iC(\cdot)$ values are used to form the novel 3D LCM representation, as follows:

$$\mathbf{I}_W(k) = \begin{bmatrix} I_{NBG}(x_c^i, y_c^i) \\ I_{NBG}(x_c^i + dx, y_c^i + dy) \\ iC(I_{NBG}(x_c^i, y_c^i)) \end{bmatrix}, \qquad \forall \ -\lfloor Q/2 \rfloor \leq dx, dy \leq \lfloor Q/2 \rfloor \tag{6}$$

As one can observe, the local contrast information of each patch is added as another feature to the previous 2D LCM, creating a novel 3D LCM feature, aiming at enhancing character outline binarization.

4.3. Binarization via MoG clustering

Once the LCM representation has been established, image binarization can be achieved by performing clustering on the data points $\mathbf{I}_W(k)$. There exist numerous methods to perform clustering. In this work, we examine the application of Mixtures of Gaussians (MoG) modeling to address the clustering problem. *Mixtures of Gaussians* (MoG) is a weighted sum (mixture) of different

multidimensional Gaussians that can be used to model any arbitrary probability density function (pdf) that does not follow a particular known distribution.

$$p(\mathbf{x}) = \sum_{1}^{K} a_i \mathcal{N}(\mathbf{m}_i, \Sigma_i) \tag{7}$$

where $\mathbf{x}$ is a random vector that is observed from the data, a_i are the mixing coefficients, $\mathbf{m}_i$ is the mean vector and Σ_i is the covariance matrix of the i-th multivariate Gaussian $\mathcal{N}(\mathbf{m}_i, \Sigma_i)$. In the special case that the arbitrary data distribution features relatively disjoint clusters of data, MoG can be employed to perform clustering by fitting each individual Gaussian of the mixture to each data cluster. The essentials of general multidimensional MoG were established in [38, 39], where the estimation of the MoG's parameters are performed using the *Expectation-Maximization* (EM) algorithm. The MoG estimation is sensitive to the initialisation of its parameters. To accelarate MoG training, one can initialize the EM using the result of a simple clustering algorithm, including the K-Means, the Fuzzy C-Means and the Harmonic K-Means algorithm.

We will employ the EM algorithm, as described in [39], to perform clustering of the LCM data $\mathbf{I}_w(k)$. Our clustering problem is very constrained and these constraints should be used in the initialisation of the EM algorithm. First of all, we are looking at identifying 2 clusters (characters-artifacts), thus $K = 2$. This implies that the mixing coefficients should be initialised by $a_i = 0.5$. The initialization of the means $\mathbf{m}_i$ is also very important. As previously observed, the desired clusters are usually centered on the main diagonal. In addition, the character cluster should be centered near the beginning of the main diagonal (dark intensities) whereas the artifacts cluster should be placed in the opposite part of the main diagonal (lighter intensities). Consequently, the character mean can be initialised as e.g. $\mathbf{m}_1 = [20\ 20\ 20]^T$, whereas the artifacts mean can be initialised as e.g. $\mathbf{m}_2 = [230\ 230\ 230]^T$. Finally, to simplify calculations, we can assume that the Gaussians' covariance matrices are diagonal and use random initialization for their variances. In the previous section, we mentioned the case of discovering more than 2 concentrations of LCM points, due to significant paper stains, or different text color. In this case, we can initialise the EM using 3 or more Gaussians (as necessary) and equidistant initialisation on the main diagonal. After the convergence of the EM, we can merge the new middle clusters with either the text or the background cluster, depending on the distance between their means.

Once the EM algorithm has converged, we have to use the LCM points that correspond to the character cluster (the cluster with the lowest mean vector) to form the binarised image $I_{BN}(x, y)$. The classification rule is straightforward: "*if any LCM data point in each $Q \times Q$ neighborhood is classified to the character cluster, then the corresponding central pixel (x_c^i, y_c^i) is set to black, i.e. $I_{BN}(x_c^i, y_c^i) = 0$. The remaining pixels are set to white i.e. $I_{BN}(x^i, y^i) = 255$.*"

The proposed algorithm is summarized, as follows:

Document Image Binarization Algorithm

1. Use the proposed Background Removal algorithm to create the image $I_{NBG}(x, y)$.
2. For every $Q \times Q$ neighborhood in the image, create the 3D LCM representation $\mathbf{I}_w(k)$ using (6). Neighborhoods whose central pixel has been classified as background are not used in the LCM representation.
3. Identify two clusters on the LCM representation using the MoG-EM algorithm and the initialisation discussed earlier.
4. Initialise the $M \times N$ matrix $I_{BN}(x, y) = 255$.
5. If any pixel in each $Q \times Q$ neighborhood is classified to the character cluster, then the corresponding central pixel (x_c^i, y_c^i) is set to zero, i.e. $I_{BN}(x_c^i, y_c^i) = 0$.

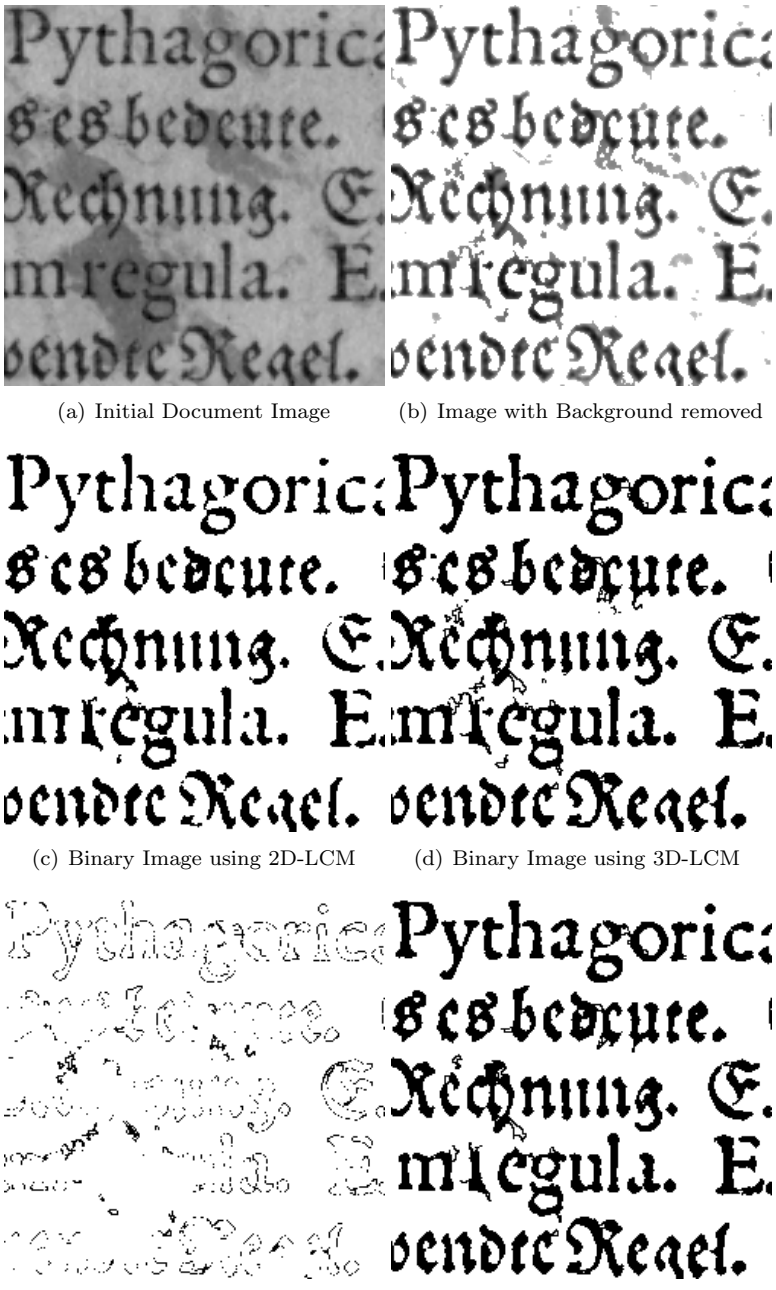
5. Post-Processing

The final stage aims at removing artifacts from the previous binarization stage. Isolated blobs or small misclassified noisy items can be removed using a mathematical morphology step. We use MATLAB's `bwlabel` command to identify connected objects with 8-connectivity in the binary output of the 3D LCM algorithm. The command returns an annotated image containing all the different connected components with 8-connectivity that exist in the image. If some of these components are small in size, they should be noisy artifacts, as described earlier. Therefore, we remove all those connected components that contain less than 20 pixels. Of course, this threshold relates to the image's resolution and has to be adapted accordingly. This choice seemed to work well for the DIBCO image databases that were our main experimental ground. In Fig. 9 (f), we can see the result of post-processing on the previous 3D LCM binary image (d). Many of the previous binarization errors have been removed. Of course, there are several more complicated post-processing methods, one can use to improve the binarization output, such as those proposed in [6] and [4], however, we wanted to keep the computational complexity of our algorithm as low as possible.

6. Evaluation

In this section, we evaluate the performance of the proposed image document binarization approach. In the first section of the evaluation process, we investigate several properties of the proposed binarization algorithm. In the second section, we compare its performance with other well-established approaches in the field and several evaluation datasets of historical machine-printed and handwritten document images. For the training and evaluation of MoGs, we used the functions `gaussmix` and `gaussmixp` respectively available freely from Voicebox[2]. All experiments were conducted on an Intel Core i5-460M (2.53 GHz) PC

[2] `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`



(a) Initial Document Image (b) Image with Background removed

(c) Binary Image using 2D-LCM (d) Binary Image using 3D-LCM

(e) Difference between 2D and 3D LCM (f) Binary 3D LCM after post-processing

Figure 9: A document image (a) through the three steps of the proposed binarization approach. Background is removed from the image (b) and then it is binarized through the LCM-MoG approach using either a 2D LCM (c) or a 3D LCM (d). The difference between 2D and 3D-LCM (e) clearly demonstrates that the added contrast features highlights the characters' outline. Post-processing of 3D-LCM removes several artifacts (f).

with 6GB DDR3 SDRAM running Windows 7 Professional 64-bit and MATLAB R2013a. Our MATLAB implementations were not optimised in terms of execution speed.

The document images used in our study, were publicly available by the document image binarization community in previous open competitions, including *DIBCO2009, DIBCO2011, H-DIBCO2010, H-DIBCO2012* and *DIBCO2013* [32, 33, 34, 35, 36]. In these competitions, datasets including both machine-printed (P) and hand-written historical (H) document images were publicly provided, along with their hand-annotated Ground Truth binarization result. All these images have very challenging noise and degradations due to the document's wear.

6.1. Evaluation Metrics

There are many metrics available for the evaluation of image binarization algorithms [32, 33, 34, 35]. Let $I_{BN}(x, y)$ be the binary image result of a binarization algorithm and $I_{GT}(x, y)$ be the hand-annotated ground truth binary result. Some commonly used evaluation measurements for the evaluation of image binarization algorithms, that will be used in our study, are the following:

- Mean-Square Error (MSE)

$$MSE = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} (I_{BN}(x, y) - I_{GT}(x, y))^2 \tag{8}$$

- Picture Signal-to-Noise Ratio (PSNR)

$$PSNR(dB) = 10 \log \frac{255^2}{MSE} \tag{9}$$

One can also count the number of *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) and *False Negative* (FN) matches between the two binary images and calculate the following metrics.

- Recall

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- Precision

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

- F-Measure (FM)

$$FM = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{12}$$

- Negative Rate Measurement (NRM)

$$NRfn = \frac{FN}{FN + TP} \tag{13}$$

$$NRfp = \frac{FP}{FP + TN} \tag{14}$$

$$NRM = \frac{NRfn + NRfp}{2} \tag{15}$$

6.2. Evaluation of the proposed method's performance

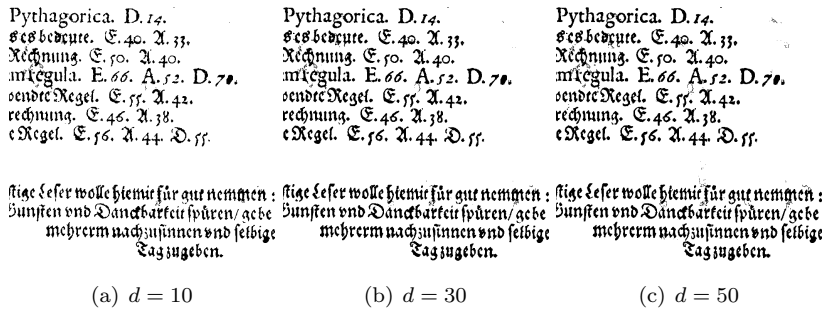
In this section, we discuss the effect of contrast information in the algorithm's performance, as well as the effect of the parameters d and q in the binarization performance.

6.2.1. Effect of local contrast feature

Firstly, we demonstrate the positive effect of incorporating the contrast information in the LCM implementation. In Fig. 9, we demonstrate the algorithm's performance on a document image. In Fig. 9(c), we depict the output of the 2D LCM algorithm and in Fig. 9(d), we depict the output of the 3D LCM algorithm incorporating contrast. The difference between the two outputs is depicted in 9(e). It is evident that the inclusion of local contrast information has enhanced the presence of character outlines, which was missing from 2D LCM. To perform objective evaluation of the effect of local contrast, we measured the aforemention binarization performance metrics for the two cases. The results are reported in Table 1. It can be observed that contrast information improves all performance indices compared to 2D LCM. 3D LCM improves Recall but reduces precision; however the FM measurement is improved. Thus, the inclusion of contrast information improves the performance of the proposed algorithm both subjectively and objectively.

Table 1: Effect of contrast information in the algorithm's performance.

	Without Contrast	With Contrast
PSNR (dB)	15.29	**15.99**
MSE	0.0295	**0.025**
Recall	0.8331	0.9178
Precision	0.9466	0.9016
FM	0.8862	**0.9096**
NRM	0.0872	**0.0491**



(a) $d = 10$ (b) $d = 30$ (c) $d = 50$

Figure 10: Effect of point rejection from the main diagonal for various values of d.

6.2.2. Effect of threshold d

Next, we evaluate the effect of rejecting LCM points that are far from the main diagonal. Points that are close to the main diagonal should belong to character pixels. Points far from the main diagonal may belong either to character outlines or background noise. Rejecting points close to the main diagonal usually results into character thinning. Fig. 10 shows the algorithm outputs for various values of threshold d. The difference between character outlines can be seen in Fig. 10 (a) and (b), whereas in (c), we can see the inclusion of background noise and oversize characters.

In an attempt to find an optimal value for d via experimentation, we evaluated the algorithm's average PSNR and FM for all the available P and H-DIBCO datasets for various values of d. Fig. 11 (a) and (b) contains the average PSNR and FM for the printed (P) images and Fig. 11 (c) and (d) the same measurements for the handwritten (H) images. It appears that in most cases for low and great values of d, the binarization result is much inferior. The characters in this case appear very thin or too much noise has been incorporated in the binarization result or the character appear much thicker. Unfortunately, we can not automate the optimal selection of d and thus has to be manually selected. Judging from the results, we can pick a value of $d = 40$, which seems to perform better in most printed and handwritten datasets. This value is not adapted any further in our experiments.

6.2.3. Effect of background removal threshold q

In this section, we evaluate the effect of background removal in the binarization result. This is controlled via the parameter q, which defines the threshold after which some pixels are considered text or background. Lower values of q denote stronger background removal, whereas higher values leave more background pixels classified as text. In a similar effort to the previous section, we evaluated the algorithm's average PSNR and FM for all the available P and H-DIBCO datasets for various values of q. Fig. 12 (a) and (b) contains the average PSNR and FM for the printed (P) images and Fig. 12 (c) and (d) the same measurements for the handwritten (H) images. Here, we can see some difference

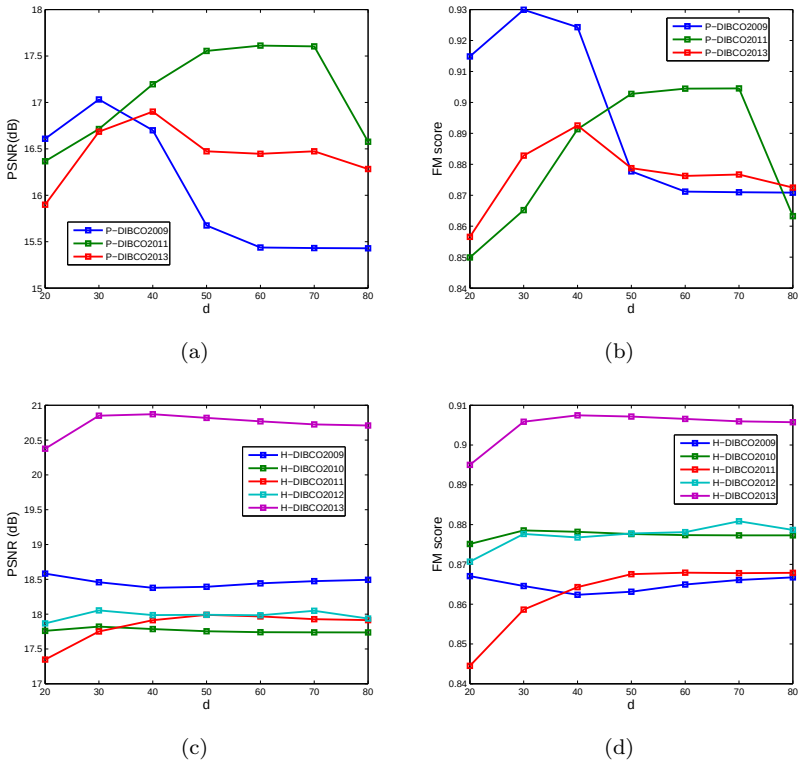


Figure 11: Effect of parameter d on average PSNR and FM measurements for various printed (P) and handwritten (H) datasets.

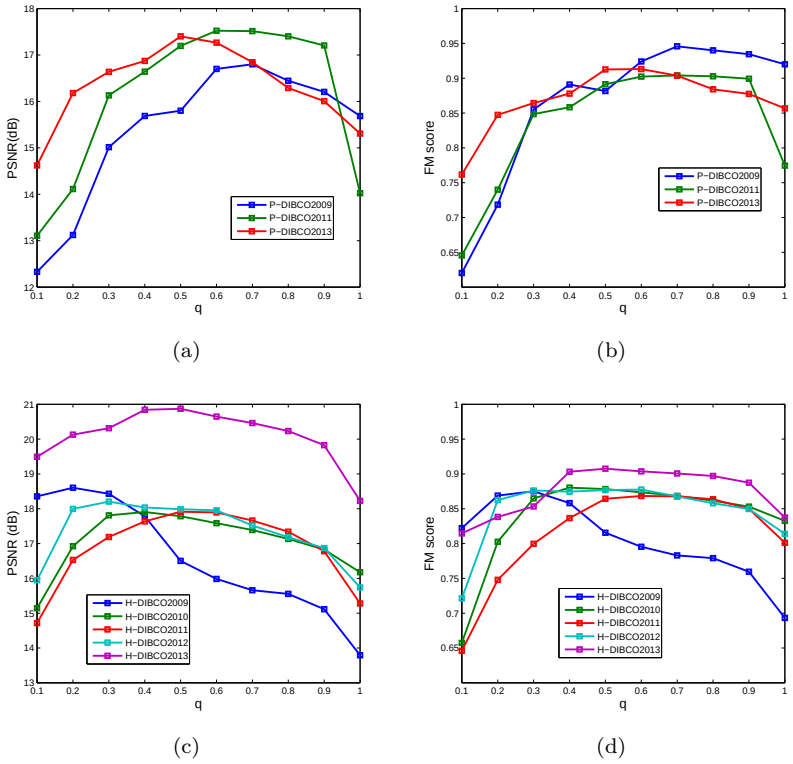


Figure 12: Effect of parameter q on average PSNR and FM measurements for various printed (P) and handwritten (H) datasets.

between hand-written and printed documents. Hand-written documents seem to give better performance at lower values of q compared to the printed ones. Again, automation of the optimal selection of q seems not possible at this stage and thus has to be hand-picked. Hence, we use a value of $q = 0.6$ for the printed documents and a value of $q = 0.4$ for the handwritten ones.

6.3. Comparison with other binarization methods

In this section, we compare the proposed LCM binazation method with other common binarization methods. In our benchmarking exercise, we use Otsu's thresholding method [40] and the local binarization techniques of Sauvola (Sau) [11] and Bernsen (Bern) [12]. For the Sauvola method, we use a value of $k = 0.4$ and a window size of 21×21 to calculate the local statistics. We also use the Adaptive Logical Level Technique (ALLT) and the Improvement of Integrated Function Algorithm (IIFA), as proposed by Badekas and Papamarkos [13]. We use the GPP binarization method, as proposed by Gatos et al. [18], the

binarization method of Howe[2][3] with automated threshold selection. Finally, we include Ramirez-Ortegon et al method (Ramir.)[4, 24, 25][4] and Su et al method (Su) [20][5]. For the proposed LCM method, we use a value of $d = 40$, a value of $q = 0.6$ for the machine-printed documents and a value of $q = 0.4$ for the handwritten documents. This implies that stronger background removal was essential for the handwritten documents. It was not our intention to develop the best performing binarization algorithm, however, we can see that the proposed algorithm performs favorably with the tested approaches and those scores reported at image binarization competitions at a reasonable running time. We employed the images from the available DIBCO datasets in our study. The objective evaluation metrics of PSNR, MSE, Recall, Precision, FM and NRM were calculated from all the results.

6.3.1. Algorithm's speed

Table 2: Average algorithm's running time for all datasets

Dataset	*sec* per image	*msec* per pixel
P-DIBCO2009	13.74	0.0321
P-DIBCO2011	13.96	0.0232
P-DIBCO2013	33.32	0.0294
H-DIBCO2009.	23.34	0.0299
H-DIBCO2010	14.08	0.0209
H-DIBCO2011	10.9	0.019
H-DIBCO2012	33.76	0.025
H-DIBCO2013	43.27	0.0176
Average	-	**0.0246**

Firstly, we estimate the algorithm's running time by using MATLAB's commands `tic-toc` on the aforementioned PC system. We estimate the average running time in *sec* per image for each dataset. Since the algorithm's running time depends also on the image size, we calculated a normalised average running time in *msec* per pixel, in order to get a clearer performance overview. The results are summarised in Table 2. We can understand that the algorithm's running time is on average 0.0246 msec per pixel. This implies that for a 640×480 image the algorithm requires an average 7.7 sec on an average PC. It was only possible to compare the proposed algorithm's running time with Howe's approach, since they were both implemented in MATLAB, whereas the other methods were implemented in different platforms. Howe's algorithm is the best performing algorithm in our later experiments, therefore it is sensible

[3]Code kindly provided at `http://www.cs.smith.edu/~nhowe/research/code/`
[4]Code kindly provided at `https://sites.google.com/site/martehomepage/`
[5]Code kindly provided at `https://sites.google.com/site/flydreamersu/`

to compare with the best. We also normalised the two algorithms' running time to the running time of Sauvola's algorithm (implemented in MATLAB as well). The results are shown in Table 3. The LCM algorithm is on average 63.83 times slower than Sauvola's algorithm but is much faster than Howe's approach, which is 254.77 times slower than Sauvola. This implies that LCM is about 4 times faster than Howe's approach.

Table 3: LCM and Howe's algorithm running time normalised to Sauvola's algorithm.

Dataset	LCM	Howe
P-DIBCO2009	87.36	233.93
P-DIBCO2011	59.43	252.47
P-DIBCO2013	83.43	310.37
H-DIBCO2009	79.47	265.76
H-DIBCO2010	50.79	243.16
H-DIBCO2011	48.96	259.02
H-DIBCO2012	57.1	231.97
H-DIBCO2013	44.11	241.5
Average	63.83	254.77

6.3.2. Objective Evaluation

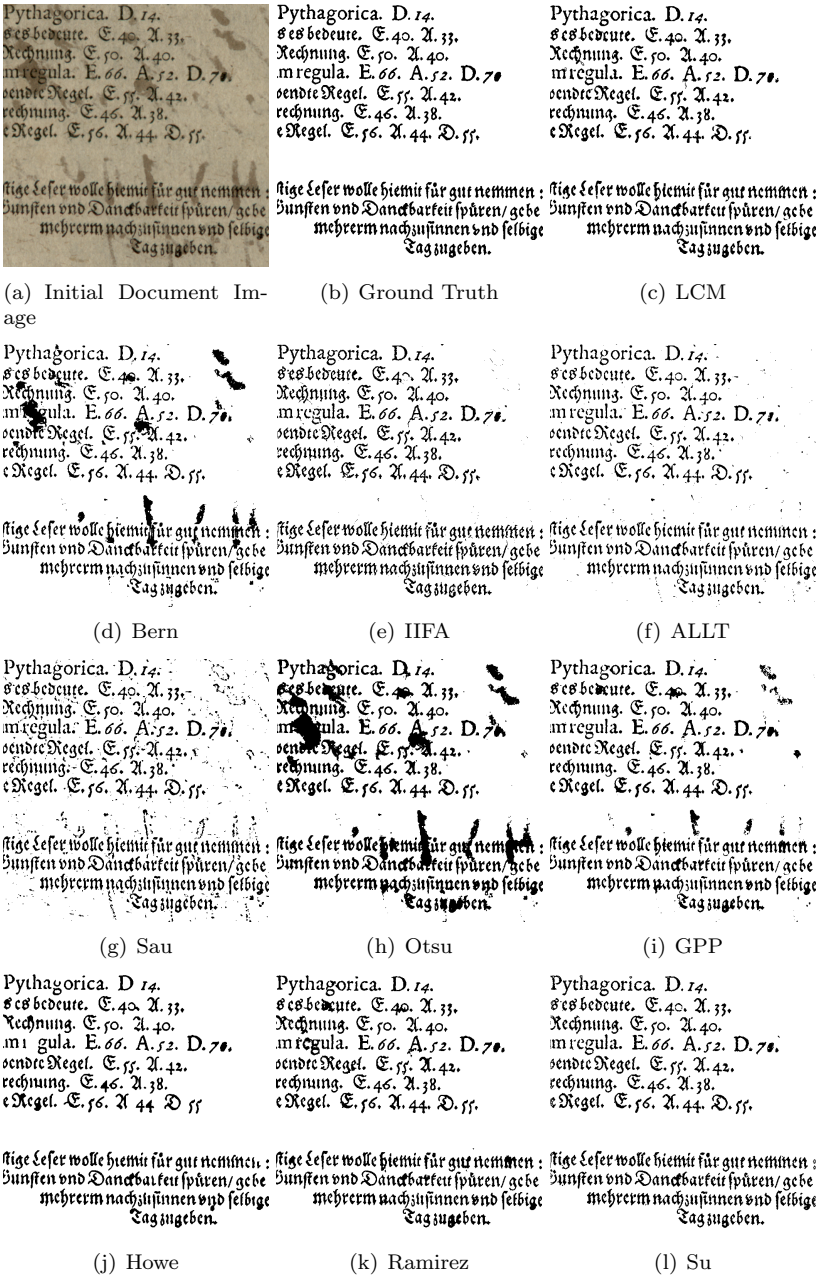
In Table 4, we present the results of binarization of machine-printed historical document images of DIBCO2009 DIBCO2011 and DIBCO2013. Some typical document images and the respective binarization results are depicted in Figs. 13, 14, 15. For the printed document images, we used a $q = 0.6$ for the LCM approach. Howe's method seems to give the best performance in P-DIBCO 2009 both in terms of PSNR and FM. For the P-DIBCO2011, Ramirez et al seems to give the best performance both in terms of PSNR and FM with Su et al being the winner at P-DIBCO2013. The LCM approach seems to be third best in P-DIBCO 2011 and 2013 in terms of PSNR and second best in terms of FM. This is not the case for the DIBCO2009 dataset where the LCM is fifth best in terms of PSNR and FM, since it contains images with multiple colour characters. The LCM approach is not calibrated to work for multi-colour documents, as it was described before. However, it can be easily adapted to handle multi-colour document images, simply by increasing the number of desired clusters in the MoG clustering module. The lower intensity centered clusted can then be merged to form the text cluster. This justifies the lower performance of the algorithm in some of the images, which undermines the average scores. The result images of all datasets can be downloaded from the following url[6].

In Table 5, we depict the results of binarization of handwritten historical

[6] `http://utopia.duth.gr/~nmitiano/machine.rar`

Table 4: Average results for the machine-printed historical document images of DIBCO2009, DIBCO2011 and DIBCO2013.

	Average machine-printed DIBCO2009									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	16.73	15.1868	13.4289	13.6385	13.5102	16.1885	16.9338	**18.7866**	18.0311	17.8061
MSE	0.0216	0.0329	0.0551	0.0506	0.0497	0.0268	0.0209	0.0147	0.0177	0.0172
Recall	0.9202	0.8371	0.7276	0.7258	0.8161	0.9440	0.9004	0.9504	0.9387	0.9220
Precision	0.9295	0.9175	0.9172	0.9263	0.8499	0.8737	0.9539	0.9434	0.9348	0.9591
FM	0.9243	0.8723	0.7976	0.8036	0.8290	0.9044	0.9261	**0.9467**	0.9366	0.9393
NRM	0.0460	0.0870	0.1418	0.1418	0.1042	0.0386	0.0533	**0.0295**	0.0360	0.0423
	Average machine-printed DIBCO2011									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	17.7245	14.3946	12.8877	13.6065	12.9166	14.8470	15.1522	17.7619	**18.0838**	15.5592
MSE	0.0203	0.0401	0.0553	0.0442	0.0526	0.0403	0.0316	0.0240	0.0190	0.0437
Recall	0.8806	0.8448	0.7140	0.7637	0.8279	0.9305	0.8400	0.9300	0.9138	0.8835
Precision	0.9363	0.8733	0.8915	0.9096	0.8131	0.8355	0.8822	0.8855	0.9226	0.8357
FM	0.9068	0.8556	0.7819	0.8284	0.8177	0.8708	0.8486	0.9007	**0.9162**	0.8160
NRM	0.0634	0.0881	0.1508	0.1247	0.1024	0.0723	0.0868	**0.0438**	0.0479	0.0736
	Average machine-printed DIBCO2013									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	17.59554	14.4270	13.9487	14.5029	14.7016	14.7271	16.3588	18.4252	17.0493	**18.8973**
MSE	0.0197	0.0604	0.0481	0.0398	0.0377	0.0595	0.0263	0.0249	0.0266	0.0189
Recall	0.8979	0.8742	0.6988	0.7107	0.8353	0.9200	0.8445	0.9368	0.9107	0.9404
Precision	0.9277	0.7875	0.8896	0.9174	0.8396	0.7758	0.9244	0.8966	0.8848	0.9183
FM	0.9107	0.8063	0.7681	0.7745	0.8284	0.8186	0.8778	0.9077	0.8902	**0.9243**
NRM	0.0566	0.0895	0.1571	0.1494	0.0927	0.0694	0.0825	0.0419	0.0538	**0.0368**



(a) Initial Document Image (b) Ground Truth (c) LCM

(d) Bern (e) IIFA (f) ALLT

(g) Sau (h) Otsu (i) GPP

(j) Howe (k) Ramirez (l) Su

Figure 13: Binarization results of a machine-printed document image from the DIBCO2011 dataset.

(a) Initial Document Image (b) Ground Truth

(c) LCM (d) Bern

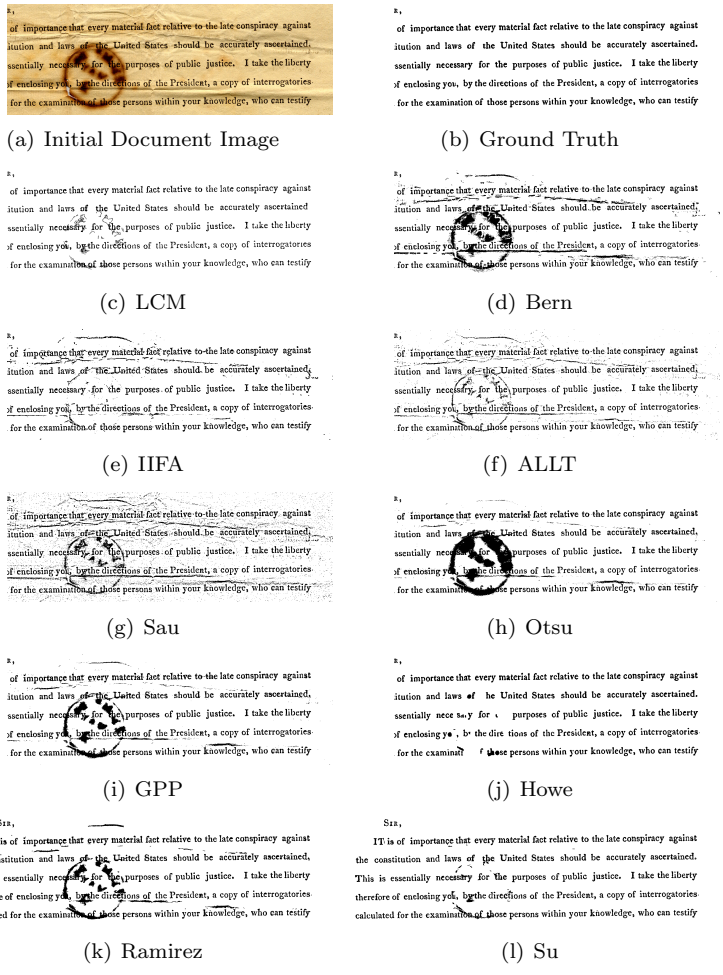
(e) IIFA (f) ALLT

(g) Sau (h) Otsu

(i) GPP (j) Howe

(k) Ramirez (l) Su

Figure 14: Binarization results of a machine-printed document image from the DIBCO2011 dataset.



(a) Initial Document Image (b) Ground Truth

(c) LCM (d) Bern

(e) IIFA (f) ALLT

(g) Sau (h) Otsu

(i) GPP (j) Howe

(k) Ramirez (l) Su

Figure 15: Binarization results of a machine-printed document image from the DIBCO2013 dataset.

document images of DIBCO2009, DIBCO2010, DIBCO2011, DIBCO2012 and DIBCO2013. Here, we used a value of $q = 0.4$ for the document background removal, implying that there was more need for degradation removal in these document images. Howe's method seems to be the winner in all datasets in terms of PSNR. In terms of FM, Howe's method is the winner in H-DIBCO2009, H-DIBCO2010, H-DIBCO2012 with LCM being the winner in H-DIBCO 2011 and H-DIBCO 2013. LCM is fourth in terms of PSNR in H-DIBCO2009, H-DIBCO2010, H-DIBCO2012, H-DIBCO2013 and second in H-DIBCO2011. In terms of FM, LCM ranks 4th in H-DIBCO2009, H-DIBCO2010 and 3rd in H-DIBCO2012. In Figs. 16, 17, typical examples of the images are shown. These images were heavily contaminated by ink from the opposite page. As it is evident, in this case the LCM approach performs well at removing these contamination artifacts, especially for document images with bleed-through contamination. The result images of all datasets can be downloaded from the following url[7].

In Table 6, the average score of all available printed and handwritten datasets is presented. Howe's method is the winner, while LCM ranks fourth both in terms of PSNR and FM.

6.3.3. Comparing with results reported in DIBCO competitions

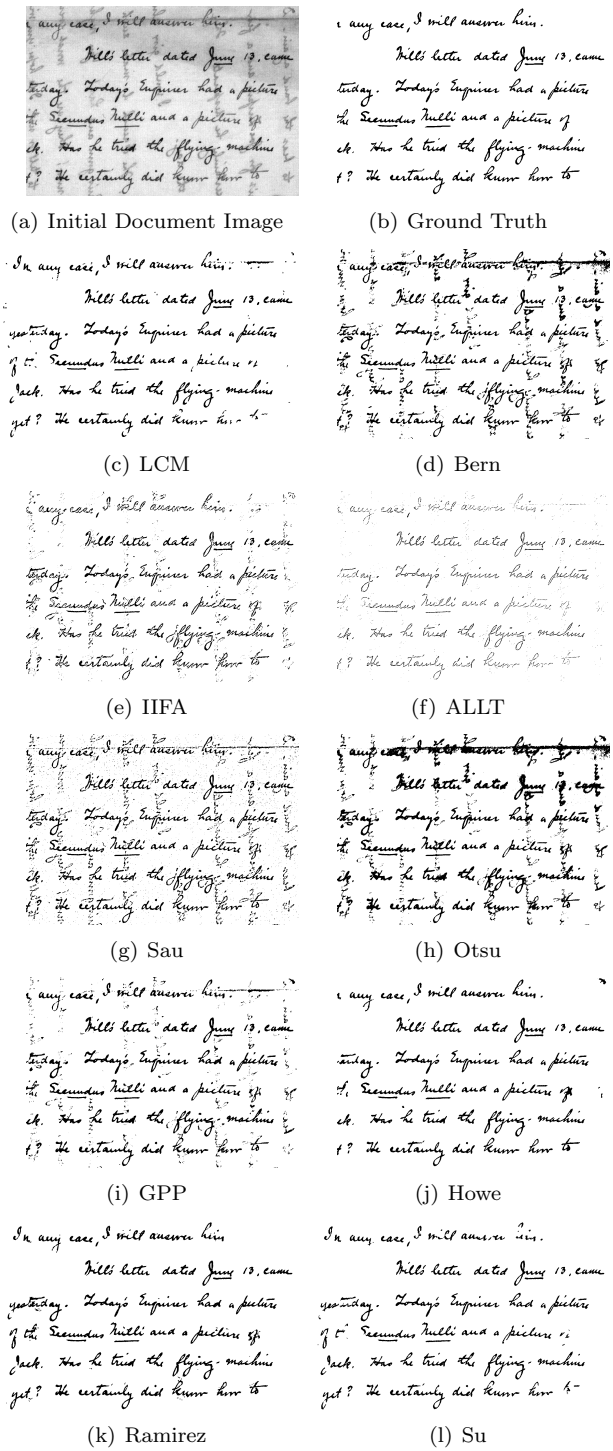
In this section, we compare LCM's scores with those reported in DIBCO competitions. More methods than the ones examined here have taken part in these competitions, therefore, it is important to know LCM's standing compared to a wider range of techniques. Comparing with the results reported in DIBCO2009 [32], the method would rank 3rd in terms of PSNR and 2nd in terms of FM for the combined printed and handwritten dataset. Comparing with the results, reported in H-DIBCO2010 [33], the method would rank 7th in terms of PSNR and 6th in FM. Comparing with the results, reported in DIBCO2011 [34], the method would rank 1st both in terms of PSNR and FM for the printed and 4th in terms of PSNR and 2nd for the FM for the handwritten dataset (no more measurements were provided in the paper). Looking at the H-DIBCO2012 results [35], the method would get the 13th position in terms of PSNR and 8th for the FM, but will be at the top faster methods at this performance on a slighter faster machine. For the H-DIBCO2013 results [36], the method would get the 5th position in terms of PSNR and FM for both handwritten and printed dataset. Finally, the LCM method was submitted to H-DIBCO 2014 [19], getting the 5th position in terms of PSNR and the 4th in terms of FM. These results are summarized in Table 7.

In summary, the method performs relatively well in terms of binarization, of course lacking in performance compared to state-of-the-art methods, such as Howe's method. Nevertheless, the method is not very complicated, compared to the best-performing ones described earlier. Thus, this lower complexity can be an advantage to use this method in an environment where computational

[7] `http://utopia.duth.gr/~nmitiano/handwritten.rar`

Table 5: Average results for the handwritten historical document images of DIBCO2009, DIBCO2010, DIBCO2011, DIBCO2012 and DIBCO2013.

	Average Handwritten DIBCO2009									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	19.5717	14.0212	17.3101	16.0695	16.7753	13.9286	17.7434	**22.6143**	20.1885	22.0044
MSE	0.0133	0.0783	0.0192	0.0254	0.0232	0.0907	0.0176	0.0064	0.0109	0.0072
Recall	0.8826	0.8989	0.8088	0.6378	0.8516	0.9450	0.8508	0.9579	0.9332	0.9314
Precision	0.8879	0.5815	0.8324	0.8765	0.7763	0.5806	0.8465	0.9357	0.8870	0.9427
FM	0.8836	0.6531	0.7990	0.6922	0.7859	0.6594	0.8365	**0.9467**	0.9092	0.9369
NRM	0.0621	0.0887	0.0999	0.1839	0.0818	0.0741	0.0792	**0.0231**	0.0374	0.0361
	Average Handwritten DIBCO2010									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	18.0848	16.8626	16.8331	14.5122	16.0394	17.4412	15.9639	**21.0505**	19.2781	20.1279
MSE	0.0191	0.0216	0.0271	0.0390	0.0280	0.0186	0.0288	0.0084	0.0125	0.0104
Recall	0.8280	0.7586	0.6992	0.4870	0.7398	0.8184	0.6575	0.9224	0.8996	0.8816
Precision	0.9307	0.9211	0.9263	0.9525	0.8833	0.9016	0.9434	0.9528	0.9148	0.9645
FM	0.8758	0.8196	0.7543	0.6051	0.7874	0.8527	0.7494	**0.9369**	0.9059	0.9207
NRM	0.0890	0.1238	0.1529	0.2580	0.1350	0.0944	0.1733	**0.0405**	0.0533	0.0605
	Average Handwritten DIBCO2011									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	18.3152	14.9752	16.2537	14.9554	14.5512	15.0349	16.7432	**19.3230**	18.0845	15.5592
MSE	0.0157	0.0451	0.0302	0.0393	0.0439	0.0539	0.0238	0.0233	0.0224	0.0437
Recall	0.9173	0.8101	0.7672	0.6158	0.8400	0.8461	0.8091	0.8753	0.9139	0.8835
Precision	0.8673	0.7606	0.8804	0.8690	0.7154	0.7471	0.8719	0.9101	0.8659	0.8357
FM	**0.8897**	0.7658	0.8098	0.7133	0.7556	0.7671	0.8318	0.8721	0.8838	0.8160
NRM	**0.0467**	0.1126	0.1232	0.1969	0.0975	0.1013	0.1011	0.0704	0.0522	0.0736
	Average Handwritten DIBCO2012									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	18.7265	15.6106	16.4953	14.9111	16.2999	15.5702	17.0446	**22.2778**	20.2841	19.6261
MSE	0.0142	0.0564	0.0284	0.0390	0.0267	0.0638	0.0219	0.0064	0.0100	0.0159
Recall	0.9077	0.8383	0.6745	0.4766	0.7777	0.8647	0.7481	0.9485	0.9218	0.8692
Precision	0.8922	0.7789	0.9235	0.9374	0.8535	0.7775	0.9399	0.9560	0.9314	0.9355
FM	0.8971	0.7666	0.7456	0.5956	0.8008	0.7748	0.8178	**0.9521**	0.9258	0.8887
NRM	0.0502	0.1051	0.1649	0.2623	0.1169	0.0970	0.1282	**0.0273**	0.0416	0.0692
	Average Handwritten DIBCO2013									
	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	21.5742	17.3243	17.8128	17.2365	16.3032	18.4383	18.7736	**23.8068**	21.5897	22.8477
MSE	0.0075	0.0222	0.0199	0.02	0.0285	0.0178	0.0160	0.0055	0.0073	0.0066
Recall	0.8948	0.7502	0.7725	0.6470	0.8030	0.7714	0.7744	0.8685	0.9123	0.8724
Precision	0.9492	0.8560	0.8694	0.9035	0.7629	0.8844	0.9012	0.9738	0.9289	0.9833
FM	**0.9209**	0.7456	0.7901	0.7281	0.7413	0.7769	0.7927	0.8945	0.9184	0.9207
NRM	0.0540	0.1305	0.1183	0.1785	0.1087	0.1183	0.1159	0.0664	**0.0457**	0.0642



(a) Initial Document Image (b) Ground Truth

(c) LCM (d) Bern

(e) IIFA (f) ALLT

(g) Sau (h) Otsu

(i) GPP (j) Howe

(k) Ramirez (l) Su

Figure 16: Binarization results of a handwritten document image from the DIBCO2012 dataset.

Table 6: Average results for all available datasets.

	LCM	Bern	IIFA	ALLT	Sau	Otsu	GPP	Howe	Ramir.	Su
PSNR (dB)	18.0117	14.9349	15.3583	14.6245	14.8921	15.2352	16.6014	**19.8331**	18.5061	18.9581
FM	0.8986	0.7823	0.7800	0.7126	0.7921	0.7970	0.8457	**0.9214**	0.9072	0.9046
NRM	0.0614	0.0993	0.1425	0.1892	0.1041	0.0771	0.0984	**0.0399**	0.0470	0.0542

power is constrained, without losing much in quality.

It appears that the proposed algorithm performance depends on the choice of the parameter q, which defines the amount of background that needs to be removed from the initial image. Removing much of the background may remove character information, whereas removing less background may leave stains that may not be sorted later by MoG clustering. The next task will be to automate this parameter choice in order to optimize the performance of the algorithm. Our previous study can give rough guidelines for the optimal value of q. Nonetheless, we have observed that every image may benefit from a different value of q during binarization. Thus, it would be very important to automate the choice of this threshold.

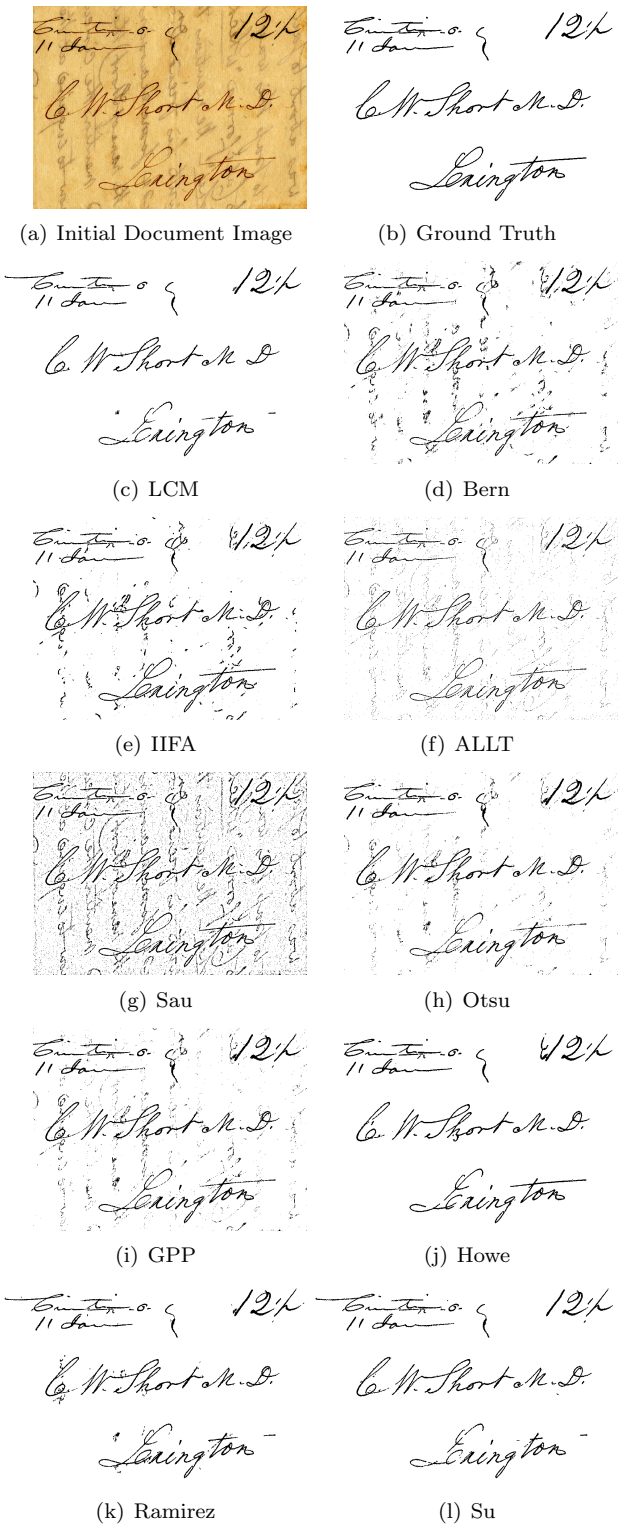
Table 7: LCM's ranking based on results reported in DIBCO competitions.

Dataset	PSNR	FM
DIBCO2009	3rd	2nd
P-DIBCO2011	1st	1st
H-DIBCO2010	7th	6th
H-DIBCO2011	4th	2nd
H-DIBCO2012	13th	8th
DIBCO2013	5th	5th
H-DIBCO2014	5th	4th

7. Conclusions

In this paper, the authors propose a novel document image binarization system that can be applied on both machine-printed and handwritten document images. The system consists of three stages. During the background removal stage, an estimate of the background image is calculated via adaptive median filtering. The background is removed by statistical thresholding of the differences between the estimated background and the document image. In the next stage, Local Co-occurance map (LCM) is calculated as described earlier. This representation aims at grouping together pixels of similar intensity value and similar contrast, thus creating two dominant clusters: character and remaining background. Clustering is performed using a Mixture-of-Gaussian (MoG) model of two Gaussians. In the last stage, some isolated binary artifacts are removed by morphological 8-connected object segmentation.

The proposed approach is robust to severe degradation of the document images. The inclusion of contrast seems to improve the inclusion of character outlines in the binarization results. The method performs quite well in our experiments and DIBCO benchmarks. Although, it is not the best performing method, it is a low-complexity good performing method that can be used in environments, where computational power use is important. Nonetheless, the



(a) Initial Document Image (b) Ground Truth

(c) LCM (d) Bern

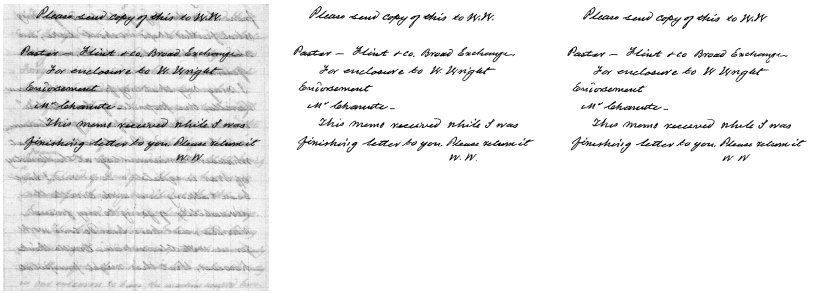
(e) IIFA (f) ALLT

(g) Sau (h) Otsu

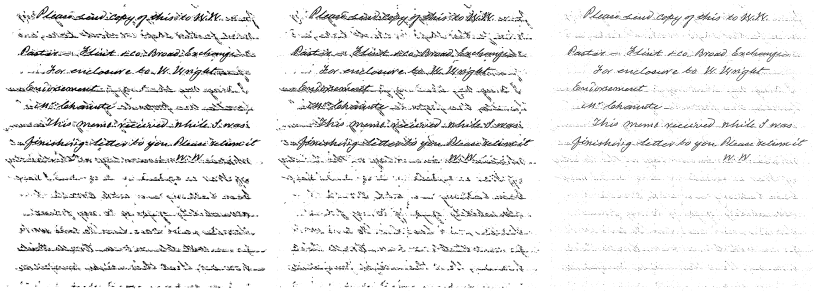
(i) GPP (j) Howe

(k) Ramirez (l) Su

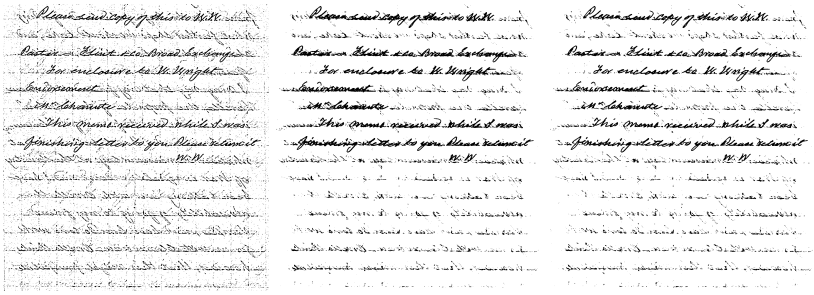
Figure 17: Binarization results of a handwritten document image from the DIBCO2011 dataset.



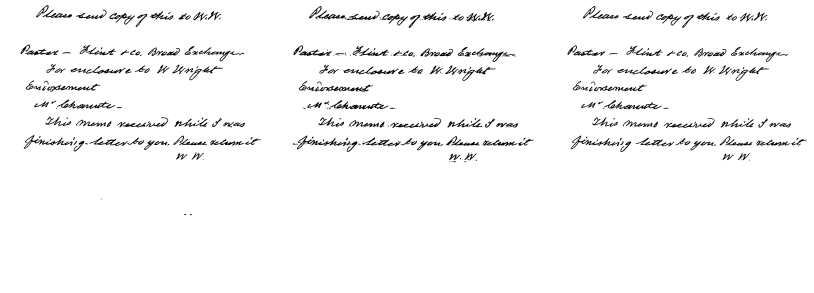
(a) Initial Document Image (b) Ground Truth (c) LCM



(d) Bern (e) IIFA (f) ALLT



(g) Sau (h) Otsu (i) GPP



(j) Howe (k) Ramirez (l) Su

Figure 18: Binarization results of a handwritten document image from the DIBCO2013 dataset.

method is very sensitive to the amount of background removal performed in the first stage, which is controlled by the parameter q. In this study, we have used a value of $q = 0.6$ for the printed images and a value of $q = 0.4$ for handwritten images, that seemed to work well in our experiments.

The authors would also like to extend the method to work for multi-colour documents. Although it is trivial to extend the number of clusters in the MoG model, it would be preferable if the system could automatically identify the number of colours and configure the number of clusters accordingly. In addition, the authors would like to look into a more automated method to define the value of q in the background removal stage and the cluster size in the post-processing stage. Another extension can be to change the Gaussian distribution assumption for the background and character cluster for skewed distributions, including the log-normal distribution, as observed by Ramirez-Ortegon et al [24].

References

[1] N. Papamarkos, A neuro-fuzzy technique for document binarisation, Neural Comput. Appl. 12 (3-4) (2003) 190–199.

[2] N. Howe, Document binarization with automatic parameter tuning, Int. Jour. on Document Analysis and Recognition 16 (2013) 247–258.

[3] T. Lelore, F. Bouchara, FAIR: A fast algorithm for document image restoration, IEEE Trans. on Pattern Analysis and Machine Intelligence 35 (8) (2013) 2039–2048.

[4] M. Ramirez-Ortegon, V. Margner, E. Cuervas, R. Rojas, An optimization for binarization methods by removing binary artifacts, Pattern Recognition Letters 34 (2013) 1299–1306.

[5] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, S. Perantonis, An old greek handwritten OCR system based on an efficient segmentation-free approach, Int. Jour. on Document Analysis and Recognition (IJDAR), Special issue on the analysis of historical documents 9 (2–4) (2007) 179–192.

[6] S. Lu, B. Su, C. Tan, Document image binarization using background estimation and stroke edges, Int. Jour. on Document Analysis and Recognition 13 (2010) 303–314.

[7] R. Hedjam, R. Moghaddam, M. Cheriet, A spatially adaptive statistical method for the binarization of historical manuscripts and degraded images, Pattern Recognition 44 (2011) 2184–2196.

[8] R. Moghaddam, M. Cheriet, AdOtsu: An adaptive and parameterless generalization of otsu's method for document image binarization, Pattern Recognition 45 (2012) 2419–2431.

[9] N. Papamarkos, B. Gatos, A new approach for multithreshold selection, Computer Vision, Graphics, and Image Processing - Graphical Models and Image Processing 56 (5) (1994) 378–388.

[10] W. Niblack, An introduction to digital image processing, Prentice-Hall, Englewood Cliffs, NJ, 1986.

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

[12] J. Bernsen, Dynamic thresholding of grey-level images, in: Proc. 8th Int. Conf. on Pattern Recognition, Paris, France, 1986, pp. 1251–1255.

[13] E. Badekas, N. Papamarkos, Document binarization using kohonen SOM, IET Image Processing 1 (1) (2007) 67–84.

[14] E. Badekas, N. Papamarkos, Optimal combination of document binarization techniques using a self-organizing map neural network, Engineering Applications of Artificial Intelligence 20 (1) (2007) 11–24.

[15] E. Badekas, N. Nikolaou, N. Papamarkos, Text binarization in color documents, Int. J. of Imaging Systems and Technology 16 (6) (2006) 262–274.

[16] M. Makridis, N. Papamarkos, An adaptive layer-based local binarization technique for degraded documents, Int. Jour. of Pattern Recognition and Artificial Intelligence 24 (2) (2010) 1–35.

[17] S. Reddi, S. Rudin, H. Keshavan, An optimal multiple threshold scheme for image segmentation, IEEE Trans. System Man and Cybernetics 14 (4) (1984) 661–665.

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

[19] K. Ntirogiannis, B. Gatos, I. Pratikakis, A combined approach for the binarization of handwrittern document images, Pattern Recognition Letters 35 (2014) 3–15.

[20] B. Su, S. Lu, C. Tan, Binarization of historical document images using the local maximum and minimum, in: Proc. Int. Workshop on Document Analysis Systems, Boston, MA, USA, 2010, pp. 159–166.

[21] M. Valizadeh, E. Kabir, Binarization of degraded document image based on feature space partitioning and classification, Int. Jour. on Document Analysis and Recognition 15 (2012) 57–69.

[22] M. Ramirez-Ortego, E. Tapiaa, L. Ramirez-Ramirez, R. Rojas, E. Cuevas, Transition pixel: A concept for binarization based on edge detection and gray-intensity histograms, Pattern Recognition 43 (2010) 1233–1243.

[23] M. Ramirez-Ortego, E. Tapiaa, R. Rojas, E. Cuevas, Transition thresholds and transition operators for binarization and edge detection, Pattern Recognition 43 (2010) 32433254.

[24] M. Ramirez-Ortegon, L. Ramirez-Ramirez, V. Margner, I. Messaoud, E. Cuevas, R. Rojas, An analysis of the transition proportion for binarization in handwritten historical documents, Pattern Recognition 47 (8) (2014) 2635 – 2651.

[25] M. Ramirez-Ortegon, L. Ramirez-Ramirez, I. Messaoud, V. Margner, E. Cuevas, R. Rojas, A model for the gray-intensity distribution of historical handwritten documents and its application for binarization, International Journal on Document Analysis and Recognition 17 (2) (2014) 139 – 160.

[26] A. Gooch, S. C. Olsen, J. Tumblin, B. Gooch, Color2Gray: salience-preserving color removal, ACM Trans. Graphics 24 (2005) 634–639.

[27] M. Grundland, N. Dodgson, The decolorize algorithm for contrast enhancing, color to gray-scale conversion, Tech. rep., Technical Report, No. 649, Computer Laboratory, Cambridge University (2005).

[28] M. Qiu, G. Finlayson, G. Qiu, Contrast maximizing and brightness preserving color to gray-scale image conversion, in: Proc. 4th European Conference on Colour in Graphics, Imaging, and Vision, Paris, France, 2008, pp. 347–351.

[29] C. Kanan, G. Cottrell, Color-to-grayscale: Does the method matter in image recognition, PLoS ONE 7 (1).

[30] R. F. Moghaddam, M. Cheriet, A multi-scale framework for adaptive binarization of degraded document images, Pattern Recognition 43 (2010) 2186–2198.

[31] A. Hyvarinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley, New York, 2001.

[32] B. Gatos, K. Ntirogiannis, I. Pratikakis, ICDAR 2009 document image binarization contest (DIBCO2009), in: Proc. Int. Conf. on Document analysis and Recognition (ICDAR09), Barcelona, Spain, 2009, pp. 1375–1382.

[33] I. Pratikakis, B. Gatos, K. Ntirogiannis, H-DIBCO 2010 handwritten document image binarization competition, in: Proc. Int. Conf. on Frontiers in Handwriting Recognition, Kolkata, India, 2010, pp. 727–732.

[34] B. Gatos, K. Ntirogiannis, I. Pratikakis, DIBCO 2011 - document image binarization contest, International Journal of Document Analysis and Recognition 14 (1) (2011) 35–44.

[35] I. Pratikakis, B. Gatos, K. Ntirogiannis, ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012), in: Proc. Int. Conf. on Frontiers in Handwriting Recognition, Bari, Italy, 2012, pp. 817 – 822.

[36] I. Pratikakis, B. Gatos, K. Ntirogiannis, ICDAR 2013 document image binarization contest (DIBCO 2013), in: Proc. 12th Int. Conf. on Document Analysis and Recognition, Washington DC, USA, 2013, pp. 1471 – 1476.

[37] A. Michelson, Studies in Optics, U. of Chicago Press, 1927.

[38] A. P. Dempster, N. Laird, D. Rubin, Maximum likelihood for incomplete data via the EM algorithm, J. of the Royal Statistical Society, ser. B 39 (1977) 1–38.

[39] J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Mixture Models, Tech. rep., Department of Electrical Engineering and Computer Science, U.C. Berkeley, California (1998).

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