Local Co-occurrence and Contrast Mapping for Document Image Binarization

Nikolaos Mitianoudis and Nikolaos Papamarkos
Image Processing and Multimedia Lab
Democritus University of Thrace
Xanthi, Greece
Email: nmitiano@ee.duth.gr

***Abstract*—Document Image Binarization refers to the task of transforming a scanned image of a handwritten or printed document into a bi-level representation containing only characters and background. Here, we address the historic document image binarization problem using a three-stage methodology. Firstly, we remove possible stains and noise from the document image by estimating the document background image. The remaining background and character pixels are separated using a Local Co-occurrence Mapping, local contrast and a two-state Gaussian Mixture Model. In the last stage, possible isolated misclassified blobs are removed by a morphology operator. The proposed scheme offers robust and fast performance, especially for handwritten documents.**

***Index Terms*—Binarization; historic documents; background estimation**

I. Introduction

The advance of modern imaging devices has urged many libraries throughout the world to digitize their book collection, which is now available online as document images. These may contain printed or handwritten material. In order to exploit the content of these document images more efficiently, one should be able to extract their text information automatically, i.e. perform Optical Character Recognition (OCR). To enhance the performance of OCR algorithms, a number of preprocessing steps are usually applied, including page skew detection, artifact and noise removal, document page layout analysis and document image binarization [1]. In this paper, we address the problem of background removal and document image binarization.

Background estimation and removal is a preparatory step that enhances the document quality which is beneficial for many binarization techniques [2]. For example, historic document images often suffer from different degradation types that hinder efficient document image binarization and character recognition. Scanned documents often contain undesired textual noise, such as specks, dots, black borders, lines, and hole-punch marks. Handwritten document images are more challenging, since the character pixels may vary in intensity levels (depending on the ink type) and thus can not be easily identified using intensity information.

After the original document images have been enhanced and background artifacts have been removed, the output of a document preprocessing system is a bi-level image containing characters and background. Image binarization can then be performed either on a global or a local basis. The local binarization techniques of Niblack [3], Sauvola [4] and Bernsen [5] have been efficiently used by the community. In 2006, Gatos et al. (GPP) [6] introduced the efficient GPP method. The document background is estimated by an adaptive threshold, classifying each pixel either as text or background. Sauvola's binarization algorithm is used to roughly extract the text pixels and calculate the background surface from them by interpolation of neighboring background pixels intensities. For other pixels, background surface is set to the gray level of the original image. In [7], 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 introduced a novel 2D image representation that aimed at creating 2D concentrations (clusters) of neighboring pixels of similar intensity, i.e. document characters and background. Binarization was then performed by identifying 2 clusters using the multi-thresholding technique of Reddi et al. [8]. In [9], Su et al. demonstrated the use of local contrast image thresholding in estimating the text stroke width more accurately. In [10], 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. Recently, Howe [11] 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.

In this paper, the authors extend the previous work of Makridis and Papamarkos [7]. In the first stage, the background removal technique in [7] is enhanced by automating the median-filter window size selection and the threshold selection between the document image and the background estimate. In the second stage, the local neighborhood representation is redesigned to include local contrast information to enhance the inclusion of character outlines. Binarization is then performed by separating two clusters of document characters and background artifacts using Mixtures of Gaussians (MoG) clustering. Finally, small-size 8-connected clusters are removed to eliminate possible binarization noise.

II. Background Removal

Let $\mathbf{I}(x, y)$ be the initial color document image of size $3 \times M \times N$, where x, y denote image coordinates. In the case of degraded and poor quality documents, a background removal stage of the source image is essential for the elimination of noisy areas and artifacts [6], as these artifacts may easily be misclassified as text by most binarization algorithms.

Firstly, the three-channel RGB image is converted to an one-channel intensity grayscale image by performing Principal Component Analysis on the multichannel image. The principal component image is then retained as the grayscale image.

The aforementioned artifacts can then be isolated from the original image, by performing low-pass filtering of long window size. This long-window low-pass filtering can fade out document characters, leaving only an image containing artifacts and the document background that needs to be removed. Median filtering is preferable to low-pass filtering, since character intensity values will be replaced by existing background or artifact intensity values. Nonetheless, the median filter size needs to be defined. In [7], Makridis and Papamarkos used a fixed user-defined window size. Here, we 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. $S_I = 6$), it implies that the image still contains character information and thus the median filter size needs to increase by 5 , i.e. $G \leftarrow G + 5$. This procedure is repeated, until the previous criterion is satisfied. The final image $I_{MED}(x, y)$ is an estimate of the document background.

To remove the document background from the document image and form the $I_{NBG}(x, y)$ image, a simple comparison procedure classifies every pixel (x, y) as background or not. 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, otherwise it should be a character pixel. To automate the selection of T, we calculate a normalised histogram (density estimate) p of $|I(x, y) - I_{med}(x, y)|$. In most document images, this density tends to be a decreasing function. A threshold value of T can thus be set at the point, where the probability $p(T) \leq q \max(p)$ with $q \in [0, 1]$. In Fig. 1, a document image is depicted along with its background estimate and background removal results using various values of q.

III. Binarization

Document Image Binarization is the task of transforming a grayscale document image $I(x, y)$ into a bi-level image $I_{BN}(x, y)$, attributing one level to document characters and one to background pixels. In this work, we exploit the following local character properties: a) pixels belonging to any character are geometrically close, b) pixels belonging to any character should feature similar intensity values, c) any local area (neighborhood) that includes the outline of a character should have increased contrast. In [7], the authors employed the first two properties to propose a Symmetrical Frequency Map (SFM) that was used to perform binarization. Here, we extend this framework to emphasize character outlines.

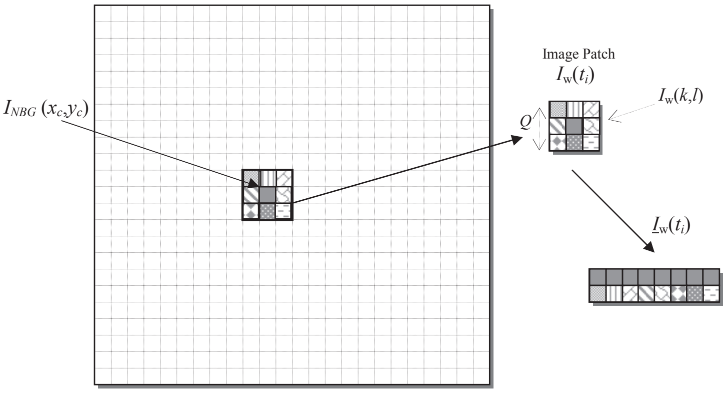


Fig. 2. Creating the Local Co-occurrence points for $Q = 3$.

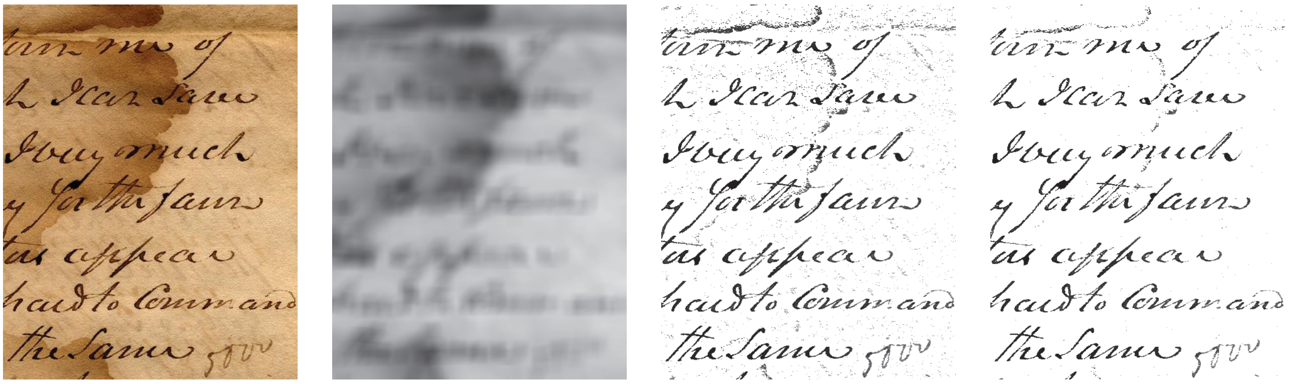
A. Local Co-occurence Mapping (LCM)

To emphasize proximity and connectivity between neighboring character pixels, we devise a co-occurrence map in the following manner. The image is divided into every distinct $Q \times Q$ patch. 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:

$$\left[\begin{array}{c} I_{NBG}(x_c^i, y_c^i) \\ I_{NBG}(x_c^i + dx, y_c^i + dy) \end{array} \right], |dx|, |dy| \leq \lfloor Q/2 \rfloor \quad (1)$$

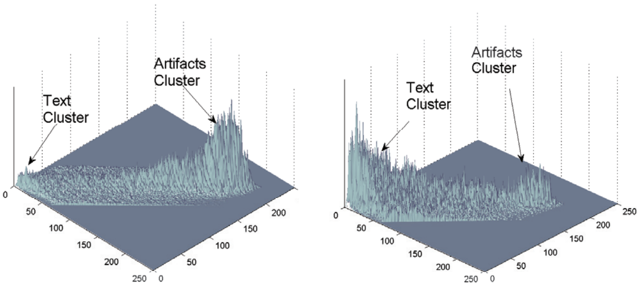
Thus, each pixel in the i-th patch is transformed to a 2D point, containing the intensity of the central patch pixel and the pixel's intensity. The whole procedure is visualised in Fig. 2. The combination of the center pixel with itself is not included in the formation of this group of 2D points. Repeating 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.

Observing the original LCM histograms [7] in many document images, we reached to the following conclusions. First, the character cluster is usually smaller compared to the background cluster. This is logical since the character pixels constitute only a very small part of the image, compared to the background pixels. This will hinder any clustering attempt to estimate the character cluster, since the background cluster dominates the LCM 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



(a) Degraded Handwritten Image (b) Background estimate (c) Background Removal 1 (d) Background Removal 2

Fig. 1. A typical degraded handwritten image (a) with its background estimate (b) and the proposed method's removal results for $q = 0.3$ (c) and $q = 0.1$ (d).



(a) Initial 2D map (b) Improved 2D map

Fig. 3. Improvement in the 2D LCM histograms, compared to the previous in [7].

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 the background identified points from $\mathbf{I}_W(t_i)$. After removing these points, the LCM histograms change considerably. The character clusters are more visible compared to the background cluster. In addition, the actual task is now amended to discriminate between the character and the misclassified background / artifact pixels. In Fig. 3 (a), (b), the improvement in the previous LCM histograms is shown.

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.

To incorporate the third local character property, we should emphasize the existence of strong contrast in the $Q \times Q$ neighborhood, which denotes the existence of character outlines. The use of contrast to estimate text stroke width was also discussed in [9]. We also experimented with similar textural measures, including standard deviation an entropy, but the use of contrast seemed to be more stable. To include this information in the LCM, we simply calculate the 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 as below:

$$C(I_W(t_i)) = \frac{\max(I_W(t_i)) - \min(I_W(t_i))}{\max(I_W(t_i)) + \min(I_W(t_i))} \tag{2}$$

Favourable patches are those with high contrast values, whereas favourable patches in the 2D LCM feature lower values (dark intensities-characters). To move the contrast 3rd dimension 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 of (2).

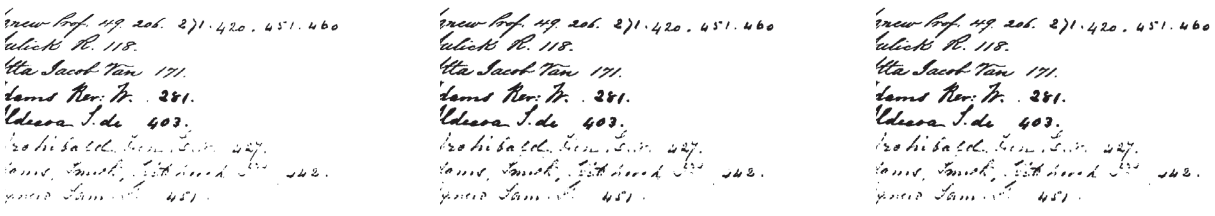
$$iC(u) = 255(1 - \tanh(2C(u))) \tag{3}$$

The nonlinear function $\tanh(\cdot)$ serves as a method of increasing separation between the two clusters: *character outlines* and *low-contrast patches*. The new $iC(\cdot)$ values are used to form the novel 3D LCM representation, as follows:

$$\mathbf{I}_W(k) = \begin{bmatrix} I_{NBG}(x_c^i, y_c^i) \\ I_{NBG}(x_c^i + dx, y_c^i + dy) \\ iC(I_{NBG}(x_c^i, y_c^i)) \end{bmatrix}, \ |dx|, |dy| \leq \lfloor Q/2 \rfloor \tag{4}$$

B. Binarization via MoG clustering

Once the LCM representation has been established, image binarization can be achieved by performing clustering on the data points $\mathbf{I}_W(k)$. In this work, we examine the application of Mixtures of Gaussians (MoG) modeling to address this clustering problem. *Mixtures of Gaussians* (MoG) is a weighted sum (mixture) of different multidimensional Gaussians that can be used to model any arbitrary pdf that does not follow a particularly known distribution [12]. In the special case that the arbitrary data distribution show 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 MoG were established in [12], [13], where



(a) 2D-LCM output (b) 3D-LCM output (c) 3D-LCM, Post-Processing

Fig. 4. Binarization improvement via 3D-LCM and post-processing

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, which is commonly accelarated by using the K-Means, the Fuzzy C-Means or the Harmonic K-Means algorithm for initialisation.

We will employ the EM algorithm, as described in [13], to perform clustering of the LCM data $\mathbf{I}_w(k)$. Our clustering problem is very constrained and these constraints can be used in the initialisation of the EM algorithm to expedite and guide its convergence. First of all, we are looking at identifying 2 clusters (characters-artifacts), thus the order of MoG is $K = 2$. The MoG's mixing coefficients should be initialised as $a_i = 0.5$, as both clusters are equally probable. The initialization of the Gaussian 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 characters 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.

Once the EM algorithm has converged, we 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 the following: "*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. $I_{BN}(x^i, y^i) = 255$.*"

In Fig. 4, we can see the binarization result of the 2D-LCM scheme, compared to the novel 3D-LCM. The 2D-LCM gave an F-Measure (FM) score [14] of 0.822, while the 3D-LCM gave an FM score of 0.839. The strengthening of character outlines is more visible in the bottom part of the image. This improvement is consistent with all the handwritten images in our experiments.

IV. Post-Processing

The final stage aims at removing artifacts from the previous binarisation stage. Isolated blobs or small misclassified noisy items can be removed using a mathematical morphology step. We identify connected objects with 8-connectivity in the binary output of the 3D LCM algorithm. All these connected components that contain less than 20 pixels are removed, as they should be noisy artifacts. Of course, this threshold is related to the image resolution and has to be adapted accordingly. This choice seemed to work well for the H-DIBCO image databases that were our main experimental ground. In Fig. 4 (c), we can see the result of post-processing on the previous 3D LCM binary image (b), giving an FM score of 0.842.

V. Experiments

In this section, we compare the proposed LCM binazation method with other common binarization methods. In our benchmarking exercise, we used Otsu's thresholding method [15] and the local binarization techniques of Sauvola (Sau) [4] and Bernsen (Bern) [5]. We used the GPP binarization method, as proposed by Gatos et al. [6] and the earlier work of Makridis and Papamarkos (MakPap) [7]. For the proposed LCM method, we used a value of $q = 0.3$ for the background removal step. For the training and evaluation of MoGs, we used the functions `gaussmix` and `gaussmixp` respectively available freely from Voicebox[1]. All experiments were conducted on an Intel Core i5-460M (2.53 GHz) with 4GB DDR3 SDRAM running Windows 7 Professional 64-bit and MATLAB R2013a. Our MATLAB implementations were not optimised in terms of execution speed. It was not our intention to develop the best performing binarization algorithm, but to introduce the LCM structure for local document image modelling. However, we can see that the proposed algorithm performs favorably with the tested approaches and scores reported at image binarization competitions.

The document images used in our study, were publicly available by the document image binarization community in previous open competitions, including DIBCO2009 [16], DIBCO2011 [14], H-DIBCO2010 [17] and H-DIBCO2012 [18]. In these competitions, hand-written historical 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. Images from all these datasets

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

were used in our study. The objective evaluation metrics of PSNR, MSE, Recall, Precision, FM and NRM [14] were calculated from all the resulting images.

Firstly, we estimate the algorithm's running time by using MATLAB's commands `tic-toc` on the aforementioned computer. 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. We thus estimate that the algorithm's running time is on average 0.025 msec per pixel. It was not possible to compare the proposed algorithm's runnning time with those of the other benchmarking approaches, since they were implemented on different, more optimised platforms (Delphi and C++).

In Table I, binarization results of handwritten historical document images from DIBCO2009, H-DIBCO2010, DIBCO2011 and H-DIBCO2012 are presented. Here, we used the values of $S_I = 6$, majority of pixels = 98% and $q = 0.3$ for the document background removal. This value of q implies that there was more need for degradation removal in these document images. In all cases, the proposed LCM method tends to outperform the other tested methods and improve considerably compared to the previous MakPap method. In Fig. 5, some typical examples from H-DIBCO 2012 is 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 and binarize the image, compared to the other methods. The result images of all datasets can be downloaded from the following url[2]. Comparing with the results of other methods reported in DIBCO2009 [16], the method would rank 2nd in terms of PSNR and 12th in FM. Comparing with the results, reported in H-DIBCO2010 [14], the method would rank 6th in terms of PSNR and 6th in FM. Comparing with the results, reported in DIBCO2011 [17], the method would rank 4th both in terms of PSNR and FM. Finally, looking at the H-DIBCO2012 results [14], the method would get the 15th position in terms of PSNR and FM, but will be at the top faster methods at this performance on a slighter faster machine.

VI. Conclusions

In this paper, the authors propose a novel document image binarisation system. The system consists of an image background removal step, an image binarization step and a post-processing morphological step. During the background removal, 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. The Local Co-occurance map (LCM) is then calculated, where the image is segmented into distinct $Q \times Q$ blocks, and for every pixel in the block, a 3D vector is created containing the central block pixel value, the actual pixel value, and the local block contrast value. Clustering is performed on the LCM data using a Mixture-of-Gaussian (MoG) model of two Gaussians. One Gaussian representing the character cluster and the other the background cluster. Finally, some isolated binary artifacts are removed by morphological 8-connected object segmentation. The proposed approach is robust to severe degradations of the document images. The inclusion of contrast seems to improve the inclusion of character outlines in the binarization result and outperform the previous offering of [7]. In our benchmarks, the LCM approach tends to perform better in most cases than the other benchmarking methods and compare favourably with other state-of-the-art approaches, as benchmarked in open competitions.

References

[1] N. Papamarkos, "A neuro-fuzzy technique for document binarisation," *Neural Comput. Appl.*, vol. 12, no. 3 - 4, p. 190 199, 2003.

[2] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, and 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*, vol. 9, no. 2–4, pp. 179–192, 2007.

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

[4] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, pp. 225–236, 2000.

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

[6] B. Gatos, I. Pratikakis, and S. Perantonis, "Adaptive degraded document image binarization," *Pattern Recognition*, vol. 39, pp. 317–327, 2006.

[7] M. Makridis and N. Papamarkos, "An adaptive layer-based local binarization technique for degraded documents," *Int. Jour. of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 2, pp. 1–35, 2010.

[8] S. Reddi, S. Rudin, and H. Keshavan, "An optimal multiple threshold scheme for image segmentation," *IEEE Trans. System Man and Cybernetics*, vol. 14, no. 4, pp. 661–665, 1984.

[9] B. Su, S. Lu, and 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.

[10] S. Lu, B. Su, and C. Tan, "Document image binarization using background estimation and stroke edges," *Int. Jour. on Document Analysis and Recognition*, vol. 13, pp. 303–314, 2010.

[11] N. Howe, "Document binarization with automatic parameter tuning," *Int. Jour. on Document Analysis and Recognition*, vol. 16, pp. 247–258, 2013.

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

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

[14] I. Pratikakis, B. Gatos, and 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.

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62 – 66, 1979.

[16] B. Gatos, K. Ntirogiannis, and 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.

[17] ——, "DIBCO 2011 - document image binarization contest," *International Journal of Document Analysis and Recognition*, vol. 14, no. 1, pp. 35–44, 2011.

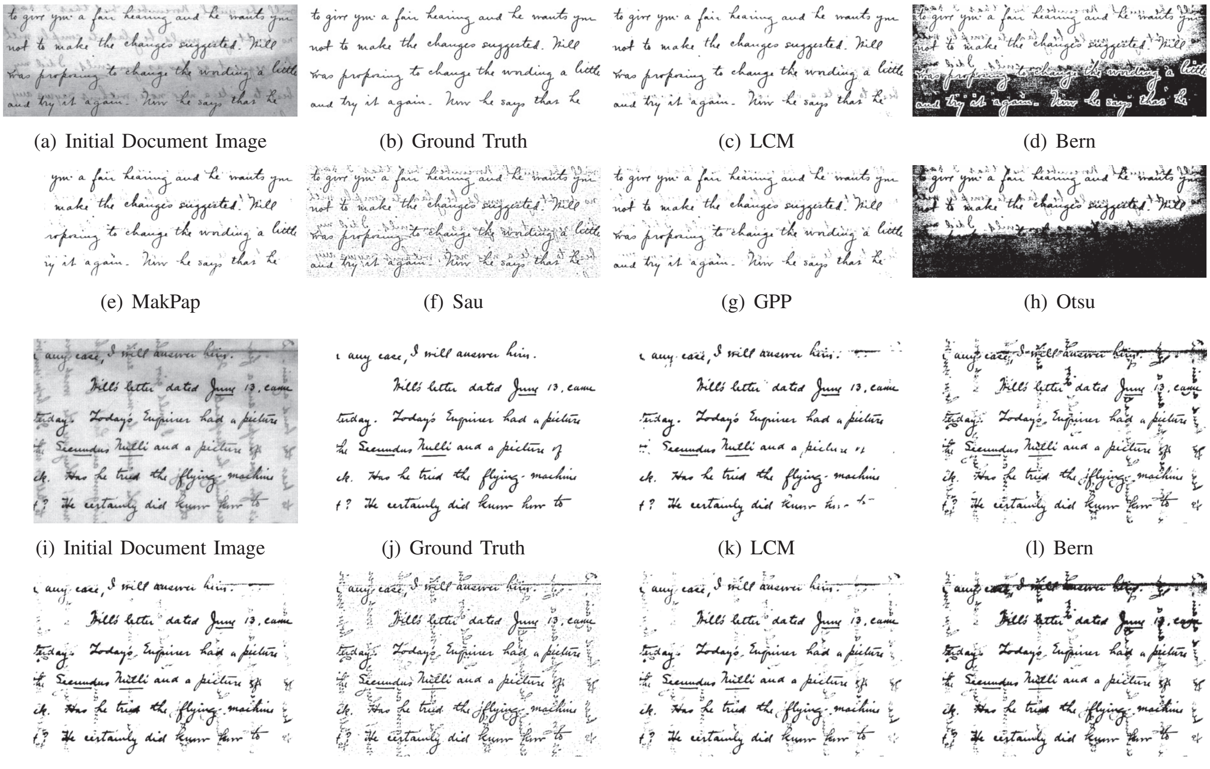
[18] I. Pratikakis, B. Gatos, and K. Ntirogiannis, "ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012)," in *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Bari, Italy, 2012.

[2] `http://utopia.duth.gr/∼nmitiano/handwritten.rar`

TABLE I
AVERAGE RESULTS FOR THE HANDWRITTEN HISTORICAL DOCUMENT IMAGES OF DIBCO2009, DIBCO2010, DIBCO2011 AND DIBCO2012.

	Average handwritten DIBCO2009						*Average handwritten DIBCO2010*					
	LCM	Bern	MakPap	Sau	Otsu	GPP	LCM	Bern	MakPap	Sau	Otsu	GPP
PSNR (dB)	**18.30**	14.02	16.34	16.78	13.93	17.74	**17.84**	16.86	14.67	16.04	17.44	15.96
MSE	**0.018**	0.078	0.025	0.023	0.090	**0.018**	**0.017**	0.022	0.040	0.028	0.0186	0.029
Recall	0.9031	0.8989	0.6775	0.8516	0.9450	0.8508	0.9203	0.7586	0.4778	0.7398	0.8184	0.6575
Precision	0.8285	0.5815	0.8754	0.7763	0.5806	0.8465	0.8481	0.9211	0.9500	0.8833	0.9016	0.9434
FM	**0.8617**	0.6531	0.7236	0.7859	0.6594	0.8365	**0.8793**	0.8196	0.56	0.7874	0.8527	0.7494
NRM	**0.0552**	0.0887	0.1643	0.0818	0.0741	0.0792	**0.0458**	0.1238	0.2628	0.1350	0.0944	0.1733

	Average handwritten DIBCO2011						*Average handwritten DIBCO2012*					
	LCM	Bern	MakPap	Sau	Otsu	GPP	LCM	Bern	MakPap	Sau	Otsu	GPP
PSNR (dB)	**17.63**	14.98	15.42	14.55	15.03	16.74	**18.55**	15.61	14.94	16.30	15.57	17.04
MSE	**0.0178**	0.0451	0.034	0.0439	0.0539	0.0238	**0.01486**	0.0564	0.038	0.0267	0.0638	0.0219
Recall	0.9322	0.8101	0.580	0.8400	0.8461	0.8091	0.8971	0.8383	0.92	0.7777	0.8647	0.7481
Precision	0.8293	0.7606	0.57	0.7154	0.7471	0.8719	0.8943	0.7789	0.91	0.8535	0.7775	0.9399
FM	**0.8748**	0.7658	0.67	0.7556	0.7671	0.8318	**0.8914**	0.7666	0.63	0.8008	0.7748	0.8178
NRM	**0.0409**	0.1126	0.208	0.0975	0.1013	0.1011	**0.0553**	0.1051	0.214	0.1169	0.0970	0.1282



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

(e) MakPap (f) Sau (g) GPP (h) Otsu

(i) Initial Document Image (j) Ground Truth (k) LCM (l) Bern

(m) MakPap (n) Sau (o) GPP (p) Otsu

Fig. 5. Binarization results for the "H05" and the "H06" handwritten document image from the DIBCO2012 dataset.