Binarization Algorithms for Historical Text Documents: A Survey and Implementations

Oliver A. Nina
olivernina@gmail.com

Matthew Stofflet
Matthew.Stofflet@knights.ucf.edu

University of Central Florida
Orlando, FL

Abstract

Binarization is an important part of reading text documents automatically through Optical Character Recognition. However, binarization of historical documents is difficult and is still an open area of research.

There have been great advances in the binarization of historical text documents in previous years. These advances have been seen in recent binarization contests such as the Document Image Binarization Contest (DIBCO) in 2009 [6] and 2011 [7] and the Handwritten Document Image Binarization Contest (HDIBCO) in 2010 [8] and 2012 [9]. These contests have brought new a new state of the art in the accuracy of binarization algorithms to text documents and to hand written documents.

In this paper we present a survey of the top ranked methods that participated in the DIBCO and the HDIBCO competitions. More specifically, we will look at the Lu algorithm, that won the DIBCO competition in 2009, the Su Algorithm, that won the HDIBCO competition in 2010, and the Smith and Nina algorithms that finished first and fourth at HDIBCO 2012, respectively.

In this paper, we review the previous state of the art methods. Since at this point there are no open source implementations available of some of these methods, we will also make available through this paper our implementation of the Lu and Su algorithms in C++, as an open source project.

Introduction

Before performing any Optical Character Recognition to read the text in historical images, we need to perform binarization. Binarization is usually one of the first steps of preprocessing images in order to read the text.

Binarization is a difficult problem when the images come from old documents where the images have a lot of noise; artifacts that make the text blurry, faint or not legible.

During last few years there have been improvements on the binarization of typewritten text and handwritten text images due to several events such as the DIBCO and HDIBCO competitions.

These competitions have brought binarization methods to new levels, including state of the art. Some of them are complicated and achieve high accuracy; others are more simple and faster. But there is a tradeoff between speed and accuracy.

In this paper we will provide a survey of the four methods that have participated in the DIBCO and the HDIBCO competitions, mainly the Lu [3], Su [4], Howe [5] and Nina algorithms.

Lu Algorithm

The Lu algorithm [3] won the DIBCO 2009 competition and was one of the first methods to make a breakthrough in text binarization.

This algorithm consists of several parts that better distinguish the text from the background noise. The first is approximating the background.

Background approximation: According to the paper, the approximation of the background through this algorithm is done by polynomial fitting on the entire document.

This is done by sampling every line from the image and creating a curve based on the intensity values of that line. Then a polynomial fitting algorithm is used to reconstruct the curve of intensities of the sampled line. By doing this we can better isolate the pixels that are darker than the background, at every line. We do the polynomial fitting for both the horizontal and the vertical axis.
Once we have a good approximation of the background, we will do the next step of the algorithm which is contrast compensation.
The image contrast compensation is done by using the image from the background approximation with the formula:

$$I' = \frac{C}{BG} I$$

where I is the original image C is a constant value, BG is the background approximated value and I' is the compensated value. The result of this step will give us an image with a more uniform background. This step will help remove the lighting and contrast artifacts from the original image.

Once we compensate the image with the background approximated image, we will create a histogram of the edges of neighboring pixels. This resembles calculating the first derivative of the pixels in x and y. When we add these differences to create a histogram of differences we use Otsu to approximate the value of the text stroke edges.

A local threshold estimation approach is used by a sliding window approach. This is accomplished by looking at the neighboring pixels of a window and comparing the center of the window with any edge pixels within the window. If there are any edge pixels within the window above a minimum number and if the center pixel intensity is smaller than the mean of edge pixels intensity inside the window, then the center pixel is classified as text.
The size of the window is calculated by estimating the stroke width of the text through a measuring of distances between edge pixels and picking the most frequent distance in the histogram. The final step of the algorithm is despeckling the binarized image using morphological operators.

Su algorithm

The Su algorithm [4] won the HDIBCO 2010 competition. It uses the subtraction of images calculated by the dilation and erosion morphological operators, or the max and min operators as they are called in their paper.
The subtraction of the two images simulates an edge detector that results into a high response on the edges of the text. In order to avoid an artifact of uneven background, the algorithm uses a normalization term that is equal to the sum of the dilation and erosion results plus a small constant. The edge and non-edge pixels get separated through the Otsu [1] algorithm based on the histogram's final value intensities.
Once the edges have been approximated through the previous step, the whole image is then thresholded using the sliding window approach, in a similar way to Lu although the thresholds are set differently. In this case the pixel in the center of the window is classified as text if the intensity of the pixel is less than half of the standard deviation, above the average of edge pixel intensities. As in the previous method, the number of edge pixels within the pixel has to be greater than a minimum number.

Nina Algorithm

The Nina algorithm placed 4th in the HDIBCO competition 2012 [9]. It is an improvement of the Lu algorithm. The steps have been enhanced from the Lu algorithm are the background approximation, the edge detection, and the thresholds. The background approximation was changed by not only using polynomial fitting but also applying a median filter over the original image with a large kernel. Both the polynomial fitted and the median filter approximations are averaged to obtain the background approximation. The edge detection step was changed by smoothing the compensated image with a median filter and calculating the L2 norm gradient magnitude. The final edge detection is obtained by averaging the Otsu [1] and the Kittler [2] thresholds and using it as the final threshold for the high gradient values.

Howe Algorithm

The Howe algorithm [5] won the HDIBCO 2012 [9] competition and is the state of the art algorithm for binarizing text images. This method uses local information of pixels from a Laplacian image combined with information of a Canny [10] edge detector to binarize an image base on a local energy function. The energy function uses six parameters of which only two of them strongly influence the binarization. The algorithm uses a way to automatically calibrate these two parameters for better results.
More details about the algorithm can be found in Howe's paper to appear at IJDAR
Implementation of this algorithm can be found at the author's website:
http://cs.smith.edu/~nhowe/research/code/

Source Code

We are providing implementations for the Lu, Su and Nina algorithms at the following link:
https://code.google.com/p/binatool/
The code is free of charge for educational and research purposes. Please contact Oliver Nina if you would like to use the code for commercial purposes.
If you use the code for research purposes please make a reference to our implementation.
The Lu and Su algorithm have been implemented following the papers published by the authors. These implementation somewhat vary from the original implementations. However, according to our experiments the scores are very similar to the ones reported by the author as seen the following table. The Lu* and Su* are our implementations of the algorithms.

DIBCO2009	Recall	Precision	FMeasure	DIBCO2009	Recall	Precision	FMeasure
Lu 1	95.4074	89.9925	92.6209	LU* 1	85.3333	98.7011	91.5317
2	97.3744	88.0486	92.477	2	85.206	97.3306	91.2139
3	96.0056	84.6147	89.9509	3	85.8865	93.027	89.3142
4	85.7177	89.0082	87.332	4	86.9091	93.2633	89.9742
5	78.861	81.9241	80.3634	5	79.7937	94.4722	86.5148
SU 1	85.7908	96.7441	90.9388	SU* 1	92.1493	94.0648	93.0972
2	86.5825	95.6946	90.9108	2	86.1246	95.9549	90.7744
3	91.8889	90.3542	91.1151	3	94.0516	92.5594	93.2995
4	84.038	96.565	89.8671	4	83.0702	97.099	89.5385
5	87.5816	89.4915	88.5263	5	87.6749	89.7151	88.6833

Results

In this section we provide quantitative and qualitative results of the methods explained in this paper.

Quantitative Results

We first present the tables of Measurements of precision, recall and F-measures of the methods described previously.

We run the algorithm on the different datasets from previous competitions and in some cases use the result images from the authors. We calculated the scores for all the algorithms in all the datasets except for Su which we did not include in the HDIBCO 2012 dataset.
As we can see in the Results the best performing algorithm is the Howe algorithm which usually has the highest F-Measure score, except in a few cases such as in image 1 in DIBCO 2011 which can be seen in the appendix. However, for the most part, the algorithm performs better than other algorithms.
In the appendix, we provide the results in detail of each image in all datasets.

DIBCO2009	Recall	Precision	FMeasure
Howe	95.76882	93.75534	**94.749**
Lu	90.67322	86.71762	88.54884
Nina	88.92054	89.51694	89.17564
Su	87.17636	93.76988	90.27162

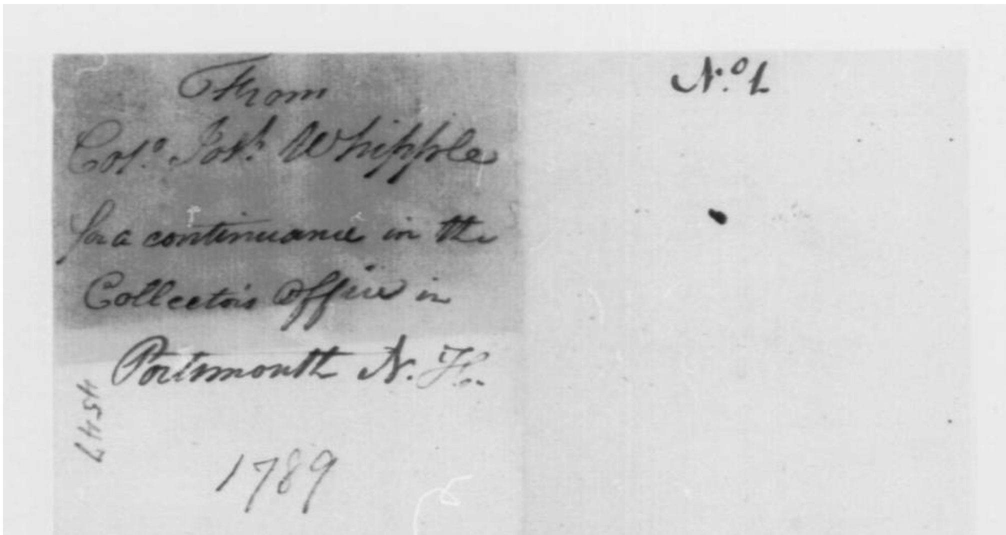
HDIBCO10	Recall	Precision	FMeasure
Howe	83.5739	86.82059	85.06971
Lu	44.8274	83.8533	55.42678
Nina	85.5897	96.16366	**90.48693**
Su	40.7378	96.23822	57.11821

DIBCO11	Recall	Precision	FMeasure
Howe	93.01155	89.412488	**91.085038**
Lu	86.866613	84.412888	83.09255
Nina	85.914075	94.482438	89.913563
Su	79.043863	90.938263	83.722138

HDIBCO12	Recall	Precision	FMeasure
Howe	91.9762929	96.02361	**93.728814**
Lu	77.5641429	76.80214	74.726671
Nina	85.2798357	96.04799	90.189486

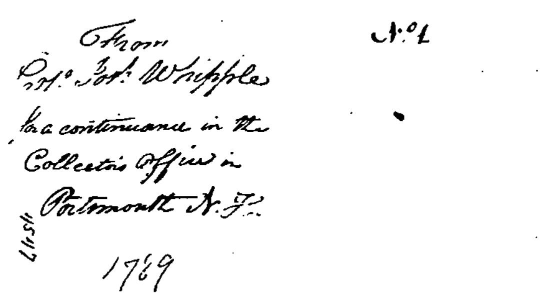
Qualitative Results

In this section, we look at a few examples where we can see the results of the algorithms before mentioned. We tried to choose representative images that were among the hardest to binarize within its dataset.

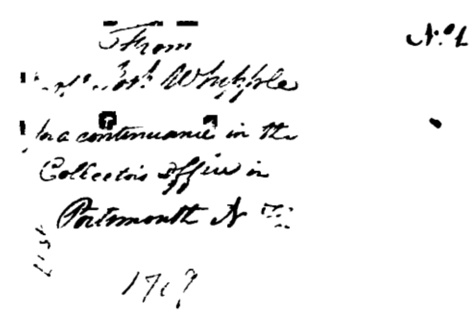


Original Image 5 – DIBCO 2009

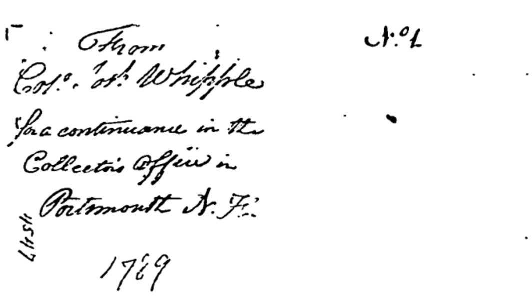
Howe



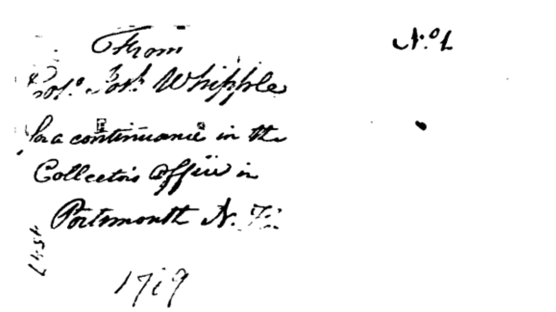
Lu

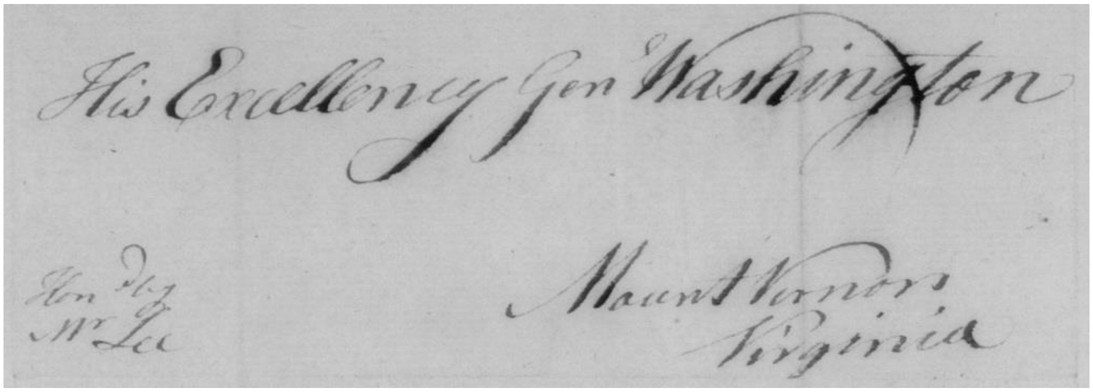


Nina



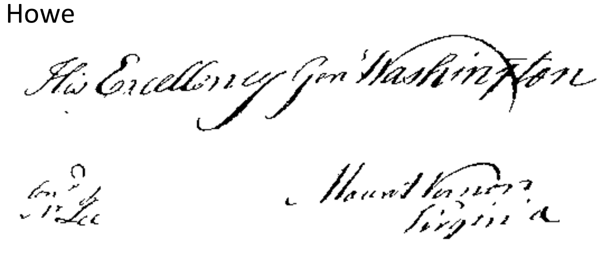
Su



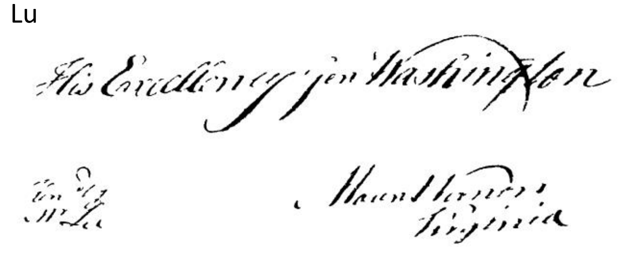


Original Image H010 – HDIBCO2010

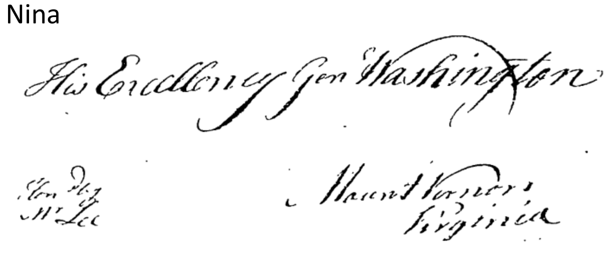
Howe



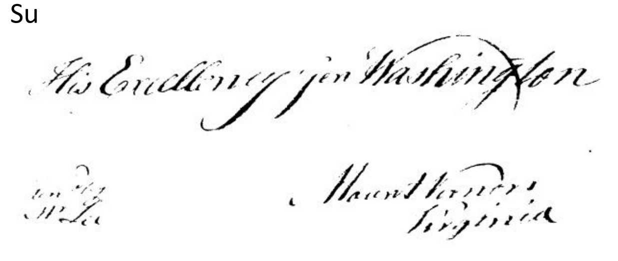
Lu

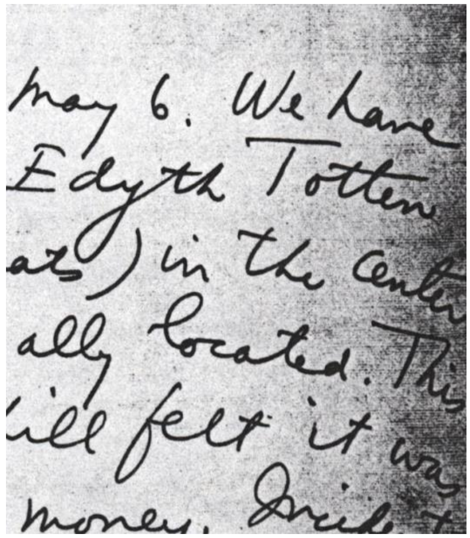


Nina

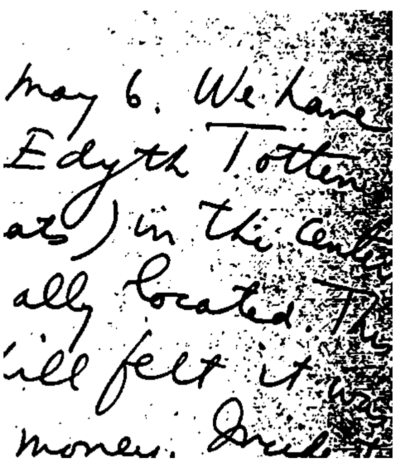
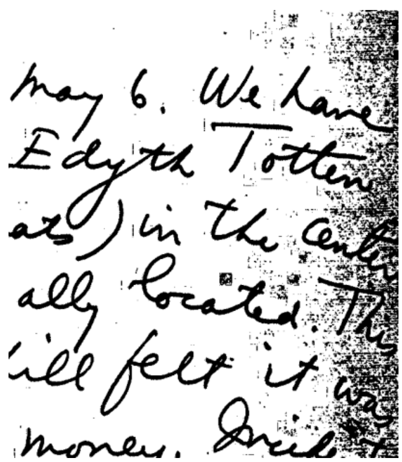
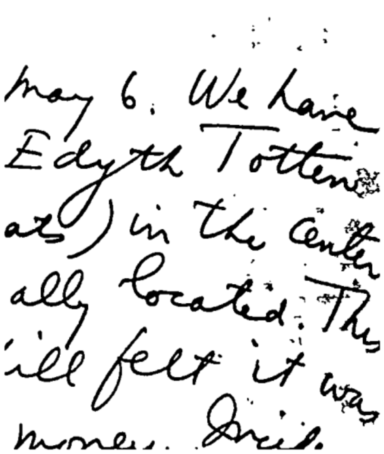
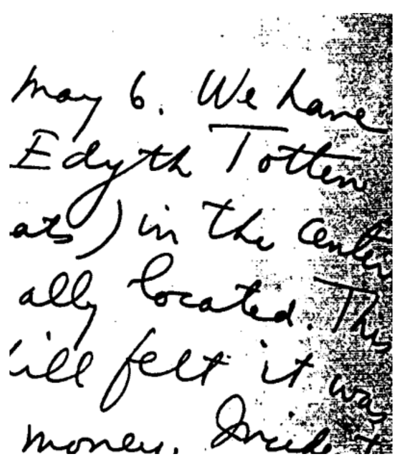


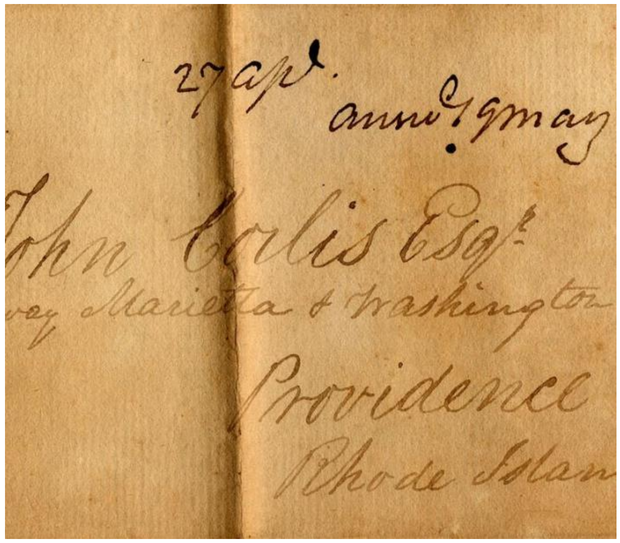
Su




may 6. We have
Edyth Totten
ats) in the center
ally located. This
ill felt it was
money. ...

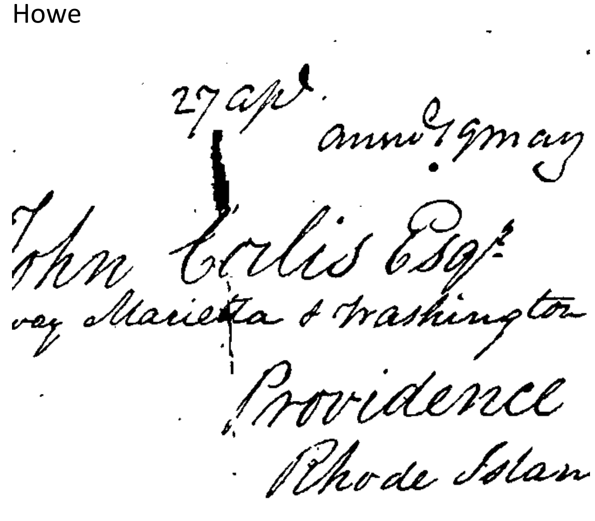
Original Image HW1 – DIBCO 2011

Howe 	Lu 
Nina 	Su 

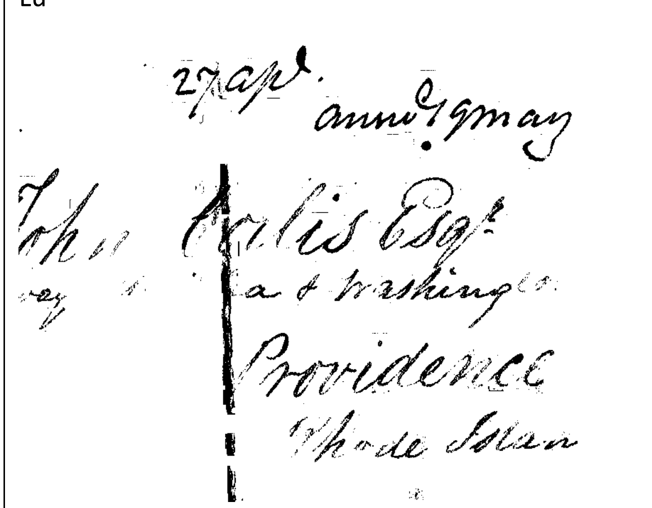


Original Image HW6 –DIBCO2011

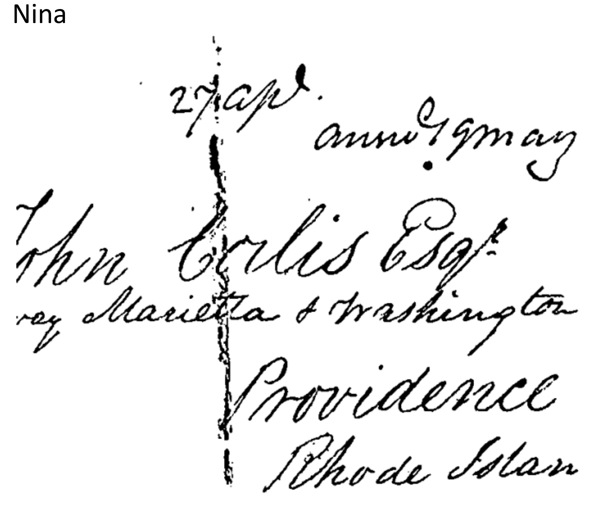
Howe



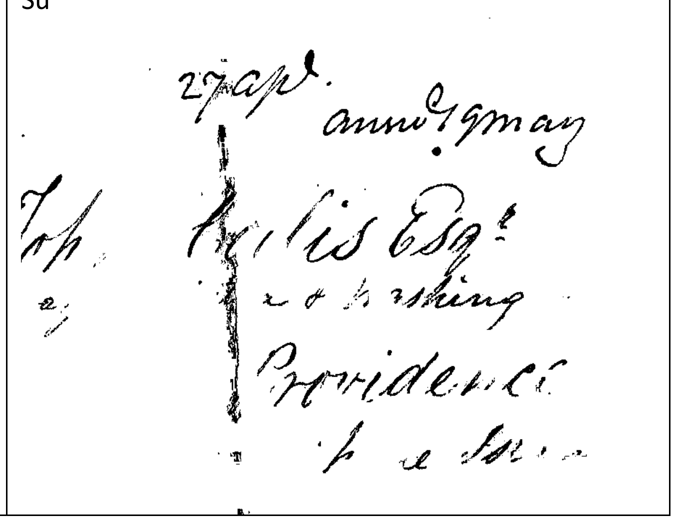
Lu

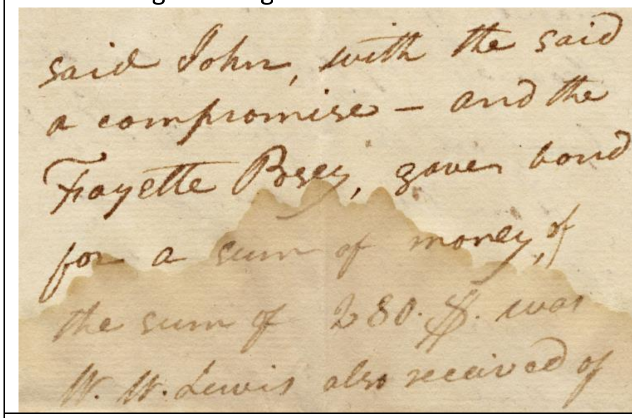
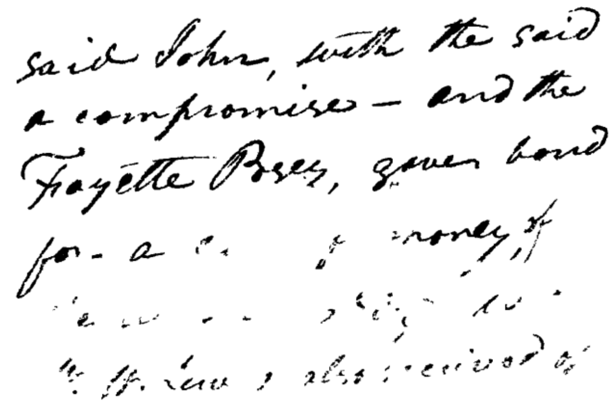
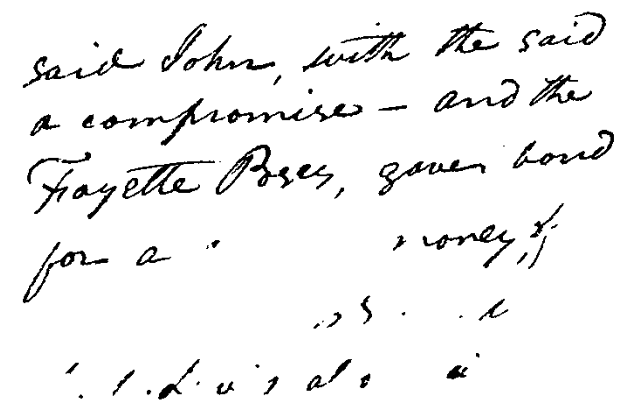
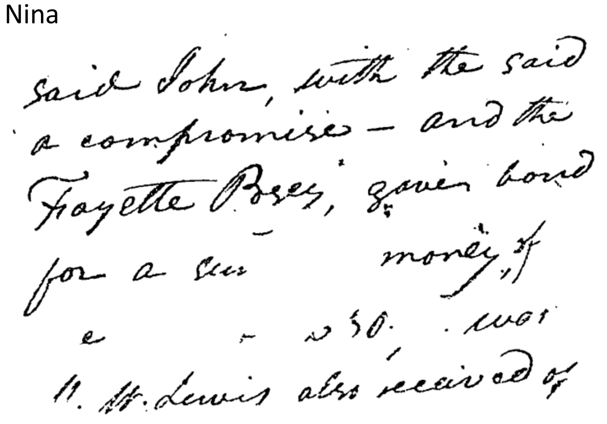


Nina



Su



Original Image H02 – HDIBCO 2012	Lu
	
Howe	Nina
	

Conclusion

Binarization of text images is a hard problem but recent competitions have advanced the state of the art to new levels.

In this paper we present a summary of state of the art binarization methods for text images.

We conclude that Howe is overall the best method so far for binarization of text images. However, there is still room for improvements since as we can see from our experiments; this algorithm still makes mistakes in classifying text and background.

Acknowledgements

We would like to thank Dr. Nicholas Howe for his help in providing details and code for his winning algorithm at HDIBCO 2012.

References

[1] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys., Man., Cyber.* 9 (1): 62–66

[2] J. Kittler and J. Illingworth. 1986. Minimum error thresholding. *Pattern Recogn.* 19, 1 (January 1986), 41-47.Lu

[3] Shijian Lu, Bolan Su, Chew Lim Tan. Document Image Binarization Using Background Estimation and Stroke Edges. International Journal on Document Analysis and Recognition, December 2010

[4] Bolan Su, Shijian Lu, Chew Lim Tan. Binarization of Historical Document Images Using the Local Maximum and Minimum. International Workshop on Document Analysis Systems, 9-11 June 2010, Boston, MA, USA.

[5] Document Binarization with Automatic Parameter Tuning, N. Howe. To appear in *International Journal of Document Analysis and Recognition*.

[6] Gatos, B.; Ntirogiannis, K.; Pratikakis, I., "ICDAR 2009 Document Image Binarization Contest (DIBCO 2009)," *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on* , vol., no., pp.1375,1382, 26-29 July 2009

[7] Pratikakis, I.; Gatos, B.; Ntirogiannis, K., "H-DIBCO 2010 - Handwritten Document Image Binarization Competition," *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on* , vol., no., pp.727,732, 16-18 Nov. 2010

[8] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. 2011. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). In *Proceedings of the 2011 International Conference on Document Analysis and Recognition* (ICDAR '11). IEEE Computer Society, Washington, DC, USA, 1506-1510. DOI=10.1109/ICDAR.2011.299 http://dx.doi.org/10.1109/ICDAR.2011.299

[9] Pratikakis, I.; Gatos, B.; Ntirogiannis, K., "ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012)," *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on* , vol., no., pp.817,822, 18-20 Sept. 2012

[10] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

Appendix

DIBCO09	Recall	Precision	FMeasure
Howe 1	96.8355	94.8031	95.8085
2	96.5481	94.7951	95.6636
3	96.4122	93.7767	95.0762
4	96.1998	92.9514	94.5477
5	92.8485	92.4504	92.649
Lu 1	95.4074	89.9925	92.6209
2	97.3744	88.0486	92.477
3	96.0056	84.6147	89.9509
4	85.7177	89.0082	87.332
5	78.861	81.9241	80.3634
Nina 1	91.801	96.103	93.9027
2	86.654	80.1303	83.2646
3	89.3555	90.5745	89.9609
4	89.0899	89.6005	89.3444
5	87.7023	91.1764	89.4056
SU 1	85.7908	96.7441	90.9388
2	86.5825	95.6946	90.9108
3	91.8889	90.3542	91.1151
4	84.038	96.565	89.8671
5	87.5816	89.4915	88.5263

HDIBCO10	Recall	Precision	FMeasure
Howe 1	95.1912	95.5134	95.352
2	95.8216	94.9347	95.3761
3	92.4684	96.9853	94.673
4	8.7129	8.5951	8.6536
5	97.1785	95.6983	96.4327
6	87.1504	95.0625	90.9346
7	94.8937	95.3043	95.0985
8	93.3965	93.4873	93.4419
9	92.1703	94.8891	93.5099
10	78.7551	97.7359	87.2248
LU 1	54.6054	15.0435	23.5885
2	36.4399	93.4428	52.4326
3	43.8312	90.9123	59.1464
4	37.3517	93.4407	53.3696
5	51.1979	87.8251	64.6865
6	44.3897	93.2963	60.1571
7	45.4856	93.1872	61.132
8	47.3699	84.9286	60.8179
9	47.338	91.3559	62.3618

10	40.2643	95.1006	56.5754
Nina 1	89.4265	96.9974	93.0582
2	89.3582	96.4318	92.7603
3	83.2046	98.2553	90.1057
4	82.5478	95.794	88.679
5	90.1447	96.0298	92.9959
6	81.7933	95.29	88.0273
7	87.18	96.3873	91.5527
8	88.5567	93.368	90.8988
9	87.9923	94.8085	91.2733
10	75.6929	98.2745	85.5181
SU 1	46.4546	94.6815	62.3283
2	46.0232	95.5459	62.1227
3	39.4583	98.2868	56.3102
4	40.0383	95.7711	56.469
5	43.2771	98.8806	60.2045
6	36.783	94.8799	53.0137
7	40.0641	97.7777	56.8387
8	43.6059	92.0377	59.1755
9	36.5787	95.4965	52.8962
10	35.0949	99.0245	51.8233

DIBCO11	Recall	Precision	FMeasure
Howe 1	98.452	57.1744	72.3391
2	97.0087	97.7943	97.4
3	90.6568	96.5083	93.4911
4	88.3931	96.0073	92.043
5	96.9853	85.3901	96.1811
6	93.5512	90.6267	92.0657
7	85.5264	94.5222	89.7996
8	93.5189	97.2766	95.3607
LU 1	97.9942	94.0632	77.4765
2	86.7076	91.7181	89.1425
3	87.5892	86.8382	87.2121
4	85.8671	72.266	78.4816
5	95.8634	86.6538	91.0262
6	67.3571	81.6947	73.8363
7	85.1381	82.3058	83.698
8	88.4162	79.7633	83.8672
Nina 1	84.9337	94.9153	89.6475
2	88.3989	98.8686	93.3411
3	84.5993	97.8251	90.7327
4	80.6079	96.036	87.6482
5	89.1912	96.6732	92.7816
6	87.4782	88.4106	87.942
7	85.7601	84.656	85.2045

8	86.3433	98.4747	92.0109
SU 1	91.9588	63.9539	75.4413
2	77.8276	99.114	87.1904
3	76.8954	98.5046	86.3689
4	79.3277	88.293	83.5706
5	89.2563	95.3456	92.2005
6	58.8293	87.2936	70.2891
7	78.0747	96.704	86.3965
8	80.1811	98.2974	88.3198

HDIBCO12	Recall	Precision	FMeasure
Howe 1	97.3289	97.8537	97.5906
2	65.4087	96.5806	77.9954
3	83.219	96.0301	89.1667
4	92.7382	97.0159	94.8289
5	97.6825	94.5194	96.0749
6	96.1459	94.0467	95.0847
7	85.8526	92.8843	89.2302
8	97.5318	96.0988	96.81
9	96.0185	97.1268	96.5695
10	97.2674	97.7241	97.4952
11	94.2983	95.7061	94.997
12	96.4812	95.6594	96.0685
13	90.0844	96.096	92.9932
14	97.6107	96.9886	97.2986
LU 1	95.0556	90.089	92.5057
2	68.4208	91.5807	78.3246
3	0	0	0
4	88.3013	94.1577	91.1355
5	96.9292	82.0285	88.8585
6	96.7602	78.5157	86.6884
7	88.8866	90.3222	59.5986
8	97.7652	89.2223	93.2986
9	85.9397	97.0308	91.1491
10	89.961	96.1081	92.933
11	90.8993	89.8816	90.3876
12	93.58	91.5589	92.5584
13	87.7077	77.9772	82.5567
14	5.6914	6.7572	6.1787
Nina 1	73.5879	98.1715	84.1204
2	72.4464	96.8108	82.875
3	85.1452	96.5827	90.5041
4	84.678	97.2826	90.5437
5	91.1706	92.8871	92.0209
6	90.4315	88.1579	89.2802
7	81.7177	93.8222	87.3526

8	92.1477	95.7858	93.9315
9	86.0432	99.2642	92.1821
10	88.1715	98.3316	92.9748
11	86.651	96.5517	91.3338
12	89.861	95.942	92.8016
13	83.7641	96.3176	89.6033
14	88.1019	98.7641	93.1288