# BinarizationShop: A User-Assisted Software Suite for Converting Old Documents to Black-and-White

Fanbo Deng Zheng Wu Zheng Lu Michael S. Brown
School of Computing
National University of Singapore
{dfanbo, wuz, luzheng, brown}@comp.nus.edu.sg

## ABSTRACT

Converting a scanned document to a binary format (black and white) is a key step in the digitization process. While many existing binarization algorithms operate robustly for well-kept documents, these algorithms often produce less than satisfactory results when applied to old documents, especially those degraded with stains and other discolorations. For these challenging documents, user assistance can be advantageous in directing the binarization procedure. Many existing algorithms, however, are poorly designed to incorporate user assistance. In this paper, we discuss a software framework, *BinarizationShop*, that combines a series of binarization approaches that have been tailored to exploit user assistance. This framework provides a practical approach for converting difficult documents to black and white.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; J.m [**Computer Applications**]: Miscellaneous

## General Terms

Algorithms, Human Factors, Design

## Keywords

Binarization, document processing, user-assisted software

## 1. INTRODUCTION AND RELATED WORK

Binarization is a crucial step in document imaging and a large body of literature is available on this topic (e.g. [3, 5, 6, 7, 9, 10, 11]). While simple binarization approaches (e.g. [5, 10]) can be used in a fully automated manner for documents that have uniform appearance and little noise, for older documents with discolored backgrounds, stains, and biological damage, more sophisticated "adaptive thresholding" algorithms are required (e.g. [3, 6, 9, 11]).

For difficult documents, even these more sophisticated adaptive thresholding techniques are unable to be automated and require some type of user assistance to help guide the binarization process. The vast majority of such methods, however, are poorly designed to effectively exploit user guidance. For example, the approaches in [3, 6, 9, 11] allow the user to adjust algorithmic parameters, but are still applied in a global fashion thus producing suboptimal results on documents with non-uniform image characteristics. The only exception that we are aware of is the recent work by Lu et al. [7] that provided a parameter-free binarization approach where the user was required only to denote regions in the image that were binarized incorrectly. These user specified regions could then be further processed in a local manner. While this approach helps improve results for many documents, this "lazy-evaluation" style may not be the preferred way to perform binarization by all users.

The impetus of this work was to develop a comprehensive software for user-guided binarization of old documents. To this end, we have built a software suite that combines several effective binarization approaches that can be used in a flexible manner. In particular, our software, *BinarizationShop*, is based on three different methods: the parameter-tuning adaptive thresholding method by Sauvola and Pietikainen [11], the interactive thresholding method by Lu et al. [7], and a new example-based method that uses previous binarization results on documents with similar image characteristic. For the parameter-tuning and example-based methods, our software has been designed to allow for local processing, thus allowing all three methods to be used in a global or local manner. For completeness, we also provide manual cleanup tools for final touch-up as well as a blending tool to help verify results.

While the idea of utilizing user assistance is not unique in digital library applications [1, 4, 8, 12, 13], we are unaware of any document binarization technique similar to our software. We believe this strategy of coupling several binarization techniques with local and global processing facilities provides a practical tool for use in the digitization of old documents. The remainder of this paper overviews our BinarizationShop software (Section 2) and provides a short discussion (Section 3).

## 2. BINARIZATIONSHOP

This section first overviews the BinarizationShop framework and its workflow, followed by details to the individual binarization approaches and additional features.

### 2.1 Software Overview

Fig. 1 shows the workflow of the BinarizationShop soft-


*JCDL'10,* June 21–25, 2010, Gold Coast, Queensland, Australia.

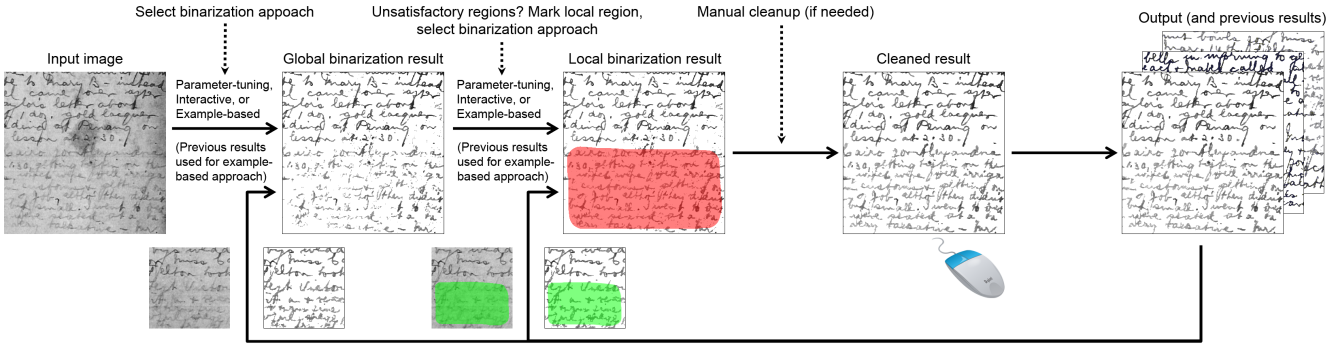



**Figure 1: An overall view of our system's workflow, in the sequence of image loading, global and local binarization using one of three methods, manual editing and output.**

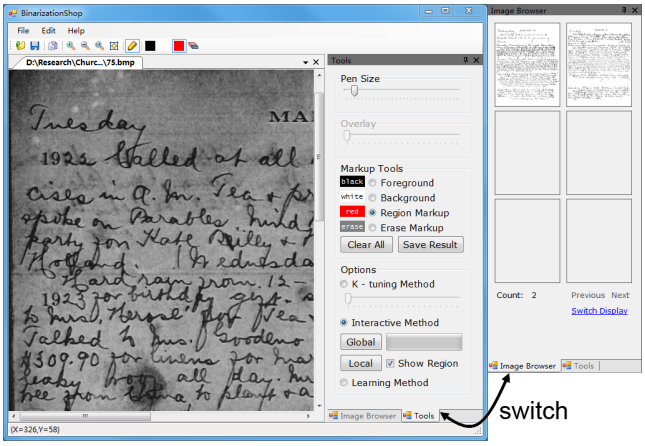



**Figure 2: Interface of BinarizationShop, an image-based GUI with markup and binarization tools on the control panel, together with the result browser.**

ware. With a document loaded, the user first chooses one of the three binarization methods to generate an initial global binary result. A blending tool allows the results to be overlayed with adjustable transparency with the original image to help visually verify the results. For documents with regions where the global result is unsatisfactory, the user can specify the unsatisfactory regions by drawing on the image to denote where to perform local binarization. The user can manually touch-up the results (if necessary) to produce the final results. As binarized documents are produced they are made available to be used in the example-based approach. Our software ranks previous results' similarity to a newly loaded document based on image characteristics (i.e. histogram similarity).

Fig. 2 shows a screen shot of our software's interface. The interface is a typical image-editing GUI that allows an image to be loaded and supports standard manipulation, such as zoom in, zoom out, and panning. Markup tools and binarization methods are provided in the *tools* tab of the control panel on the right side of the interface. The user can choose the desired method by toggling the corresponding option. The final binarized results are maintained in the *image browser* tab of the control panel.

## 2.2 Global/Local Parameter-Tuning

One available approach in the BinarizationShop software is the parameter-tuning method [11], that uses the equation

$$T(x,y) = m(x,y) \cdot (1 - k \cdot (1 - s(x,y)/R)), \qquad (1)$$

to decide the threshold $T(x,y)$ for each image pixel $(x,y)$ with intensity $I(x,y)$. This approach uses the local mean $m(x,y)$ and the local standard deviation $s(x,y)$ of the intensities inside a window centered at $(x,y)$[1]. The term $R$ is fixed to 128 for $0-255$ grayscale values. The term $k$ is a positive parameter that needs be tuned. A slider-tool is provided on the control panel for the user to try different $k$ values and evaluate the result. The procedure can be performed quickly ($\approx 2s$) to provide quick visual feedback.

For well-kept documents this method typically produces satisfactory results for the whole image, while for difficult documents (as shown in Fig. 3) local processing may be necessary. For local $k$-tuning, our software provides a *region markup* tool (red brush) for the user to specify the region in need of a better $k$. By re-adjusting the $k$ value for the selected region, the local result can be improved.

## 2.3 Interactive Binarization

The interactive method [7] is based on the following thresholding formula adaptive to the local mean:

$$T(x,y) = m(x,y) - s_b, \qquad (2)$$

where $s_b$ is the estimated standard deviation of the background of the entire document or a specified local region. When using this method, the formula with the globally-estimated $s_b$ is first applied. An initial result with some erroneous regions may be obtained, as the example shown in the Fig. 3. For local processing, the user needs to only roughly indicate the region that is incorrect, by drawing coarse markup. The region is automatically extracted using an Markov Random Field segmentation technique (see [7] for more details), and highlighted in yellow if the user chooses to display it. The region is then re-binarized using Eq. 2 with the locally-estimated $s_b$. The software provides *global* and *local* buttons on the control panel to run the corresponding operations.

[1]The local window should cover at least 1–2 written characters. For our documents, the window size is fixed to $31 \times 31$ for images of 150 dpi resolution. The window size for higher-resolution images is adjusted using the same ratio.

Compared with the parameter-tuning method, the interactive approach is easier to operate because it does not require a "try and see" procedure as need with parameter-tuning. However, the performance of this method depends on the estimation of $s_b$, which may be less accurate than a result that can obtained with careful global and local parameter-tuning.

## 2.4 Global/Local Example Learning

Digitization typically involves imaging a series of documents from the same source (e.g. a book or a diary). As more and more binary images are generated, a set of training-data is obtained. The example-based approach exploits this availability of examples. The approach works as follows. For each pixel in a new image, we compute a feature vector with the following five (5) features: the pixel's intensity, local intensity mean, local intensity standard deviation, local mean of the image gradient magnitude, and the image local contrast (the difference between the maximum and minimum intensity values inside a local window). The user chooses a similar image from the previous results as an example. For all pixels in this example image, we also compute the same five (5) features. To decide how to binarize the input image, a KNN classifier [2] is used as follows: for each pixel in the new image, we search for the $K$ nearest pixels in the training-examples based on the Euclidean distance between the input and examples feature vectors. In our implementation we use $K = 9$. A pixel in the input image is colored either black or white based on the majority color of its $K$ nearest neighbors (e.g. if the majority of the $K$ nearest neighbors are white, the input pixel is colored white).

We sort the example images selected by the user based on how similar they are to the input image using histogram ranking, as shown in the third row of Fig. 3. In the result browser, the candidate examples are displayed in the similarity-decreasing order, with the similarity score defined as the Euclidean distance between the intensity histograms of two images. With the ranking order as reference, the user can select the desired example image by dragging it onto the working image. A global result is then generated by KNN, with the training data from the entire example image.

This method can also be locally implemented. An example is shown the fourth row of Fig. 3. The user needs to separately markup the local regions in the example image (using a green brush tool), as well as in the working image (using a red brush tool). Once the user drags the selected example region onto the working region, a local result is generated, with the training data limited to the example region. As the pixels in two local regions share higher similarity, a better result can be obtained.

## 2.5 Manual Editing

Although the provided global and local tools can correctly binarize the majority of the degraded document, errors are unavoidable, especially for difficult regions. To correct errors, *foreground*, *background* and *erase* brushes are provided. The *foreground* brush allows the user to label any pixel as foreground. Conversely, the *background* brush is used to remove any pixel mistakenly binarized as foreground. At last, the *eraser* brush helps remove any modification made with the other brush.

# 3. SUMMARY

We have presented a comprehensive software for converting scanned documents to black and white. Our approach is based loosely after commercial softwares, such as Photoshop, and provides a set of tools at the user's disposal. This allows flexibility and personal preference as how to best perform the binarization task based on a particular input. Since binarization is often performed on a series of similar documents, we have also incorporated an example-based approach that can leverage previous results. One fundamental idea in our approach is the ability to process parts of the document locally to achieve a better result. We believe this software framework provides a valuable tool for use in the digitization of old documents.

# 4. REFERENCES


[1] A. Dekhtyar, I. E. Iacob, J. Jaromczyk, K. Kiernan, N. Moore, and C. Porter. Building image-based electronic editions using the edition production technology. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2005.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[3] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39:317–327, 2006.

[4] Y. Huang and M. S. Brown. User-assisted ink-bleed correction for handwritten documents. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2008.

[5] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong. A new method for graylevel picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29:273–285, 1985.

[6] C. G. Leedham, C. Yan, K. Takru, J. H. N. Tan, and L. Mian. Comparison of some thresholding algorithms for text/background segmentation in difficult document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

[7] Z. Lu, Z. Wu, and M. S. Brown. Interactive degraded document binarization: An example (and case) for interactive computer vision. In *IEEE Workshop on Application of Computer Vision (WACV)*, 2009.

[8] C. Monroy, R. Furuta, and G. Stringer. Digital donne: workflow, editing tools, and the reader.s interface of a collection of 17th-century english poetry. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2007.

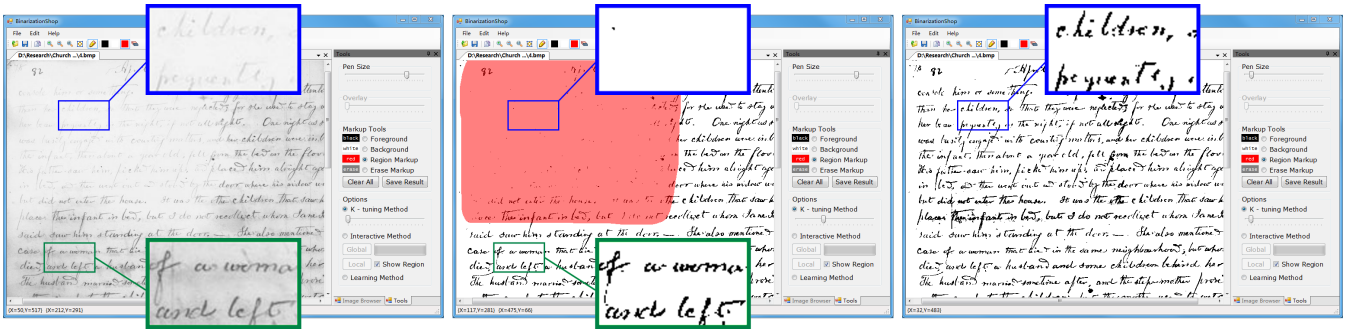
[9] W. Niblack. *An Introduction to Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[10] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66, 1979.

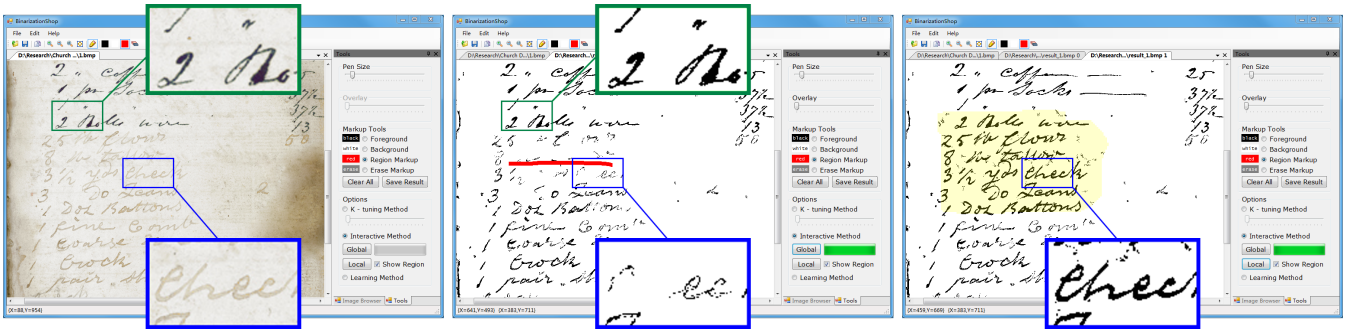
[11] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[12] W. B. Seales and Y. Lin. Digital restoration using volumetric scanning. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2004.

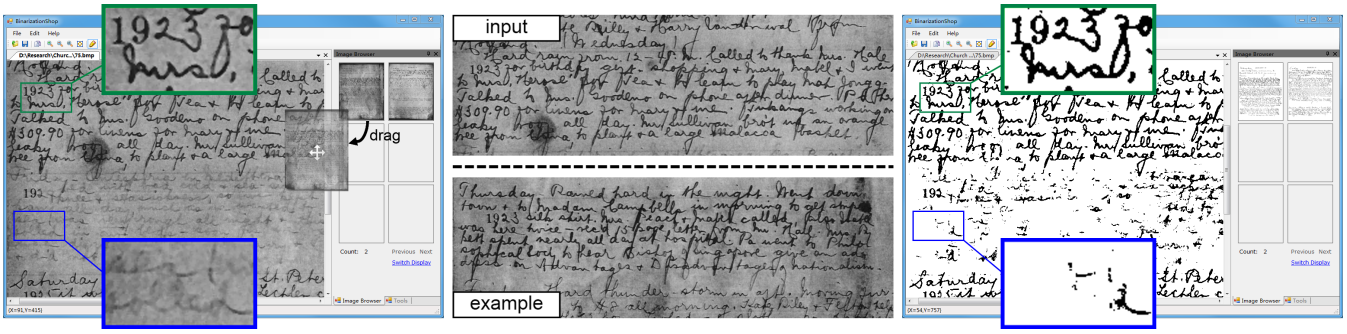
[13] B. Wingenroth, M. Patton, and T. DiLauro. Enhancing access to the levy sheet music collection. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2002.



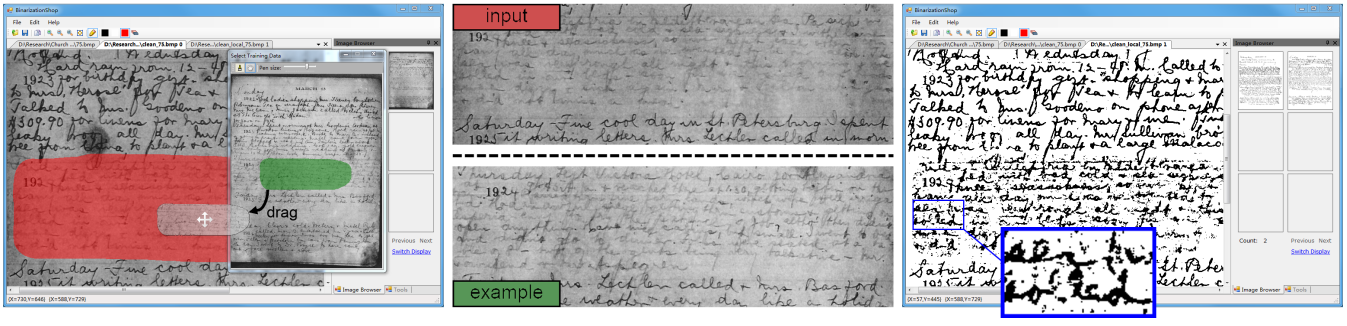
[**Global/Local Parameter-Tuning**] The first image shows the input document with two zoomed regions that exhibit different characteristics. The second image shows the results obtained by tuning the adaptive thresholding parameter globally. The user can highlight the region that should be processed locally. The third image shows the local result.



[**Interactive Thresholding**] The first image shows the input document with two zoomed regions that exhibit different characteristics. The second image shows the results obtained automatically. The result is not satisfactory in the middle of the region. The user can draw a line to denote this region. The third image shows the complete automatically-extracted region and the local result.




[**Global Example-Based**] The first image shows the user selecting an example from a set of previously cleaned results to process the new input. The second image shows a comparison of the example and input image. The third image shows the global results, note that one part of the image is not satisfactory.




[**Local Example-Based**] The first image shows the user selecting only a local region of the example and applying to a local region in the input document. The second image shows a comparison of the example and input image. An improved result is shown in the third image.

**Figure 3: BinarizationShop provides various approaches to achieve a binarized image. The approaches are demonstrated here.**