A Watershed-Based Thresholding Approach for Image Binarization

Andrei Tigora

Abstract— In this paper a watershed image segmentation combined with a thresholding technique is used to produce a parameter-free binarization with very good preservation of fine details, whilst maintaining a reasonable signal-to-noise ratio.

Keywords—image binarization, image processing, image segmentation, local thresholding, Niblack, Otsu, watershed

I. Introduction

With the increasing availability in photographing and scanning devices also comes an increase in the need for processing the generated images. Whether it is identifying family members in photos for cataloging purposes, or extracting characters from a scanned image or even a poor camera shot, [1] everybody wants some kind of image processing done.

One important element of the processing chain that many of the above mentioned tasks require is binarization [2]. Though fairly trivial when performed on “good” input images, binarization can prove to be quite a challenge when performed on images that are not exactly test-lab reference. One of the most common usages for binarized images is the automatic “content-conversion system”. For these systems, in order to process large image databases, a robust and parameter free binarization algorithm is a must [1], [3]. Also the used algorithm must strike a good balance between local and global image processing because of the large input images and their increasing heterogeneity in both foreground contrast and background texture. Methods for background reconstruction in order to distinguish the foreground are useful for statistics estimation[1] and filtering is usually used in conjunction with binarization[4].

II. Background

The current paper is based on the works of Fernand Meyer and Nobuyuki Otsu, combining two very popular algorithms in image processing: *watershed* and *Otsu thresholding*.

Over the years, several segmentation algorithms have been developed [5], but none are perfect; segmentation is important in that it identifies regions of importance in the image, based on certain characteristics deemed relevant for the processing. Lately, the watershed algorithm has received a lot of attention, being used in a wide range of domains [6]-[9].

The watershed algorithm was introduced by Fernand Meyer in 1991 [10] and it is widely used for image segmentation, due to its innate simplicity. The grayscale image can be viewed as a topological relief, which, in a heavy rain scenario, is flooded, starting from the local minimum areas. As the water levels rise, pools tend to merge, but instead of letting such a thing happen, barriers are constructed to keep the original pools separated. The resulting infrastructure of barriers represents a watershed by flooding, while the areas determined by them become segments of the image.

Otsu’s method [11] is a highly used algorithm that has spawned numerous variants and adaptations [12], [13], due to its robustness and high speed. It is a threshold based binarization algorithm that aims to maximize the inter-class variance (or minimize the intra-class variance). The inter-class variance σ_b^2 is defined as follows:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{1}$$

where $\omega_x(t)$ represents the class probability and $\mu_x(t)$ represents the class mean. These two are given by the following formulas:

$$\omega_x(t) = \sum_{i=0}^{t} c(i) \tag{2}$$

$$\mu_x(t) = \sum_{i=0}^{t} c(i)i \tag{3}$$

with $c(i)$ representing the number of pixels in the image whose value is i. Intuitively, equation (1) tries to separate the means of the cluster, while also keeping each cluster with a high probability of occurrence.

The aim of Otsu’s method is to find the threshold t^* such that:

$$t^* = \arg\max_t \sigma_b^2(t) \tag{4}$$

An exhaustive search is employed. For multiple optimal thresholds the value taken into consideration is the mean value. This is very important for the quality of the binarization after the post-processing thresholds stage.

III. Algorithm Description

The watershed binarization algorithm aims to improve upon the original Otsu method by computing several localized thresholds instead of a single global value [14]. The algorithm has several distinct steps that are described in the following paragraphs.

Step 0: Preparing the image

The process begins by smoothing the original image using a bilateral filter [15]. This kind of filter reduces noise and still manages to preserve the edges, making it a perfect choice for the algorithm [16].The algorithm is run on the resulting image.

Step 1: Detecting local minimum areas

A local minimum area (named from now on LMA) is defined as a connected area in the page with all pixels having the same value, surrounded in all directions by pixels with higher grayscale values. Such an area can be either a single pixel, or a group of pixels.

In the following, for clarity reasons, the input image is considered to be a grayscale one, represented using 8-BPP unsigned values (every pixel is represented as an integer value in the $0-255$ range).

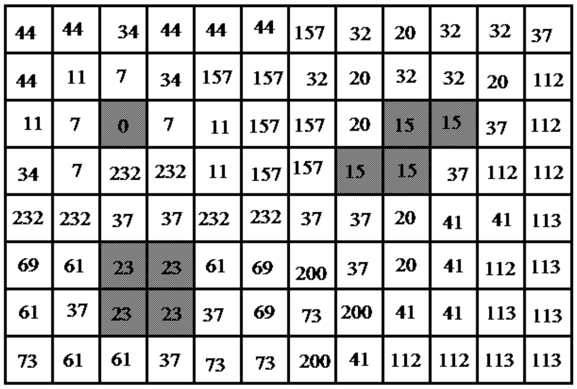
The basic idea behind the entire processing is to split the original image into segments for which a unique threshold is relevant. It makes sense to have a unique threshold in areas with a “2D monotony” like watershed areas. This is because, when you think of a binarization by performing thresholding as a cutting in two (up and down) pieces, if a simple function has a certain monotony, a single cut (an unique threshold) is enough. The purpose of determining a local minimum area (LMA) is to generate such a unique threshold area, an equivalent of a “2D monotony” area for which only one threshold is necessary. Splitting an image into LMA-s means in fact allocating areas of local thresholding all over the input image.

Determining a LMA is done by repeatedly applying a global threshold on the original document, for every value in the range $0-255$. A LMA will be a connected area in the intermediary conversion (at a specific threshold level) which fulfills the following rule: **none of the pixels belonging to it were classified as black in the previous intermediary conversion*. For example, if the global conversion is performed at the threshold level of 30, a LMA will be any connected area of black pixels having the corresponding grayscale value of 30 and which is not surrounded by any pixel with lower values than 30. If at least one such pixel existed, that particular pixel would become part of the connected area, and the previously defined rule would not be fulfilled.

Step 2: Extending the LMA-s into segments

A segment is as a LMA that has been extended with pixels that do not respect the LMA definition; each such segment (originating in a LMA) is uniquely identified by a numerical value. Using the intermediary global threshold conversions, the previously unclassified pixels in the image are assigned to one of the segments as follows:

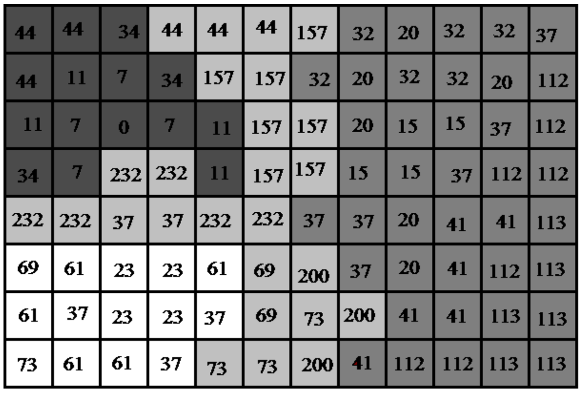
- if an area (adjacent equal value pixels) connects with a single segment, then all the pixels belonging to that area are assigned to that particular segment;
- if an area connects with at least two different segments, it is classified as a *boundary area* and will be processed in the following step.



44	44	34	44	44	44	157	32	20	32	32	37
44	11	7	34	157	157	32	20	32	32	20	112
11	7	0	7	11	157	157	20	15	15	37	112
34	7	232	232	11	157	157	15	15	37	112	112
232	232	37	37	232	232	37	37	20	41	41	113
69	61	23	23	61	69	200	37	20	41	112	113
61	37	23	23	37	69	73	200	41	41	113	113
73	61	61	37	73	73	200	41	112	112	113	113

Fig. 1 Image pixel values and the identified LMA-s represented with gray

By the end of this step, the pixels in the original image will either be assigned to one of the segments, or be part of the boundary area.



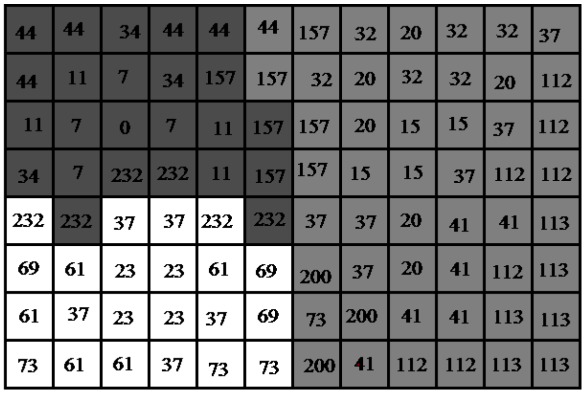
44	44	34	44	44	44	157	32	20	32	32	37
44	11	7	34	157	157	32	20	32	32	20	112
11	7	0	7	11	157	157	20	15	15	37	112
34	7	232	232	11	157	157	15	15	37	112	112
232	232	37	37	232	232	37	37	20	41	41	113
69	61	23	23	61	69	200	37	20	41	112	113
61	37	23	23	37	69	73	200	41	41	113	113
73	61	61	37	73	73	200	41	112	112	113	113

Fig. 2 The unassigned pixels act a s a boundary separating the three segments (bottom left, top left and right)

Step 3: Assigning boundary pixels

In order to finish the segmentation process, boundary pixels must also be assigned to one of the segments as well. Given a pixel of coordinates (X,Y) that is part of the boundary area, the closest classified pixels are determined using the Manhattan distance. Following this procedure, there are two possible situations:

- there is a unique closest classified pixel at coordinates (A,B); in this case, the previously unclassified (X,Y) pixel is assigned to the segment that contains (A,B);
- there are a number of classified pixels (A_1,B_1), (A_2,B_2), ..., (A_n,B_n), each situated closest to (X,Y); in this case the pixel is randomly assigned to the segment to which $(A_i,B_i), i=\overline{1,n}$, belongs.



44	44	34	44	44	44	157	32	20	32	32	37
44	11	7	34	157	157	32	20	32	32	20	112
11	7	0	7	11	157	157	20	15	15	37	112
34	7	232	232	11	157	157	15	15	37	112	112
232	232	37	37	232	232	37	37	20	41	41	113
69	61	23	23	61	69	200	37	20	41	112	113
61	37	23	23	37	69	73	200	41	41	113	113
73	61	61	37	73	73	200	41	112	112	113	113

Fig. 3 Extended segments with assigned boundary pixels

Step 4: Computing local thresholds

For each one of previously determined segments, an individual local threshold is computed using Otsu's method. If there are several thresholds that satisfy the maximum inter-class variance criteria, the mean of these values is used instead.

Step 5: Correction

As no image comes without noise, despite the smoothing performed in the initial stages, it is essential that its impact is diminished. To do that, a height map of thresholds is constructed and afterwards adjusted using surface approximation algorithms in order to correct the threshold level of each segment. This adjustment takes into account both the area of the segment and threshold values of neighboring segments. To keep things simple for the current implementation, the smoothing was done using a Gaussian kernel [17] G with size equal to a tenth of the image diagonal, and the standard deviation equal to its size.

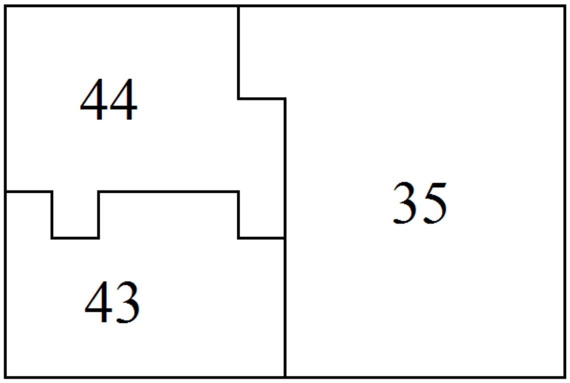


Fig. 4 Individual Otsu thresholds for the determined segments

Step 6: Local conversion

Each segment is converted to grayscale using the previously determined thresholds.

Step 7: Final conversion

The final conversion is obtained by putting together all the converted segments.

IV. Results

Obtaining a good segmentation of the image through the watershed method is the key to the success of the algorithm. If the original image is processed without any prior intervention, too many segments are obtained, as proven by Fig. 5; the high number of fragments determines a very "sharp" binarization, which is highly sensitive to noise.

Instead, a blurred version of the image is preferred for segmentation, to eliminate some of the artifacts introduced in the image creation process as seen in Fig. 5. However, the blur should not be too aggressive, otherwise the image becomes too bland, losing its features, which in turn results in too few segments being generated (see Fig. 6).

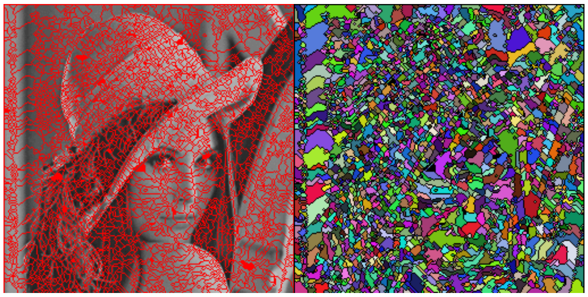


Fig. 5 Watershed segmentation for unprocessed Lena image: contours only (left) and segments (right)

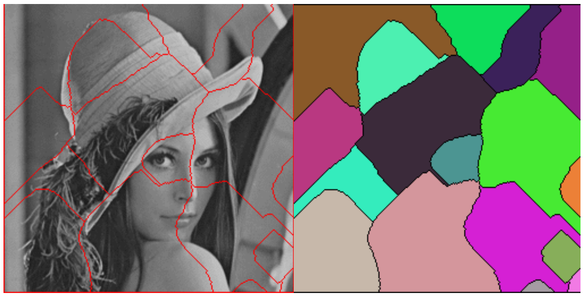


Fig. 6 Watershed segmentation of over blurred Lena image: contours only (left) and segments (right)

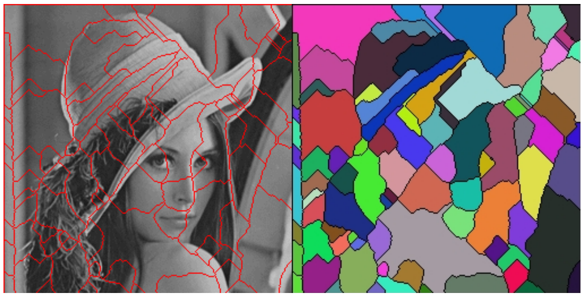
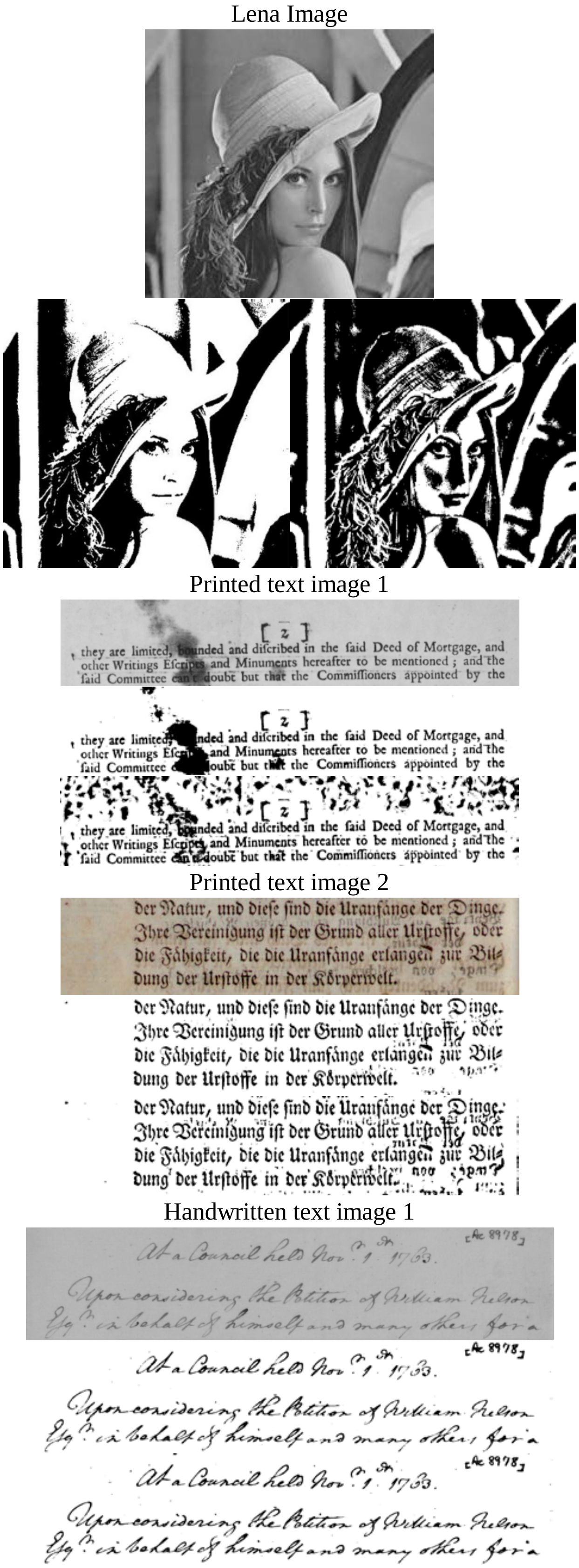


Fig. 7 "Ideally" watershed segmented Lena image: contours only (left) and segments (right)



Lena Image

Printed text image 1

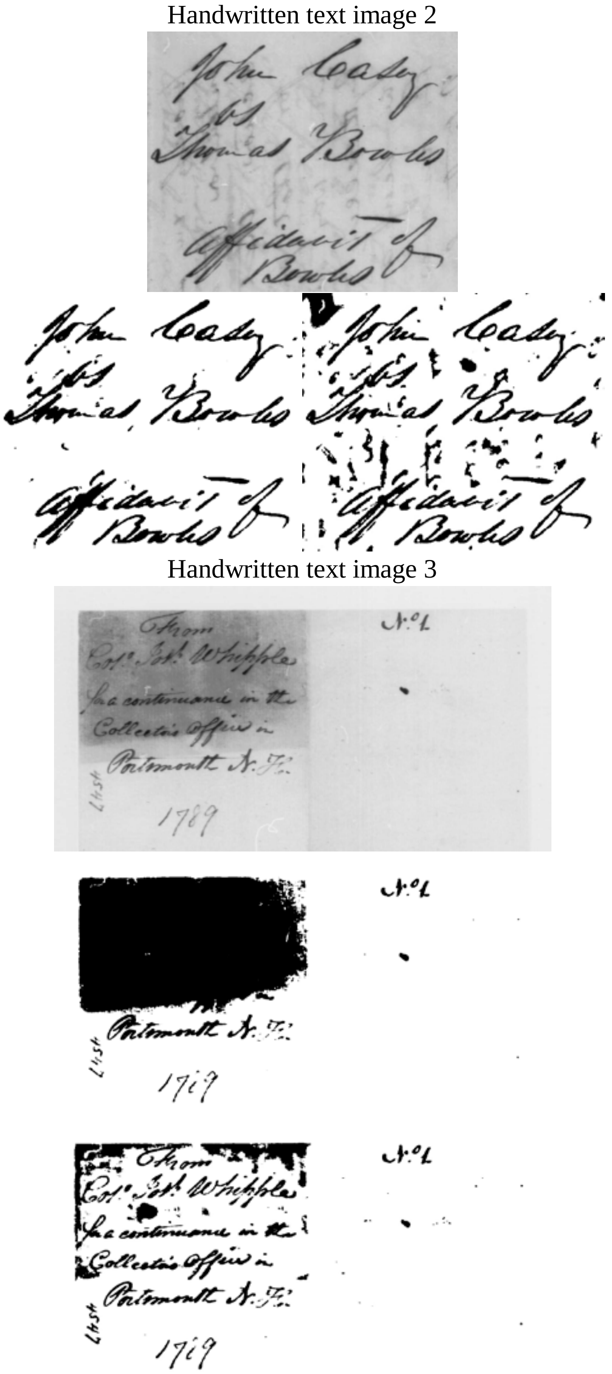
[2]
they are limited, bounded and diſcribed in the ſaid Deed of Mortgage, and other Writings Eſcripts and Minuments hereafter to be mentioned; and the ſaid Committee can't doubt but that the Commiſſioners appointed by the

Printed text image 2

der Natur, und dieſe ſind die Uranfänge der Dinge. Ihre Vereinigung iſt der Grund aller Urſtoffe, oder die Fähigkeit, die die Uranfänge erlangen zur Bildung der Urſtoffe in der Körperwelt.

Handwritten text image 1

At a Council held Nov.r 1st 1783. Ac 8978
Upon considering the Petition of William Nelson Esq.r in behalf of himself and many others for a



Handwritten text image 2

John Cady
vs
Thomas Bowles
Affidavit of
Bowles

Handwritten text image 3

From
Col. Jos. Whipple
for a continuance in the
Collector's Office in
Portsmouth N.H.
1789
N.o 1

Fig. 8 Original images followed by simple Otsu and the proposed algorithm

The full algorithm was tested on several images containing printed and handwritten text, as well as some photographs in order to evaluate its potential in different situations. Otsu's algorithm was also run on the same input to provide a reasonable reference for the results.

Each of the following sections presents the original image, followed by the Otsu binarization and finally the watershed binarization.

As the previous images indicate, the output of the algorithm varies greatly in its quality compared to Otsu. For those images that hardly have any noise, the outputs are almost identical; on the other hand, for more complex images, while Otsu tends to create huge areas of identical pixels, the watershed creates more nuanced results, revealing even some hardly noticeable features, that may or not be of interest to the viewer.

However, even on those occasions when watershed appears to be introducing more noise than Otsu, the quality of certain regions of the watershed binarized image is definitely better, thus allowing additional software systems like OCR-s to extract more valid information.

The obvious downside of watershed is that it detects black pixels even for those regions that should be white, based solely on slight pixel differences. Such a situation occurs for the top left corner of the second handwritten image. Although the region would be considered white by a human reader, without giving it much thought, on closer inspection it becomes obvious that the region is a bit darker than the surrounding area. Ideally, this should not happen, because the region is way too white to be cataloged as black.

On the other hand, if the entire pixel group were evaluated as being white, many features that appear in the Lena image would probably disappear as well.

V. Conclusion and Future Work

The main advantage of this conversion method is the fact that it computes a local threshold for each segment of the document. It tends to reveal many hidden features of the original image, which may or not be desired in the final binarized image.

Comparing with well-known local binarization methods like Niblack [18] and Sauvola [19] the presented approach has the advantages of both better quality and parameter-free approach, since all of the localized thresholding algorithms are depending on the window size input parameter, which cannot be computed or estimated other than by primitive heuristics.

The current research concentrated on evaluating the algorithm performance on images obtained with highly specialized devices, such as professional cameras or document scanners. However, most images are created with low end gadgets in less than ideal conditions by untrained users [20]. As proven in previous research [21], [22], adaptive binarization can yield better results for a wide range of inputs than the fixed parameter counter parts. It would be worthwhile to evaluate the algorithm's performance with this kind of input and compare with similar approaches.

Future work will be aimed at deciding upon the best way to estimate the threshold surface in the Correction stage. Also, it would be interesting to evaluate whether incorporating a global threshold could somehow limit the wrongful labeling of some obvious white regions as black. A proposed method for incorporating globalization is to use a pyramid of scales, each binarized locally with the same dimension window. The process is explained in Fig. 9. In the current implementation, the threshold for a region is constant, and local to the region. In order to include a global influence, the threshold for a specific pixel will be a weighted sum between the local calculated threshold and the one obtained from the smaller resolution. In the image, we will want to raise or lower the

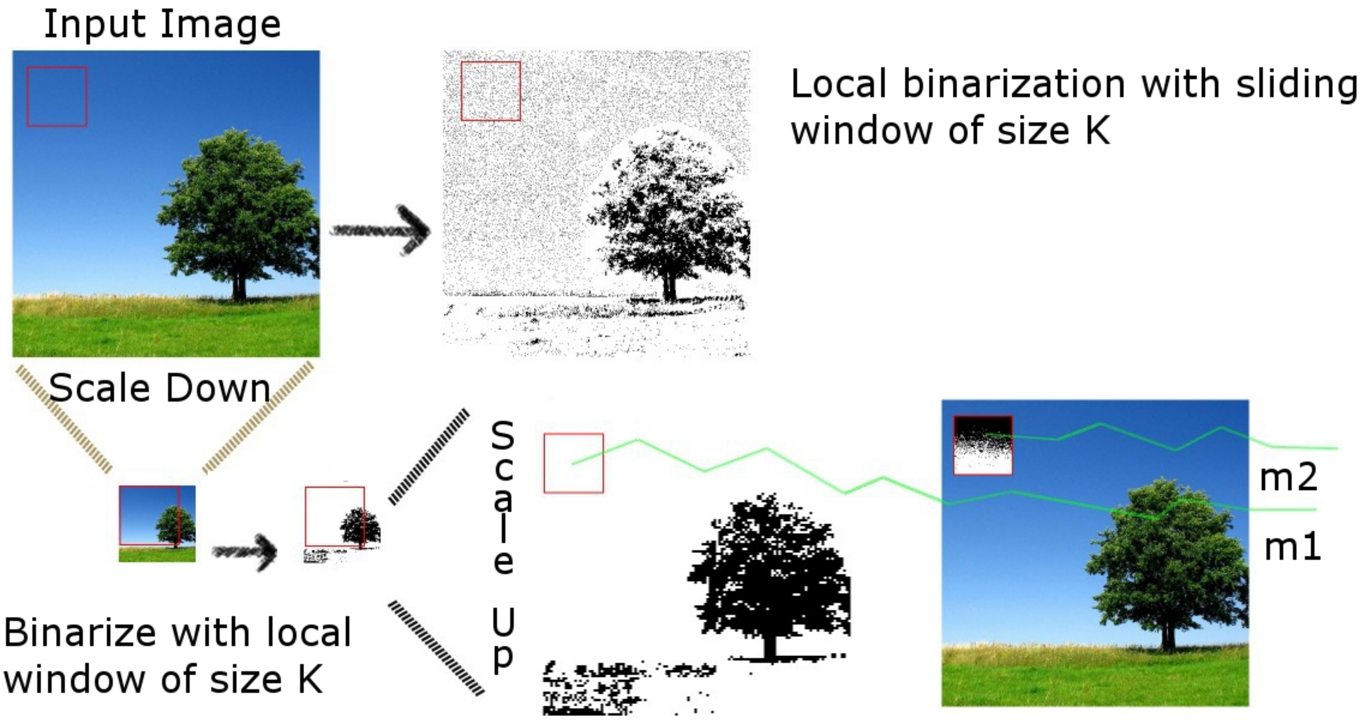

Input Image
Local binarization with sliding window of size K
Scale Down
Scale Up
Binarize with local window of size K
m2
m1

Fig. 9 Exemplification of the pyramidal binarization method

threshold for the final image, in order to get the mean value m2 closer to the mean value from the lower resolution, m1, with a certain factor. The technique is not limited to using only one other scale, like exemplified, but a number of scales and weights for each scale can be used. The significance is that of the impact larger and larger localities will have on the local pixel.

Another future work direction will be to generalize the LMA-s to act more as localities for a certain point and not as a segment for an image. In other words the generalized LMA-s will be overlapping regions (many points are sharing the same LMA). This approach is more general and also more intuitive, acting more as a local image processing. Thus local statistic computed on generalized LMA-s may be successfully combined with global statistics (like weighting local and global Otsu threshold), allowing us to take the best advantages from both worlds with basically no drawbacks. One basic idea to compute the generalized LMA-s is to construct an image-blur stack: every stack level is a different blur level (in increasing order from the original, non-blurred image to an infinite-radius blur, single-shade image level). Using LMA-s on every level and identifying a confidence measure for every LMA (for example one good confidence measure is the color extent in the grayscale space), may enable us to select for each pixel the "best LMA" as being the LMA on every stack level that contains the pixel inside and has the best confidence. This definition of the "generalized LMA" will automatically resolve almost every problem with the current approach, but further study should be carried so that a successful detection of the "generalized LMA-s" may be done in a reasonable amount of time and the technique described above may become usable.

The usage of the "generalized LMA" as described above will remove the problems (related to over-sensitivity to color in the intra-cluster space) in the resulted binarizations presented in Fig. 10.

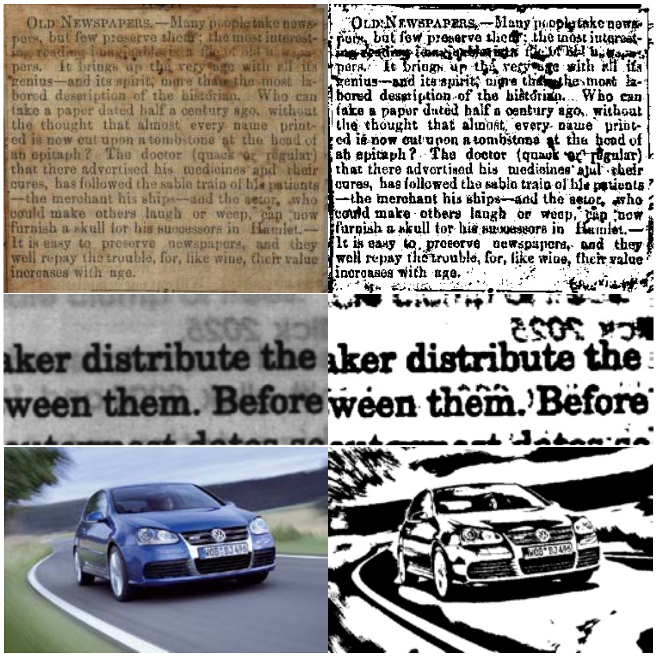


Fig. 10 Watershed segmentation results (right)

Other work may be directed at finding segmentation alternatives to the watershed algorithm, perhaps color or texture based [4]. Perhaps watershed itself could be used slightly differently: first compute the boundaries with a boundary detection algorithm, then threshold the boundaries, compute a distance transform and finally applying watershed on the resulting image. This should perform well for scanned printed images.

REFERENCES

[1] C. A. Boiangiu, A. I. Dvornic, D. C. Cananau, "Binarization for digitization projects using hybrid foreground-reconstruction," in *Proc. 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, August 27-29, 2009, pp.141-144.

[2] C. A. Boiangiu, A. I. Dvornic. "Methods of bitonal image conversion for modern and classic documents," *WSEAS Transactions on Computers*, vol. 7, no. 7, pp. 1081 – 1090.

[3] C. A. Boiangiu, A. V. Stefanescu, D. Rosner, N. Tapus, M. Andreica, "Local thresholding algorithm based on variable window size statistics," in *Proc. CSCS-18*, *The 18-th International Conference on Control Systems and Computer Science*, vol. 2, 2011, pp. 647-652.

[4] C. A. Boiangiu, A. I. Dvornic. "Bitonal image creation for automatic content conversion," in *Proc. 9th WSEAS International Conference on Automation and Information*, WSEAS Press, pp. 454 – 459.

[5] C. Gonzalez, R. E. Woods, *Digital image processing* (Third Edition), Prentice Hall, August 31, 2007

[6] A. Mustaqeem, A. Javed, T. Fatima, "An efficient brain tumor detection algorithm using watershed & thresholding based segmentation," *International Journal of Image, Graphics and Signal Processing*, vol. 4, no. 10, 2012, pp. 34-39.

[7] A. Pinaki, P. Acharjya, D. Ghoshal, "An effective human fingerprint segmentation method using watershed algorithm,"*International Journal of Computer Applications*, vol. 53, no. 13, 2012, pp. 23-26.

[8] A. B. Huguet, M. C. de Andrade, R. L. Carceroni, A. A. Araujo, "Color-based watershed segmentation of low-altitude aerial images," *Proceedings of 17th Brazilian Symposium on* Computer Graphics and Image Processing, 2004, pp. 138-145.

[9] S. Lingala, M. Adjouadi, M. Mourad, N. Rishe, "Modified watershed transform without gradient," *4th WSEAS International Conference on Electronics, Control and Signal Processing, Miami*, Florida, USA, 17-19 November, 2005, pp.93-98.

[10] F. Meyer, "Un algorithme optimal pour la ligne de partage des eaux," *8^{me} congrès de reconnaissance des formes et intelligence artificielle*, vol. 2, Lyon, France, 1991, pp. 847-857.

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

[12] S. Bag, P. Bhowmick, P. Behera, G. Harit, "Robust binarization of degraded documents using adaptive-cum-interpolative thresholding in a multi-scale framework," *International Conference on Image Information Processing* (ICIIP), 2011, pp. 1-6.

[13] L. Jianzhuang, L. Wenqing, T. Yupeng, "Automatic thresholding of gray-level pictures using two-dimension Otsu method," in *Proc. 1991 International Conference on Circuits and Systems*, 1991, pp. 325-327.

[14] A. Tigora, “An image binarization algorithm using watershed-based local thresholding”, Proceedings of 1st International Conference on Image Processing and Pattern Recognition (IPPR '13) Budapest, Hungary, December 10-12, 2013, Volume “Recent Advances in Image, Audio and Signal Processing”, pp. 154-160

[15] K.N. Chaudhury, D. Sage, M. Unser, “Fast O(1) bilateral filtering using trigonometric range kernels,” *IEEE Transactions on Image Processing*, vol. 20(12), 2011, pp. 3376 – 3382.

[16] D. Li, G. Zhang, Z. Wu, L. Yi, “An edge embedded earker-based watershed algorithm for high spatial resolution remote sensing image segmentation,” *Image Processing, IEEE Transactions on*, vol. 19, no 10, 2010, pp. 2781-2787.

[17] L. G. Shapiro, G.C. Stockman, *Computer Vision*, Prentice Hall, Upper Saddle River, NJ, 2001

[18] W. Niblack, *An Introduction to Digital Image Processing*, Prentice Hall, Englewood Cliffs, 1986, pp.115-116.

[19] J. Sauvola, T, Seppanen, S. Haapakoski, M. Pietikainen, Adaptive Document Binarization, *ICDAR'97 4th Int. Conf. On Document Analysis and Recognition*, Ulm, Germany, 1997, pp.147-152.

[20] C. A. Boiangiu, A. V. Stefanescu, D. Rosner, A. Olteanu, A. Morar, “Automatic slanted edge target validation in large scale digitization projects,” *Annals of DAAAM for 2010, Proceedings of the 21st International DAAAM Symposium*, 2010, pp. 131-132.

[21] S. Bukhari, F. Shafait, T. Breuel, “Adaptive binarization of unconstrained hand-held camera-captured document images,” *Journal of Universal Computer Science*, vol. 15, 2009, pp. 3343—3363.

[22] S. Bukhari, F. Shafait, T. Breuel, “Foreground-background regions guided binarization of camera-captured document images,” *Proceedings of the Third International Workshop on Camera-Based Documents*, 2009, pp.18-25.