Adaptive scene-text binarization on images captured by smartphones

Amira Belhedi, Beatriz Marcotegui

Adaptive scene-text binarization on images captured by smartphones

Amira Belhedi Beatriz Marcotegui

MINES ParisTech, PSL Research University, CMM - Centre for Mathematical Morphology, 35 rue Saint Honoré - Fontainebleau, France

{amira.belhedi,beatriz.marcotegui}@mines-paristech.fr

Abstract

We address, in this paper, a new adaptive binarization method on images captured by smartphones. This work is part of an application for visually impaired people assistance, that aims at making text information accessible to people who cannot read it. The main advantage of the proposed method is that the windows underlying the local thresholding process are automatically adapted to the image content. This avoids the problematic parameter setting of local thresholding approaches, difficult to adapt to a heterogeneous database. The adaptive windows are extracted based on ultimate opening (a morphological operator) and then used as thresholding windows to perform a local Otsu's algorithm. Our method is evaluated and compared with the Niblack, Sauvola, Wolf, TMMS and MSER methods on a new challenging database introduced by us. Our database is acquired by visually impaired people in real conditions. It contains 4000 annotated characters (available online for research purposes). Experiments show that the proposed method outperforms classical binarization methods for degraded images such as low-contrasted or blurred images, very common in our application.

Keywords. Smartphone-captured images, adaptive binarization, scene-text, ultimate opening, area stability, Otsu algorithm, visually impaired people, assistive application.

1 Introduction

Smartphones are opening new possibilities for enhancing user's view of the world, providing applications such as geo-localization, augmented reality, *etc.* This has become possible thanks to the high sensor resolution and computational power of these devices. In the framework of LINX project, we develop a smartphone application allowing visually impaired people to get access to textual information in their every-day life. A critical step for this project is to identify regions of interest in the images. One way to do this is to produce a binary image. However, image binarization in this context is hard: in addition to the absence of prior information on image content, acquired images can be of low quality. In fact, image acquisition conditions are not under control: taken by visually impaired people, several issues can arise such as blur, noise, bad lighting conditions, *etc.*

Several works have been devoted to finding a relevant and efficient binarization method. Some of them perform globally applying the same threshold to the whole image. One of the best known in literature is Otsu's algorithm [1]. Despite its performance on clean documents, it is not well suited to uneven illumination and to the presence of random noise. Other works perform locally, adapting the threshold to each image region. A popular method in literature is proposed by Niblack [2]. It is based on calculating a pixel-wise threshold by gliding a rectangular window over the gray level image. The threshold T for the center pixel of the window is defined as:

$$T = m + k.s, \tag{1}$$

with m and s respectively the mean and the variance of the gray value in the window and k a negative constant. The main limitation of this method is the noise created in regions that do not contain any text, since a threshold is also applied in these regions. Sauvola [3] addresses this problem by normalizing the standard deviation by its dynamic range. This method outperforms the latter one, except in case of low-contrast text regions. A solution is proposed by Wolf [4] to overcome this drawback. The threshold formula is changed in order to normalize the contrast and the mean gray level of the image. More recent local

methods have been proposed (*e.g.* [5, 6]) An interesting method that solves Sauvola's algorithm limitations is proposed by Lazzara [7]. It is a multiscale version of Sauvola's method. According to the authors, this method outperforms the previous ones, particularly in case of documents with text of various sizes. However, it is not robust to very blurred text regions: they can be entirely removed or only partially detected with this method (this is demonstrated in section 4).

Local approaches give better results compared to global ones but often require more parameters to be tuned. The most difficult part is to find the optimal parameters' values for the set of images to be processed. However, adjusting those parameters with no prior knowledge of image content is difficult. In particular, adjusting the window size parameter with no information about the text size is not possible. That is the main reason why we propose a new adaptive scene-text binarization method that does not require a window size adjustment, since adaptive local windows (regions of interest) are automatically extracted from the image. Other reasons motivate us to propose this method: (1) the amount of false alarms created in regions that do not contain any text and (2) the missing detection problem produced by existing methods in blurred and low-contrasted text regions.

Our approach is mainly based on a simple local Otsu's algorithm [1] performed on regions of interest automatically extracted from the image. Regions of interest are detected with ultimate opening (UO) [8] weighted by the area stability of regions. The UO is a residual morphological operator that detects regions with the highest contrast and has been used successfully for text detection [9]. In addition to grayscale information used by UO, the proposed method uses area stability information, derived from Maximally Stable Extremal Regions (MSER) method, to favor the detection of regions with a more stable area. The MSER [10, 11, 12] method is commonly used for scene text segmentation purposes [13, 14, 15, 16, 17, 18]. It detects regions that are stable over a range of thresholds. It performs well most of the time but has problems on blurry images and characters with very low contrast [13]. This constraint (area stability weight) is introduced in order to avoid the detection of regions with great changes in areas that are probably related to the merging of different

characters or to the presence of noise.

Our binarization technique is the first step in the LINX processing chain. Further steps consist in characterizing extracted regions and classifying them in text or non-text regions. Therefore, our objective is to maximize the recall (the number of detected characters).

2 Linx dataset and ground truth generation

Some public text databases with ground truth are available. However, most of them provide ground truth at the level of words (*e.g.* ICDAR'15 database [19]): they are suitable to evaluate text localization, but not for text binarization. Other databases with character-level ground truth exist, such as DIBCO [20], EPITA [21], IIIT 5K-word [22], *etc.*, but they only contain text documents. In LINX project, we are not limited to text documents. Therefore, we had to produce our own annotated dataset (from LINX database).

2.1 LINX dataset

The dataset used is a subset of LINX database. It contains 16 images acquired with smartphone cameras by visually impaired people. Some images are shown in figure 1. In spite of the reduced number of images, an important number of words is present: about 1200 words with 4000 characters. It varies from text documents to products exposed in a supermarket, very blurred text, noisy regions, shadow regions, high saturated regions, small and large texts, light and dark texts, *etc.*

Figure 1: Some images from LINX dataset.

2.2 Ground truth generation

In order to validate our method (described in the following section) and to compare it to existing methods, it is crucial to generate an accurate ground truth. This is a challenging problem, especially for the blurred regions of the dataset. For that, we used a semi-automatic approach. For each image of the dataset, a manually selected global threshold that maximizes the number of detected characters (*i.e.* maximizing the recall) is first applied and a binary image is generated. After that, all splitted or merged characters are respectively manually connected or disconnected, and all false detected regions are manually deleted. We note that, for blurred regions, the global thresholding does not work well. To solve that, different local thresholding methods [2, 3, 4] are performed on these regions and the best binarization result, for each region, is selected. However, even after applying several local thresholding methods, some words are not correctly segmented into characters by any of them. We then chose to keep them as such and to add a flag ($GTLevel$) in the ground truth file indicating the ground truth level (if $GTLevel = 1$ it is a character-level, otherwise it is a word-level). Note that the overall procedure is applied on the image and on its inverse in order to detect both light and dark characters. Then results are merged in the same binary image and another flag ($textPolarity$) is added in the generated ground truth (if $textPolarity = 1$ it is light text, otherwise, it is a dark one). Finally, rectangular bounding boxes are computed from each connected component of the obtained binary images, since we have chosen a rectangle based validation (described in section 4). Even if a pixel-wise validation seems a better approach, we gave up the idea as it requires an accurate pixel-level ground truth that is extremely difficult to obtain in practice and may favor the method used for its generation.

The generated ground truth contains 3913 rectangular bounding boxes: 3253 characters and 660 words, 2216 bounding boxes with dark text and 1697 with light text. Some ground truth rectangles, cropped from our dataset, are shown in figure 2. LINX dataset and ground truth are available online[1].

[1] http://cmm.mines-paristech.fr/Projects/LINX

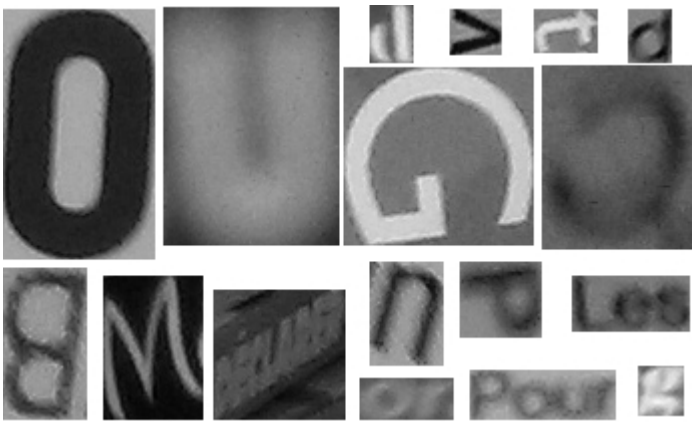


Figure 2: Ground truth rectangles cropped from our dataset. It is a very challenging dataset that varies from very large and high-contrast text regions to very small, noisy and blurred text regions. Its annotation was a hard task, especially for low-contrasted or blurred regions. As explained in the text, all local binarization methods failed to segment some words into single letters. They were kept as such in the generated ground truth. Some examples are shown here.

3 Adaptive scene-text binarization

We propose a new adaptive scene-text binarization that does not require manual adjustment of the window size to the image content, and reduces the number of false alarms. The proposed method performs a local Otsu's algorithm on adaptive windows that are automatically extracted from the image. The adaptive windows extraction is mainly based on mathematical morphology.

In this section, we first describe the dataset pre-processing, then we detail the adaptive windows detection and finally we present the binarization method based on adaptive windows.

3.1 Pre-processing

The dataset used in this study is a set of smartphone images, usually color images. Given in the one hand that luminance contains essential information for binarization process, and on the other hand that we need a computationally efficient method, we start by converting color to grayscale images. For this purpose, we chose to use the following conversion formula: $Luma = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$. A pre-processing filtering step is also required

to reduce image noise. We chose to perform a bilateral filter [23] of size 3 and σ_{gray} empirically fixed to 20.

3.2 Adaptive windows detection

In general, the text is contrasted compared to the background (clear text on dark background or dark text on clear background), otherwise it cannot be read. The proposed method is based on this feature: it detects the high-contrast regions in the image and obviously, in its inverse, based on the ultimate opening operator (UO) introduced by Beucher [8]. The UO is a morphological operator based on numerical residues that detects the highest contrasted connected components in the image. The operator successively applies a series of openings γ_i (opening of size i) with structuring elements of increasing sizes, i. Then, the residues between successive openings are computed: $r_i = \gamma_i - \gamma_{i+1}$ and the maximum residue is kept for each pixel. Thus, this operator has two significant outputs for each pixel x: $R(I)$ which gives the value of the maximal residue (contrast information), called the transformation in literature, and $q(I)$ which indicates the size of the opening leading to this residue (the structure size that contains the considered pixel) called associated function:

$$\begin{aligned} R\left(I\right) &= \max(r_i(I)) = \max\left(\gamma_i(I) - \gamma_{i+1}(I)\right), \forall i \geq 1 \\ q(I) &= \begin{cases} max\left\{i+1 \mid r_i(I) = R(I)\right\} & \text{if } R(I) > 0 \\ 0 & \text{if } R(I) = 0 \end{cases} \end{aligned} \tag{2}$$

The UO has been extended by Retornaz [24] to use an attribute opening [25] such as width, height, *etc.* The new definition of the transformation R and the associated function q are obtained by replacing, in equation (2), γ_i by the considered attribute opening. In this case, the associated function q indicates information linked with the considered attribute. An example is shown in figure 3 to illustrate the intermediate steps of UO calculation.

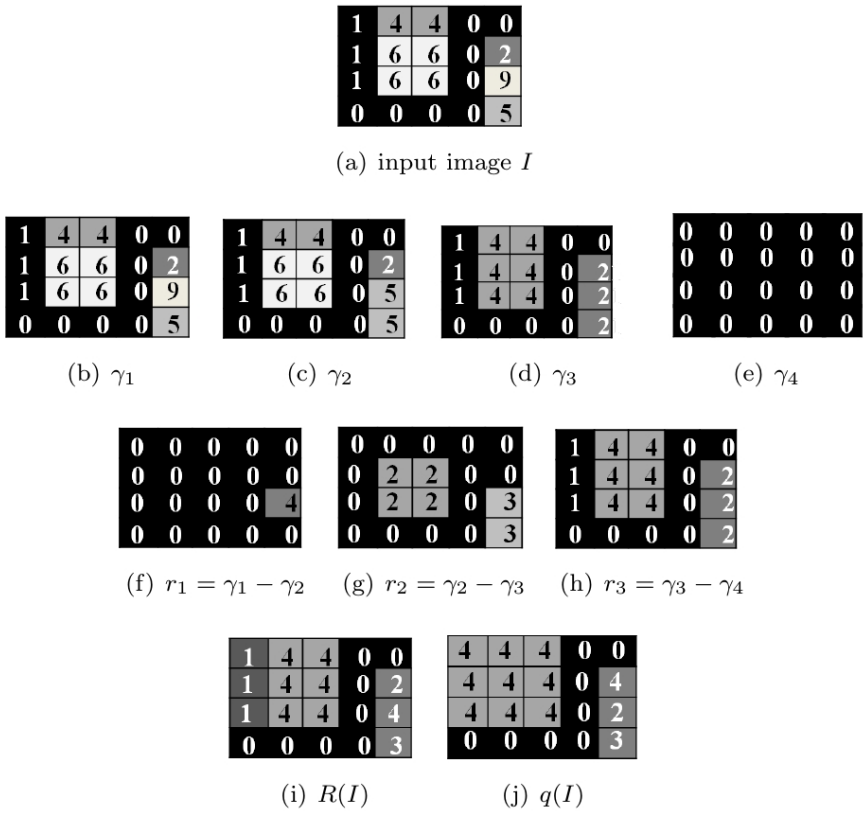


Figure 3: UO computation step by step. The UO attribute used in this example is the height of the connected component. (a) Input image I. (b-e) results of height openings with size $1, 2, 3$ and 4. (f-h) computed residues r_1, r_2 and r_3. An opening of size 1 (γ_1) does not change the image, γ_2 removes one maxima and generates the first residue r_1. An opening of size 3 (γ_3) removes larger regions and generates the second residue r_2. At the end, γ_4 removes all regions and generates the residue r_3. The last step of UO computation consists in generating the two resulting images: (i) the transformation $R(I)$ and (j) the associated function $q(I)$. For each pixel, the maximum residue r_i is selected and recoded in $R(I)$ and the size of the opening leading to this residue is recorded in $q(I)$. For example, the maximum residue of the pixel located in the third line of the last column (= 4) was selected from r_1 and the opening size leading to r_1 is equal to 2.

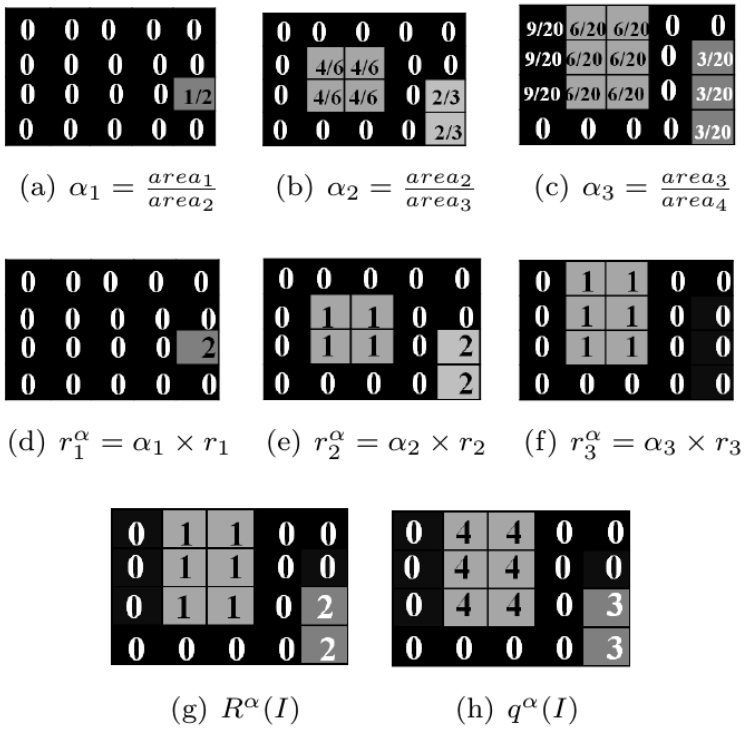


Figure 4: UO weighted by area stability step by step computation . The same input image I of figure 3 is used here. First, the weighted functions (area stability of regions) α_1, α_2 and α_3 are computed (images (a-c)). Then, they are used with their corresponding opening (figure 3 (b-e)) to generate weighted residues r_1^α, r_2^α and r_3^α (images (d-f)). Then, the outputs of the UO weighted by area stability function $R^\alpha(I)$ and $q^\alpha(I)$ are deduced (images (g) and (h)). Comparing figure 3(i) and figure 4(g), we can observe that R^α contains less noise than R.

For our application, we chose to use the extension of the UO with the height attribute, since it is the most suited one for text detection. We set the largest opening considered equal to $\frac{1}{3}$ of the image height, since characters are rarely larger. This choice is made in order to avoid artifacts that occur with a larger opening size. Very small regions (area < 15) are also not considered in order to avoid useless process. Note that the associated function q is not used in this study, we only use the transformation function R.

We chose to use this morphological operator for many reasons. First, it has the capacity of highlighting regions with the highest contrast which is suited for text detection. Then, it has the advantage to be a non parametric multi-scale

operator and does not require any prior knowledge of image content. Finally, it can be performed in real time using the fast implementation based on image maxtree representation [26].

Despite its performance, this operator has the limitation to produce a lot of connected components, most of them do not correspond to characters. To reduce the number of false positives appearing with the UO, we propose to introduce a weighted function α_i within the residue computation (equation 3), α_i being the region area stability inspired from MSER method [10] (used successfully as weighted function in [27]). The weighed residue r_i^α is defined as:

$$r_i^\alpha = \alpha_i \times r_i, \tag{3}$$

with α_i the weighted function computed, for each pixel x, as follows:

$$\alpha_i = \frac{area_i}{area_{i+1}}, \tag{4}$$

with $area_i$ the structure area (containing x) obtained from opening γ_i. Using this weight function, the residue of connected component with low area stability is artificially reduced, compared to that with high area stability. The computation of the UO with area stability weight is illustrated on simulated data in figure 4.

An example on real images is shown in figure 5(c): R produces many spurious regions in the background (in red) that do not correspond to text regions. A thresholding of R can help removing these regions but may also remove low-contrast text regions. The use of area stability weight avoids the detection of regions with important changes in area that are probably related to the presence of noise, artifacts and contrast variation in the background (figure 5(c)) or an unintended connection between components (see for example characters "INE" in figure 5(g)). Another example of the obtained transformation R^α is shown in figure 5(d). The number of false alarms is considerably reduced. We also observe that characters are better separated with R^α. An example is shown in the same figure. The transformation R connects 3 characters ("INE") as shown in figure 5(g), whereas R^α disconnects them (figure 5(h)). Note that the area stability weight function is easily introduced with the UO implementation based

on maxtree. In the tree, we only need to add to each node n, the area of its corresponding region and then, to weight in each UO iteration the computed residue with the node area stability value (see [26] for more details about tree based UO computation).

The transformation R^α (area stability weighted) is thresholded with a global low threshold (set to 1) to remove regions with very low residues (low contrast or low area stability) leading to the image B^α defined as follows:

$$B^\alpha(I) = \begin{cases} 255 & \text{if } R^\alpha(I) > 1, \\ 0 & \text{otherwise} \end{cases}$$

The binary image B^α is used to extract the adaptive windows used in the next step of binarization. These adaptive windows correspond to the rectangular bounding boxes of B^α connected components. An example is shown in figure 6 (c).

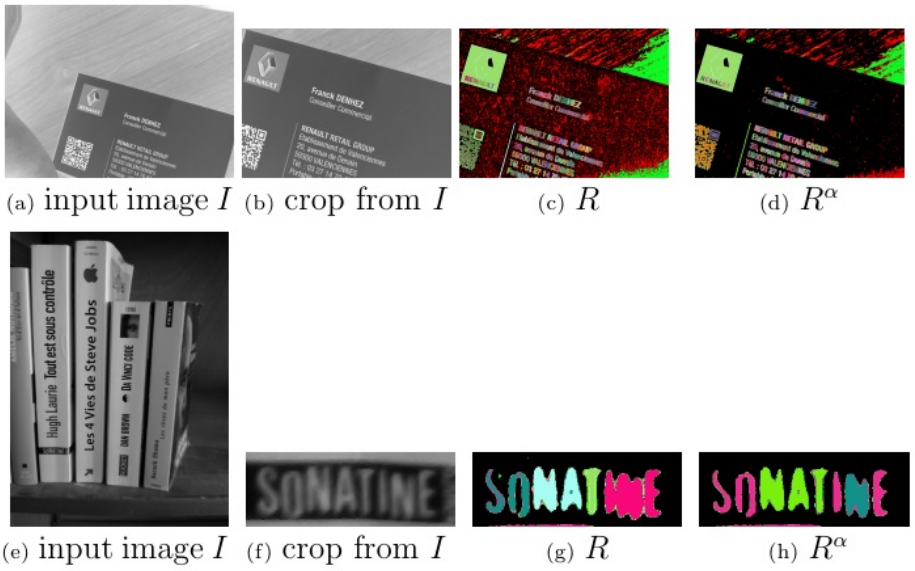
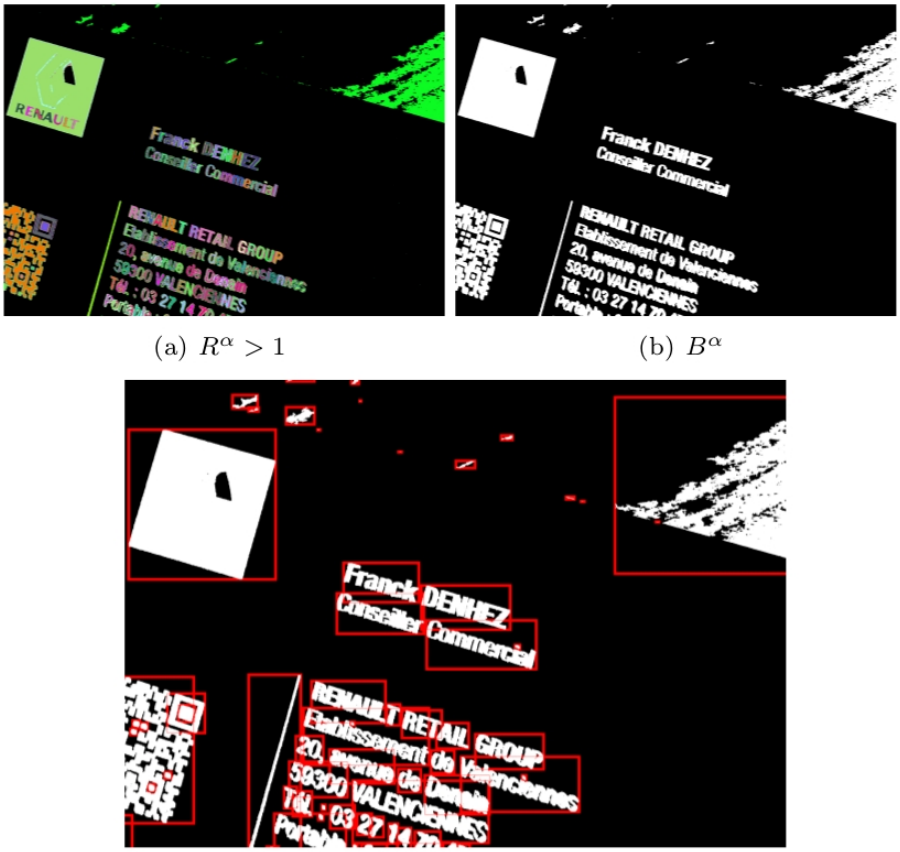


Figure 5: Comparison between the transformation obtained (c)(g) with UO and (d)(h) with UO weighted by area stability function. These images are converted to color images for a better illustration (random colors are used to better illustrate the different residues values). The number of false alarms is reduced with area stability weight and characters are better segmented.



(a) $R^{\alpha} > 1$ (b) B^{α}

(c) Adaptative windows

Figure 6: An example of (a) $R^{\alpha} > 1$, (b) B^{α} and (c) adaptive window: rectangular bounding boxes (in red) (results obtained on the image shown in figure 5(b)).

The substantial improvement brought by the adaptive windows based on the UO is shown in section 4. The fixed size windows used by the Niblack, Sauvola and Wolf methods are replaced by adaptive windows and better results are obtained.

3.3 Binarization

A binarization process is applied to the original image on adaptive windows defined in previous section. In the simplest case, an adaptive window contains a character in its background. A simple Otsu thresholding process performs well in these cases. More complex windows may correspond to a set of characters

gathered in their containing support. This may happen due to the fact that we have chosen a very low threshold ($R^\alpha > 1$) in order to ensure the detection of low-contrast characters. An example of this situation is shown in figure 7. Applying an Otsu threshold on such adaptive windows does not detect the characters but the region containing them (see for example figure 7(d)). A multi-level Otsu algorithm, with three classes (one for the characters, a second one for the merged region and the third one for the background), is required in this case. In order to detect this merging situation we analyze R^α in each connected component (CC) of B^α, that we note R^α_{CC}. If the CC contains very different R^α values and the most common value (the mode) is significantly lower than its maximum value, we assume that the CC has merged significant regions. Thus, a merging situation is declared if the following two conditions are satisfied:

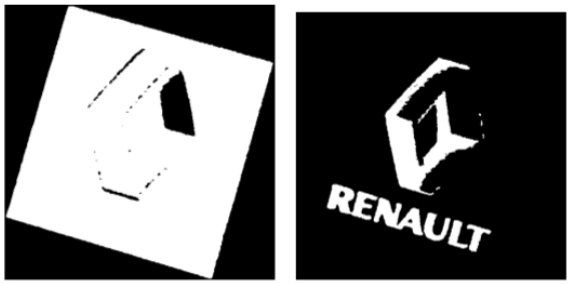
- $mode \leq \frac{max}{2}$, with $mode$ the value that appears most often in R^α_{CC} and max the maximum value of R^α_{CC}. If this condition holds, the CC contains regions with a contrast twice higher than the contrast of the largest part of it (the mode). This is the first hint indicating that significant regions are missed.

- $modePercentage > 0.7$, with $modePercentage$ the percentage of pixels with R^α value equal to $mode$. This condition confirms that the low-contrasted region of the CC covers a significant part of it. This is in general the case when low contrasted region surround significant area.

An example of this process is illustrated in figure 7. The input is a crop from the original image corresponding to an adaptive window 7(a). The corresponding R^α_{CC} is shown in figure 7(b). If Otsu's algorithm is performed in this window, these characters will not be detected, as shown in figure 7(d). Analyzing R^α_{CC}, a merging situation is detected. The maximum value in R^α_{CC} is equal to 22 ($max = 22$) and characters are surrounded by a low-contrast region, its value in R^α_{CC} is equal to 8 ($mode = 8$). Thus, both merging conditions are satisfied. Then, the multi-level Otsu with three classes is performed in this adaptive window of the input image. The obtained result is shown in figure 7(e). The word "RENAULT" is well segmented.

We observe that character edges after performing the binarization step are cleaner than B^α (figure 6(b)) and merged characters of B^α are correctly segmented.

(a) crop from figure 5(a) (b) R^α_{CC} (crop from R^α) (c) crop from B^α



(d) Otsu (e) multi-level Otsu

Figure 7: Example of binarization merging problem illustrated on (a) a crop from the original image (figure 5(a)). This crop corresponds to an adaptive window defined by B^α of figure (c) The word ("RENAULT") is contained in a low-contrast region. b) R^α_{CC}. c) B^α d) Otsu algorithm applied to (a), all characters are missing. Analizing R^α_{CC}, the merging conditions are satisfied ($mode = 8 \leq \frac{max}{2} = \frac{22}{2}$ and $modePercentage = 0.8 > 0.7$). (e) Multi-level Otsu with three classes is performed, all characters are correctly segmented.

3.4 Post-processing

A post-processing is applied to the image obtained from the previous step in order to remove very small regions. We use a small area opening of size 15 (the image resolution is about 3200×2400). We show in figure 8 an example of the final result. Comparing this final result with B^α image (figure 6(b)), we can state that the binarization step improves characters detection.

Figure 8: Binary image obtained with our method (input image shown in figure 5(a)).

4 Validation

In this section, we validate the performance of the proposed method and compare it to the best known methods in the literature. The dataset used and its ground truth generation are detailed in section 2. In the following, the experimental protocol is first presented and the obtained results are then discussed.

4.1 Evaluation Protocol

The evaluation is performed comparing a list G of ground truth bounding boxes $G_{i,i=1..|G|}$ with a list D of binarized objects bounding boxes $D_{j,j=1..|D|}$ (with $|G|$ and $|D|$ the number of bounding boxes respectively in G and D). The rectangle based evaluation method presented by Wolf [28] is used. This choice is made for many reasons. First, it supports one-to-one, as well as one-to-many (splits) and many-to-one matches (merges). Then, the precision and recall measures are computed at the object level by imposing quality constraints (recall constraint t_r and precision constraint t_p) to the matched rectangles. This gives a better estimation of false alarms and correct detections than the direct accumulation of rectangle overlaps. This is briefly explained in the following. The matching between G and D rectangles are determined according to conditions of different matching types (one-to-one, split and merge) based on t_r and t_p. Then for each

G_i, the recall value r_i is defined as follows:

$$r_i = \begin{cases} 1 & \text{if } G_i \text{ matches against a single } D_j, \\ 0 & \text{if } G_i \text{ does not match against any detected rectangle,} \\ f_{sc}(k) & \text{if } G_i \text{ matches against } k \text{ detected rectangles } (k > 1), \end{cases}$$

and for each D_j, the precision value p_j is defined as follows:

$$p_j = \begin{cases} 1 & \text{if } D_j \text{ matches against a single } G_i, \\ 0 & \text{if } D_j \text{ does not match against any ground truth rectangle,} \\ f_{sc}(k) & \text{if } D_j \text{ matches against } k \text{ ground truth rectangles } (k > 1), \end{cases}$$

with $f_{sc}(k)$ a parameter that controls the punishment amount. In our experiments, merges and splits are severely punished by setting $f_{sc}(k) = \frac{1}{1+\log(k)}$ which corresponds to the fragmentation metric introduced in [29]. The recall constraint t_r is set to 0.7 and the precision constraint t_p to 0.4 (value recommended by Wolf). Obviously, splits in case of word-level annotation are not punished *i.e.* if G_i matches against several detected rectangles and its ground truth flag $GTlevel$ (introduced in section 2.2) is not equal to 1, then its recall r_i is set to 1. Another A flag which is saved in the ground truth file is used for recall and precision computation: the $textPolarity$ flag. As mentioned above, the same image can contain dark and clear texts. Then we perform each tested method twice, on the image and on its inverse. Then, for each G_i, we compute the recall value from the suitable resulting image according to the $textPolarity$ flag value and we select its(their) matching rectangle(s) from D, for precision computation. The false alarms that correspond to D_j that do not match against any G_i must be taken into account in precision computation. They can be selected from one of the two resulting images. In our experiments, we select them from the dominant polarity (based on the $textPolarity$ flag). This choice seems appropriate and obviously does not influence the recall measure.

4.2 Results

For all methods presented in this section, the images are pre- and post-processed as described in sections 3.1 and 3.4.

We first verify that the use of adaptive windows based on the ultimate opening, extracted from B^α, enhances substantially the results of local binarization methods. For that, three methods from the literature that use fixed window sizes (Niblack [2], Sauvola [3] and Wolf [4]) are tested on our dataset. Note that these methods are better adapted for document binarization purposes, even if they are currently cited in scene-text localisation approaches too. We use the implementation provided by Wolf [30]. For each of them, the optimal k value recommended by its author is used *i.e.* -0.2, 0.34 and 0.5 for respectively Niblack, Sauvola and Wolf methods. Concerning the window size value, we set it to 40×40 (default values of the distributed code [30]). Applying Niblack and Wolf methods on adaptive windows, B^α, instead of fixed ones, enhances substantially the results and mainly the mean precision (table 1). It is increased from 8.4% to 50.2% in the case of Niblack's algorithm, and from 41.9% to 52.4% in case of Wolf's algorithm. The mean recall is also improved for both of them and exceeds 90% in case of Wolf's algorithm. However, we do not observe an improvement of Sauvola's algorithm with adaptive windows. This is probably due to its main limitation: the miss detection of low-contrast regions.

	fixed windows size			adaptive windows		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
Niblack	86.1	8.4	15.3	89.8	50.2	64.4
Sauvola	65.4	36.9	47.2	62.1	37.3	46.7
Wolf	67.7	41.9	51.7	93.0	52.4	67.0

Table 1: Mean recall (R) precision (P) and F-measure (F) comparison of fixed and adaptive windows binarization methods.

We compare now our approach with three methods that do not require a window size adjustment to the image contents. The multiscale version of Sauvola's method presented by Lazzara [7], the morphological algorithm based on the toggle mapping operator TMMS [31] (ranked 2^{nd} out of 43 in DIBCO 2009 challenge [32]) and the MSER method [10] (the most cited for scene-text localisation [33]).Here, for TMMS and multiscale version of Sauvola's methods,

the implementation provided by the respective authors with their recommended parameters is used. For multiscale Sauvola's algorithm, $k = 0.34$, the window size at scale 1 is $w = 101$ and the first subsampling ratio $s = 3$. For TMMS algorithm, the hysteresis thresholds $cminL = 20, cminH = 45$ and the thickness parameter $p = 50$. For MSER, the OpenCV implementation [34] is used with the parameters leading to the best results on our dataset: the Δ value = 0, the minimum area = 15, the maximum variation between areas = 0.25 and the minimum MSER diversity = 0.2. The obtained results are presented in table 2. A recall and precision values of respectively 73.3% and 34.6% are obtained with multiscale Sauvola's method. The corresponding f-measure is similar to the f-measure obtained with the single-scale version of this method, but with a higher recall (73.3% instead of 65.4%). The popular MSER leads to a relatively low recall and precision ($R = 61.7\%$ and $P = 47.6\%$) while TMMS method performs relatively well ($R = 88\%$ and $P = 61.3\%$), but some low-contrast texts are missing in the result.

		R (%)	P (%)	F (%)
Multiscale Sauvola		73.3	34.6	47.0
TMMS		88.0	61.3	72.2
MSER		61.7	47.6	53.7
Otsu on MSER-based adaptive windows		90.7	57.5	70.4
our approach	without area stability	95.4	35.9	52.2
	with area stability	95.3	58.2	72.3

Table 2: Mean recall (R), precision (P) and F-measure (F) comparison of methods that do not require a window size parameterization.

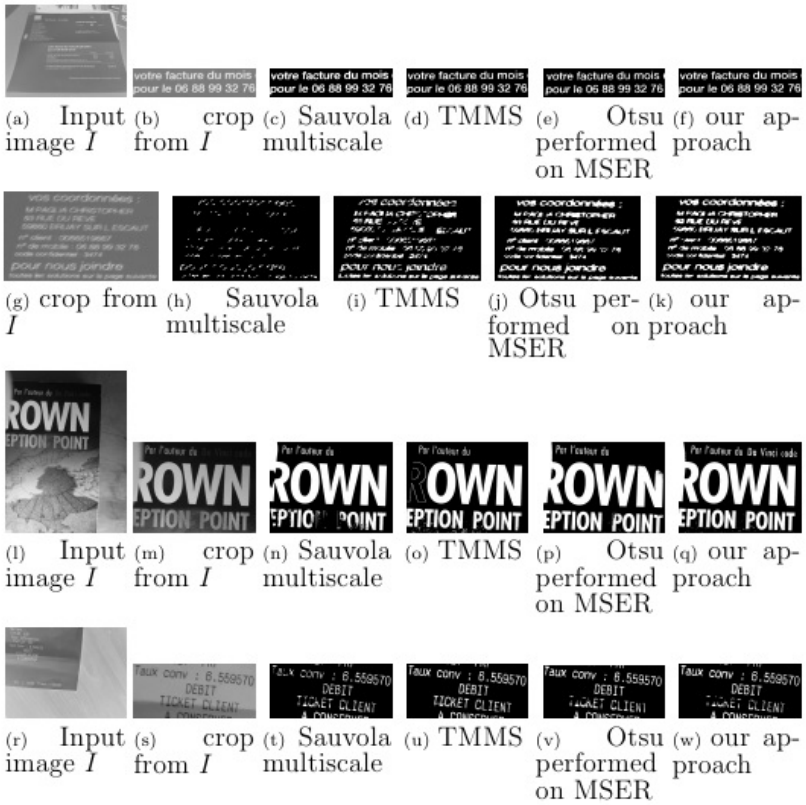
Our approach gets much better results: $R = 95.3\%$ and $P = 58.2\%$). Regarding the contribution of area stability weight, we observe a significant precision increase (58.2% compared to 35.9%) with very similar recall figures (95.3% instead of 95.4%). Thus, the area stability weight considerably reduces the false alarms without missing regions of interest. The use of a simple binarization technique such as Otsu on adaptive windows leads to the best recall figures. Our adaptive windows rely on the UO approach, as explained in section 3.2. Other techniques can be used for this purpose, for example bounding boxes of MSER resulting regions. Applying Otsu technique on MSER-based adaptive windows a recall $R = 90.7\%$ and a precision $P = 57.5\%$ are reached. The binarization step

significantly improves the result (the f-measure is about 17% higher). Moreover UO technique outperforms MSER for adaptive windows extraction.

Our approach and TMMS lead to very similar f-measures ($\sim 72\%$), followed by binarization on MSER-based adaptive windows ($\sim 70\%$). Note that our binarization technique is the first step in the LINX processing chain. Further steps consist in characterizing extracted regions and classifying them in text or non-text regions. Therefore, our objective is to maximize the recall (the number of detected characters). In these conditions our approach is the best suitable solution for our application.

When observing the obtained results with more details, we see that for relatively good quality text regions, all methods give good results. An example is shown in the first row of figure 9. However, for degraded or low-contrasted text regions, we obtain better results with our approach. This is shown in the second and third rows of figure 9. Characters are removed or partially detected with TMMS and multiscale Sauvola's methods. With our approach we are able to detect even very low-contrasted, blurred and degraded text regions. However, it is not robust to shadows and high-saturated regions. An example is shown in the fourth row of figure 9: the word "TICKET CLIENT" is truncated by a shadow. It is not well segmented with our approach and multiscale Sauvola's method, TMMS method gives better results in that region.

We compare now the execution time of our method and the state of the art methods that give good results on our database (multiscale Sauvola's and TMMS methods). Given times do not include pre-processing and post-processing operations, only processing and I/O operations are included. The whole LINX dataset (16 images with a resolution of 3264×2448) is processed. For our method and the multiscale Sauvola's method, each image is processed twice (both polarities). Tests are carried out on a DELL PC with 3.4GHz (8 physical cores) Intel processor. Obtained results show that our method is slower (1.34s per image) compared to multiscale Sauvola's algorithm (0.86s per image) and TMMS algorithm (0.62s per image). There are also the post-processing operation that is time-consuming: about 0.22s per image and the pre-processing takes about 0.08s per image. Our code will be optimized in order to reduce the execution time and to be able to use it in real time.



(a) Input image I (b) crop from I (c) Sauvola multiscale (d) TMMS (e) Otsu performed on MSER (f) our approach

(g) crop from I (h) Sauvola multiscale (i) TMMS (j) Otsu performed on MSER (k) our approach

(l) Input image I (m) crop from I (n) Sauvola multiscale (o) TMMS (p) Otsu performed on MSER (q) our approach

(r) Input image I (s) crop from I (t) Sauvola multiscale (u) TMMS (v) Otsu performed on MSER (w) our approach

Figure 9: Comparison with the multiscale version of Sauvola's, TMMS and Otsu performed on MSER adaptive window methods. The MSER is used for adaptive windows detection, then Otsu's algorithm is performed on these windows. First row, a good quality image. Good results are obtained with all methods. Second row, a blurred image. Multi-scale Sauvola approach misses most characters, TMMS misses part of them and our method and MSER based methods obtain better results . Third row, characters with very low contrast ("Da Vinci code") are detected by our method but not by TMMS, Sauvola and MSER based methods. Fourth row, an example of image with a shadow that leads to a missed detection of "TICKET" word with our approach, multi-scale Sauvola and MSER based methods while TMMS detects it correctly.

In conclusion, we can say that the proposed approach outperforms the methods from the state of the art on our challenging dataset.

5 Conclusion and perspectives

We have presented, in this paper, a new binarization method of text images acquired with smartphones. It has the advantage to be efficient, even on low

quality images, common in our application. In addition, it does not require any prior knowledge of image content, thus avoiding the parameters adjustment problem of local thresholding approaches (such as Niblack's and Wolf's) and giving good results. The proposed method is composed of two steps: 1) adaptive windows detection based on an original combination of the UO operator and the region area stability, 2) local binarization with Otsu's algorithm on these detected windows.

We have compared our method with several methods on a new challenging dataset introduced by us (containing 4000 characters). It is a subset of LINX images, acquired in real conditions, by visually impaired people. The heterogeneity of this database makes difficult the parameter tuning of popular local thresholding approaches.

The obtained results demonstrate: 1) the benefit of weighting the UO by the area stability function: the number of false alarms is significantly reduced 2) the efficiency of our method: more characters are detected compared to the tested methods, in particular in case of low-contrasted or blurred images, very common in our application and a good precision rate is obtained. The new database introduced in this paper and its ground truth are available online for research purposes.

Future work will be to optimize our algorithm in order to be able to use it in real time. It would also be interesting to improve its robustness to the presence of shadow and saturation regions, as it tends to detect contrasted shadows or saturated regions rather than the possible characters included in them.

Acknowledgments.

The work reported in this paper has been performed as part of Cap Digital Business Cluster LINX Project.

References

[1] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. IEEE Transactions on Systems, Man and Cybernetics, 9(1):62–66, 1979.

[2] Wayne Niblack. An Introduction to Digital Image Processing. Strandberg Publishing Company, Birkeroed, Denmark, 1985.

[3] J. Sauvola, T. Seppänen, S. Haapakoski, and M. Pietikäinen. Adaptive document binarization. In International Conference on Document Analysis and Recognition, volume 1, pages 147 – 152, 1997.

[4] Christian Wolf, Jean-Michel Jolion, and Françoise Chassaing. Text localization, enhancement and binarization in multimedia documents. In International Conference on Pattern Recognition, pages 1037–1040. IEEE Computer Society, 2002.

[5] T. Romen Singh, Sudipta Roy, O. Imocha Singh, Tejmani Sinam, and Kh. Manglem Singh. A new local adaptive thresholding technique in binarization. CoRR, abs/1201.5227, 2012.

[6] He Xiao and Yunbo Rao. An efficient method of text localization in complicated background scenes. Journal of Software, 9(6), 2014.

[7] Guillaume Lazzara and Thierry Géraud. Efficient multiscale Sauvola's binarization. International Journal of Document Analysis and Recognition (IJDAR), 17(2):105–123, June 2014.

[8] Serge Beucher. Numerical residues. Image Vision Computing, 25(4):405–415, April 2007.

[9] Thomas Retornaz and Beatriz Marcotegui. Scene text localization based on the ultimate opening. In International Symposium on Mathematical Morphology, volume 1, pages 177–188, Rio de Janeiro, 2007.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Proceedings of the British Machine Vision Conference, pages 36.1–36.10. BMVA Press, 2002.

[11] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput., 22(10):761–767, 2004.

[12] Ron Kimmel, Cuiping Zhang, Alexander M Bronstein, and Michael M Bronstein. Are MSER features really interesting? Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(11):2316–2320, 2011.

[13] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3538–3545. IEEE, 2012.

[14] Honggang Zhang, Kaili Zhao, Yi-Zhe Song, and Jun Guo. Text extraction from natural scene image: A survey. Neurocomputing, 122:310–323, 2013.

[15] Hyung Il Koo and Duck Hoon Kim. Scene text detection via connected component clustering and nontext filtering. Image Processing, IEEE Transactions on, 22(6):2296–2305, 2013.

[16] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 36(5):970–983, 2014.

[17] Lei Sun, Qiang Huo, Wei Jia, and Kai Chen. A robust approach for text detection from natural scene images. Pattern Recognition, 48(9):2906–2920, 2015.

[18] Qixiang. Ye and David Doermann. Text detection and recognition in imagery: A survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 37(7):1480–1500, 2015.

[19] ICDAR 2015, robust reading competition. http://2015.icdar.org.

[20] HDIBCO 2014, handwritten Document Image Binarization Contest. http://users.iit.demokritos.gr/ kntir/HDIBCO2014/.

[21] Guillaume Lazzara and Thierry Géraud. LRDE document binarization dataset. https://www.lrde.epita.fr/wiki/Olena/DatasetDBD.

[22] A. Mishra, K. Alahari, and C. V. Jawahar. The IIIT 5k-word dataset. http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html.

[23] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In Computer Vision, 1998. Sixth International Conference on, pages 839–846. IEEE, 1998.

[24] Thomas Retornaz and Beatriz Marcotegui. Ultimate Opening Implementation based on a flooding process. In International Congress for Stereology, pages 0–7, Saint Etienne, 2007.

[25] Edmond J. Breen and Ronald Jones. Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding, 64(3):377–389, 1996.

[26] Jonathan Fabrizio and Beatriz Marcotegui. Fast implementation of the ultimate opening. In Proc. of the 9th Intern. Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, ISMM '09, pages 272–281. Springer-Verlag, 2009.

[27] Andrés Serna, Beatriz Marcotegui, Etienne Decencière, Thérèse Baldeweck, Ana-Maria Pena, and Sébastien Brizion. Segmentation of elongated objects using attribute profiles and area stability: Application to melanocyte segmentation in engineered skin. Pattern Recognition Letters, 47:172–182, 2014.

[28] Christian Wolf and Jean-Michel Jolion. Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms. International Journal of Document Analysis and Recognition, 8(4):280–296, April 2006.

[29] V.Y. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, D. Doermann, and T. Drayer. Performance Evaluation of Object Detection Algorithms. In International Conference on Pattern Recognition, pages 965–969, 2002.

[30] Christian Wolf. C++ code for document image binarization. http://liris.cnrs.fr/christian.wolf/software/binarize/.

[31] J. Fabrizio, B. Marcotegui, and M. Cord. Text segmentation in natural scenes using toggle-mapping. In Proceedings of the 16th IEEE International Conference on Image Processing, 2009.

[32] Basilios Gatos, Konstantinos Ntirogiannis, and Ioannis Pratikakis. Icdar 2009 document image binarization contest (dibco 2009). In ICDAR, pages 1375–1382. IEEE Computer Society, 2009.

[33] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pages 1156–1160. IEEE, 2015.

[34] OpenCV MSER. http://docs.opencv.org/master/d3/d28/classcv_1_1MSER.html.