TEXT SEGMENTATION IN NATURAL SCENES USING TOGGLE-MAPPING

J. Fabrizio[1,2], B. Marcotegui[2], M. Cord[1]

[1]UPMC Univ Paris 06,
Laboratoire d'informatique de Paris 6, 104 av du Président Kennedy, 75016 Paris, France
[2]MINES Paristech, CMM- Centre de morphologie mathématique, Mathématiques et Systèmes
35 rue Saint Honoré - 77305 Fontainebleau cedex, France.

ABSTRACT

We offer, in this paper, a new method to segment text in natural scenes. This method is based on the use of a morphological operator: the *Toggle Mapping*. The efficiency of the method is illustrated and the method is compared, according to various criteria, with common methods issued from the state of the art. This comparison shows that our method gives better results and is faster than the state of the art methods. Our method reduces also the number of segmented regions. This can lead to time saving in a complete scheme (executing time of multiple processing steps usually depends on the number of regions) and proves that our algorithm is more relevant.

1. INTRODUCTION

Automatic text localization in images is a major task in computer vision. Various applications depend on this task (automatic image indexing, visual impaired people assistance or optical character reading...). We are currently working, in Itowns project [1], on text extraction in an urban environment. The aim of the projet is to automatically enhance cartographic databases and to allow the user to make high level queries on them. Another target of the projet is to allow the user to navigate freely within the image flow in the city but our work is not related to this task. To achieve this work, geolocalized pictures of every streets are taken, every meter. All images are processed off line to extract as much semantic data as possible. Text extraction in this context is hard because 1. there is a wide variety of text in a city and no hypothesis can be made (style, position, orientation...) and 2. the amount of data is huge. Today 2 TB for a part of a single district in Paris. More than 4 TB for more districts (next year). Then, we want a fast and easy method to segment the image that keeps as much characters as possible.

Multiple solutions already exist (even if major works on text segmentation are focused on documents or on constrained contexts, such as license plate localizations). We can cite first the *stroke filter* defined by Liu et al. in [2] and specifically used for text detection [3]. This filter is an edge detector, similar to Canny's or Sobel's, but more efficient for character segmentation [2]. Second, we can cite many local thresholding methods. There is a wide variety of criteria [4]. Two local criteria seem to be better for character segmentation: Niblack criterion [5] and Sauvola criterion [6]. In Niblack criterion, threshold $T(x)$ for a given pixel x, is given according to its neighborhood by:

$$T(x) = m(x) + ks(x) \tag{1}$$

with m and s the mean and the standard deviation computed on the neighborhood and $k \in \mathbf{R}$ a parameter. In Sauvola criterion, threshold $T(x)$ is found by:

$$T(x) = m(x)\left(1 + k\left(\frac{s(x)}{R} - 1\right)\right) \tag{2}$$

with R the dynamic of standard deviation $s(x)$.

Last, we can cite the segmentation exposed by Retornaz [7] based on the *ultimate opening*. This operator, introduced by Beucher [8], is an unparametered morphological operator that highlights the most contrasted areas in an image.

We explore, in this article, a new application of the morphological operator *toggle mapping* to segmentation. We offer to associate a local contrast-based approach to the *Toggle Mapping* framework. The resulting segmentation operator (*TMMS*) is fast and may be easily tuned to get any kind of characters. In the first part of the paper, we explain briefly *toggle mapping*. In the second part, we expose our method and explain how to set up toggle mapping to perform a segmentation. In the last part we present some results and we compare the method with Sauvola and Niblack thresholding method as well as with an ultimate opening method.

2. TOGGLE MAPPING

Toggle Mapping is a morphological operator introduced by Serra [9]. Given a function f (defined on D_f) and a set of n

We are grateful for support from the French Research National Agency (A.N.R.)

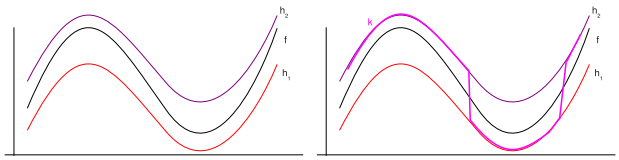

Fig. 1. On the left, function f and a set of 2 functions h_1 and h_2. On the right, function k computed by toggle mapping.

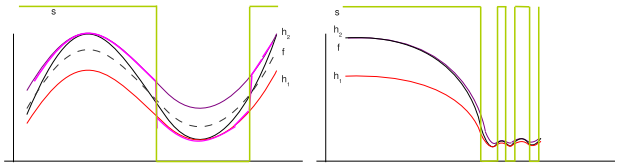

Fig. 2. Result of eq. 6 (function s) on an edge and in homogeneous noisy regions.

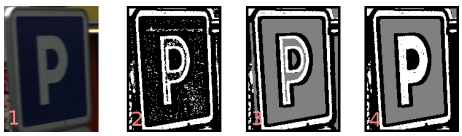

Fig. 3. From left to right: 1. Original image, 2. Binarization (function s from eq. 6), 3. Homogeneity constraint (eq. 7), 4. Filling in small homogeneous regions.

functions $h_1, ..., h_n$, a new function k is defined by (Fig. 1):

$$\forall x \in D_f k(x) = h_i(x); \forall j \in \{1..n\} \quad |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \quad (3)$$

The result depends on the choice of the set of functions h_i. A classical use of toggle mapping is contrast enhancement: if the initial function f is an image, taking a set of 2 functions h_1 and h_2 extensive and anti-extensive respectively, the toggle mapping result will be a new image k, which looks like f with an enhanced contrast.

3. TOGGLE MAPPING MORPHOLOGICAL SEGMENTATION (TMMS)

Toggle mapping is a generic operator which maps a function on a set of n functions. It has been used, as we saw before, for contrast enhancement but also for noise reduction. We propose here to use toggle mapping to perform image segmentation. To segment a gray scale image f by the use of toggle mapping, we use a set of 2 functions h_1 and h_2 with h_1 the morphological erosion of f and h_2 the morphological dilatation of f. These two functions are computed by:

$$\forall x \in D_f \quad h_1(x) = \min f(y); y \in v(x) \quad (4)$$

$$\forall x \in D_f \quad h_2(x) = \max f(y); y \in v(x) \quad (5)$$

with $v(x)$ a small neighborhood (the structuring element) of pixel x. Then, instead of taking the result of toggle mapping k (eq. 3), we define function s:

$$\forall x \in D_f s(x) = i; \forall j \in \{1..n\} |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \quad (6)$$

Function $s(x)$ takes two values and may be seen as a binarization of image f with a local criterion (Fig. 2 left). Our function efficiently detects boundaries but may generate salt and peeper noise in homogeneous regions (Fig. 2 right): even very small local variations generate an edge. This leads us to refine the definition of s by the introduction of c_{min}, a minimal contrast:

$$s(x) = \begin{cases} 0 & if |h_1(x) - h(2)| < c_{min} \\ 1 & if |h_1(x) - h(2)| >= c_{min} \\ & \& \, |h_1(x) - f(x)| < p * |h_2(x) - f(x)| \\ 2 & otherwise \end{cases} \quad (7)$$

Then, no boundaries will be extracted from homogeneous areas. s is a segmentation of f (notice that now we have 3 possible values instead of 2: a low value, a high value and a value that represents homogeneous regions).
To use this method efficiently, some parameters must be set up: the size of the structuring element used to compute a morphological erosion (h_1) and a dilation (h_2), the minimal contrast c_{min} and an additional parameter p. Variations of p influence the thickness of detected structures.
Getting three values in output instead of two can be embarrassing. Many strategies can be applied to assign a value to homogeneous regions (to determine whether the region belongs to low value area or high value area): it is possible to study boundaries of such a region to see if the region is surrounded by a low or a high valuated boundaries and assign it to the region. Another strategy consists in dilating all boundaries onto homogeneous regions. In our case, this is not a real issue as characters have low thickness, it is not common to have homogeneous regions into characters and if it occurs, such regions are small. Then, our strategy consists in studying boundaries of small regions in order to fill a possible hole in characters followed by a small dilation.

4. RESULTS

The context of our work is the text detection in an urban environment (Fig. 5). Before measuring the efficiency of the algorithm, let us see some results in this context. First, the application of definition (eq. 6) extracts boundaries but also salt and peeper noise (Fig. 3). Adding constraint c_{min} (eq. 7), *cleans* the image result (Fig. 3) but may introduce some holes. Removing small homogeneous regions fills in holes in interesting regions (Fig. 3). Various results are shown in figure 4.

Fig. 4. Six results of our method on various texts from IGN [10] image database. Segmentation is difficult as there is a wide variety of text: text style, illumination and orientation may vary. Decorations (illustrated background, relief effect on characters...) may decrease readability.

Fig. 6. Results of various algorithms on different images. From top to bottom and left to right: Original image, Niblack thresholding, Sauvola thresholding and our method. More characters are segmented with our method.

5. EVALUATION

We have seen the results of our method. Now we measure its efficiency. To do so, we compare our method with two thresholding methods (that uses Niblack [5] and Sauvola [11] criterion respectively as both are references in text segmentation) and the ultimate opening [8] (because of its efficiency). To perform a comparison, we focus on different aspects: the segmentation speed and quality. According to our context, we perform the comparison on a randomly taken subset of Itowns project [1] image database provided by IGN [10].

Before processing the comparison, each method is set up. Different sets of parameters have been used for each method. The parameters leading to the best result (in terms of segmentation quality) are selected for each method and all comparisons are performed using selected parameters. For all methods we select the size of the mask 9x9. k parameter is set to -0.05 and 0.05 for Niblack (eq. 1) and Sauvola (eq. 2) criterion respectively. For our method, the lowest contrast (c_{min} in eq. 7) is set to 16 and p parameter (eq. 7) is set to 80%. Results of all methods can be seen in figure 6.

Segmentation quality assessment

The segmentation evaluation is always difficult as it is, for a part, subjective. It is, frequently, impossible to have a ground truth to be use with a representative measure. To evaluate segmentation as objectively as possible for our application, we segment the image database and we count every properly segmented characters. For us, *properly segmented* means that the character is not split or linked with other features around it. The character must also be readable. The thickness may vary a little provided that its shape remains correct. The evaluation image database contains 501 readable characters. The following table gives the result of each method:

	% of properly segmented characters
Ultimate Opening	48,10
Sauvola	71,26
Niblack	73,85
TMMS	**74,85**

The ultimate opening surprisingly gives bad results. This may be due to the fact that images may have motion blur (they are acquired by sensors mounted on a moving vehicle). We then cancel it from the rest of the comparison. Our method gives the best results, followed by thresholding with Niblack Criterion. Thresholding with Sauvola criterion is far less efficient on average. It fails frequently on text correctly handled with Niblack criterion or our method but, in some situations, it gives the best quality segmentation. The overall poor result is explained by the high difficulty level of the environment.

Fig. 5. Image from the itowns project.

Executing time

The other aspect of our comparison is the speed. The next table gives mean times of every method, in second, according to the size of the mask (Image size for the test is 1920x1080 and execution was performed on 2,40GHz Q6600 processor):

Mask size	3x3	5x5	7x7	9x9	11x11
Niblack	0,16	0,22	0,33	0,47	0,64
Sauvola	0,16	0,23	0,33	0,47	0,64
TMMS	**0,11**	**0,18**	**0,27**	**0,44**	**0,55**

All implementations are performed according to the definition without any optimization. Our method always gets the best execution times (Notice that Shafait et al. [12] has recently offered a faster way to compute Sauvola criteria).

The speed of the algorithm is important but the output is also a major aspect as execution time of a complete scheme usually depends on the number of regions provided by segmentation steps. The next table gives, on our database, the average number of regions generated by each method:

	number of output regions in mean
Niblack	65 177
Sauvola	43 075
TMMS	**28 992**

Reducing the number of regions in the output may save time when we process these regions. The possibility, in our method, to set up the lowest allowed contrast prevents from having oversegmented regions. Moreover, many of these regions, noticed as homogeneous, can be associated with other neighboor regions (end of section 3). This simple process may lead to a decrease in the number of regions. This low number of regions may increase the localization precision as it can decrease false positives. It is another proof that the segmentation provided by our method is more relevant.

6. CONCLUSION

We have presented a new method to segment characters in natural scenes (but it is not limited to them, as it can be used in other contexts...). This method is based on the use of toggle mapping. We compare it with most common methods and we show that our method is better in terms of 1. efficiency as it segments more characters than other methods, 2. quality as the method does not generate too many regions (over segmentation) that could slow down other processing steps, 3. speed as it is faster than others. As the size of data is huge, we have to work on an efficient implementation of our method. The use of *SIMD instructions* seems to be a good way to speed up the implementation.
In our process, all regions of the image are studied by a classifier to select textual information. All selected regions are transmitted to an OCR software. In Itowns project, the identified text automatically enhances cartographic database to improve the information retrieval task [13].

7. REFERENCES

[1] iTowns project, ," www.itowns.fr.

[2] Q.F. Liu, C.K. Jung, S.K. Kim, Y.S. Moon, and J.Y. Kim, "Stroke filter for text localization in video images," *IEEE International Conference on Image Processing*, 2006.

[3] X.J. Li, W.Q.A. Wang, S.Q.A Jiang, Q.M. Huang, and W. Gao, "Fast and effective text detection.," in *IEEE International Conference on Image Processing*, 2008.

[4] Mehmet Sezgin and Blent Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13(1), pp. 146–165, 2004.

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

[6] J. Sauvola, T. Seppänen, S. Haapakoski, and M. Pietikäinen, "Adaptive document binarization," in *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*. 1997, pp. 147–152, IEEE Computer Society.

[7] T. Retornaz and B. Marcotegui, "Scene text localization based on the ultimate opening," *International Symposium on Mathematical Morphology*, vol. 1, pp. 177–188, 2007.

[8] S. Beucher, "Numerical residues," *Image Vision Comput.*, vol. 25, no. 4, pp. 405–415, 2007.

[9] J. Serra, "Toggle mappings," *From pixels to features*, pp. 61–72, 1989, J.C. Simon (ed.), North-Holland, Elsevier.

[10] Institut Géographique National, ," www.ign.fr.

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

[12] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," *Document Recognition and Retrieval XV*, 2008.

[13] P. H. Gosselin and M. Cord, "Active learning methods for interactive image retrieval," *IEEE Transactions on Image Processing*, 2008.