TEXT EXTRACTION FROM STREET LEVEL IMAGES

J. Fabrizio[1,2], M. Cord[1], B. Marcotegui[2]
[1]UPMC Univ Paris 06
Laboratoire d'informatique de Paris 6, 75016 Paris, France
[2]MINES Paristech, CMM- Centre de morphologie mathématique, Mathématiques et Systèmes,
35 rue Saint Honoré - 77305 Fontainebleau cedex, France

KEY WORDS: Urban, Text, Extraction, Localization, Detection, Learning, Classification

ABSTRACT

We offer in this article, a method for text extraction in images issued from city scenes. This method is used in the French iTowns project (iTowns ANR project, 2008) to automatically enhance cartographic database by extracting text from geolocalized pictures of town streets. This task is difficult as 1. text in this environment varies in shape, size, color, orientation... 2. pictures may be blurred, as they are taken from a moving vehicle, and text may have perspective deformations, 3. all pictures are taken outside with various objects that can lead to false positives and in unconstrained conditions (especially light varies from one picture to the other). Then, we can not make the assumption on searched text. The only supposition is that text is not *handwritten*. Our process is based on two main steps: a new segmentation method based on morphological operator and a classification step based on a combination of multiple SVM classifiers. The description of our process is given in this article. The efficiency of each step is measured and the global scheme is illustrated on an example.

1 INTRODUCTION

Automatic text localization in images is a major task in computer vision. Applications of this task are various (automatic image indexing, visual impaired people assistance or optical character reading...). Our work deals with text localization and extraction from images in an urban environment and is a part of iTowns project (iTowns ANR project, 2008). This project has two main goals : 1. allowing a user to navigate freely within the image flow of a city, 2. Extracting features automatically from this image flow to automatically enhance cartographic databases and to allow the user to make high level queries on them (go to a given address, generate relevant hybrid text-image navigation maps (itinerary), find the location of an orphan image, select the images that contain an object, etc.). To achieve this work, geolocalized set of pictures are taken every meter. All images are processed off line to extract as many semantic data as possible and cartographic databases are enhanced with these data. At the same time, each mosaic of pictures is assembled into a complete immersive panorama (Figure 1).
Many studies focus on text detection and localization in images. However, most of them are specific to a constrained context such as automatic localization of postal addresses on envelopes (Palumbo et al., 1992), license plate localization (Arth et al., 2007), text extraction in video sequences (Wolf et al., 2002), automatic forms reading (Kavallieratou et al., 2001) and more generally "documents" (Wahl et al., 1982). In such context, strong hypothesis may be asserted (blocks of text, alignments, temporal redundancy for video sequences...). In our context (*natural scenes* in an urban environment), text comes from various sources (road sign, storefront, advertisements...). Its extraction is difficult: no hypothesis can be made on text (style, position, orientation, lighting, perspective deformations...) and the amount of data is huge. Today, we work on 1 TB for a part of a single district in Paris. Next year, more districts will be processed (more than 4 TB). Different approaches already exist for text localization in natural scenes. States of the art are found in (Mancas-Thillou, 2006, Retornaz and Marcotegui, 2007, Jung et al., 2004, Jian Liang et al., 2005). Even if preliminary works exist in natural scene (Retornaz and Marcotegui, 2007, Chen and Yuille, 2004), no standard solution really emerges and they do not focus on urban context.
The paper presents our method and is organized as follows: the text localization process is presented and every step is detailed followed by the evaluation of main steps. In the last part, results are presented. Then comes the conclusion.

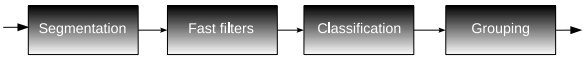

Figure 2: General principle of our system.

2 SEGMENTATION BASED STRATEGY

The goal of our system is to localize text. Once the localization is performed, the text recognition is carried out by an external O.C.R. (but the system may improve the quality of the region by correcting perspective deformations for example). Our system is a region based approach and starts by isolating letters, then groups them to restore words and text zones. Region based approach seems to be more efficient, such approach was ranked first (Retornaz and Marcotegui, 2007) during ImagEval campaign (ImagEval, 2006). Our process is composed of a cascade of filters (Figure 2). It segments the image. Each region is analysed to determine whether the region corresponds to text or not. First stages during selection eliminate a part of non text regions but try to keep as many text region as possible (at the price of a lot of false positives). At the end, detected regions that are close to other text regions are grouped all together. Isolated text regions are canceled.

Figure 1: Image from iTowns project.

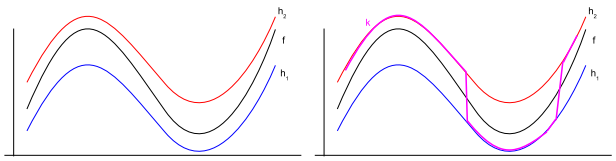

Figure 3: On the left, function f and a set of 2 functions h_1 and h_2. On the right, function k computed by toggle mapping.

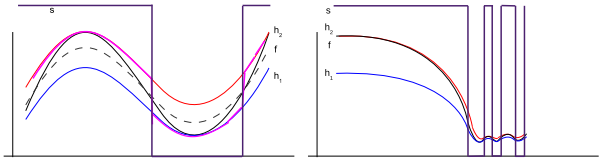

Figure 4: Result of eq. 4 (function s) on an edge and in homogeneous noisy regions.

Figure 5: From left to right: 1. Original image, 2. Binarization (function s from eq. 4), 3. Homogeneity constraint (eq. 5), 4. Filling in small homogeneous regions.

3 TEXT SEGMENTATION

Our segmentation step is based on a morphological operator introduced by Serra (Serra, 1989): *Toggle Mapping*. Toggle mapping is a generic operator which maps a function on a set of n functions: given a function f (defined on D_f) and a set of n functions $h_1, ..., h_n$, this operator defines a new function k by (Fig. 3):

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

The result depends on the choice of the set of functions h_i. A classical use of toggle mapping is contrast enhancement: this is achieved by applying toggle mapping on an initial function f (an image) and a set of 2 functions h_1 and h_2 extensive and anti-extensive respectively.

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:

$$\forall x \in D_f \quad h_1(x) = \min f(y); y \in v(x) \tag{2}$$

$$\forall x \in D_f \quad h_2(x) = \max f(y); y \in v(x) \tag{3}$$

with $v(x)$ a small neighborhood (the structuring element) of pixel x. Then, instead of taking the result of toggle mapping k (eq. 1), we keep the number of the function on which we map the pixel. This leads us to define function s:

$$\forall x \in D_f s(x) = i; \forall j \in \{1..2\} |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \tag{4}$$

Function $s(x)$ takes two values and may be seen as a binarization of image f with a local criterion (Fig. 4 left). Our function efficiently detects boundaries but may generate salt and pepper noise in homogeneous regions (Fig. 4 right): even very small local variations generate an edge. To avoid this, we introduce a minimal contrast c_{min} and if $|h_1(x) - h_2(x)| < c_{min}$, we do not analyse the pixel x.

Function s is then improved:

$$s(x) = \begin{cases} 0 & if |h_1(x) - h_2(x)| < c_{min} \\ 1 & if |h_1(x) - h_2(x)| >= c_{min} \\ & \& \; |h_1(x) - f(x)| < p * |h_2(x) - f(x)| \\ 2 & otherwise \end{cases} \tag{5}$$

Then, no boundary will be extracted within 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 disturbing. Many strategies can be applied to assign a value to homogeneous regions (to determine whether the region belongs to low value areas or high value ones): if a region is completely surrounded by pixels of the same value, the whole region is assigned to this value. Another strategy consists in dilating all boundaries onto homogeneous regions. In our case, this is not a real issue: as characters are narrow, it is not common to have homogeneous regions inside characters and if it occurs, such regions are small. Then, our strategy consists in studying boundaries of small homogeneous regions in order to fill a possible hole in characters. Bigger homogeneous regions are mostly left unchanged, only a small dilation of these boundaries is performed.

Illustration of the segmentation process is given in Figure 5. In the rest of the paper, this method is called Toggle Mapping Morphological Segmentation (TMMS).

4 FILTERING

Once the image is segmented, the system must be able to select which regions contain text (letters) and which do not. A part of these regions is obviously non text (too big/too small regions, too large...). The aim of this step is to dismiss most of these obviously non text regions without loosing any good character. A small collection of fast filter (criteria opening) eliminate some regions with simple geometric criteria (based on area, width and height). These simple filters help saving time because they rapidly eliminate many regions, simplifying the rest of the process (which is a bit slower).

5 PATTERN CLASSIFICATION

Some segmented regions are dismissed by previous filters but a lot of false positives remain. To go further, we use classifiers with suitable descriptors.
Due to the variability of analysed regions, descriptors must (at least) be invariant to rotation and scale. The size and the variability of examples in training database ensure to be invariant to perspective deformations. We have tested a lot of different shape descriptors (such as Hu moments, Fourier moments...). Among them, we have selected two families of moments : Fourier moments and the pseudo zernike moments. We select them empirically as during our test, they get a better discrimination ratio than others. We choose also to work with a third family of descriptors: polar representation is known to be efficient (Szumilas, 2008) but the way this representation is used does not match our need. Then we define our own polar descriptors: the analysed region is expressed into polar coordinate space centered into the gravity center (Figure 6). The feature is then mapped into a normalized rectangle (the representation is then invariant in scale factor). To be rotation invariant, many people use this representation by computing a horizontal histogram within this rectangle but this leads to a loss of too much information. Another way to be rotation invariant if the representation used is not rotation invariant is to redefine the distance computed between samples (Szumilas, 2008). But this leads to a higher complexity. To be rotation invariant, we simply take the spectrum magnitude of Fourier transform of each line in the normalized rectangle. These results carry much more information than simple histograms, and are easier than changing the distance used.

Once we choose the descriptors, we train a svm classifier (Cortes and Vapnik, 1995) for each family of descriptors. To give a final decision, all outputs of svm classifier are processed by a third svm classifier (Figure 7). We tried to add more classifiers in the first step of the configuration (with other kinds of descriptors) but this makes the overall accuracy systematically decreasing.

6 GROUPING

We are able to analyse main regions in the image and extract characters. Once these characters are selected, they

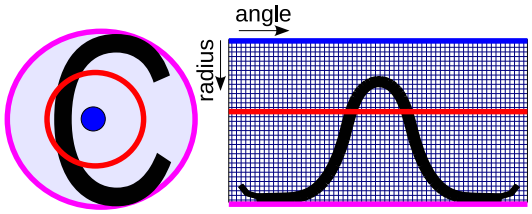

Figure 6: The region is expressed in a polar coordinate space and to have a rotation invariant descriptor we take the spectrum of Fourier transform of every line.

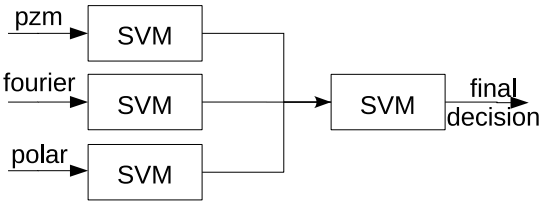

Figure 7: Our classifier is composed of 3 svm classifiers that use common family of descriptors and a svm that take the final decision.

are grouped all together with neighbour to recover text regions. The conditions to link two characters to each other are the one given in (Retornaz and Marcotegui, 2007). They are based on the distance between the two regions relatively to their height. This steps will soon be improved to handle text in every direction as this approach is restricted to nearly horizontal text. During this process, isolated text regions (single character of couple of letters) are dismissed. This aggregation is mandatory to generate words and sentences to integrate as an input in an O.C.R. but it also suppresses a lot of false positive detections.

7 LETTER DETECTION EXPERIMENTS

In this section, we evaluate segmentation and classification steps.

Segmentation The segmentation evaluation is always difficult as it is, for a part, subjective. Most of time, it is impossible to have a ground truth to be used with a representative measure. To evaluate segmentation as objectively as possible for our application, we have constituted a test image database by randomly taking a subset of the image database provided by I.G.N. (Institut Géographique National, n.d.) to the project (iTowns ANR project, 2008). We segment all images from this database and we count properly segmented characters. We define as clearly as possible what *properly segmented* means: the character must be readable, it must not be split or linked with other features around it. The thickness may vary a little provided that its shape remains correct. We compare the result with 3 other segmentation methods:

- Niblack binarization criterion (Niblack, 1986) which

evaluates a threshold $T(x)$ for a given pixel x, according to its neighborhood by:

$$T(x) = m(x) + ks(x) \quad (6)$$

with m and s the mean and the standard deviation computed on the neighborhood and $k \in \mathbf{R}$ a parameter.

- Sauvola binarization criterion (Sauvola et al., 1997) which evaluates a threshold $T(x)$ by:

$$T(x) = m(x)\left(1 + k\left(\frac{s(x)}{R} - 1\right)\right) \quad (7)$$

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

- the segmentation exposed by Retornaz (Retornaz and Marcotegui, 2007) based on the *ultimate opening*. This operator, introduced by Beucher (Beucher, 2007), is a non-parametric morphological operator that highlights the most contrasted areas in an image.

The evaluation image database contains 501 characters. The results of each method are given in the following table:

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

Our method gives the best results. Thresholding with Sauvola criterion is far less efficient on average. It fails frequently on text correctly handled with Nilback 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. The ultimate opening surprisingly gives bad results. This may come from the fact that images are taken by sensors mounted on a moving car: images may have a motion blur, which makes the ultimate opening fail. We then cancel it from the comparison.
The other aspect of our comparison is speed. We evaluate all methods on the set of images and compute mean times. Times are given in seconds for 1920x1080 image size and according to the size of the mask of every method:

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. (Shafait et al., 2008) have recently offered a faster way to compute Sauvola criterion).

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. On our database, on average, binarization

Figure 8: Examples of text and non text samples in learning database.

with Niblack criterion generates 65177 regions, binarization with Sauvola criterion generates 43075 regions, our method generates 28992 regions. 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 over segmented regions. Moreover, many of these regions, noticed as homogeneous, can be associated with other neighbour 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 localisation precision as it can decrease false positives. It is another proof that the segmentation provided by our method is more relevant.

Letter Classification To perform training and testing we have constituted (Fig. 8):

- a training data base composed of 32400 examples with 16200 characters from various sources (letters at different scales/points of view...) and 16200 other regions extracted from various urban images and,
- a testing base with 3600 examples.

Notice that all training are performed by tools provided by (Joachims, n.d.).
Different configurations of classifiers have been tested to get the highest classification accuracy. With the configuration we have chosen (Figure 7), the svm classifier trained with pseudo Zernike moments gives 75.89% of accuracy, the svm classifier trained with our polar descriptors gives $81,50\%$ of accuracy and last svm classifier trained with Fourier descriptors gives $83,14\%$ of accuracy. This proves that our descriptor is well defined as its accuracy is at the same level of accuracy as Fourier descriptors and pseudo Zernike moments.
To make the final decision we choose a *late fusion* architecture. Different tests are performed: from a simple vote of the three previous classifiers to the use of another classifier. The best result has been reached by the use of a SVM classifier which gets, $87,83\%$ of accuracy with the confusion matrix :

%	Letter	Background
Letter	91,56	8,44
Background	15,89	84,11

The unbalanced result is interesting for us, as the most important for us is not to lose a character.

Figure 9: The system localizes correctly text in the image (even with rotated text) but it detects aligned windows as text.

Figure 10: Text is correctly localized, but the classification step fails on the end of the word *courant* in red and zebra crossing sign is seen as text.

We also test different combinations of classifiers and descriptors. When we try early fusion architecture, we give all descriptors to a unique svm classifier ; the result does not even reach 74% of accuracy. On the contrary, if we add a collection of simple geometric descriptors (compacity, surface, concavity...) to the svm classifier that must take the final decision in our architecture, the overall accuracy reaches $88,83\%$. These measures seem to help the classifier to select which classifiers are the most reliable depending on the situation.
The overall accuracy seems to be a bit low but the variability of text in our context is so huge that the real performance of the system is not so bad.

8 TEXT LOCALIZATION IN CITY SCENES

Let us see the application of the complete scheme. We took an initial image (Figure 12). The application of our algorithm of segmentation gives the result in figure 13. All regions with a reasonable size are kept, others are dismissed (Figure 14). The classifier selects text regions among remaining regions (Figure 15). Text regions are grouped to create words and sentences (Figure 16).
The system is efficient: instead of a variation of orientation, police and lighting condition, the system handles majority of text (Figure 9, 10 et 11). But it also generates many false positives: especially aligned windows (Figure 9 top right and Figure 11). Other results can be seen in figures 9 and 10. The system must then be improved to reduce false positives.

Figure 11: Various texts are correctly handled but periodical features are also interpreted as text.

9 CONCLUSION

We have presented a text localization process defined to be efficient in the difficult context of the urban environment. We use a combination of an efficient segmentation process based on morphological operator and a configuration of svm classifiers with various descriptors to determine regions that are text or not. The system is competitive but generates many false positives. We are currently working to enhance this system (and reducing false positives) by improving the last two steps: we keep on testing various configurations of classifiers (and selecting kernels of svm classifiers) to increase the accuracy of the classifier and we are especially working on a variable selection algorithm. We are also working on the grouping step of neighbour text regions and its correction to send properly extracted text to O.C.R.

ACKNOWLEDGEMENTS

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

REFERENCES

Arth, C., Limberger, F. and Bischof, H., 2007. Real-time license plate recognition on an embedded DSP-platform. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '07) pp. 1–8.

Beucher, S., 2007. Numerical residues. Image Vision Comput. 25(4), pp. 405–415.

Chen, X. and Yuille, A. L., 2004. Detecting and reading text in natural scenes. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2, pp. 366–373.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine Learning 20(3), pp. 273–297.

ImagEval, 2006. www.imageval.org.

Institut Géographique National, n.d. www.ign.fr.

iTowns ANR project, 2008. www.itowns.fr.

Jian Liang, David Doermann and Huiping Li, 2005. Camera-Based Analysis of Text and Documents: A Survey. International Journal on Document Analysis and Recognition 7(2+3), pp. 83 – 104.

Joachims, T., n.d. svm. http://svmlight.joachims.org/.

Jung, K., Kim, K. and Jain, A., 2004. Text information extraction in images and video: a survey. Pattern Recognition 37(5), pp. 977–997.

Kavallieratou, E., Balcan, D., Popa, M. and Fakotakis, N., 2001. Handwritten text localization in skewed documents. In: International Conference on Image Processing, pp. I: 1102–1105.

Mancas-Thillou, C., 2006. Natural Scene Text Understanding. PhD thesis, TCTS Lab of the Facult Polytechnique de Mons, Belgium.

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

Palumbo, P. W., Srihari, S. N., Soh, J., Sridhar, R. and Demjanenko, V., 1992. Postal address block location in real time. Computer 25(7), pp. 34–42.

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

Sauvola, J. J., Seppänen, T., Haapakoski, S. and Pietikäinen, M., 1997. Adaptive document binarization. In: ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition, IEEE Computer Society, Washington, DC, USA, pp. 147–152.

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

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

Szumilas, L., 2008. Scale and Rotation Invariant Shape Matching. PhD thesis, Technische universitt wien fakultt fr informatik.

Wahl, F., Wong, K. and Casey, R., 1982. Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing 20(4), pp. 375–390.

Wolf, C., michel Jolion, J. and Chassaing, F., 2002. Text localization, enhancement and binarization in multimedia documents. In: In Proceedings of the International Conference on Pattern Recognition (ICPR) 2002, pp. 1037–1040.

Figure 12: The initial image used for the test. This image is provided by the french ign (Institut Géographique National, n.d.).

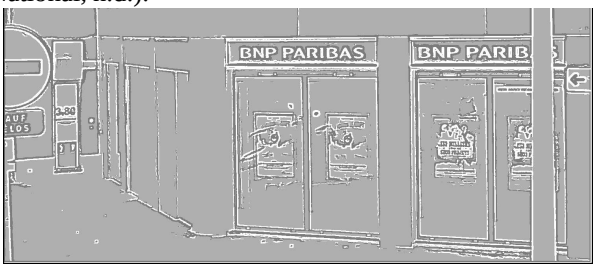

Figure 13: The image segmented by our algorithm TMMS.

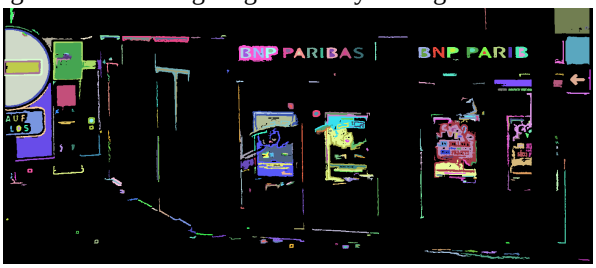

Figure 14: All big regions are removed. Only the regions of reasonable size are kept.

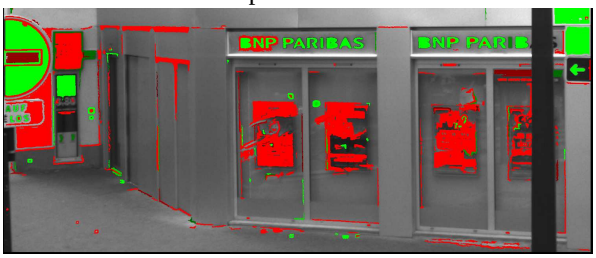

Figure 15: Remaining regions are classified by our system. Text region (in green) are kept, non text region (in red) are removed.

Figure 16: Isolated text regions are removed and remaining regions are grouped.