# Text Detection and Recognition in Urban Scenes

R. Minetto[1,2], N. Thome[1], M. Cord[1], J. Stolfi[2], F. Précioso[1], J. Guyomard[1], N.J Leite[2] *
(1) UPMC Univ Paris 6, LIP6, 4 place Jussieu, 75005 Paris, France
(2) University of Campinas UNICAMP, Av. Albert Einstein, 1251, Campinas-SP, Brazil
rodrigo.minetto@gmail.com , nicolas.thome@lip6.fr

## Abstract

Text *detection and recognition in real images taken in unconstrained environments, such as street view images, remain surprisingly challenging in Computer Vision.*

*In this paper, we present a comprehensive strategy combining bottom-up and top-down mechanisms to detect* Text *boxes. The bottom-up part is based on character segmentation and grouping . The top-down part is achieved with a statistical learning approach based on box descriptors. Our main contribution consists in introducing a new descriptor,* ***F****uzzy* ***HOG*** *(F-HOG), fully adapted for text box analysis. A thorough experimental validation proves the efficiency of the whole system outperforming state of the art results on the standard ICDAR text detection benchmark.*

*Another contribution concerns the exploitation of our text extraction in a complete search engine scheme. We propose to retrieve a location from a textual query: combining our text box detection technology with OCR on georeferenced street images, we achieved a GIS*[1] *system with a fully automatic textual indexing. We demonstrate the relevance of our system on the real urban database of [10].*

## 1. Introduction

Text detection is still a very challenging task in Computer Vision. Many approaches have been proposed, but most of them are dedicated to specific contexts, such as automatic localization of postal addresses on envelopes [11], license plate localization [14], or Optical Character Recognition (OCR) for scanned documents. They are very successful when applied to scanned pages of well formatted printed text, but quickly failed in many other contexts. To be convinced, just think to the successful visual CAPTCHA[2] application, where text image is explicitly constructed to fool computers.

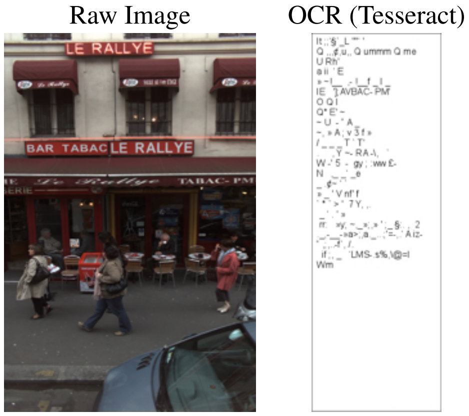


Figure 1. Text Detection & Recognition in Urban Context with a public OCR (Tesseract): lists of characters that contain almost no (piece of) words but a lot of noise.

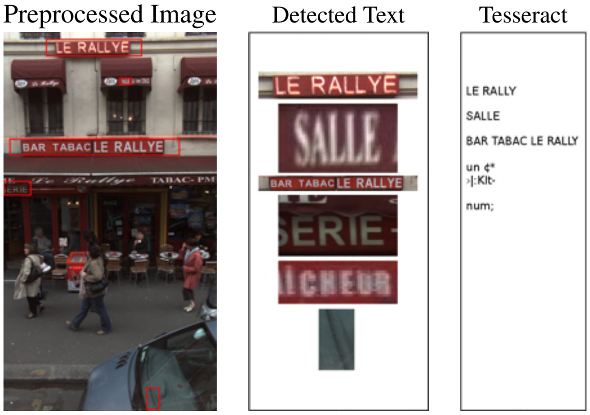


Figure 2. Our strategy for Text Detection & Recognition in Urban Context. There are many readable words and few noise.

For street view images as the one presented in figure 1, applying an OCR is even a complete failure. Actually, street images makes the context of text detection and recognition especially hard, with the main challenges of extreme text

*This work was partially supported by ANR ITOWNS (07-MDCO-007-03) and FAPESP (Phd grant 07/54201-6), CAPES/COFECUB 592/08.

[1] **G**eographic **I**nformation **S**ystems

[2] **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part

size and font variations, strong background clutter and difficult illumination conditions, *etc*. In this context, applying off-the-shelf Optical Character Recognition (OCR) tools on raw input street images is very unlikely to succeed. For example, when processing the street image in figure 1 with a publicly available OCR (Tesseract), the OCR results contain very few readable (piece of) words.

We propose here a complete system dedicated to text detection and recognition on such complex images. Basically, we process the input images in order to locate text box candidates that we connect to OCR input. For detection, our strategy combines bottom-up and top-down mechanisms to detect based on a new HOG-like descriptor dedicated to text analysis. The strategy is illustrated on the same raw image in figure 2. Thanks to our focus on relevant image patches, OCR produces a much more relevant list of words coming from the input image. For example, applying a simple OCR (Tesseract) on boxes pre-selected by our text detector makes the whole recognition process successful (figure 2).

## 2. Related Works and contributions

In this section, we give a brief overview of state of the art text detection approaches.

Regarding methodology, the different approaches can be classified into bottom-up and top-down strategies. Bottom-up methods first attempt at detecting individual characters and then merge neighboring positive detections. The main difficulty is to efficiently isolate single characters due to the ambiguity of local and low-level visual features. Contrarily, top-down approaches directly look for text in image (sub)regions, mostly using a scanning window mechanism. However, this brute force strategy is extremely computationally demanding since the number of windows to analyze exponentially grows with respect to the deformation parameters (*e.g.* scale, rotation, apse ct ratio). Therefore, most practical system must resort to approximations: a coarse discretization of the parameters or limiting some degrees of freedom (fixed aspect ratio or text orientation, for example).

ICDAR conference organized *robust reading competitions* [9] which goal is to establish a common benchmark giving a clear understanding of the current state of the art of such algorithms. This dataset became a standard baseline for comparing text detection systems. Interestingly, the two leading systems of the last ICDAR challenge [9] rely on different methodologies. Hinnerk Becker [9] (best system) developed a bottom-up approach that uses an adaptive finalization scheme to extract character regions which are then combined fulfilling some geometrical constraints to create text lines. Alex Chen et al [9] (rank 2) developed a top-down approach that makes use of a statistical analysis over text regions to select relevant features for characterizing text. Then, they use a cascade of classifiers trained over the chosen features to select regions candidates. Finally, connected components are extracted over these regions, and analyzed to recover each text word.
Many approaches have been evaluated in the ICDAR dataset since 2005. We focus here on two recent papers that are closely connected to our approach. In [5], Epshtein *et.al.* propose a new operator, the so-called Stroke Width Transform (SWT) to detect characters in images. Each character is supposed to have a nearly constant stroke width. In addition, the authors provide a new annotated with urban scenes taken with hand-held cameras. In [1], Chen *et.al.* propose to apply a connected component labeling algorithm after pre-processing each input image with MSER. The letter candidates are then filtered out using stroke width information. Both approaches are clearly bottom-up: character candidates are aggregated to generate text line hypotheses, and eventually decomposed into words. Both system report similar results in the ICDAR dataset [9], achieving state of the art performances.

In this paper, we propose a text detection scheme efficient in urban context. The proposed approach, depicted in figure 3, combines bottom-up and top-down mechanisms. The bottom-up stage (hypothesis generation) consists in character segmentation, classification and grouping, and is inspired from [10]. The top-down phase is dedicated to validate each text box hypothesis using global descriptors, *i.e.* complementary features from the bottom-up steps that only analyze single characters locally. This hypothesis validation is significant improved with respect to [10], and we propose here a novel descriptor, that we denote as Fuzzy **HOG** (F-HOG), to accurately represent each text box hypothesis. The second paper contribution concerns the integration of our approach into a real GIS search engine application.

The remainder of the paper decomposes as follows. Section 3 gives a brief overview of the bottom-up steps of the proposed detector, while section 4 describes the F-HOG. A experimental validation of the proposed text detector is proposed in section 5, showing that our system outperforms state of the art results in the ICDAR database. In addition, section 6 evaluates the whole system (text detection and recognition) in urban images. Finally, section 7 concludes the paper and gives directions for future works.

## 3. Bottom-Up Text Hypotheses Generation

Regarding hypothesis generation, our algorithm is composed of three main steps: image segmentation, character classification, and character grouping.

The segmentation step is based on a morphological operator, *toggle mapping*, introduced by Serra [13]. Toggle mapping is a generic operator which maps a function on a set of $n$ functions and is generally used for contrast enhancement, noise reduction and successfully applied to image segmentation [7]. In order to be effective in complex images, such as urban images with large character size

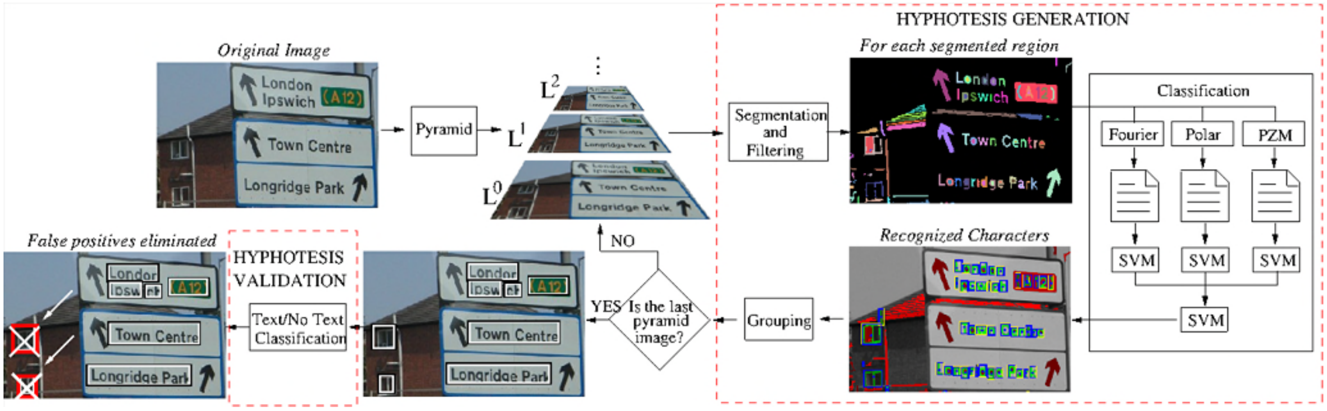
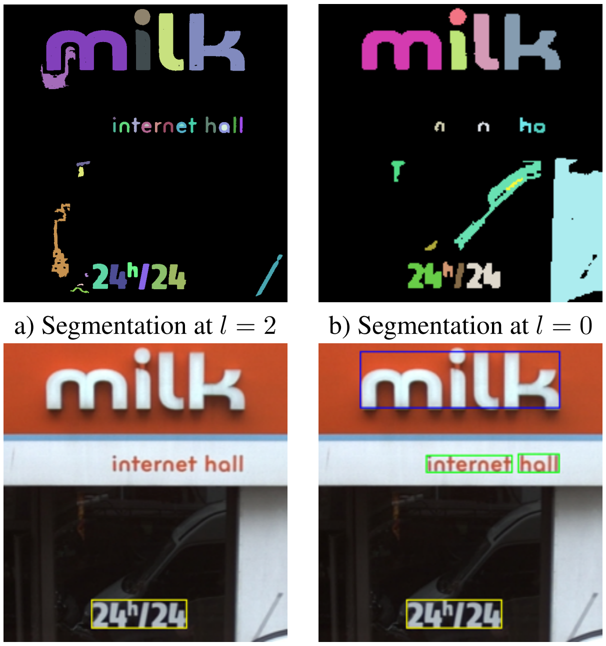


Figure 3. The proposed text detection scheme.

variations and strong background clutter, we have extended the segmentation algorithm [7] in a multiresolution fashion [10]. This is illustrated in figure 4. Each resolution level $l$ is dedicated to detect a given range of text regions scales. At coarser levels (*e.g.* $l = 2$, figure 4a)), we aim at detecting large text areas, and ignoring texture details (high frequencies). At finer levels (*e.g.* $l = 0$, figure 4b)), our goal is to detect smaller regions, analyzing more accurately the local image content. As shown in figures 4c) and 4d), using our multi-resolution scheme is able to properly detect text with large size variations, whereas a mono-resolution cannot.


a) Segmentation at $l = 2$

b) Segmentation at $l = 0$

c) Mono-resolution

d) Multi-resolution

Figure 4. Mono *v.s.* Multi-resolution segmentation.

The segmentation produces a set of homogeneous regions. We now aim at discriminating regions that contain text (characters) from those that do not. To achieve this goal, we use a classification strategy based on the extraction of shape descriptors in each image region. We have selected three families of descriptors: fourier moments, pseudo zernike moments and a new definition of a polar representation [6]. These descriptors are appealing since they are scale and rotation invariant. Then, a hierarchical SVM classifier [3] is used to discriminate characters from non-character regions. Thus, we train three different classifiers at the first level with each family of descriptors. The final decision is given by merging the previous outputs into a third SVM classifier (Figure 5).

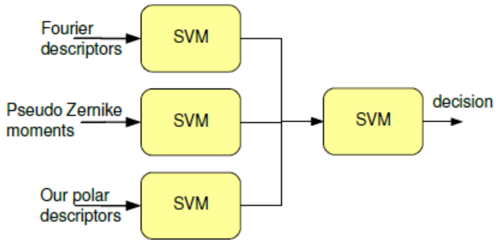


Figure 5. Character Classification.

In order to build text hypotheses, we merge neighboring recognized characters all together to recover text regions. The conditions to link two characters to each other are inspired from [12]. They are based on the distance between the two regions relatively to their height. During this process, isolated text regions (single characters) are eliminated. At the end *rectangular windows* are detected in the image. These windows are the input for the hypothesis validation step fully described in section 4.

## 4. Hypothesis validation: Fuzzy HOG

The hypothesis generation outputs a set of text window candidates. Since the classification step only analyzes the local image content around each character, false positives occur in complex urban scenes where geometric objects might be confused with characters. Some common false positives are shown in figure 6: windows, guardrail, cobblestone, *etc*. For example, the bars of the guardrail have a similar shape to a series of i's or l's. To filter out these understandable false positives, we apply an hypothesis validation step which extract a global descriptor that is complementary to those used in the hypothesis generation process. For example, in the guardrail case of figure 6, we aim at extracting features encoding periodical patterns that are not present in text regions. We train a SVM classifier having the proposed descriptor as input for our hypothesis validation purpose.

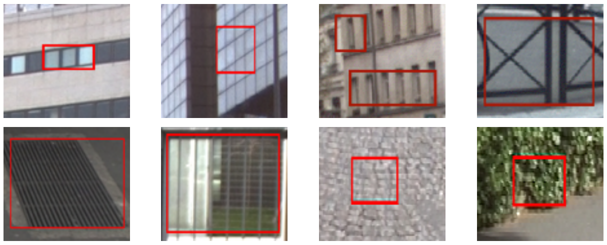


Figure 6. Hypothesis validation: example of understandable false positives from the bottom-up part of the system.

Our solution builds upon the general-purpose texture descriptor known as histogram of oriented gradients (HOG), developed by Dalal and Triggs [4], denoted as DT-HOG. The HOG descriptor is based on the idea that a particular texture can often be characterized by the distribution of the directions of the image gradient. HOG-based descriptors have been successfully used for the recognition of pedestrians [4], objects [17] and for text detection [15, 16].

We propose a novel text region descriptor, **F**uzzy **HOG** (F-HOG), that presents three main improvements with respect to HoG. As observed by Chen and Yuille [2], the top, middle and bottom parts of Roman characters have distinctive distributions of edge directions. Therefore, in the F-HOG algorithm we split the normalized image into three horizontal slices (section 4.1). For each pixel we compute the local image gradient, and we build a histogram of the gradient directions for each slice. Furthermore, the slices are not sharp-edged but "fuzzy", defined by smoothly varying weight functions $w_0, w_1, w_2$ (section 4.2).

Figure 7 gives an overview of F-HOG computation. As F-HOG is based in concatenating standard HOG in different sub-regions, we briefly recall HOG computation. HOG is based on the local gradient $\nabla I$ estimated at each pixel. Then one builds a histogram of the directions $\theta(\nabla I)$ of the gradients, weighted by $|\nabla I|$. Figure 8 shows standard HOGs of a few isolated letters, with orientation discretised to 8 bins. Note that the HOG gives the predominant orientation of the letter strokes. In particular the histogram of a rounded letter like 'O' is almost uniform over the whole range $[0, \pi]$, while that of 'I' has a spike at the directions perpendicular to the letter's stem.

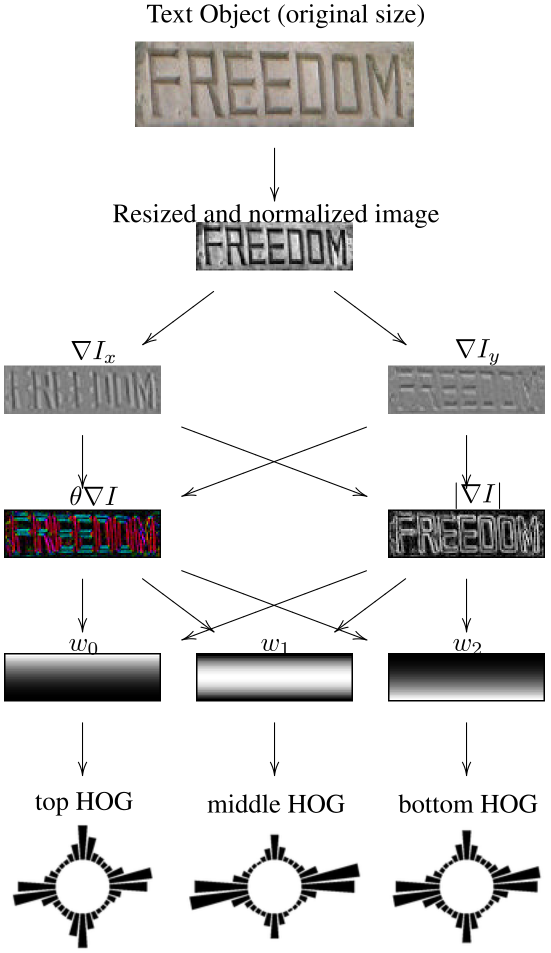


Figure 7. Fuzzy HOG text descriptor scheme. For clarity, the histograms are duplicated so as to cover the full range 0 to $2\pi$.

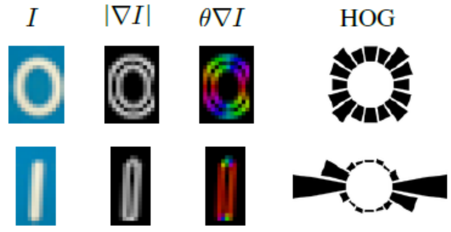


Figure 8. HOGs of some isolated letters.

### 4.1. Multi cell HOGs

Images of complex objects typically have different HOGs in different parts. Images of humans, for example have different gradient orientation distributions in the head, torso and leg regions. Therefore, in many applications, the candidate region is partitioned into an array of *cells*, and a HOG is computed separately for each cell. The concatenation of those HOGs is taken to be the descriptor of the full region.

For single-line text images, formed by Roman characters, it makes sense to analyze separately the distributions of edge directions in the top, middle and bottom parts. As shown in figure 9, it is expected that all three parts contain mostly vertical or horizontal strokes, so that the gradients orientations are predominantly 0 (or 180) and 90 (or 270) degrees. The top and bottom parts should contain a larger proportion of horizontal strokes, so that the gradients there are pointing mostly in the vertical direction. The middle part is expected to have a larger proportion of vertical strokes. In all three parts we expected to have a small amount of diagonal strokes from letters such as 'A', 'V', 'Y', etc, and from rounded parts of letters such as 'O', 'S', 'B', etc.

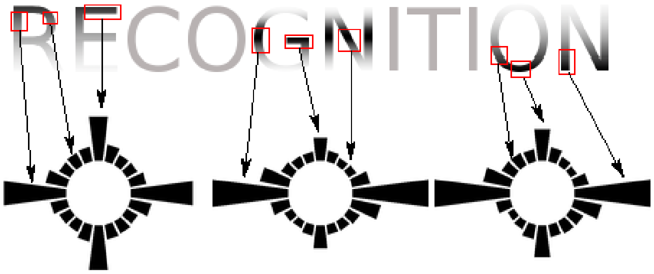


Figure 9. From left to right we have the top, middle and bottom HOGs for the text "RECOGNITION". The arrows show the contribution of specific letters strokes to the final descriptor.

### 4.2. Fuzzy cells

However, if the cells are defined by sharp boundaries, the HOG may change drastically with small vertical displacements of the text inside its bounding box. To avoid this problem, we use "fuzzy" cells, defined by *weight functions* $w_0, w_1$ and $w_2$. The weight $w_k$ of each cell is a function of the relative vertical coordinate

$$z = \frac{y - y_{\text{top}}}{y_{\text{bot}} - y_{\text{top}}} \tag{1}$$

where $y$ is the vertical coordinate of the pixel, and $y_{\text{top}}$ and $y_{\text{bot}}$ are the estimated $y$ coordinates of the top and bottom contours of the text in the image. The value of $w_k(z)$ is a number that vary smoothly between 0 and 1. When computing the histogram for cell number $k$, each pixel $(x, y)$ of the normalized text image is assumed to have mass $|\nabla I(x, y)|w_k(z)$.

Recall that each HOG is normalized to unit sum. Therefore, only the shape of each weight function $w_k$ is important. Scaling each $w_k$ by any positive factor will have no effect on the final HOG.

This model is a generalization of hard-edged cells. These can be emulated by defining each $w_k$ to be the appropriate step function. See figure 10. For fuzzy cells we tested different sets of weight functions (Gaussian bell functions, Hann, etc). The best found consists of clipped and scaled Bernstein polynomials. Namely, for $n+1$ horizontal stripes, we use

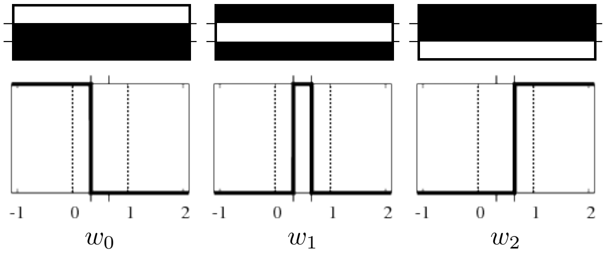


Figure 10. The step weight functions for hard-edged cells. The tic marks and the dotted lines show the ordinates $y_{\text{top}}, y_{\text{bot}}$.

$$w_k(z) = \begin{cases} 1 & \text{if } k = 0 \text{ and } z \leq 0 \\ 1 & \text{if } k = n \text{ and } z \geq 1 \\ \beta_k^n(z)/\beta_k^n(k/n) & \text{if } 0 < z < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where

$$\beta_k^n(z) = \binom{n}{k} z^k (1-z)^{n-k} \tag{3}$$

for $k = 0, 1, \ldots, n$. See figure 11.

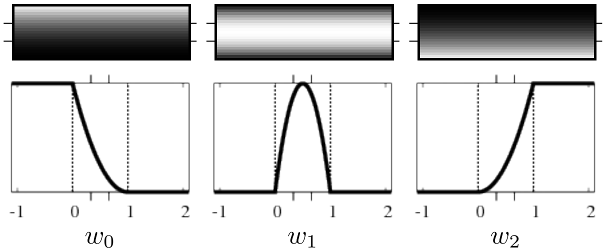


Figure 11. The Bernstein weight functions for $n = 2$.

Dalal and Triggs also used (Gaussian) weight functions in the DT-HOG [4], but in a different and more limited way. Experimentally, we verify that the F-HOG descriptor outperforms the DT-HOG in our text filtering context.

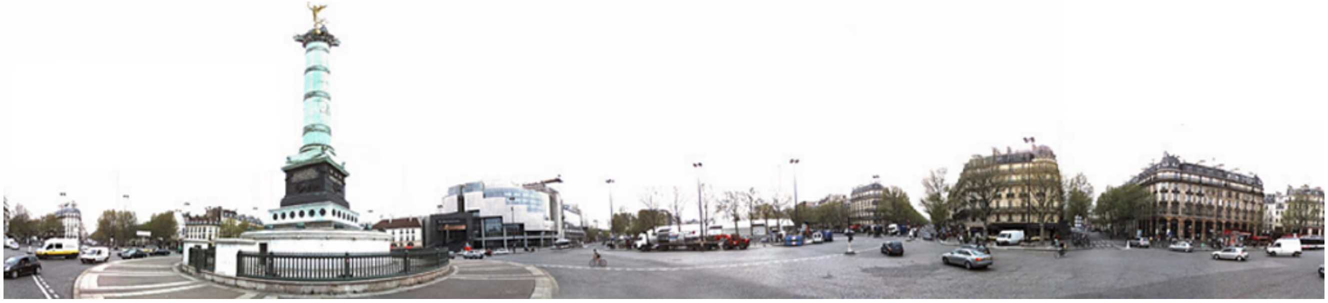


Figure 12. Panoramic street images generated in the project [10].

## 5. Text Detection: State of the Art Comparison

To compare the proposed text detector to state of the art systems, we evaluate our performances on the ICDAR 2005 dataset. It is composed of 499 color images, captured with different digital cameras and resolutions, of book covers, road signs, household objects, posters ,*etc*. Some images are shown in figure 13. To evaluate performances, we follow the protocol and use the metric described in [9]. The precision and recall were defined as: $p = (\sum_{r_e \in E} m(r_e, T))/|E|$ and $r = (\sum_{r_t \in T} m(r_t, E))/|T|$, where $m(r, R)$ defines the best match for a rectangle $r$ in a set of rectangles $R$, $T$ and $E$ are the groundtruth sets and estimated rectangles respectively. To combine the precision and recall we use the $f$ measure defined as [9]: $f = 1/(\alpha/p + (1 - \alpha)/r)$ where $\alpha$ is a weighting coefficient (set to 0.5).

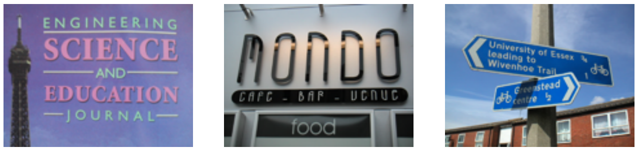


Figure 13. ICDAR 2005 Images.

The performance of our system in the ICDAR database are shown in table 1. As we can see, our system compares favorably with respect to state of the art methods. Indeed, we output the best precision (73%, results similar to Epshtein *et.al.* [5] and Chen*et.al.* [1]). In addition, we get a better recall (61% *vs* 60%). Note that the best recall reported in this database is 67% (Hinnerk Becker), but the algorithm has never been published, making a relevant comparison difficult. Our overall performance is $f = 67\%$, one 1% above the best results reported ever (66% for [5] and [1]).

| System | Precision (p) | Recall (r) | f |
|---|---|---|---|
| Our System | **0.73** | **0.61** | **0.67** |
| Epshtein [5] | **0.73** | 0.60 | 0.66 |
| Chen [1] | **0.73** | 0.60 | 0.66 |
| [10] | 0.63 | 0.61 | 0.61 |
| Hinnerk Becker [3] | 0.62 | 0.67 | 0.62 |
| Alex Chen | 0.60 | 0.60 | 0.58 |
| Ashida | 0.55 | 0.46 | 0.50 |
| HWDavid | 0.44 | 0.46 | 0.45 |
| Wolf | 0.30 | 0.44 | 0.35 |
| Qiang Zhu | 0.33 | 0.40 | 0.33 |
| Jisoo Kim | 0.22 | 0.28 | 0.22 |
| Nobuo Ezaki | 0.18 | 0.36 | 0.22 |
| Todoran | 0.19 | 0.18 | 0.18 |
| Full | 0.01 | 0.06 | 0.08 |

Table 1. ICDAR performance results.

## 6. Application to real keyword search engine

We propose to use the proposed text detector to provide a application dedicated to retrieval semantically information, through keyword search, in a real database of high-resolution street images. The image dataset has been collected in line with the project [10].

### 6.1. Project description

This project [10] 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 orphanimage, select the images that contain an object, *etc*). To achieve this work, geo-localized 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. On example of such panoramic image is shown in figure 12.

[3] the algorithm is not published

### 6.2. Performances

Figure 14 gives some detection results in the street images database from [10].

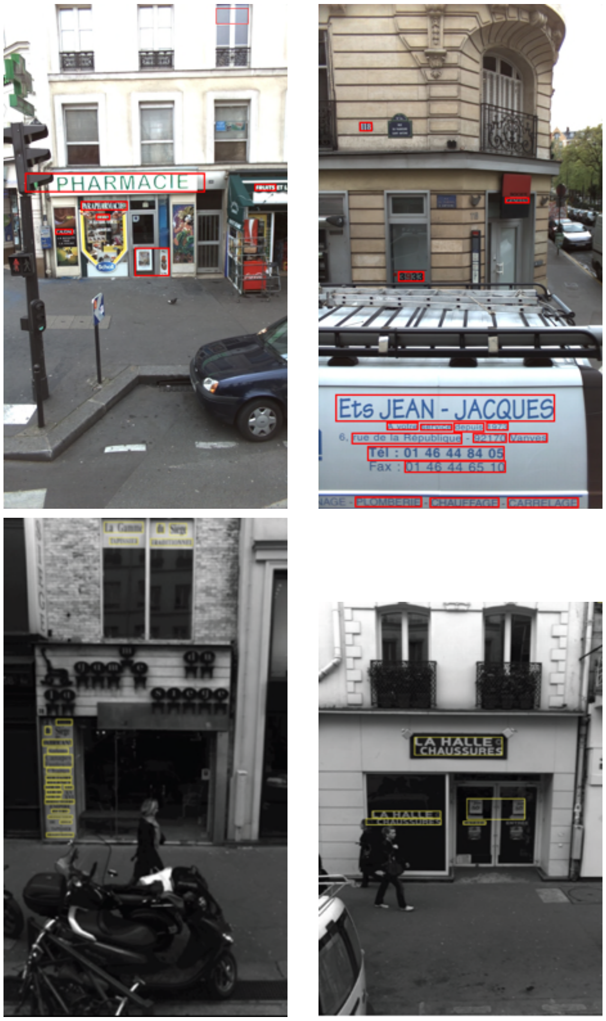


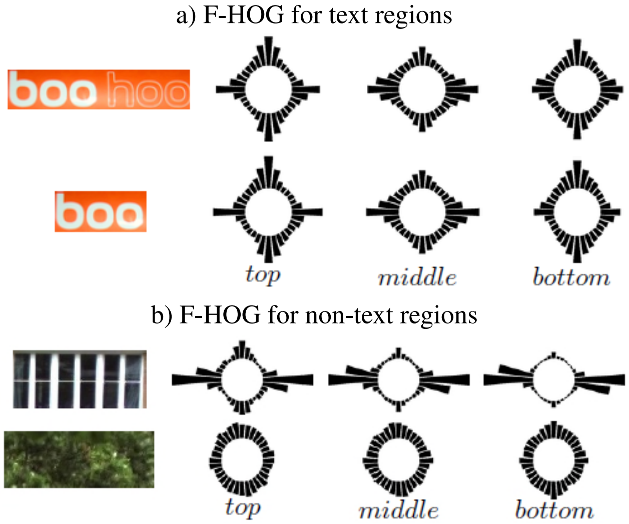
Figure 14. Detection Results on Images from [10] database

| System | Precision (p) | Recall (r) | f |
|---|---|---|---|
| Our System | **0.69** | **0.49** | **0.55** |
| [10] | 0.46 | 0.49 | 0.48 |

Table 2. [10] Project performance results.

Regarding quantitative evaluation, the improvement with respect to the results previously reported in [10] is impressive: for the same recall (r=49%), we increase the precision of 23 pt (69% *vs* 46%) As our system and [10] mainly differ with respect to the validation part, this illustrates the relevance of using F-HOG in a text filtering context, particularly its superiority with respect to the standard HOG used in [10]. This justifies the two improvements stated in section 4.1 and 4.2, namely the horizontal splitting and the fuzzy cells.

Figure 15 presents a visualization of some F-HOG descriptors for text and non-text regions. In these examples, we can notice that F-HOG clearly discriminates true positives from false positives. In addition, figure 15a) specifically points out the invariance of F-HOG with respect to text line splitting (note that the HOG's to different images are very similar). This property would not hold for a standard HOG with a vertical splitting, which clearly argues in favor of only performing a horizontal decomposition. Therefore, we claim that the proposed descriptor presents the adequate balance between invariance and discriminability for our text filtering purpose.


a) F-HOG for text regions

b) F-HOG for non-text regions

Figure 15. F-HOG of some text and non-text boxes.

### 6.3. Keyword Search Results

We run our text detector on each image, and process each extracted text box with a publicly available OCR (Tesseract). Thus, each image is represented by a set a words (strings). The user can then makes a textual query to retrieval images semantically relevant to it. The text query is then matched against each word of the database, by computing the Edit distance [8]. Each image containing a matching word is considered as relevant to the query. Figure 16 shows an example of processing the database with the query ”sushi”. We can notice that the system is able to output positive images, even with very small relevant text areas.

## 7. Conclusion

We have proposed a complete system for text detection and recognition in urban context. One contribution of the paper is to provide an efficient descriptor dedicated to text

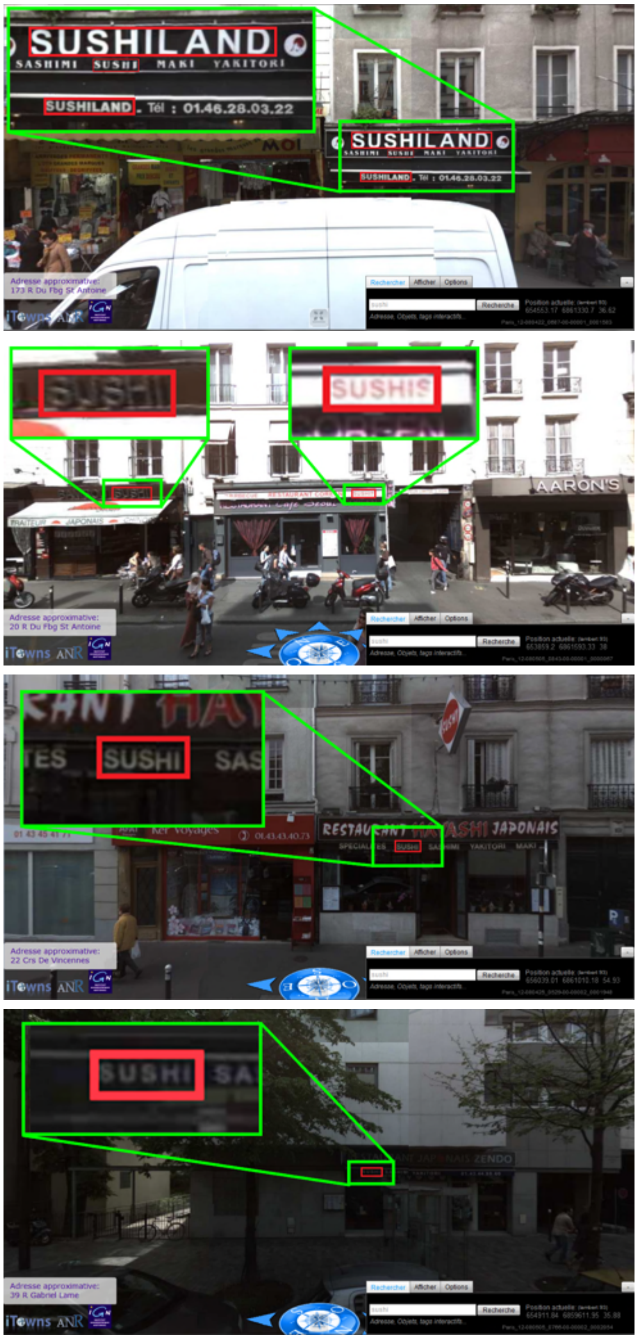

SUSHILAND
SASHIMI SUSHI MAKI YAKITORI
SUSHILAND . Tél : 01.46.28.03.22
Adresse approximative: 173 R Du Fbg St Antoine
iTowns ANR
Rechercher Afficher Options
sushi
Recherche
SUSHI
SUSHIS
AARON'S
Adresse approximative: 20 R Du Fbg St Antoine
SUSHI
RESTAURANT
JAPONAIS
Adresse approximative: 22 Crs De Vincennes
SUSHI
Adresse approximative: 39 R Gabriel Lame

Figure 16. Example of keyword search for the query ”sushi”.

filtering. The proposed F-HOG proves to be relevant to discriminate text from non-text boxes. The second contribution is to apply OCR to each detected text box to provide a keyword search tool. A thorough experimental validation proves the efficiency of the system to manage real street image databases. The main direction for future works is to optimize the computation cost of the bottom-up steps.

## References

[1] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *2011 IEEE International Conference on Image Processing*, Brussels, Sep 2011. 2, 6

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:366–373, 2004. 4

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

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, pages 886–893. IEEE Computer Society, 2005. 4, 5

[5] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2010. 2, 6

[6] J. Fabrizio, M. Cord, and B. Marcotegui. Text extraction from street level images. *City Models, Roads and Traffic (CMRT)*, 2009. 3

[7] J. Fabrizio, B. Marcotegui, and M. Cord. Text segmentation in natural scenes using toggle-mapping. *IEEE ICIP*, 2009. 2, 3

[8] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady.*, 10(8):707–710, February 1966. 7

[9] S. Lucas. Icdar 2005 text locating competition results. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 80–84 Vol. 1, 2005. 2, 6

[10] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui. Snoopertext: A multiresolution system for text detection in complex visual scenes. pages 3861–3864, 2010. 1, 2, 3, 6, 7

[11] P. W. Palumbo, S. N. Srihari, J. Soh, R. Sridhar, and V. Demjanenko. Postal address block location in real time. *Computer*, 25(7):34–42, 1992. 1

[12] T. Retornaz and B. Marcotegui. Scene text localization based on the ultimate opening. *ISMM*, 1:177–188, 2007. 3

[13] J. Serra. Toggle mappings. *From pixels to features*, pages 61–72, 1989. J.C. Simon (ed.), Elsevier. 2

[14] N. Thome, A. Vacavant, L. Robinault, and S. Miguet. A cognitive and video-based approach for multinational license plate recognition. *Machine Vision and Applications*, March 2010. 1

[15] X. Wang, L. Huang, and C. Liu. A new block partitioned text feature for text verification. *International Conf. on Document Analysis and Recognition (ICDAR)*, 0:366–370, 2009. 4

[16] J. Zhang and R. Kasturi. Text detection using edge gradient and graph spectrum. *International Conference on Pattern Recognition (ICPR)*, 0:3979–3982, 2010. 4

[17] W. Zhang, G. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 4