Robust Text Detection.

Abstract—In this paper, we describe SnooperText, a robust and accurate multiresolution approach to detect and classify text regions in photos and videos of urban scenes. At each resolution scale, our detector first segments the image to detect candidate character regions, based on their size and aspect ratio. Neighboring character-like regions are then merged into candidate text lines by size and position criteria. These regions are then validated by a texture classifier, based on F-HOG, a fuzzy version of the histogram of oriented gradients (HOG) descriptor. SnooperText outperforms other published methods on standard image benchmarks.

Index Terms—Text recognition, histogram of oriented gradients, text descriptor, text tracking.

1 Introduction

Text detection is a challenging task in computer vision, with many potential applications such as traffic monitoring, geographic information systems, road navigation, and scene understanding. No efficient solution yet exists for arbitrary text that is part of an unstructured 3D environment (such as store names, traffic signs, and indoor sign).

In this paper, we describe SnooperText, a robust and accurate text detector for these tasks. Our detector was developed in the context of the iTowns project [1], which aims to build tools for virtual navigation of urban environments. Towards this goal, a car-mounted camer is used to obtain a geo-localized set of high-resolution digital photos, with mean viewpoint spacing of one meter between sets. Each set of images is assembled offline into a complete imersive panorama. See figure 1. The purpose of SnooperText within this project is to extract semantic information (offline) from the images themselves, which may then be integrated with cartographic databases, geo-localization data, business directories, and other information sources. This data is necessary for intelligent navigation and high-level queries, such as locating a street or storefront view given the address, store name, or other textual information.

1.1 OCR for unstructured 3D scenes

OCR algorithms designed for scanned documents perform very poorly on photos of 3D scenes. See figure 2. The reasons include extreme text size and font variations, tilted or curved baselines, strong background clutter and difficult illumination conditions. Much better results are obtained by applying the OCR algorithm to the output of a generic text detector, as illustrated in figure 3.

- *R. Minetto, N.J Leite and J. Stolfi are with Institute of Computing of University of Campinas (UNICAMP), 13984-971, SP, Brazil. E-mail: rodrigo.minetto@ic.unicamp.br, neucimar@ic.unicamp.br, stolfi@ic.unicamp.br*
- *R. Minetto, N. Thome and M. Cord are with Laboratoire d'Informatique Paris 6 (LIP6), Universite Pierre et Marie Curie, UPMC-Sorbonne Universities, 4 place Jussieu, 75005, Paris, France. E-mail: minettor@lip6.fr, nicolas.thome@lip6.fr, matthieu.cord@lip6.fr.*

This work was partly supported by FAPESP (Phd grant 07/54201-6), CAPES/COFECUB (Brazil/France cooperation) 592/08 and ANR 07-MDCO-007-03.

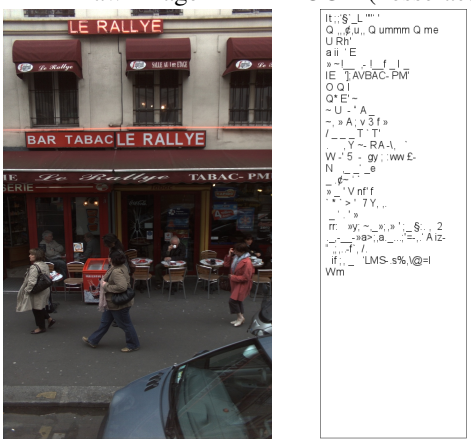

Fig. 2. Attempt to extract text from a storefront photo using a public OCR (Tesseract).

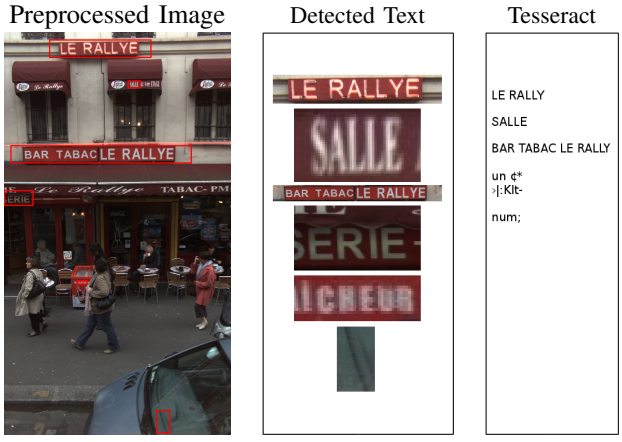

Fig. 3. Our proposed text detector & Recognition (Tesseract) in Urban Context. There are many readable words and few noise.

1.2 Our detector

Our text detector, called SnooperText [7], is an hybrid scheme combining bottom-up and top-down strategies (as defined in section **??**). Regarding methodology, our approach relies on the hypothesis generation/validation paradigm. The hypothesis

Fig. 1. Panoramic street images generated in the iTowns project [1].

generation is carried out using a bottom-up approach. Candidate character regions are obtained by adaptive segmentation and identified by their size and aspect ratio, then by a binary shape recognizer, based on a combination of three geometric descriptors and a generic support vector machine (SVM) classifier [**?**]. Tentative character regions are then grouped by position and size criteria into rectangular regions believed to contain single words or lines of text. These regions are further filtered by a novel text/non-text classifier, based on *histogram of oriented gradients* (HOG) descriptors.

In order to deal with the strong character size variations in urban context, we run the basic algorithm at multiple scales of resolution. A final step eliminates or merges any conflicts (overlapping candidate text regions that were found at different scales).

The remainder of the paper is organized as follows. ⋆[**Check!**] Section 2 presents the overall system for text detection. Section 2.2 and 2.3 point out the two methodological area of novelty of the paper. Section **??** shows that SnooperText is very competitive being among the state-of-the art systems in the ICDAR dataset. Finally section 4 concludes the paper and proposes directions for future works.

2 Overview of the Method

The whole scheme of SnooperText is shown in figure 4. As previously mentioned, our system generates a set of text hypotheses, and validates them using a complementary strategy. Regarding hypothesis generation our algorithm is composed of three main steps: image segmentation, character classification, and character grouping.

2.1 Segmentation

The segmentation step is based on a morphological operator, *toggle mapping*, introduced by Serra [**?**]. Toggle mapping is a generic operator which maps a function on a set of n functions and is generally used for contrast enhancement and noise reduction. Our segmentation procedure is based on the algorithm of Fabrizio *et al.* [**?**], which obtained second place in the recent ICDAR 2009 Document Image Binarization Contest [**?**]. The segmentation with the toggle mapping is done by means of morphological erosions and dilations. The advantage of this approach is that it can efficiently detect image boundaries necessary to recognize each image character.

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 [4]. These descriptors are appealing since they are scale and rotation invariant. Then, a hierarchical SVM classifier [**?**] is used to discriminate characters from non-character regions. Thus, we train three different classifiers at the fisrt level with each family of descriptors. The final decision is given by merging the previous outputs into a third SVM classifier (Figure 4).

In order to build text hypotheses, we developped a grouping step where all recognized characters are grouped all together with their neighbours to recover the text regions. The conditions to link two characters to each other are those given in [**?**]. They are based on the distance between the two regions relatively to their height. During this process, isolated text regions (single characters) are eliminated. This aggregation is mandatory to generate words and sentences to integrate as an input in an OCR, but it also suppresses a lot of false positive detections. At the end *rectangular windows* are detected in the image. These windows are the input for the hypothesis validation step fully described in section 2.3.

⋆

[**To be expanded**]

2.2 Multiresolution Scheme

⋆

[**To be fixed/updated**]

In principle, fonts of different sizes could be handled by processing the image with morphological structuring elements of different sizes. In complex scenes like urban images where text scale may highly vary, this method is likely to fail, especially in cluttered background with textured areas or local illumination variations. Thus, large text areas with texture are prone to over-segmentation, while small text regions might be missed.

In order to efficiently handle fonts of arbitrary size, we apply the basic algorithm (segmentation, text hypothesis generation, and filtering) in a multiresolution fashion. Each resolution level is dedicated to detect text regions with a given range of font sizes. This appraoch has the added benefit that, at coarser

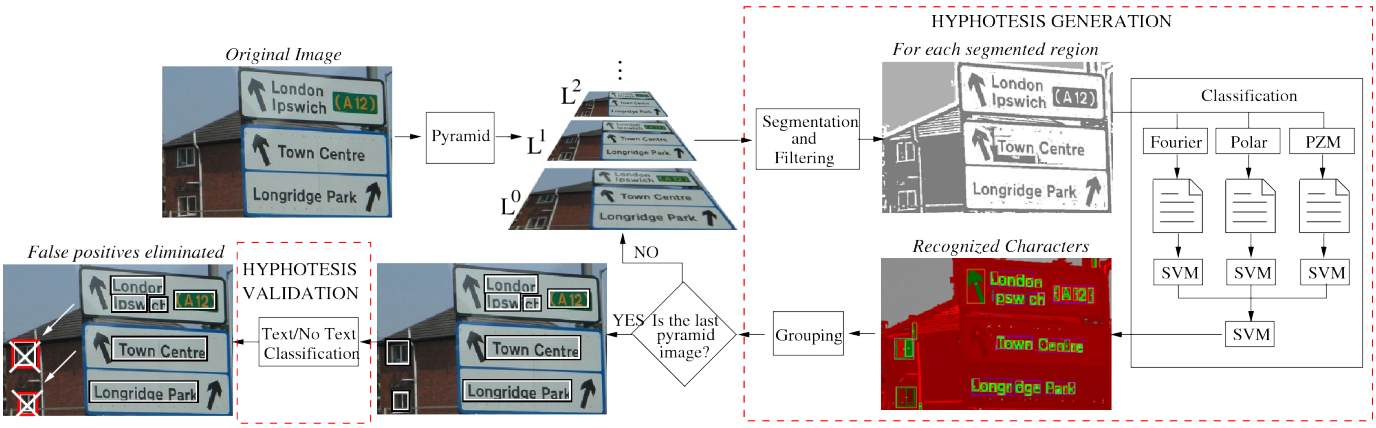

Fig. 4. SnooperText scheme.

levels, small texture details (high frequencies) are eliminated. In addition, the multiscale processing is much more efficient than using larger structuring elements in the segmentation. At finer levels, our goal is to detect smaller regions, analyzing more accurately the local image content.

More formally, let us consider a pyramid image I^l, $l \in \{0; L-1\}$, as defined in the multiresolution framework [**?**]. At resolution level l, the image is decimated with a ratio of 2, leading to an image size decreased by 4^l (with respect to the initial image). Many solutions can be adopted for relating resolution levels to text region scales. In this paper, we propose a simple yet effective technique, using a fixed size over scales. Thus, at resolution level l, we aim at detecting text regions with size $s_l \in [m_l; m_l + \delta_l]$. In addition, we define a constant c_l corresponding to the overlap between two consecutive scales s_l and s_{l+1} (see figure 5), so that:

$$m_{l+1} = \frac{m_l + \delta_l - c_l}{4} \quad (1)$$

As δ and c are constant over scales, we have: $\delta_l = \delta_0/4^l$ and $c_l = c_0/4^l$, and m_l can be computed as follows:

$$m_l = \frac{m_0 + l(\delta_0 - c_0)}{4^l} \quad (2)$$

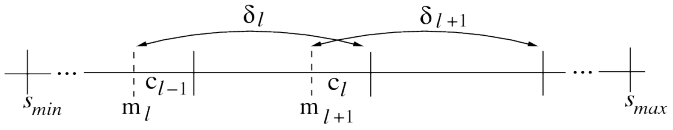

Fig. 5. Region sizes managed at different resolution levels.

Therefore, if we require to detect text regions with size $s \in [s_{min}; s_{max}]$ (where $m_0 = s_{min}$), with a given overlap c_0 in each level and L pyramid images, δ_0 must fulfill : $\delta_0 = (s_{max} - s_{min} + c_0(L-1))/L$.

Practically, our multiresolution algorithm processes as follows. First, we build a set of L pyramid levels, and run the segmentation algorithm of [**?**] in each downsampled image. At level l, a set of regions r_l^i ($i \in \{1; N_l\}$) are extracted. Only regions whose area s_l^i fall into the bound $[m_l; m_l + \delta_l]$ are considered for the following processing steps, others are ignored. Figure 6 illustrates our multiresolution algorithm. Figure 6(a) shows the image segmentation in the original image resolution ($l = 0$), while Figure 6(b) shows the segmentation in a coarser image resolution ($l = 2$). As we can see in Figure 6(c), when $l = 0$, small text regions are found (yellow windows), when $l = 1, 2$ bigger text regions are found due to the scale intervals computed in equation (2) and thanks to the texture removeal. Figure 6(d) shows the results of the monoresolution approach [4], that fail at detecting the word RIESCOPAM, due to its texture and color.

2.3 Hypothesis validation

⋆

[Insert short summary of F-HOG in this section]

The F-HOG descriptor is then converted to a binary text/non-text decision by an SVM classifier with a Gaussian χ^2 SVM kernel K.

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 false positives are shown in figure 7: *e.g.* the bars of the guardrail have a similar shape to a sequence of i's.

To deal with these false positives, we apply an hypothesis validation step relying on global image descriptors over the detected windows. These global descriptors are complementary to those used in the hypothesis generation process. For example, in the guardrail case of figure 7, we aim at extracting features encoding periodical patterns that are not present in text regions. In this work, we used the Histograms of Oriented Gradients (HOG) descriptors [**?**]. HOG descriptors are based on the idea that local object appearance and shape can be well characterized by the distribution of local intensity gradients or edge directions. HOG descriptors proved to reach state of the art performances for object recognition (*e.g.* pedestrian detection [**?**]), and have been recently used for text detection [**?**].

To achieve good performances, the HOG extraction in [**?**] is based on splitting a given window into $N \times M$ cells. This tiling allows to capture spatial information. In addition, in order to be robust to local illumination variations, a contrast normalisation

(a) Image segmentation when $l = 0$

(b) Image segmentation when $l = 2$

(c) Multiresolution results: detections at scale $l = 0$ in yellow, $l = 1$ in green and $l = 2$ in white

(d) Monoresolution results

Fig. 6. Multiresolution system results, 3 scales and $L = 2$, (a, b and c) and monoresolution results (d). The word "RIESCOPAM" is detected in the multiresolution approach but not in the monoresolution (d).

is applied. Groups of $N' \times M'$ adjacent cells are grouped into larger spatial blocks, with an overlap of K cells. Each cell is then normalised with respect to each of its surrounding block. In our experiments, we use $N = M = 4$, $N' = M' = 2$ and $K = 2$. From the extracted feature, we train a SVM classifier to discriminate text from non-text windows. We use a Gaussian chi2 kernel, and perform a cross-validation to optimize its standard deviation σ parameter. Figure 7 illustrates some non-text windows that have been successfully discarded after our hypothesis validation step.

3 Experimental Validation

In this section we determine performance of SnooperText on three publicly available benchmarks, and compare it with other state-of-the-art text detector systems.

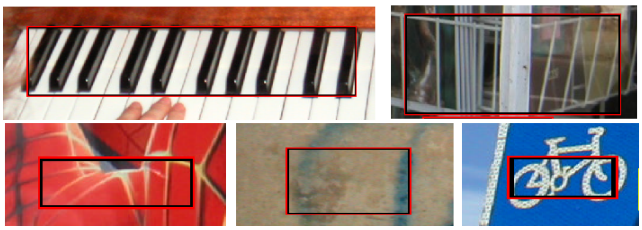

Fig. 7. Hypothesis validation: the windows above were correctly identified as non-text by our validation step.

3.1 Benchmarks

Specifically, we use:

1) The 2005 ICDAR challenge collection [6], consisting of 499 color images, captured with different digital cameras and resolutions, of book covers, road signs, household objects, posters, *etc*.
2) The Epshtein *et al.* benchmark [3] with 307 color images of urban scenes, ranging from 1024×1360 to 1024×768 pixels, taken with hand-held cameras.
3) A subset of the iTowns Project collection [1], [7], consisting of a hundred 1080×1920 color images of Parisian façades taken by a camera-equipped vehicle (similar to Google's Street View images).

3.2 General parameters

For all tests, the character recognizers were trained on the same database of single characters provided by Fabrizio *et al.*. The character recognition and grouping parameters [7] were set as follows: minimum character size $\lambda = 5$ pixels, minimum number of characters per group $GOC = 2$. For the F-HOG text recognizer, we use the following parameters [**?**]: candidate regions were extracted and rescaled to a fixed height $H = 21$ pixels with Lanczos interpolation [**?**], and normalized for contrast and brightness by assuming noise deviation $\varepsilon = 0.02$. The histograms of gradient orientation (assumed to range over $[0, 2\pi]$) were computed for a grid of fuzzy cells with $n_x = 1, n_y = 7$ and $n_b = 9$, resulting in a descriptor of size $N = 63$. The full descriptor was then scaled to unit L_1 norm.

For each benchmark, we first executed the SnooperText algorithm without the F-HOG filter on the given training set, which produced a large number of false as well as true positives. We then manually classified the reported regions as text or non-text, and used this dataset to train the SVM classifier of the F-HOG filter. The standard deviation parameter σ of the SVM kernel was optimized by cross-validation.

3.3 Performance metrics

To quantify the performance of the various algorithms, we used the protocol and metrics described in [6]. We start with the ICDAR 2005 measure of similarity [**?**] between two rectangles r, s, defined as

$$m(r, s) = \frac{S(r \cap s)}{S(r \cup s)} \tag{3}$$

Fig. 8. On left the SnooperText* text detection results (boxes). On right the results of SnooperText* + F-HOG text filtering. Images from Epshtein et al and iTowns datasets.

where $S(t)$ is the area of the smallest rectangle enclosing the set t. The function $m(r, s)$ ranges between 0 (if the rectangles are disjoint) and 1 (if they are identical). The metric m is extended to a set of rectangles Z by the formula

$$m(r, Z) = \max\{m(r, s') : s' \in Z\} \tag{4}$$

From this indicator one derives the ICDAR *precision* p and *recall* r scores [**?**]

$$p = \frac{\sum_{r \in E} m(r, T)}{\#E} \qquad r = \frac{\sum_{r \in T} m(r, E)}{\#T} \tag{5}$$

where T is the set of manually identified text regions in the input images, and E is the set of text regions reported by the detector. For ranking purposes, the ICDAR 2005 committee used the f *measure* [**?**] which is the harmonic mean of precision and recall,

$$f = 2/(1/p + 1/r) \tag{6}$$

There are several ways of averaging these metrics over a multi-image database. The approach used by the ICDAR 2005 scoring program (method I) is to evaluate p, r and f separately for each image, and then compute the arithmetic mean of the f-scores over all images. Another approach (II) is to compute p and r for each image, then take the arithmetic means of all p and r values, and compute f from these means. We note that the first method suffers from higher sampling noise and a negative bias compared to the other two. These points must be considered when comparing f values reported by different authors.

3.4 Results

3.4.1 ICDAR benchmark

We compared the performance of SnooperText on the ICDAR collection with the published scores of all detectors that that took part in the ICDAR 2003 and 2005 Challenges [**?**]. We also considered the detectors of Tian *et al.* [9], H. Chen *et al.* [2] and Epshtein *et al.* [3], which did not take part in the Challenge but reported even higher f-scores (higher than the best ICDAr entry) on that collection.

For this benchmark, we followed the ICDAR Challenge protocol: namely, used the "training" subset of the ICDAR

collection to train the SnooperText algorithm (specifically, the SVM of the F-HOG filter), and the ICDAR "testing" subset to evaluate its performance. The results are shown in table 1. All p and r scores were computed with the ICDAR scoring program [6]. The scores in the f_I and f_{II} columns were averaged by methods I and II, respectively. Observe that

System	p	r	f_I	f_{II}
ST2	0.73	0.61	0.65	0.67
Yi and Tian [9]	0.71	0.62	0.62	0.66
H. Chen *et al.* [2]	0.73	0.60	—	0.66
Epshtein *et al.* [3]	0.73	0.60	—	0.66
Hinnerk Becker†	0.62	0.67	0.62	0.64
ST3	0.64	0.59	0.59	0.61
Alex Chen†	0.60	0.60	0.58	0.60
ST2	0.42	0.65	0.47	0.51
Ashida†	0.55	0.46	0.50	0.50
HWDavid†	0.44	0.46	0.45	0.45
Wolf†	0.30	0.44	0.35	0.36
Qiang Zhu†	0.33	0.40	0.33	0.36
Jisoo Kim†	0.22	0.28	0.22	0.25
Nobuo Ezaki†	0.18	0.36	0.22	0.24
Todoran†	0.19	0.18	0.18	0.19
Full†	0.01	0.06	0.08	0.02

TABLE 1
Performances of various text detectors on the "testing" subset of the ICDAR image collection. The competitors of the ICDAR 2003 and 2005 Challenges are marked with †.

SnooperText achieves the same precision (73%) as the best published methods [3], [2], [9], but with a better recall (61% *vs.* 60%).

3.4.2 Epshtein benchmark

For the Epshtein benchmark, we compared SnooperText against the stroke-width transform algorithm of Epshtein *et al.* [3], the only one for which performance data was available. Since the reported performance was measured over the entire image collection, we used ICDAR and iTowns collections to train our F-HOG filter. The results are shown in table 2.

System	p	r	f_I	f_{II}
ST2	0.59	0.47	0.49	0.52
Epshtein *et al.* [3]	0.54	0.42	—	0.47

TABLE 2
Performances the SnooperText detector and of the Epshtein *et al.* detector on the whole Epshtein image collection.

3.4.3 iTowns benchmark

For the iTowns benchmark, there are no published performance figures for any other algorithm, so we give those of our algorithm only. For this test, we used the ICDAR and Epshtein image collections to train F-HOG filter, and the entire iTowns collection to evaluate the SnooperText performance. The results are shown in table 3.

System	p	r	f_I	f_{II}
ST2	0.72	0.50	0.56	0.59

TABLE 3
Performance the SnooperText on the whole iTowns image collection.

4 Conclusion

We have proposed a complete system for text detection in complex natural images. Focussing first on character segmentation, filtering and grouping, we generate text region hypotheses. This process is embedded in a multiresolution scheme to handle text regions of various sizes. A validation step exploiting region signature based on texture analysis allows to filter a lot of false positives. We have evaluated our scheme in two databases, achieving very good results. As shown in experiments, our multi-scale approach significantly improves text detection performances with respect to a single-scale approach.

References

[1] iTowns ANR project. http://www.itowns.fr.

[2] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. *IEEE International Conference on Image Processing (ICIP)*, pages 1–4, 2011.

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

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

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

[6] Jonathan Fabrizio, Matthieu Cord, and Beatriz Marcotegui. Text extraction from street level images. *City Models, Roads and Traffic (CMRT)*, 2009.

[7] Jonathan Fabrizio, Beatriz Marcotegui, and Matthieu Cord. Text segmentation in natural scenes using toggle-mapping. *IEEE ICIP*, 2009.

[8] Basilis Gatos, Konstantinos Ntirogiannis, and Ioannis Pratikakis. Document image binarization contest (dibco). In *ICDAR*, pages 1375–1382, 2009.

[9] S.M. Hanif, L. Prevost, and P.A. Negri. A cascade detector for text detection in natural scene images. In *19th ICPR*, pages 1–4, Dec. 2008.

[10] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision: Multiresolutional Computer Vision*. 1994.

[11] Simon M. Lucas. Text locating competition - icdar (2003-2005). http://algoval.essex.ac.uk:8080/icdar2005/.

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

[13] Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Jonathan Fabrizio, and Beatriz Marcotegui. Snoopertext: A multiresolution system for text detection in complex visual scenes. *IEEE International Conference on Image Processing (ICIP)*, pages 3861–3864, 2010.

[14] Thomas Retornaz and Beatriz Marcotegui. Scene text localization based on the ultimate opening. *ISMM*, 1:177–188, 2007.

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

[16] Ken Turkowski. Graphics gems. Filters for common resampling tasks. pages 147–165. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[17] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing (TIP)*, PP(99):1, 2011.