17^{th} *Computer Vision Winter Workshop*
Matej Kristan, Rok Mandeljc, Luka Čehovin (eds.)
Mala Nedelja, Slovenia, February 1-3, 2012

Text Localization in Unconstrained Images

Georg Poier, Jürgen Hatzl, and Stefan Kluckner
Siemens Corporate Technology, Graz, Austria
{georg.poier,juergen.hatzl,stefan.kluckner}@siemens.com

Peter M. Roth and Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{pmroth,bischof}@icg.tugraz.at

Abstract. *Text localization is the first step when automatically reading text in images. Since existing methods often fail when applied to unconstrained images, in this paper we propose a more robust approach exploiting different kind of information. In particular, we first extract textural features, a combination of a stroke filter with a super-pixel segmentation, and then search for connected components. To finally obtain a text localization, these are subsequently analyzed for unary character properties, binary character similarities, and text line properties. To demonstrate the benefits of the proposed method, we evaluate it on three different data sets, showing promising results.*

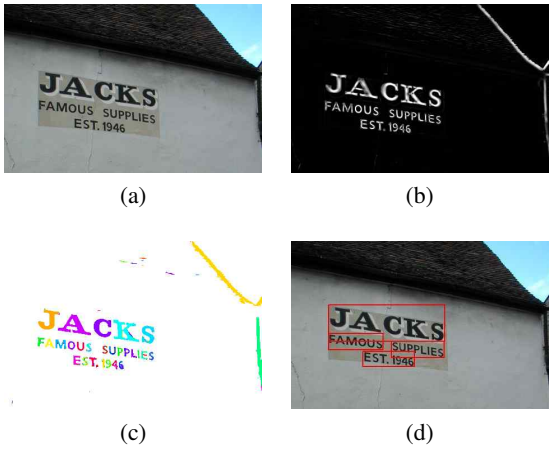


Figure 1: Text localization process: The input image (a) is filtered for strokes (b) and is segmented into small coherent regions. The stroke regions (c) are then analyzed for connected components to extract the textual properties (d).

1. Introduction

Text extraction or *Optical Character Recognition* (OCR) has been studied since the 1990s [24], however, the focus was primary on reading scans of printed documents. Thus, only little work has been done on reading text in natural images, but this topic gained more and more attention during the last five to ten years [3, 7, 16, 23, 24]. In contrast to OCR from documents, extracting text from unconstrained images is rather challenging and suffers from a lot of problems. These include low resolution, low contrast, different text colors, unknown text size, unknown orientation, color bleeding, or unconstrained backgrounds [23, 24].

The most successful approaches can be subdivided into two groups: methods that describe texture features [8, 26] and approaches that are based on extraction and analysis of connected components (CCs) [3, 18, 25]. Methods of the first group can cope better with noisy, degraded, or complex text and background, however, they usually encounter difficulties in exact localization of the text regions, which is crucial for the subsequent character recognition. Thus, for standard benchmark data sets such as ICDAR 2003 typically approaches of the second group yield the better results. However, all of these methods often fail for small or blurred text as well as transparent background [3, 16, 25], which are typical scenarios in practice.

To make the text localization more robust, we propose to combine both ideas, i.e., texture and region based methods. A stroke filter, based on the ratio of eigenvalues of the *Hessian* matrix, is used to

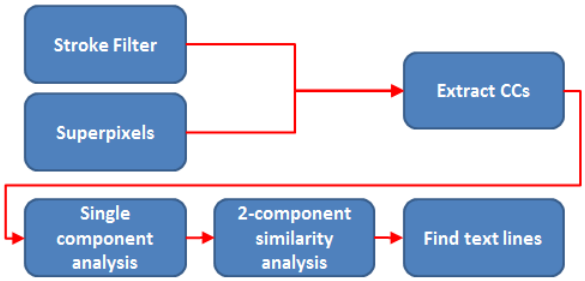


Figure 2: Flowchart of the proposed algorithm: we use a stroke filter in combination with coherent super-pixels to extract the connected components which finally form the text lines.

enhance text regions, whereas an unsupervised segmentation provides coherent image segments. The segments in enhanced regions are merged based on semantically meaningful clusters in order to extract CCs. These are then analyzed for single character properties as well as relations between neighboring characters, forming text lines. The whole process is illustrated in Figure 1 and a flowchart of our algorithm is given in Figure 2.

Bearing in mind that text localization is only the first step when reading text in natural images, we use a fairly conservative setup, focusing on high recall instead of building a rather perfectly specific detector. False positives may be easily rejected in later steps but not localized text is retrieved laboriously only.

The paper is structured as follows. First, we summarize the related work in Section 2. Then, in Section 3 we introduce our approach. Experimental results and a discussion are given in Section 4 and Section 5, respectively. Finally, we summarize and conclude the work in Section 6.

2. Related Work

Although there are some slightly varying categorizations [18, 21, 26], methods for text localization are usually divided into two categories [3, 7, 8, 24]: texture- and region-based methods.

Texture-based methods work in a *top down* manner, extracting texture features in order to discriminate between text and background. Thereby, e.g., spatial variance of intensity, Fourier transform, Wavelets, and machine learning methods are used [2, 7, 8, 24]. These methods perform well with noisy, degraded, or complex text and background. However, since an image usually has to be scanned at several scales, major restrictions arise from computational complexity and integration of results from different scales. In addition, such methods suffer from difficulties to provide exact localizations.

Region-based methods work in a *bottom up* fashion. They utilize the observation that text can be seen as sets of separate CCs. The image is separated into small regions, e.g., by using color features, edge features, or CC methods [16, 18, 24]. Next, text regions are separated from background by extracting and analyzing region properties – often including an analysis of relations between neighboring regions. These methods are efficient in extracting text, especially when the background is homogeneous. Region-based methods allow for simultaneously detecting text at different scales. However, the main drawbacks of such approaches come from noisy, multicolored, textured as well as very small texts. Problems also arise from coalesced character regions or similarly colored objects.

One very prominent region-based approach is that of Epshtein et al. [3]. It is based on the observation that strokes are nestled in edges exhibiting roughly opposite gradients. CCs are extracted from the output of the so called *Stroke Width Transform* (SWT), which seeks to find the value of stroke width for each image pixel. Components are then analyzed for character and text line properties.

An approach based on similar ideas is the text extraction scheme of Zhang and Kasturi [25]. After extracting closed boundaries by an edge detector, "character energy" and “link energy” are calculated and subsequently combined to retrieve the probability of a candidate text region to be a true positive. Therefore, "character energy" is based on gradient directions at two "opposing" points on the boundary, and “link energy” is computed to model the similarity between two neighboring character regions.

Matas and Zimmermann [15] proposed to use character classification based on MSERs [14], which were later also used by Neumann and Matas [16, 17] in their comprehensive framework for text localization and recognition. In [16] a *Support Vector Machine* (SVM) is trained on manually annotated MSERs for character classification and again for text line formation. In [17] the text line formation stage is replaced by *Gaussian Mixture Models* (GMMs) describing relational properties of two, three or more consecutive characters. In order to find the text lines a graph energy, computed from the distances between the extracted properties and the corresponding GMMs, is minimized.

Another approach based on similar ideas, where parts of our work build on, was presented by Tran et al. [21]. The authors compute the ratio of the eigenvalues of the *Hessian* matrix to detect character strokes and text lines, respectively. But in contrast to our work, they aim to extract ridges [20]. Text is required to induce a ridge at a coarse scale representing its center line and an amount of short ridges at smaller scales representing skeletons of characters.

Liu et al. [11] define *Haar*-like stroke filters, where three adjacent rectangular regions are analyzed. To compute a response for stroke regions, they estimate the three regions intensity means and the standard deviation of the central region. An application to text localization based on the stroke filter is proposed in [6]. For this purpose, the filter is combined with an SVM trained on normalized gray values and constant gradient vector.

Another remarkable approach is that of Chen and Yuille [2]. They use a boosted cascade of weak classifiers and carefully designed different kinds of features for different stages of the cascade.

Of course, even more machine learning based approaches exist (see [7, 24]). However, it is difficult to learn an appropriate model, having to incorporate such a vast amount of variation. This is also outlined by Zhang and Kasturi in [25], where they claim that such approaches still suffer from the insufficiency of the training samples from different lightnings, distortions, and languages.

3. Proposed Approach

In the following, we introduce our approach for text localization, which can be divided into three main parts: the stroke filter, extraction of coherent regions, and connected component analysis. First, we explain how the stroke filter is modeled (Section 3.1). We then give a brief description of the mechanism for super-pixel segmentation (Section 3.2). Finally, we discuss how the filter response is combined with the image segmentation in order to obtain connected components, which are subsequently analyzed for textual properties (Section 3.3).

3.1. Stroke Filtering

The most elementary and constitutive parts of text are strokes. This applies to all types of text, independently of character set or font style. Furthermore, every kind of stroke exhibits a few general properties: it is an elongated region, which is nearly homogeneous and different from its lateral regions. Based on this properties a generic stroke filter can be built. We thus use a filter based on the analysis of the eigenvalues of the Hessian matrix.

Local Curvature Analysis The Hessian represents the second order derivatives, and, thus, describes the local curvature associated to a point in an image. By analyzing its eigenvalues and eigenvectors, the principal directions of the local curvature can be extracted. This directly gives the direction of smallest curvature, which corresponds to the direction along a potential stroke. In such a case the corresponding eigenvalue is strongly dominated by the eigenvalue corresponding to the orthogonal direction.

Thus, to analyze the local structure of an image I, we consider the Taylor expansion in the neighborhood of a point $\mathbf{x}$:

$$I(\mathbf{x}+\delta\mathbf{x}, s) \approx I(\mathbf{x}, s) + \delta\mathbf{x}^T\nabla_s + \delta\mathbf{x}^T\mathcal{H}_s\delta\mathbf{x}, \quad (1)$$

where ∇_s and $\mathcal{H}_s$ are the gradient vector and Hessian matrix of the image computed in $\mathbf{x}$ at scale s. This represents an approximation of local image structure up to second order.

Given the eigenvalues λ_1 and λ_2, where $|\lambda_1| \leq |\lambda_2|$, for pixels within a stroke region λ_1 will be low whereas λ_2 will be high. In contrast, if both eigenvalues have a similar magnitude, this indicates that the corresponding region belongs to the background. Furthermore, the sign of λ_2 specifies whether the pixel belongs to a dark region on bright background or vice versa.

Hessian Based Stroke Filtering Using the properties of the Hessian, Frangi et al. [4] proposed a so called "vesselness" measure, which – for 2D images – is based on two values: the ratio between the eigenvalues, and their magnitude. The ratio $\mathcal{R}_\mathcal{B} = \lambda_1/\lambda_2$ gives a measurement of "blobness", whereas the magnitude of the eigenvalues $\mathcal{S} = \|\mathcal{H}\|_F = \sqrt{\lambda_1^2 + \lambda_2^2}$ is used to minimize the influence of random background noise. In regions of low contrast (e.g., background regions) eigenvalues will be small and, therefore, $\mathcal{S}$ will be low. In high frequency regions, on the contrary, at least one of the eigenvalues will be high and, thus, the norm becomes larger.

For computing a normalized response, the following combination of the mentioned components was proposed [4]:

$$\mathcal{V}(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp\left(-\frac{\mathcal{R}_{\mathcal{B}}^2}{2\beta^2}\right)\left(1 - \exp\left(-\frac{\mathcal{S}^2}{2c^2}\right)\right) & \text{otherwise,} \end{cases} \quad (2)$$

where β and c are parameters to control the sensitivity of the filter to the measures of $\mathcal{R}_{\mathcal{B}}$ and $\mathcal{S}$.

When analyzing $\mathcal{V}(s)$ at different scales s, it will be maximum at a scale that approximately matches the stroke to detect. The integration of the responses for different scales therefore considers only the maximum response for each image pixel: $\mathcal{V} = \max_s \mathcal{V}(s)$.

Using (2) enables us to enhance bright strokes on dark background. In contrast, when searching for dark strokes on bright background, we have to consider the sign of λ_2. It is positive for dark-on-bright and negative for bright-on-dark structures. Hence, for filtering dark strokes on bright background, we simply have to exchange the corresponding constraint in (2).

A stroke filter response is computed over two separate scale-ranges. This is necessary since a single response for all scales would blend responses in an undesirable manner, and, thus, causes problems when extracting CCs; especially, for finer structures. See Figure 1(b) for an output of our stroke filter at the smaller scale-range.

3.2. Extract Coherent Segments

The subsequent OCR demands for accurately localized text regions. This is only achievable by a rather exact knowledge about the expansion of single characters, and therefore requires segmentation on the pixel level. State-of-the-art methods utilize local adaptive binarization [18], edge based segmentation [3, 25], or MSERs [16, 17] to this end.

Since, in many situations this is a fairly critical task, we propose to use small, coherent regions from a conservative unsupervised segmentation (*superpixels*). In this way, we get nearly perfectly segmented characters, while drastically reducing computational cost in subsequent steps.

There exist a great number of superpixel segmentation methods, but in particular, we use *quick shift*[1] [5, 22], because it can easily be parallelized on a graphics processing unit (GPU). Figure 1(c) and Figure 3(c) show examples for thus obtained segmentations. Please be referred to [22] for further details.

[1]For our implementation we used the code provided by Andrea Vedaldi and Brian Fulkerson (http://www.vlfeat.org/).

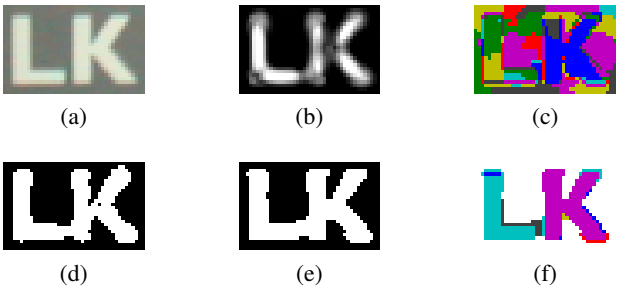


Figure 3: Example for extraction of CCs, showing images of intermediate steps. The original image (a) is filtered for strokes, obtaining a normalized response (b) where lighter colors correspond to higher responses. Furthermore, an unsupervised segmentation is applied yielding small coherent regions. Here, for better visualization a color is randomly assigned to each segment (c). From these segments – in combination with regions exhibiting a minimum filter response (d) – foreground regions are extracted (e), which are subsequently refined/disconnected based on color statistics (f).

3.3. Connected Component Analysis

Knowing the location of strokes as well as having a superpixel segmentation on hand enables us to restrict subsequent computations to the superpixels on stroke regions.

Extraction of Connected Components When considering CCs of superpixels on stroke regions, we observe that components are likely to contain more than one character (See Figure 3 for an example). Thus, merged character regions are disconnected based on color statistics of the compound superpixels.

More specifically, a k-means clustering of the superpixels' mean colors is employed, followed by a region growing on the superpixels based on calculated cluster memberships. Assuming that the color values in high contrasted regions are much wider spread than in regions of low contrast, different contrast conditions are taken into account by defining the merging criterion for region growing based on the expansion of data points within a cluster.

Examination for textual properties Having obtained the connected components, where each single character is represented by a single connected component, we sift out text regions in three steps. First, the components are filtered based on single character properties. Then, neighboring components are linked and filtered based on two component similarities, respectively. Finally, links which do not exhibit relational properties of three or more characters (i.e., text lines) are removed.

For identifying components which certainly do not contain characters, stroke width measurements bear great potential [3, 25]. We are able to obtain such information directly from the stroke filter.

This is based on the observation that the filter response will be maximum at a scale that approximately matches the stroke to detect. Thus, the finally considered measurements for single character properties include aspect ratio, minimum/maximum stroke width to area ratio, area of bounding box to median stroke width ratio, and variance of stroke widths.

From the remaining components a neighborhood graph is built. Components are considered as neighbors if their geometric and spatial relationship fulfills certain conditions. Similar to [18, 25] any two components i and j obeying the following rule are considered to be neighbors:

$$\|c_i - c_j\|_2 \leq 2\ min\big(max(w_i, h_i), max(w_j, h_j)\big), \tag{3}$$

where c specifies the centroid, and w and h are component width and height, respectively.

Neighboring components are now checked against a number of binary relationship properties, including color, height, width, stroke width, overlap of bounding boxes, and number of pixels. Furthermore, inspired by [21], we utilize the relationship between angle of the link between the two components and directions of their strokes.

For every three subsequently linked components we then evaluate the *centroid angle* as proposed in [17]. This is based on the observation that the angle between lines connecting the centroids of three following letters in a text line is limited. Finally, links between text lines are removed by checking the angle between two components against the median angle of the graph it belongs to.

Integration of Results

Results from the four cues (bright and dark strokes, each at two scale-ranges) are finally integrated. Thereby, each text region is assigned a confidence based on stroke filter response. With highly overlapping text regions, the region exhibiting smaller confidence is removed.

4. Experimental Results

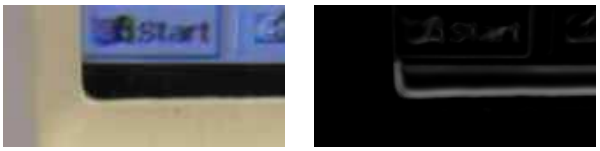
To demonstrate the benefits of the proposed approach, we performed experiments on three different data sets: the *ICDAR 2003* evaluation dataset[2], a dataset consisting of TV screen captures (in the remainder referred to as "*VideoOCR*"), and a third one containing images of railroad wagons ("*WagonID*").

The text regions in all data sets are subject to many different sources of variation, such as contrast, size, color, orientation, or degradation. In addition, we are facing different challenges with the different data sets. For instance, images from *VideoOCR* dataset are often of low quality or low resolution, and, thus, the text is small and degraded. The *WagonID* dataset is acquired from a line scan camera, as a result, color channels may exhibit varying displacements throughout an image. Furthermore, the text sometimes shows very low contrast, or is even strongly contaminated.

Since there were no annotations available for the *VideoOCR* dataset, we performed a quantitative evaluation on the *ICDAR 2003* and the *WagonID* dataset only. The annotations for *ICDAR 2003* contain rectangles representing the bounding boxes for each single word. Since our task doesn't require separation of text lines into single words, the output of our algorithm is a set of rectangles containing whole text lines. Hence, an evaluation as proposed for the *ICDAR 2003* dataset [13] wouldn't give meaningful results, as the ground-truth rectangles are thereby paired with the detected ones.

Instead, in our evaluation we compare the ground-truth and detection on pixel level. For each image we have a set of ground-truth pixels $\mathcal{T}^p$ (i.e., the pixels within the annotated rectangles) and a set of detected pixels $\mathcal{D}^p$. The recall is then defined as the number of correctly detected pixels divided by the number of ground-truth pixels: $Recall = \frac{|\mathcal{T}^p \cap \mathcal{D}^p|}{|\mathcal{T}^p|}$.

[2] http://algoval.essex.ac.uk/icdar/Datasets.html



(a) (b)

Figure 4: Text region ("Start") not localized by our approach due to the low response of the stroke filter. The original image (a) and the filter response for corresponding scale-range (b) obtained with standard parameters (used throughout all evaluations) are shown. This is mainly due to the strong degradation.

Since, in terms of precision, we are not interested in the number of falsely detected pixels, but the number of falsely detected regions (i.e., the rectangles), the precision is defined on a region basis. A detected rectangle is counted as correct if the overlap between the rectangle and at least one ground-truth rectangle is more than 50%. With respect to this, the precision score represents the number of correctly detected rectangles, denoted c, divided by the number of detected rectangles $\mathcal{D}^r$: $Precision = \frac{c}{|\mathcal{D}^r|}$.

In this way, for the *WagonID* dataset a recall of 0.84 at a precision of 0.42 is obtained. Detailed results for the ICDAR 2003 dataset are given in Table 1. Please note that the results of our method are not directly comparable to the other ones listed, due to the different evaluation procedure. Still, those should act as reference points. Selected results from all data sets are shown in Figures 5, 6, and 7.

Algorithm	Precision	Recall
Pan et al. [18]	0.67	0.71
Zhang and Kasturi [25]	0.73	0.62
Epshtein et al. [3]	0.73	0.60
Neumann and Matas [17]	0.60	0.60
proposed method	**0.48**	**0.71**

Table 1: Detection results on the ICDAR 2003 dataset.

5. Discussion

The evaluation shows that our method is able to handle a lot of the mentioned variations, even in cases where existing methods fail. This is observed especially with very small, degraded or blurred text, where edge based approaches often encounter problems [3, 25].

Some false negatives come from transparent background, showing similar colors as the foreground (Figure 5), too low stroke filter response (Figure 4), and single or strongly connected characters, respectively (Figure 6). The latter one results from the fact that a minimum number of characters is required for a text line. Text lines, where even the filter encounter problems may be addressed through performing checks on the frequency of strokes within the stroke filter response instead on extracted components, only. Missed text regions on the *WagonID* dataset are mainly due to the high level of degradation, primarily caused by the shift of color channels, in combination with small text size.

The arising high number of false positives could be reduced by a number of measures. Specificity of the stroke filter may be increased by using contour information from the border of strokes, where gradients must roughly be opposite[3]. The analysis of components against character and text line properties could be extended further and has potential to significantly improve the overall performance. More specifically, in analysis of single components one could make use of further simple features as well as more complex descriptions, e.g., based on an *Implicit Shape Model* [10] for the specially shaped elements, characters are composed of. Repetitive patterns (Figure 5) may be removed by the shape filter proposed in [12] for this task, or some even more sophisticated method as those regarding local *self similarities* [19].

6. Conclusion

We presented a hybrid method for text localization, based on textural features and a connected component analysis. We aimed to overcome problem cases, where recently proposed methods, based on a preceding edge detection step, are likely to fail [3, 25]. Thus, we used a filter design which doesn't obligatorily require edges to be present at stroke borders and utilized its output in combination with a superpixel segmentation of the image in order to obtain connected components. We finally end up by analyzing them against single character properties, as well as text line properties. Promising results prove the potential of the presented approach.

[3]Similar ideas are used with eigensystem analysis of the Hessian in the field of medical image analysis [1, 9].

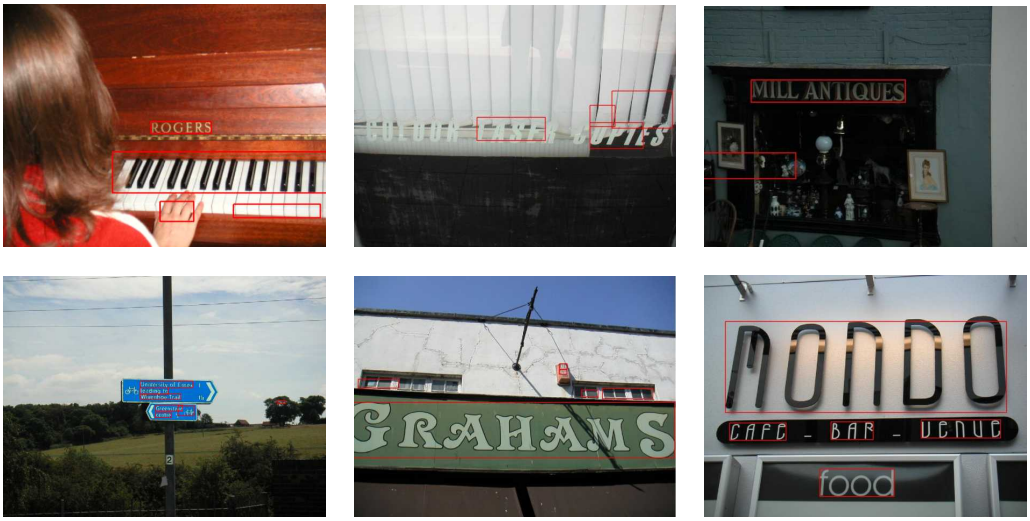


Figure 5: Selected qualitative results obtained for the evaluation on the ICDAR dataset [13].

Figure 6: Selected results on the VideoOCR dataset. Note that even the timestamps of only one pixel stroke width, added to some of the images (upper left corners), as well as text exhibiting a slightly curved bottom line are correctly localized.

Figure 7: Selected results on the WagonID dataset. Results are generated on images downscaled by half.

References

[1] C. Bauer, T. Pock, E. Sorantin, H. Bischof, and R. Beichel. Segmentation of interwoven 3d tubular tree structures utilizing shape priors and graph cuts. *Medical Image Analysis*, 14(2):172–184, 2010. 6

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proc. CVPR*, 2004. 2, 3

[3] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. CVPR*, 2010. 1, 2, 4, 5, 6

[4] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale Vessel Enhancement Filtering. In *Proc. MICCAI*, 1998. 3, 4

[5] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proc. ICCV*, 2009. 4

[6] C. Jung, Q. Liu, and J. Kim. A stroke filter and its application to text localization. *Pattern Recognition Letters*, 30(2):114 – 122, 2009. 3

[7] K. Jung. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977–997, 2004. 1, 2, 3

[8] K. I. Kim, K. Jung, and J. H. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Trans. PAMI*, 25:1631–1639, 2003. 1, 2

[9] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-based detection of tubular structures in 3d images. *Computer Vision and Image Understanding*, 80(2):130–171, 2000. 6

[10] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008. 6

[11] Q. Liu, C. Jung, S. kyun Kim, Y. Moon, and J. Y. Kim. Stroke filter for text localization in video images. In *Proc. ICIP*, 2006. 3

[12] Z. Liu and S. Sarkar. Robust outdoor text detection using text intensity and shape features. In *Proc. ICPR*, 2008. 6

[13] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Proc. ICDAR*, 2003. 5, 7

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, 2002. 2

[15] J. Matas and K. Zimmermann. A new class of learnable detectors for categorisation. In *Proc. SCIA*, 2005. 2

[16] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. ACCV*, 2010. 1, 2, 4

[17] L. Neumann and J. Matas. Estimating hidden parameters for text localization and recognition. In *Proc. CVWW*, 2011. 2, 4, 5, 6

[18] Y.-F. Pan, X. Hou, and C.-L. Liu. Text localization in natural scene images based on conditional random field. In *Proc. ICDAR*, 2009. 1, 2, 4, 5, 6

[19] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. CVPR*, 2007. 6

[20] T. T. H. Tran and A. Lux. A method for ridge extraction. In *Proc. ACCV*, 2004. 3

[21] T. T. H. Tran, A. Lux, H. L. Nguyen T, and A. Boucher. A novel approach for text detection in images using structural features. In *Proc. ICAPR*, 2005. 2, 3, 5

[22] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proc. ECCV*, 2008. 4

[23] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. ICCV*, 2011. 1

[24] J. Zhang and R. Kasturi. Extraction of text objects in video documents: Recent progress. In *Proc. IAPR Workshop on DAS*, 2008. 1, 2, 3

[25] J. Zhang and R. Kasturi. Character energy and link energy-based text extraction in scene images. In *Proc. ACCV*, 2010. 1, 2, 3, 4, 5, 6

[26] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang. Text from corners: A novel approach to detect text and caption in videos. *IEEE Trans. on Image Processing*, 20(3):790–799, 2011. 1, 2