A Head-mounted Device for Recognizing Text in Natural Scenes

Carlos Merino-Gracia*, Karel Lenc† and Majid Mirmehdi‡
*Neurochemistry and Neuroimaging Laboratory, University of La Laguna, Spain
†Center for Machine Perception, Czech Technical University, Czech Republic
‡Visual Information Laboratory, University of Bristol, UK

***Abstract*—We present a mobile head-mounted device for detecting and tracking text that is encased in an ordinary flat-cap hat. The main parts of the device are an integrated camera and audio webcam together with a simple remote control system, all connected via a USB hub to a laptop. A near to real-time text detection algorithm (around 14fps for 640×480 images) which uses Maximal Stable Extremal Regions (MSERs) for image segmentation is proposed. Comparative results against the ICDAR 2003 text locating competition database along with performance figures are presented.**

***Keywords*-wearable device, text detection, text understanding, MSER**

I. Introduction

The area of wearable computing has seen relatively little growth over the last few years after the initial wave of enthusiasm in the area, mainly due to the miniaturisation of personal computing devices, such as mobile phones that need not be worn, but carried, that perform most of our everyday needs. Also, the focus of recent advances in wearable computing have been in specific and specialist areas, e.g. in health monitoring systems. Regardless of this, wearable devices for everyday and general purpose use are still extremely important to help those most in need of it, e.g. disabled users such as the blind, or those incapacitated by language barriers, e.g. tourists!

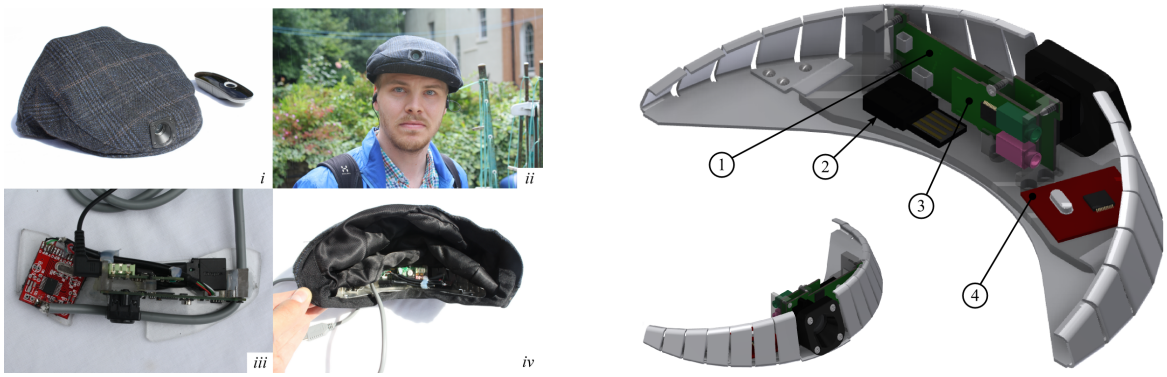
In this work, we present a simple hat, with embedded camera, speaker, and USB port (see Fig. 1) for an application that involves the real-time detection and tracking of text. The camera provides real-time video, via a discreetly hidden USB cable, to a small laptop (to be carried) where the number crunching occurs. The results of text detection and recognition is returned to the hat via an audio signal on the USB port to a speaker embedded in the hat (which can be used with earphones if necessary). All electronic components are off-the-shelf and are held in a part which is readily removable from the hat. This allows us to easily extend the device in the future just by adapting the removable part, for example with an embedded computer which will be able to handle all the computation. Since the device does not require units integrated with shades or spectacles, it does not interfere with users who have some residual vision.

Helping visually impaired people to understand the scene in their surrounding environment is a major goal in computer vision, with text detection and its communication to the user a significant aspect of it. One of the earliest approaches can be considered to be the assistive technology approach by Kurzweil's reading machine [1] in 1975 which enabled book reading for blind people using a flat CCD scanner and computer unit with optical character recognition (OCR) and text to speech synthesis (TTS) systems. Several desktop solutions with a similar design are still widely available. This layout was improved using a camera, for example in the iCARE portable reader [2] which made document manipulation less cumbersome. In Aoki et al. [3], a small camera mounted on a baseball cap was used for user navigation in an environment. Chmiel et al. [4] proposed a device comprising glasses with integrated camera and DSP-based processing unit which performed the recognition and speech synthesis tasks. However, this device was directed mainly towards document reading for the blind. The SYPOLE project [5] designed a tool primarily intended for reading text in the user's natural environment by taking snapshots of documents, e.g. banknotes, via a camera mounted on a hand-held PDA device.

In the context of other application areas, detecting and recognizing text is important for translation purposes, e.g. for tourists or robots. This is a subject of interest for the Translation robot [6] which consists of a camera mounted on reading glasses together with a head-mounted display used as the output device for translated text.

Text detection has received increasing attention in recent years, with many works surveyed in [7] and [8]. An example of a recent approach is Pan et al. [9] who combined classic region-based and connected components-based (CC) methods into a complex text detection system and achieved the best results on the ICDAR dataset yet (used for performance measurement by many text detection algorithms). Their system binarized the image in the first stage based on a text confidence map, calculated from classified gradient features of different sized regions. Segmented CCs were then classified using learned condition random field parameters of several unary and binary component features. Another recent example is Epshtein et al. [10] who used the stroke-width transform to obtain candidate text regions formed of CC pixels of similar stroke widths.

Contrary to the degree of attention enjoyed by text detection, text tracking has been hardly investigated considering that it is very important for reasonable user interaction in any



(a) (b)

Figure 1. Developed device together with remote control (a-*i*) and its shape when used (a-*ii*). Removable part (a-*iii*) is placed inside a hat (a-*iv*) in a metal framework which is visualized in image (b). The device comprises a USB camera with auto focus (1), a RC receiver (2) and a USB sound card (3) which are connected to a USB hub (4).

text detection system involving ego or object motion. In our previous work [11], we developed a real-time probabilistic tracker based on particle filtering which is used in the proposed text detection system here. We are only aware of one other work, Myers and Burns [12], who tracked text by feature correspondence across frames by correlating small patches. While we have developed our text tracking application beyond what we previously reported in [11], the focus of the work presented here is on text detection and on the hat-based communication device. Our most recent results on text tracking will be presented in a future work.

In this paper we examine the use of Maximally Stable Extremal Regions (MSER) [13] for text detection. Originally developed as a method to detect robust image features, the method responds well to text regions. MSER has been used for license plate detection [14] and more recently, Neumann and Matas [15] used MSERs in a supervised learning system for text detection and character recognition using SVM classifiers. Although this method yielded promising results it is computationally expensive. Our approach is based on MSER as a candidate text region detector but we rely on the hierarchical relationship between detected MSERs to quickly filter through them (Section III). Then a cascade of text classifying filters is applied to candidate text regions. Using much simpler text classification techniques allows us to provide a close to real-time implementation. We present single image text detection performance results evaluated against the standard ICDAR 2003 text locating competition database. Performance figures are provided to illustrate the efficiency of the algorithm (Section IV) [1].

II. Hardware Design

Placing a camera in a hat is a logical choice as it is both an unobtrusive location and an ideal position in reference to where the eyes and head point to. Mayol et al. [16] examined possible positions of wearable cameras and concluded that head mounted cameras provide the best possible link with the user's attention. The hardware proposed here allows its integration into many varieties of hats, here we have used an ordinary fashion accessory – a *flat-cap*.

The hardware was developed with emphasis on robustness, serviceability and visual appearance. Fig. 1a shows the appearance of the completed device. It is composed of a fixed part and a removable one. The fixed part is an aluminium plate, bent into a shape that very loosely follows the curves of the hat, while protecting the space used by the removable part. It has an opening in the front side, protected by a glass cover, which fits onto the camera lens. This provides dust and, to some degree, weather insulation. The removable part holds all the electronic devices. It is built out of commodity hardware, with a total cost of all the components under 100€: a high definition web camera with adjustable focus (Logitech Quickcam Pro 9000), an USB sound card used for voice feedback to the user through a pair of connected headphones, a RF transceiver and an USB hub. A view of the inner part of the hat, with the removable part and the electronic parts is shown in Fig. 1b.

The device is controlled by a hand-held remote control which acts like an ordinary USB keyboard. To minimize the number of cables, all the devices are connected to a generic USB hub which allows connecting the hat to any USB-enabled *computing device*, from tablets to fully equipped laptop computers, with a single cable.

III. Proposed System

A simplified schematic of the proposed system is shown in Fig.2. Initially, we detect candidate text regions in the image input stream using our MSER-based approach. These regions are then tracked in consecutive frames and are eventually analysed using the open source Tesseract OCR

[1] Additionally, example videos recorded using the hat can be downloaded from: http://www.cs.bris.ac.uk/home/majid/CBDAR/

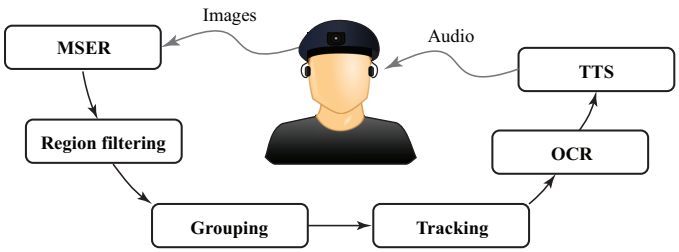


Figure 2. General structure of the text detector application.

engine[2] integrated into our software. Recognized text regions above a significant confidence measure determined by the OCR engine are then sent to a text-to-speech synthesis module (Flite TTS[3], also integrated into our software).

A. Text detection

In our previous work on text detection and tracking [11] we used adaptive thresholding to initially binarize the original image. Then a tree was constructed representing the topological relationship between CCs in the binary image. A key step of the algorithm was a hierarchical filtering of the tree nodes, which allowed the rejection of many candidate regions without classification. After that, the remaining tree nodes where filtered using a cascade of text classifiers.

The approach proposed here uses Maximally Stable Extremal Regions [13] for image segmentation along with hierarchical filtering similar to our previous work.

1) Image segmentation: MSERs are regions of interest in an image which present an extremal property of the intensity function around its contour. When applying a varying threshold level to a grey scale image, CC regions in the thresholded image evolve: new regions appear at certain levels, regions grow and eventually join others. Those regions which keep an almost constant pixel count (area) for a range of threshold levels are called MSERs. This technique, originally proposed as a distinguished region detector, also presents very desirable properties when applied to text detection, such as stability and multiscale detection.

MSERs can also be obtained by filtering the *component tree* of the source image, as shown by Donoser et al. [17]. The component tree is a representation of all the CCs which result from applying a varying threshold level to a grey scale image. The CCs are laid out in a hierarchy representing the topological relationship between them. A stability factor – i.e. the rate of change in the area of the components – is computed for each node in the component tree. MSERs are identified as local minima of the stability factor along paths in the tree towards the root.

We use the efficient, linear time MSER algorithm by Nister et al. [18], which crucially also constructs the component tree. We make two passes on the original image.

[2]Tesseract OCR: http://code.google.com/p/tesseract-ocr/
[3]Flite TTS: http://www.speech.cs.cmu.edu/flite/

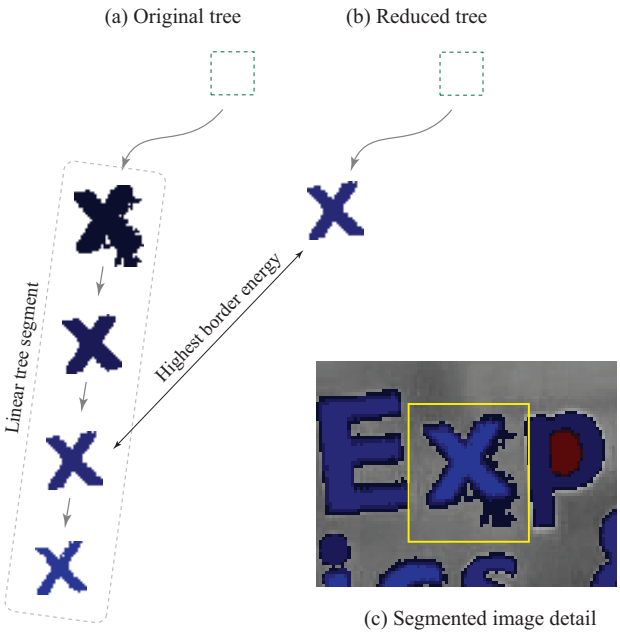


Figure 3. Linear tree segments removal.

First, MSER+ regions are obtained by applying the MSER algorithm on the image. This produces light regions inside dark ones. Then MSER- regions are obtained by applying the MSER algorithm to the inverse (negative) of the original image which produces dark regions inside light ones. The sets of regions returned by each pass are disjoint and both passes are needed to detect light text on dark backgrounds and dark text on light backgrounds. The algorithm can be easily modified to return a *hierarchical MSER* tree; an example output can be seen in Fig. 4b where blue regions were obtained by the MSER+ pass and the red regions by the MSER- pass. Darker regions represent upper tree nodes (closer to the root), while brighter regions show lower nodes (closer to the leaves). With hierarchical MSER, we have the desirable properties of MSERs as a distinguished region finder applied to text detection. Additionally, we keep the topological relationship of the CCs, which provides context information for later text filtering stages.

The resulting hierarchical MSER tree is then pruned in two stages: (1) reduction of linear segments and (2) hierarchical filtering. The first stage identifies all the linear segments within the tree where a linear segment is a maximum path between two tree nodes without any branches in between. Likewise, it is a path starting with a node with only one child, and ending with a branch node (a node with more than one child), or a leaf. Each linear segment is then collapsed into the node along the path, as shown in Fig. 3, which maximizes the *border energy* function (see below). In the second stage, the tree is walked depth-first, and a sequence of text classifying filters is applied to leaf nodes. Any non-leaf node without any descendant node classified

(a) original

(b) hierarchical MSER

(c) filtered MSER

(d) grouping results

Figure 4. Output from different stages of the text detection algorithm.

as text is also tested with the text classifying filters. This stage is similar to the hierarchical tree filtering we originally proposed in [11]. Fig. 4 shows the output of several stages of our text detection algorithm.

2) Region filtering: During the tree walk, candidate regions are passed through a cascade of filters, i.e. *size*, *aspect ratio*, *complexity*, *border energy* and *texture*. This arrangement means that most of the regions will be discarded by the simpler filters, thus reducing the number of regions examined by the more complex tests. Thus, for a region i:

Size - the simplest condition filters out regions whose boundary length falls outside an allowed interval $(l_{min}; l_{max})$. The interval limits are fixed as a function of the image size:

$$l_{\min} < |\mathbf{B}_i| < l_{\max} \tag{1}$$

where $\mathbf{B}_i$ is the set of points around the region's boundary.

Aspect Ratio - given width W_i and height H_i of candidate region i, this condition rejects regions that are too wide or too narrow:

$$a_{\min} < \frac{W_i}{H_i} < a_{\max} \tag{2}$$

Complexity - this is a simple measurement of region complexity. It measures the ratio between the region boundary length and its area A_i. This criterion filters out regions with a rough border, which are usually produced by noise:

$$\frac{|\mathbf{B}_i|}{A_i} < c \tag{3}$$

Border Energy - this is a measure of contrast against the background. It filters out regions with low average edge response (from a Sobel operator (S_x, S_y)) around its boundary set of points $\mathbf{B}_i$, i.e. the region is valid only if its border energy exceeds a threshold:

$$\frac{1}{|\mathbf{B}_i|} \sum_{(x,y)\in\mathbf{B}_i} \sqrt{(S_x(x,y)^2 + S_y(x,y)^2)} > e \tag{4}$$

Texture measure - the last filter in the sequence is a measurement of texture response, as text regions usually contain high frequencies. We found that the LU transform [19] yields good response results when applied to text regions. It is a simple transformation based on LU decomposition of square image sub-matrices A around each interest point.

$$A = P\ L\ U \tag{5}$$

where L and U are lower and upper diagonal matrices and the diagonal elements of L are equal to one. Matrix P is a permutation matrix. In the LU decomposition, the number of zero diagonal elements of U is in direct proportion to the dimensionality of the null-space of A.

The actual texture response $\Omega_p(l, w)$ is calculated as the mean value of the diagonal values of the U matrix.

$$\Omega_p(l, w) = \frac{1}{w - l + 1} \sum_{k=l}^{w} |u_{kk}|, \qquad 1 < l < w \tag{6}$$

where w is the window size and l number of skipped lower frequency values. The texture response T_i of a region i is calculated as the mean LU transform value of a sampled set of points (N_i) inside the bounding box of the region.

$$T_i = \frac{1}{|N_i|} \sum_{p\in N_i} \Omega_p(l, w) \qquad T_i > t \tag{7}$$

Figure 5. LU transform output on an example image.

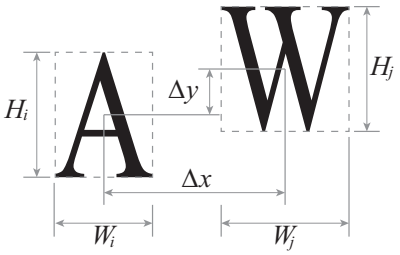


Figure 6. Variables used for text grouping.

Fig. 5 shows the output of the of LU transform on an example image. In all the filters above, the thresholds were determined empirically and fixed in all our experiments to: $a_{\min} = 0.1$, $a_{\max} = 5$, $c = 1.4$, $e = 40$ and $t = 1.9$.

3) Perceptual text grouping: After the image segmentation step, which produces a set of candidate text regions (usually representing isolated letters), a perceptual grouping step is performed to join them into candidate words and phrases. First, a planar Delaunay graph is constructed joining the centre of gravity of every text region. Each vertex of the graph represents a single text region, while the edges represent proximity relationships. Next, each edge e is then filtered using a sequence of tests.

Edge angle - The first test looks at the angle between edges and the horizontal axis ($\alpha(e)$), such that,

$$-45^\circ < \alpha(e) < 45^\circ \tag{8}$$

This is a strong limitation but the majority of text is horizontal or with a slight slope. The angle of the text is also limited by the capabilities of the OCR engine used, as for now we are not performing any perspective correction.

Relative position of adjacent regions - The following criteria were inspired by the work of Ezaki et al. [20]. Two letters appearing on the same text line are usually close together. In this test we limit the allowed distance, relative to their respective sizes.

$$\Delta x < r_x \max(H_i, H_j) \qquad \Delta y < r_y \max(W_i, W_j) \tag{9}$$

where (H_i, W_i) and (H_j, W_j) are the bounding box dimensions of both regions, and $(\Delta x, \Delta y)$ represents the distance between the centres of both regions' bounding boxes (Fig. 6). (r_x, r_y) are the *proximity coefficients*.

Size of adjacent regions - Similarly to the last test, two letters laying on the same line are assumed to have a similar size. This test limits the variance of adjacent region sizes.

$$\frac{|H_i - H_j|}{|H_i + H_j|} < r_h \qquad \frac{|W_i - W_j|}{|W_i + W_j|} < r_w \tag{10}$$

where (r_h, r_w) are the *size coefficients*, also determined experimentally.

After the edge filtering stage every remaining connected subgraph represents a text group. Text groups are tracked on consecutive frames and sent to the OCR engine for recognition.

IV. Results

To facilitate comparative analysis, we measure performance on single image text detection on the ICDAR 2003 text localisation competition 'TrialTrain' dataset [21]. The same definitions for *precision* and *recall* were used as defined by the competition. However, given that our algorithm detects whole sentences instead of isolated words, we joined the bounding boxes of the ICDAR database words into sentences, to be able to make fair evaluations. This is the same approach that Pan et al. [22] employed.

The performance result [4] of the proposed method is shown in Table I along with the reported detection results from ICDAR 2003 and ICDAR 2005 text location competitions (average, and winning entries), as well as our previous method [11] and three other recent and state-of-the-art algorithms [15], [22], and [10].

The proposed method shows a recall value of 0.67, close to the currently best performing algorithms, e.g 0.71 of [22], while not managing to obtain comparable precision performance. This means that our algorithm overestimates the number of detected regions, but indeed, it is not missing many real text locations. The lower precision rate can be compensated by the OCR engine discarding the unrecognisable regions. The text tracking step can also help in discarding the false positives as these non-text regions are unstable, while text regions are more consistently detected across several frames. In fact, by performing registration and super-resolution on tracked text regions [23], recognition accuracy can be increased. This is however beyond the scope of this paper and forms part of our future work. Some example results are shown in Fig.7.

One key advantage of our implementation is its simplicity and speed, which makes it feasible for real-time applications, including those involving text tracking. On the ICDAR database, it takes an average of 156 ms per image, but this is not representative for a real-time video text processor as every ICDAR database image has a different size and they are mostly high resolution still images. For video sequences we are able to process 14fps on 640×480 images and 9fps on 800×600 images (see Table II).

[4] All results were obtained using an Intel Core 2 Duo T9300 CPU.

Figure 7. Example results for some of the ICDAR 2003 database images.

Text localization	prec.	recall	f	time (s)
Ashida (2003 winner) [21]	0.55	0.46	0.50	8.5
ICDAR 2003 average [21]	0.32	0.32	0.31	5.3
Hinnerk Becker (2005 winner) [24]	0.62	0.67	0.64	14.4
ICDAR 2005 average [24]	0.39	0.46	0.39	4.25
Merino and Mirmehdi [11]	0.44	0.68	0.48	0.1
Neumann and Matas [15]	0.59	0.55	0.57	N/A
Epshtein et al. [10]	0.73	0.60	0.66	0.94
Pan et al. [22]	0.67	0.71	0.69	2.43
Proposed method	0.51	0.67	0.55	0.2

Table I
TEXT DETECTION PERFORMANCE ON THE ICDAR 2003 DATABASE.

	MSER	Filtering	**total**	
ICDAR database	134 ms	16 ms	156 ms	
640 × 480 video	49 ms	10 ms	61 ms	14 fps
800 × 600 video	74 ms	15 ms	95 ms	9 fps

Table II
TIME CONSUMPTIONS OF DIFFERENT STAGES OF THE TEXT LOCATOR.

V. CONCLUSION

We have reported a wearable text recognition tool that employs MSERs as the basis for real-time text detection. The proposed method refines our previous real time algorithm by exploiting hierarchical structure obtained from MSERs to yield more stable regions compared to the previous adaptive threshold method. It outperforms other published approaches computationally while maintaining similar text detection performance on the ICDAR dataset. In our future work, we plan to explore the introduction of a training stage for character recognition without reliance on third-party software, adding more cascading filters, and improving precision and recall results in general.

ACKNOWLEDGEMENTS

The Hat's construction was carried out at Bristol University based on a previous prototype and work carried out at University of La Laguna by Carlos Merino Gracia under the direction of José Luis González Mora.

Karel Lenc's contribution to this work was carried out at Bristol University as an ERASMUS student.

The Spanish Ministerio de Industria, Turismo y Comercio funds Carlos Merino Gracia through the European Regional Development Fund (project TSI-020100-2009-541).

REFERENCES

[1] R. Kurzweil, *The age of spiritual machines: when computers exceed human intelligence*. Viking Press, 1998.
[2] T. Hedgpeth, J. A. Black, and S. Panchanathan, "A demonstration of the icare portable reader," in *ACM SIGACCESS*, 2006, pp. 279–280.
[3] H. Aoki, B. Schiele, and A. Pentland, "Realtime personal positioning system for a wearable computer," in *ISWC*, 1999, pp. 37–43.
[4] J. Chmiel, O. Stankiewicz, W. Switala, M. Tluczek, and J. Jelonek, "Read IT project report: A portable text reading system for the blind people," 2005.
[5] J.-P. Peters, C. Thillou, and S. Ferreira, "Embedded reading device for blind people: a user-centred design," in *ETAIPR*, 2004, pp. 217–222.
[6] X. Shi and Y. Xu, "A wearable translation robot," in *ICRA*, 2005.
[7] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: a survey," *IJDAR*, pp. 84–104, 2005.
[8] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *IAPR Workshop on DAS*, 2008, pp. 5–17.
[9] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE TIP*, 2011.
[10] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963–2970.
[11] C. Merino and M. Mirmehdi, "A framework towards realtime detection and tracking of text," in *CBDAR*, 2007, pp. 10–17.
[12] G. K. Myers and B. Burns, "A robust method for tracking scene text in video imagery," in *CBDAR*, 2005.
[13] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions." in *BMVC*, 2002.
[14] M. Donoser, C. Arth, and H. Bischof, "Detecting, tracking and recognizing license plates," in *ACCV*, 2007, pp. 447–456.
[15] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *ACCV*, 2010, pp. 257–261.
[16] W. W. Mayol, B. J. Tordoff, and D. W. Murray, "Wearable visual robots," *Personal and Ubiquitous Computing*, vol. 6, pp. 37–48, 2002.
[17] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (MSER) tracking," in *CVPR*, 2006, pp. 553–560.
[18] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *ECCV*, 2008, pp. 183–196.
[19] A. T. Targhi, E. Hayman, and J. olof Eklundh, "Real-time texture detection using the LU-transform," in *CIMCV*, 2006.
[20] N. Ezaki, K. Kiyota, B. Minh, M. Bulacu, and L. Schomaker, "Improved text-detection methods for a camera-based text reading system for blind persons," in *ICDAR*, 2005, pp. 257–261.
[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *ICDAR*, 2003.
[22] Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *ICDAR*, 2009, pp. 6–10.
[23] C. Mancas-Thillou and M. Mirmehdi, "Super-resolution text using the teager filter," in *CBDAR*, 2005, pp. 10–16.
[24] S. Lucas, "ICDAR 2005 text locating competition results," in *ICDAR*, 2005, pp. 80–84.