Multimed Tools Appl (2014) 69:217–245
DOI 10.1007/s11042-012-1250-6

A framework for improved video text detection and recognition

Haojin Yang · Bernhard Quehl · Harald Sack

Published online: 11 October 2012

Abstract Text displayed in a video is an essential part for the high-level semantic information of the video content. Therefore, video text can be used as a valuable source for automated video indexing in digital video libraries. In this paper, we propose a workflow for video text detection and recognition. In the text detection stage, we have developed a fast *localization-verification* scheme, in which an edge-based multi-scale text detector first identifies potential text candidates with high recall rate. Then, detected candidate text lines are refined by using an image entropy-based filter. Finally, *Stroke Width Transform* (SWT)- and *Support Vector Machine* (SVM)-based verification procedures are applied to eliminate the false alarms. For text recognition, we have developed a novel skeleton-based binarization method in order to separate text from complex backgrounds to make it processible for standard OCR (*Optical Character Recognition*) software. Operability and accuracy of proposed text detection and binarization methods have been evaluated by using publicly available test data sets.

Keywords Video OCR · Video indexing · Multimedia retrieval

1 Introduction

In the last decade digital libraries and web video portals have become more and more popular. The amount of video data available on the *World Wide Web* (WWW) is

H. Yang (✉) · B. Quehl · H. Sack
Hasso-Plattner-Institute for IT-Systems Engineering, University of Potsdam,
Prof.-Dr.-Helmert Str. 2-4, 14467 Potsdam, Germany
e-mail: haojin.yang@hpi.uni-potsdam.de

B. Quehl
e-mail: Bernhard.Quehl@hpi.uni-potsdam.de

H. Sack
e-mail: Harald.Sack@hpi.uni-potsdam.de

growing rapidly. According to the official statistic-report of the popular video portal YouTube[1] more than 4 billion videos are viewed per day and about 60 h of video are uploaded every minute. Therefore, how to efficiently retrieve video data on the WWW or within large video archives has become a very important and challenging task.

Content-based video indexing and retrieval requires descriptive features, which can be used to represent the relevancy of the video content. These features can be mainly divided into two groups: the low-level perceptual features and the high-level semantic features [13]. The typical perceptual features such as, e.g., color, intensity, object shape, texture, etc. have been widely used for image classification and video indexing (as e.g., [2]). However, they can not provide an exact description of the video content.

Text in video is one of the most important high-level semantic features, which directly depicts the video content. In general, text displayed in a video can be classified into scene text and artificial text (overlay text) [13]. Compared with scene text, the artificial text can provide concise and direct description of video content, which is important for understanding. For instance, caption text in news videos often describe the speaker's name or event information; subtitles in sport videos often highlight the information about scores or players. Because of this tight coupling, in this context we mainly focus on artificial text recognition in video images.

Existing video OCR (*Optical Character Recognition*) technology is based on the combination of sophisticated pre-processing procedures for text extraction and traditional OCR engines [13]. Techniques from standard OCR, which focus on high resolution scans of printed (text) documents, have to be adapted to be also applicable for video images. For video OCR, first video frames have to be identified that obtain visible textual information, then the text has to be localized, interfering background has to be removed, and geometrical transformations have to be applied before standard OCR engines can process the text successfully.

The commonly used video OCR framework consist of three main steps: text detection, text tracking, and text recognition. Text detection process determines the location of text within the video image, i.e. bounding boxes enclosing the text are returned as results. Text tracking is intended to maintain the integrity of text positions across adjacent frames [13]. Typically, video text is embedded in very heterogeneous background with a great variety of contrast, which makes it difficult to be recognized by standard OCR software. Therefore, the text needs to be separated from interfering background by applying appropriate binarization techniques before applying the OCR process to enable high quality results.

In this paper, we address both text detection and recognition issues for video images. The major contributions of this paper are the following:

– For text detection, we propose a new localization-verification scheme. On detection stage, an edge-based multi-scale text detector is used to quickly localize candidate text regions with a low rejection rate; For the subsequent text area verification, an image entropy-based adaptive refinement algorithm can not only reject false positives that expose low edge density, but also further splits the most

[1] http://www.youtube.com

text- and non-text-regions into separate blocks. Then *Stroke Width Transform* (SWT)-based verification procedures are applied to remove the non-text blocks. Since SWT verifier is not able to correctly identify special non-text patterns such as sphere, window-blocks, garden fence, etc., we also adopt an additional *Support Vector Machine* (SVM) classifier to sort out these non-text patterns in order to further improve the detection accuracy.

– For text recognition, we have developed a novel binarization approach, in which we utilize image skeleton and edge maps to identify the text pixels. The proposed method consists of three main steps: text gradient direction analysis, seed pixel selection, and seed-region growing. After seed-region growing process, the video text images are converted into a readable format for standard OCR engines.

For reasons of comparability, the commonly used test data sets such as, e.g., ICDAR 2011 database [12] and Microsoft common test set [11] for text detection, segmentation and recognition have been applied, although the quality of Microsoft common test set is no longer up to date. We additionally compiled 4 new test sets including manual annotations with a wide variety of video resolution and genre, in order to make our evaluation as general as possible.

The rest of the paper is organized as follows: Section 2 presents related work, while in Sections 3 and 4, the proposed text detection and recognition approaches are depicted in detail. Section 5 provides the evaluation and experimental results. The last section concludes the paper with an outlook on future work.

2 Related works

In the last few years, collaborative tagging has become a popular functionality in video portals. Some approaches, as e.g., Kim [16] apply manual video tagging to generate text-based metadata for efficient context-based video retrieval. While this approach works well for recently generated video data, for legacy video data most times no tagging data or other textual metadata is available, and it is hard to attract the user's interest to annotate the outdated videos. Manual metadata annotation in general is rather time consuming and costly. On the other hand, *Automated Speech Recognition* (ASR) is able to provide text transcripts of spoken language. But often poor recording quality, speaker's dialects, or interfering noise lowers the quality of the achieved ASR results beyond further usefulness. Hence, the video OCR-based approach is suitable for processing large amounts of video data with sufficient quality.

In this section, we review existing methods for text detection and recognition respectively.

2.1 Text detection

Most of proposed text detection methods make use of texture, edge, color, motion and some other text representative features (as e.g., stroke width feature) to discriminate text from background.

Edges are a preferred feature for video text detection. Taking into account visual quality criteria for readability, most overlay text in videos comes with sufficient contrast ratio to its background. Therefore, salient edges are usually generated between text and background regions. Shivakumara et al. [29] proposed a video text detection method based on specialized edge filters for false positive elimination. Hua et al. [10] presented a text localization method based on image corner points and used edge map-based decomposition to reduce false alarms. Sato et al. [26] applied 3×3 differential filters to localize the text region and used position and size constraints for text line refinement. Chen et al. [4] as well as Anthimopoulos et al. [1] made use of Canny edge operator [3] and morphological operations to detect potential text regions. The shortcomings of those edge-based approaches are as follows: While providing high recall many false alarms may be produced for complex text background coinciding with a high edge density; to achieve an accurate detection result, a set of heuristic constraints has to be applied on detected candidate text blocks. The parameters of the constrains have to be evaluated by experiment on specified test data, which lacks of generality.

Texture features such as *Discrete Cosine Transform* (DCT) coefficients of grayscale images have been applied for text detection [25, 35, 38]. DCT feature based approaches are well suited for JPEG encoded images as well as for MPEG encoded videos. However, in practice this feature is not robust enough to distinguish text from text similar textured objects such as, e.g., bushes, or other vegetation. For this reason DCT-based approaches are not suitable to handle text detection tasks in complex natural scenes.

A large number of text detection approaches based on machine learning have been proposed such as, e.g., neural networks [20], Adaboost [24] or fuzzy logic [8]. Lienhart et al. [20] proposed an approach for text detection in both videos and images. In their system a complex-valued multilayer feedforward network is trained for text detection. Localized text lines are refined by exploiting the temporal redundancy of text in video. Their text line tracking algorithm processed every frame within the text line occurrence duration. However, their feature extraction process is executed on every pixel of the image. In contrast to hybrid approaches, which combine a fast localization algorithm and a machine learning-based text block verification method, their system is less efficient concerning computation time for both, training- and prediction-phase.

Recently, some hybrid approaches have been proposed [1, 4, 37]. Zhao et al. [37] introduced a classification approach for text detection in images using sparse representation with discriminative dictionaries. They applied wavelet transform to create an edge map and performed a block-wise classification to distinguish text and non-text regions by using two learned discriminative dictionaries. The obtained candidate text regions were further refined by projection profile analysis. Chen et al. [4] presented two-stage text detection methods that combine text localization with subsequent verification. The localization process is intended to coarsely detect text with low computation expenses, while in the verification stage, the text candidates are refined by using SVM classifier. However, the traditional machine learning-based verification can only deliver a binary decision and thus might not be robust enough to decide for the candidate bounding boxes that contain both text as well as non-text objects. Anthimopoulos et al. [1] have improved this drawback with a refinement process, which enables the machine learning classifier to refine the boundaries of the text images. This process enhanced the precision and the overall detection performance as well. But the evaluation results of the processing time reported in their work shows that the system was 6 times slower by applying this refinement algorithm.

Epshtein et al. [6] proposed a new image operator SWT for text detection in nature scene images. For each pixel the SWT operator computes the width of the most likely stroke. The output of the operator is a stroke-feature map, while each pixel represents the corresponding stroke width value of the input image. Stroke is a direct text representative feature that has demonstrated an excellent performance for distinguishing text from non-text objects. However, in practice, the computation of SWT is rather expensive for images with a complex content distribution. Hence, in this work, we have applied an adapted version of the original SWT algorithm that enables efficient text detection for video images.

2.2 Text recognition

Typically, video text is embedded in rather heterogeneous background with varying contrast ratio, which makes it difficult to be recognized by means of standard OCR software. Therefore, text pixels have to be separated from background beforehand by applying appropriate binarization techniques.

Text binarization methods can be divided into two categories: thresholding- and region-based methods. The thresholding-based methods include global thresholding, as e.g. [23], which apply a single threshold for the entire document, and adapted local thresholding algorithms [21, 27, 34], which assign a threshold for each pre-defined local region of the document. The second category sums up clustering-based [31, 33] methods and edge-based [39] methods. Since our approach is based on the region-growing strategy, it belongs to the region-based category.

Chen et al. [4] proposed a multi-hypotheses framework for text recognition. They applied several existing text binarization techniques, *Connected Component* (CC)-based text refinement method, and a print-OCR engine to create the hypotheses for text candidates. By applying a pre-trained language model they performed the scoring process for the final result selection. However, the document text binarization methods might not be robust enough when a video image combines complex background with varying contrast ratio.

Multi-frame integration techniques have also been applied to improve binarization results [18]. However, this kind of approach works well only when text lines and background are in relative motion over time.

Zhou et al. [39] proposed an edge-based binarization method for video text images. The grayscale histogram is used to determine text gradient direction. Text gradient direction determines whether the text block is of darker text on lighter background or vice-versa. The seed selection algorithm is based on the local maximum or minimum of edge values. Although parts of this approach are similar to ours, their system has several shortcomings: First, in order to determine the text gradient direction, their system processes the image histogram with a Gaussian filter recursively until only two peaks are obtained. This process is computationally complex. Furthermore, they assume that the background is always the peak with the higher value in the output histogram. However, in practice, we have realized that this assumption is not valid, when the detected text bounding boxes from the detection stage are close to the texts. Moreover, our skeleton-based seed selection method has been able to achieve more accurate results, because along the correct gradient direction the skeleton pixels will be created in the middle of the character shape. In their algorithm, the pixels surrounding the character boundaries have also been considered for the threshold determination, which might have affected the seed selection accuracy.

3 Text detection in video images

Text detection is the most challenging task in video OCR. The proposed text detection method determines, whether a single frame of a video file contains text lines, for which the appropriate bounding boxes are returned. In order to manage detected text lines efficiently, we have defined a class "text line object" with the following properties: Bounding box location (the top-left corner position of the bounding box), bounding box size, text, and average stroke width. The system starts with a coarse edge-based text localization process. If a text line has been successfully detected within a frame, a text line object is created, and the subsequent text line refinement and verification procedures ensure the validity of the achieved results in order to eliminate false alarms.

3.1 Text localization

To guarantee valid readability, artificial video texts typically obtain a sufficient contrast ratio compared to the background. Thereby, strong edges often occur at the boundaries of text and background. We have developed an edge-based text detector, subsequently referred to as *edge text detector*. The advantage of this kind of detector is its computational efficiency compared to other pure machine learning- or texture feature-based approaches, because no computationally expensive training period or feature computation process is required. However, we have to find the best suited parameters for the heuristic filtering method experimentally. In order to avoid the issue that the strong background edges often affect the detection result, we apply a gaussian filter to the images with an entropy value larger than a predefined threshold T_{entr} before starting with the text detection process. For our purpose, $T_{\mathrm{entr}} = 5.25$ has proven to serve best for our training data.

The text localization workflow for a single video frame is depicted in Fig. 1a–e. Taking into account that text regions often contain many vertical edges, a vertical edge map of the input image is computed using Sobel filter [30] (cf. Fig. 1b). Then, the morphological dilation operation is adopted to link the vertical character edges together (cf. Fig. 1c). We have defined a rectangle kernel: $1 \times MinW$ for the dilation operator, where $MinW$ denotes the detected minimal text line width. Since video texts with a height smaller than 5 pixels are hard to read for both, human and OCR engines, we assume that the minimal height $MinH$ of the recognizable text is 5 pixels for the original scale of the image. Furthermore, each detected text line has to contain at least 2 characters. Thus, $MinW$ is calculated by $MinW = 2 \times MinH$. Subsequently, a binary mask is generated by using Otsu's thresholding method [23] (cf. Fig. 1d). Finally, after CC analysis a binary map is created (cf. Fig. 1e), in which too high (with a height greater than a predefined threshold) or too large components (with a rectangle area size greater than 90 % of the image size) are removed.

The proposed text localization method depends on several size constraints of text. Therefore, characters dissatisfying the predefined size range can not be detected

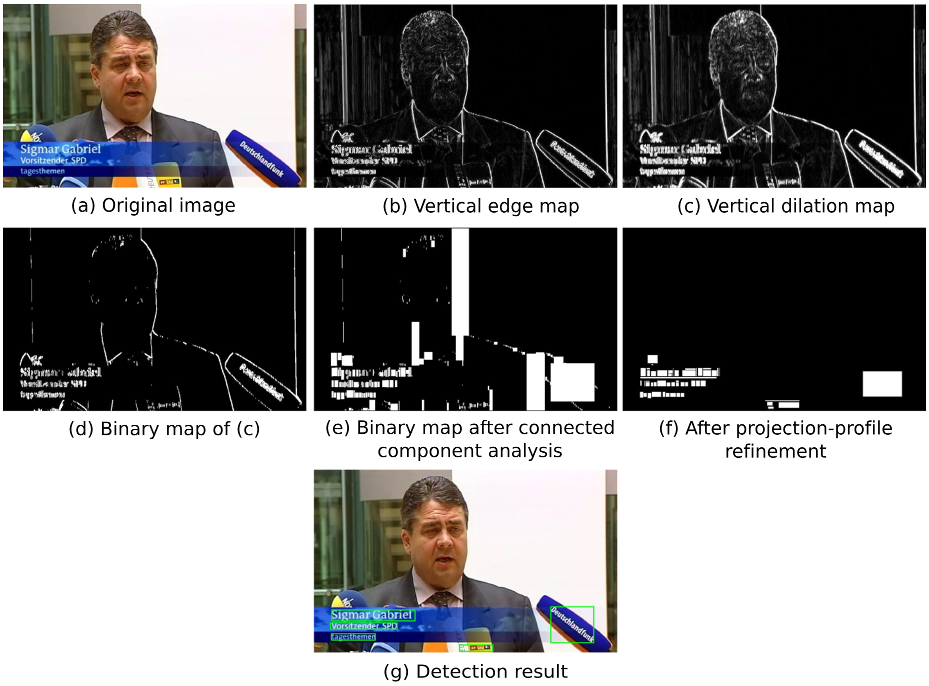


Fig. 1 Workflow of our proposed text detection method. **b** is the vertical edge map of the original image **a**, **c** is the vertical dilation map of **b**, **d** is the binary map of **c**, **e** is the result map of subsequent connected component analysis, **f** shows the binary map after the adaptive projection profile refinement, and **g** shows the final detection result

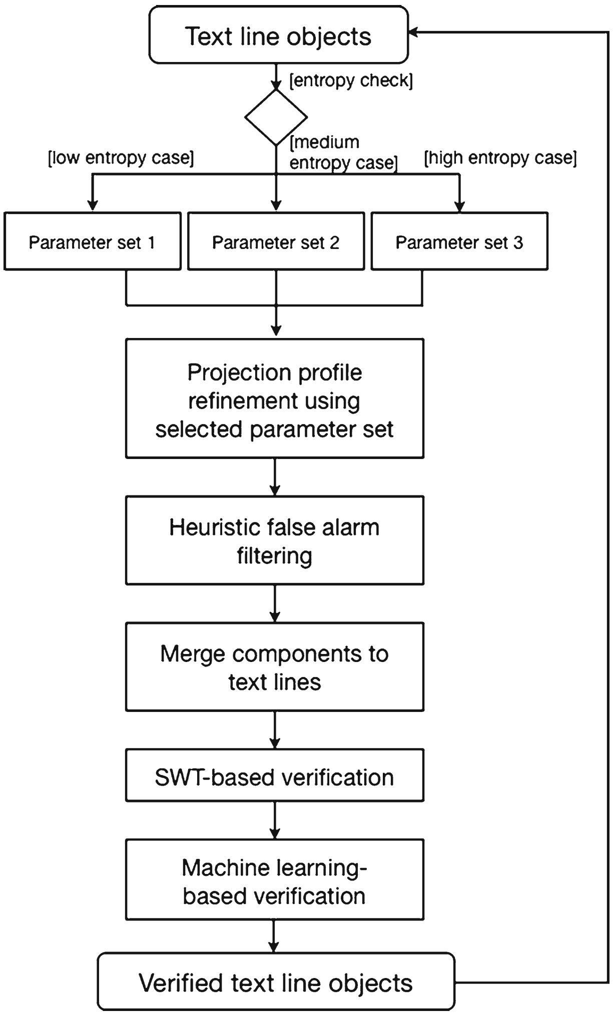
successfully. To overcome this problem, we apply the same localization method on multi-scaled images to generate corresponding binary maps after CC analysis like Fig. 1e. Then, all binary maps are scaled back to the original resolution and merged in the final mask.

To generate the initial text line object, we merge the CCs on the same horizontal text line with a neighbored distance smaller than their height. The candidate text line objects are further processed in the subsequent "text line refinement" step.

3.2 Text line refinement

Although the edge text detector results in a high detection recall rate, many false alarms might be created. In order to reject wrongly detected text line objects, we have developed an adaptive refinement procedure. Figure 2 shows its workflow.

The input text line objects have been created from the previous detection stage. Considering the detection results of the edge text detector a general robust detection result can hardly be achieved by applying a static refinement method. This is due to the general problems of thresholding-based approaches on heterogeneous input. For illustration, Fig. 3 shows two example video images together with their edge



Text line objects
[entropy check]
[low entropy case]
[medium entropy case]
[high entropy case]
Parameter set 1
Parameter set 2
Parameter set 3
Projection profile refinement using selected parameter set
Heuristic false alarm filtering
Merge components to text lines
SWT-based verification
Machine learning-based verification
Verified text line objects

Fig. 2 Workflow of the proposed adaptive text line refinement procedure

representations, where Fig. 3a contains a simple background combined with a low edge density, while Fig. 3d contains a relative high edge density in the background. Using a relatively low parameter set (as e.g., the parameter set 1 in Table 1), the result still contains too many noise as shown in Fig. 3e. On the other hand, a too strict parameter set (as e.g., the parameter set 3 in Table 1) will result in missing text edges as shown in Fig. 3c.

For classifying text line objects with varying edge complexity the image entropy can be utilized as an indicator. For text line images with a relative homogeneous background (low entropy case), we apply lower threshold parameters in order to obtain more text edges. In contrast, for text line images with a complex edge distribution (high entropy case), a more strict parameter set will be adopted. Thus, the input text line images are classified into three predefined categories according to

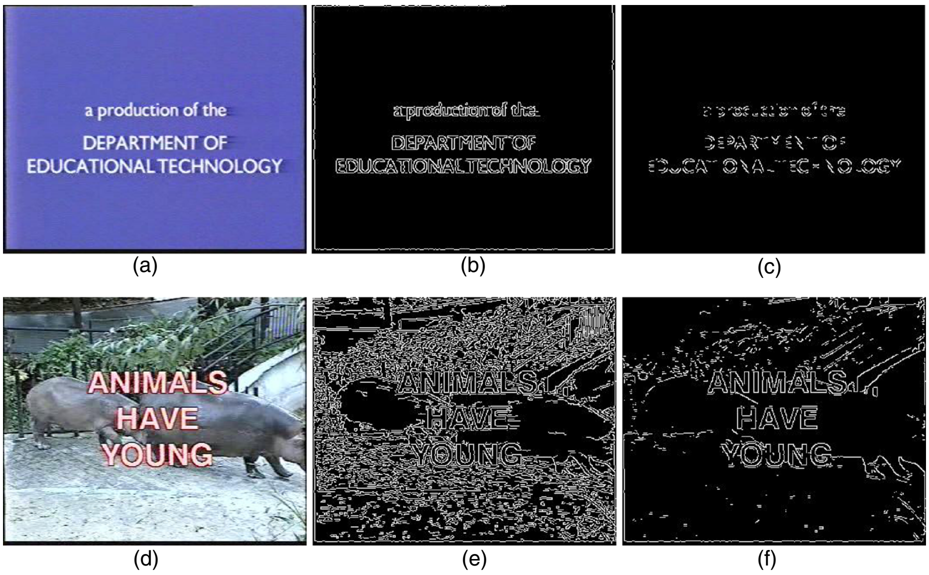


Fig. 3 Example video images with different background edge densities: **a** a low edge density image; **d** a high edge density image; **b**, **e** canny edge maps by using relative low thresholding parameters; **c**, **f** canny edge maps by using relative high thresholding parameters

their entropy. Let E denote the image entropy, whereas $Entropy_{\text{low}}$ and $Entropy_{\text{high}}$ denote the corresponding threshold ranges. The classification is processed as follows:

$$\begin{array}{lll} \textit{low entropy case:} & \text{if} & E < Entropy_{\text{low}} \\ \textit{medium entropy case:} & \text{if} & E \leq Entropy_{\text{high}} \wedge E \geq Entropy_{\text{low}} \\ \textit{high entropy case:} & \text{if} & E > Entropy_{\text{high}} \end{array}$$

For our application, the $Entropy_{\text{low}}$ and $Entropy_{\text{high}}$ have proven to serve best when set to 4.78 and 5.37. We have defined three parameter sets for the corresponding categories, which are required by the subsequent projection profile refinement process. The parameter set details are depicted in Table 1, where $Canny_{\text{low}}$ and $Canny_{\text{high}}$ indicate the threshold values for *Canny* edge detector, while min_h and min_v denote the horizontal and vertical minimal edge number. All threshold values have been compiled by experiment.

Next, the horizontal projection of the text line on the edge maps is processed (cf. Fig. 4b). A horizontal line is discarded, when its projection value is lower than $\min_h$. In this way, multi-line bounding boxes can be segmented to single lines. Then, we relocate the text line objects to the vertical projection in a similar way (cf. Fig. 4c).

Table 1 Parameter sets for projection profile analysis: $Canny_{\text{low}}$, $Canny_{\text{high}}$ denote the thresholds for the *Canny* edge filter, whereas $\min_h$, $\min_v$ denote the horizontal and vertical minimal edge number, respectively

	$Canny_{\text{low}}$	$Canny_{\text{high}}$	$\min_h$	$\min_v$
Parameter set 1	100	200	4	2
Parameter set 2	300	400	5	2
Parameter set 3	500	600	8	2

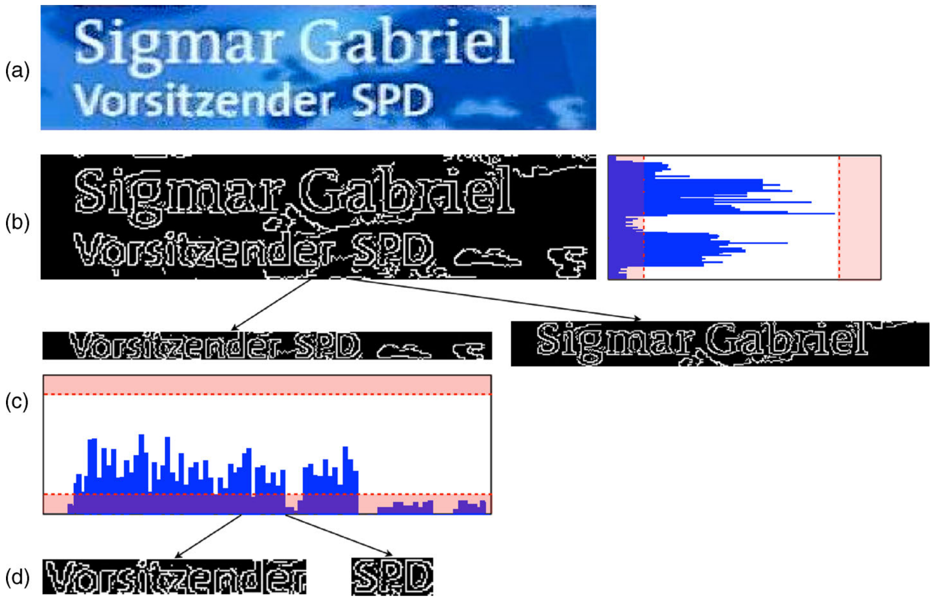


Fig. 4 Workflow of the projection profile analysis method: **a** the detected bounding box image using edge text detector, **b** horizontal projection profile analysis on edge map, **c** vertical projection profile analysis on edge map, **d** projection profiling results

We have adopted the heuristic filter methods to remove objects that are too small or too thin to be recognized as readable text. Objects with an invalid aspect ratio will also be removed. The vertical projection might split text lines into words and/or single characters. Thus, we have to merge neighbored characters with a distance less than the maximal character height of the according text line.

Finally, the SWT- and machine learning-based verification procedures have to be applied in order to remove also complex false alarms, which have a text similar pattern such as, e.g., vegetation, garden fences, etc. The refinement process is executed iteratively until no further changes occur. Figure 1g shows the result after the adaptive refinement process, which is the final output of the text detection.

In the next section, the SWT- and machine learning-based verification procedures will be described in detail.

3.3 Text line verification

We have developed two verification procedures to remove non-text objects. In general, most false alarms are sorted out by the SWT-based verification method. However, there are still a few non-text patterns, which represent a text-stroke similar structure that confuse the SWT verifier. Therefore, an additional machine learning-based verification procedure is applied in order to improve the robustness of our approach.

3.3.1 SWT-based verification

First, a SWT-based verification method is applied to candidate text line objects. Epstein et al. [6] have shown that stroke width feature is robust to distinguish text from other complex image elements with similar visual structure such as, e.g., vegetation, bushes etc. The SWT algorithm produces a feature map, where each pixel contains the most likely stroke width value of a corresponding input image pixel (cf. Fig. 5). However, the original SWT has two major shortcomings: first, the achieved recall rate does not exceed edge- or machine learning-based approaches. On the other hand it provides best precision results [6]. Second, SWT is computationally expensive for images with complex content distribution. Moreover, in order to accommodate both light text on dark background and vice-versa, the analysis process has to be applied twice to cover each gradient direction. Thus, we apply SWT on the detected candidate text line images only using constraints to verify the text line objects. A text line object is discarded if:

– its stroke width variance exceeds a threshold *MaxVar*,
– its mean stroke width exceeds a threshold range $Stroke_{\min}$ and $Stroke_{\max}$

where *MaxVar* denotes the maximal stroke width variance, which has proven to serve best when set to the half of the average stroke width in [6]. While $Stroke_{\min}$ and $Stroke_{\max}$ have been set to 1 and 10, respectively.

Subsequently, the retained text line objects are further verified as follows: first, the connected components are generated within a text line by merging pixels with similar stroke width value. These components are considered as the candidate characters. Then, connected components can be merged into chains, while the mergeable items should have a similar color and a small distance. Hence, an invalid chain is discarded, if less than two components can be detected from it, and a text line is discarded, if no valid chain can be detected from it. We apply Euclidian metric [5] to calculate the color distance of *RGB* images. The corresponding threshold value has been set to 40 (the value range is [0, 441.67]). The threshold value of the character component distance has been set to equal the height of the higher one of the comparison pair. All threshold values have been optimized in order to achieve an overall advantage for our heterogeneous video test data.

3.3.2 Classifier-based verification

Edge-based text detection methods focus mainly on edge strength and density. However, in some cases non-text regions are visually similar to edge directions and stroke width values of text and therefore might produce false alarms that the human optical system would have avoided (cf. Fig. 6). This fact has motivated us to use

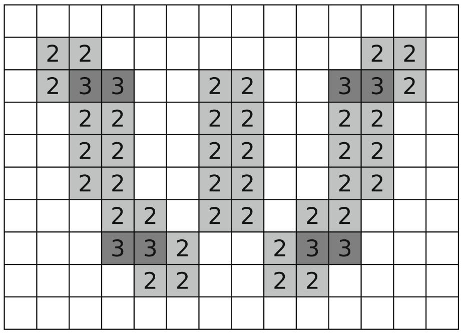


Fig. 5 An example output image from *stroke width transform* (SWT) for character *w*

	2	2									2	2	
	2	3	3			2	2			3	3	2	
		2	2			2	2			2	2		
		2	2			2	2			2	2		
		2	2			2	2			2	2		
			2	2		2	2		2	2			
			3	3	2			2	3	3			
				2	2			2	2				

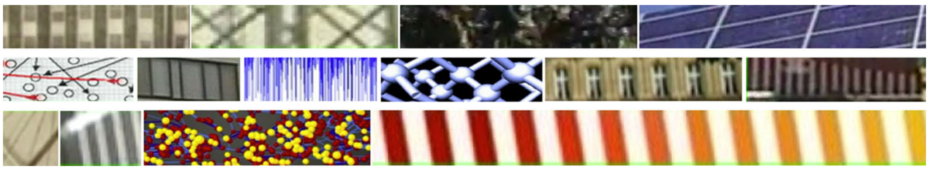


Fig. 6 Example non-text patterns which confuse the SWT-verifier, as e.g., repeating pattern like windows, other pattern like sphere cluster etc.

additional feature sets, which capture not only the edge distributions and stroke width values of the whole image, but their preselected local distribution, as well. A large number of features have been tested and evaluated such as, e.g., DCT, *Local Binary Pattern* (LBP), *Histograms of Oriented Gradients* (HOG), *Spatial Variance*, and *Constant Gradients Variance* features. For our system we use HOG feature combined with eLBP and entropy features. The HOG feature is used to describe the distribution of the orientations of gradients combined with *edge Local Binary Pattern* (eLBP), which describe the local spatial structure and the entropy features as supplement to distinguish non-text from text. Our experiments have shown that the combination of these features have achieved better results as the single features. Especially the discrimination between difficult non-text examples with repeating structures such as windows and bricks. We have applied *Support Vector Machine* (SVM) [32] for text classification task.

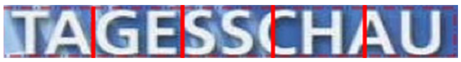
To improve the text detection result from the analytical system, which already has a high recall, we only extracted features for machine learning from a previously computed bounding box (cf. Fig. 7). The Bounding box is divided in different blocks. In our experiments the ratio of block width and height is 2:1 (i.e. the feature block width is the double of its height) has been proven as the best outcome.

Feature extraction For the classier-based verification we apply a new combination of 3 feature types HOG, LBP, and entropy feature, according to [1, 9], which have proven that LBP and HOG are highly discriminative for text segmentation. The HOG features have been successfully applied for object detection tasks as e.g., Person detection [36] or OCR [15]. HOG features count occurrences of gradient orientation in the local spatial structure of an image. Equation (1) shows the 3×3 horizontal and vertical Sobel masks where A represents the source image, and G_x and G_y contain the horizontal and vertical derivative approximations. For each pixel the resulting gradient approximations are combined to compute the gradient direction shown in (2).

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{1}$$

$$\theta = atan2(G_y, G_x) \tag{2}$$

Fig. 7 Red line separates a feature block



Every vector element is a histogram of gradients consisting of 9 values from 0 to 180°. To achieve a scale invariant feature each block is normalized in a manner so that its sum equals to one. Although HOG is well suited to capture characteristic gradient responses of text contours, there are also some objects, e.g., windows or railings, which often are confused with text patterns in HOG feature space.

The LBP is an operator (cf. (3)) which summarizes the local spatial structure of an image. It has been introduced by Ojala et al. [22], who shown the discriminative power of LBP operator for texture classification. The advantages of LBP is that invariance to monotonic gray-level changes and computational efficiency makes it most suitable for demanding text detection tasks. The original LBP operator consists of 3×3 kernel and compares the intensity value between the central pixel and its neighbors. The center pixel is used as a threshold and the 8 binarized neighbors are multiplied by the respective binomial weight, producing an integer value in a range of 0–255. Each of these 2^8 different words represents a single texture pattern. The decimal form of the resulting 8-bit word can be expressed as follows:

$$LBP(x, y) = \sum_{n=0}^{7} S_e(i_n - i_c) * 2^n \tag{3}$$

$$S_e(x) = \begin{cases} 1, x >= 0 \\ 0, x < 0 \end{cases} \tag{4}$$

where $S_e(x)$ denotes the threshold function, i_c denotes is the luminance value of the center pixel (x_c, y_c), and i_n corresponds to the luminance values of the surrounding pixels.

The LBP transforms computes an 8-bit greyscale image, where each pixel value represents the texture pattern of the corresponding pixel in the original image. Finally, the normalized histogram with 256 bins of the LBP image is computed to enable scale invariance. In our approach, we have adapted eLBP according to [1]. The edge Local Binary operator differs from the original in the definition of the threshold function (cf. (5)). To avoid noise a distance value d is introduced. Furthermore, to capture dark on light as well as light on dark text characteristics as one single LBP histogram the absolute distance is applied.

$$S_e(x) = \begin{cases} 1, |x| >= d \\ 0, |x| < d \end{cases} \tag{5}$$

According to our evaluation a distance value of 20 has proven to be most successful. Thereby, arbitrary intensity variations caused by noise have been avoided. To reduce the number of features from 256 histogram bins to 32, the vertical horizontal and the diagonal neighbors have been aggregated in eLBP operators (cf. (6) and (7)).

$$eLBP_{\text{diag}}(x, y) = \sum_{n=0}^{3} S_e(i_{2*n} - i_c) * 2^n \tag{6}$$

$$eLBP_{vh}(x, y) = \sum_{n=0}^{3} S_e(i_{2*n+1} - i_c) * 2^n \tag{7}$$

4 Text recognition

The detected text objects from the previous step serve as the input for the text recognition. Although text can be localized accurately, correct recognition of the identified text via standard OCR software often does not work with sufficient quality, because technology from standard OCR is customized for high resolution scans of printed documents with dark text on light background within properly organized text regions (paragraphs). Unfortunately, text in video data comes in different resolutions and with heterogeneous background resulting in varying contrast ratios that most times prohibit valid OCR results.

Therefore, to enable an efficient way to process the detected texts, the text images have to be further refined. To resolve the problem of low-resolution images, text line images with text height smaller than 30 pixels are enlarged via cubic interpolation. Then, a skeleton-based binarization method is applied to extract video text from its background, which is discussed in the next section.

4.1 A skeleton-based text binarization method

As previously mentioned, the detected bounding box images from the text detection stage have to be binarized for further OCR processing. In our approach, we first have to figure out the text gradient information from the text line image.

4.1.1 Text gradient analysis

The text gradient direction is required for the subsequent seed selection method and can be achieved best by analyzing the content distribution of skeleton maps. Skeletons can be considered as thin versions of object shapes that represents the structure of objects. The typical skeleton representation of a binary image can be calculated according to (8):

$$S_n(X) = (X \ominus nB) \setminus (X \ominus nB) \circ B, \tag{8}$$

where $S_n(X)$ are the skeleton subsets of a binary image containing a set of topologically open shapes X. B is a structuring element, n is the number of shapes, while $\ominus$ and $\circ$ denote the morphological erosion and opening, respectively. The skeleton map $S(X)$ of the image is the union of the skeleton subsets $S_n(X)$ [28].

Figure 8 shows example text line images and their corresponding skeleton maps. By evaluating skeleton maps, which have been created with the wrong gradient direction (cf. skeleton-*dark on light* in Fig. 8a, b and skeleton-*light on dark* in Fig. 8c), we can realize that there are always white pixels located on the image boundaries. This is due to the characteristics of the skeleton operation. The morphological skeleton can be considered as a special thinning function. Applying the skeleton operation, only the structure of the original shape is preserved. All redundant pixels are removed. Therefore, along the correct gradient direction the skeleton pixels are retained in the center line of the character shapes (cf. skeleton-*dark on light* in Fig. 8c and skeleton-*light on dark* in Fig. 8a, b). Otherwise, the skeleton pixels surround the characters and are located on the image boundaries.

Thus, we can simply obtain the text gradient direction by comparing the white pixel count at the image boundaries of two skeleton maps. This method has been able to achieve over 95 % accuracy for our test data.

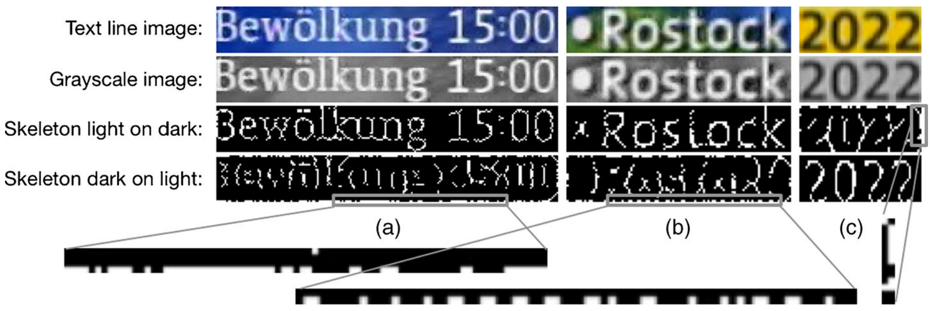


Fig. 8 Text gradient analysis: the first skeleton map is created according to *light on dark* (light text on dark background) rule, whereas the second one is created according to *dark on light* rule

Next, the seed selection and seed-region growing algorithm are described, in which the previously determined text gradient direction is applied.

4.1.2 Text seed selection

In order to distinguish text pixels from their background, we apply the skeleton map which has been created with the correct gradient direction to calculate the appropriate thresholds for the seed selection process. The thresholds can be determined by applying *global* or *adaptive* selection methods.

Global seed selection In the global selection method, we calculate the highest, lowest, and mean grayscale value of skeleton pixels for entire text line images, which can be denoted by T_h, T_l, and T_{mean}, respectively. S_p is the grayscale value of a corresponding seed pixel. In addition, we have also defined a variance seed factor σ, which is used to guarantee that the selected seed pixels are located inside the character boundaries. The seed selection process works as follows:

$$\begin{array}{ll} \textbf{if } \textit{light text on dark back} & \textbf{then } S_p > T_{\text{mean}} + \sigma \wedge S_p \leq T_h \\ \textbf{if } \textit{dark text on light back} & \textbf{then } S_p < T_{\text{mean}} - \sigma \wedge S_p \geq T_l \end{array}$$

The selection process verifies every pixel of the text line image. If an image pixel satisfies the above conditions, it will be labeled as a seed pixel. For our experiments a seed factor $\sigma = -38$ has been determined empirically.

Adaptive seed selection To perform the adaptive seed selection process, the text line image is divided into blocks. We set the block height and width equal the text line height. Subsequently, the global seed selection process is performed for each block.

A default σ value has been set to 75. In order to adaptively determine the appropriate σ for some special cases, we apply the local entropy and variance of image grayscale value as indicator. In our current implementation, the following special cases have been considered:

– if the image block has a relative upper grayscale variance (>680) with a entropy value between 4 and 5, the corresponding σ is increased by 20,
– if the image block has a relative lower grayscale variance (<200) with a entropy value smaller than 4, the corresponding σ is set to 38.

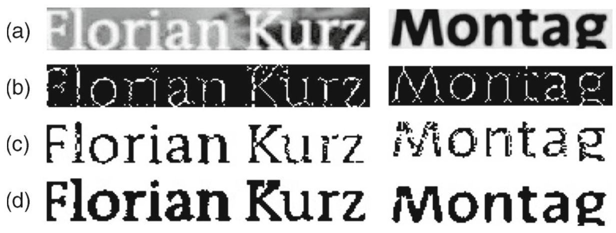


Fig. 9 Seed selection and region growing results: **a** grayscale image, **b** skeleton image, **c** seed image, **d** region-growing result

All parameters of the adaptive selection method have been optimized by using our training dataset. Figure 9c demonstrates the seed selection results.

4.1.3 Seed-region growing

After seed selection, the process continues with a seed-region growing algorithm. This process determines the pixels to be included in the growing region that forms the text area to be processed by the OCR engine. The seed region growing process starts from each seed pixel and extends the seed-region in north, south, east, and west directions. The image edge map (as e.g., *Canny edge map* [3]) is used to terminate the growing process as follows: if the seed-region reaches an edge pixel, this pixel will be labeled as a text pixel, and no further extension will be performed in this direction. To avoid open (not completely closed edge contour) text boundaries, we have applied a similar method as described in [39]: For example, let $P(x, y)$ denote the current seed pixel. To analyze its north neighbor $P(x, y-1)$, two parallel neighbors $P(x-1, y-1)$ and $P(x+1, y-1)$ are examined. If one of them is an edge pixel and the other one is not, then a potential boundary gap has been detected. Then, we label $P(x, y-1)$ as text pixel and terminate further extension in this direction. For other directions, we repeat the process in the same manner.

Figure 9d shows the final region growing result that is also the final result of the binarization. Figure 10 shows an exemplary binarization result of the adaptive skeleton binarization compared to other methods. The binarized text line images are further processed with standard OCR software.

4.2 Enhanced text recognition using multi-hypotheses framework

The text recognition result can be further improved by using a multi-hypotheses framework. In such a framework several different thresholding methods are applied to generate binarization results. The corresponding OCR hypotheses are subsequently created by applying a standard OCR engine for each binarized image.

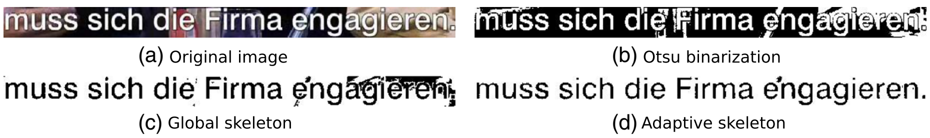


Fig. 10 An exemplary binarization result of the adaptive skeleton binarization method

Table 2 Test data sets for evaluation of text detection

Test set	Genre	Resolution (px)	Number of frames	Number of text lines
Microsoft common test set	TV news and lecture video	352×288	45	160
German TV news	TV news video	960×544	72	354
YouTube video	TV news video	1920×1080	38	226
Mediaglobe test set	Historical documentation video	720×576	31	323

Then, the spell checking process is performed on every result OCR string. The final OCR result is achieved by applying heuristic constraints. In our approach, the spell-checking is performed as following:

– first, if there are more than two hypotheses correctly recognized i.e. the result OCR string is verified via a dictionary lookup by OCR engine, the final result is determined by majority vote,
– second, if there is only one hypothesis correctly recognized, it will be applied,
– if the skeleton-based hypothesis is empty (in some cases, there is no OCR result returned from the OCR engine), we accept the hypothesis with the most valid characters, i.e. the dictionary term with the closest distance,
– otherwise, we apply the skeleton-based hypothesis as the final result.

5 Evaluation and experimental results

For the evaluation of system performance, we restricted on testing the efficiency of text detection and text binarization on single video frames. In order to achieve an evaluation as general as possible for our text detection method, we have compiled test data sets with a wide variety of genre, resolution, and other characteristics. Table 2 summarizes 4 different test data sets, whereby the Microsoft common test set [11] has also been used in [10, 37]. For reason of comparability, we have utilized this test set although its characteristics regarding quality and resolution is no longer up to date. In addition, we have compiled 3 video frame sets from German TV news program, YouTube videos and *Mediaglobe* project,[2] respectively. In order to investigate the generality, we have also performed the evaluation on ICDAR 2011 database “text localization” (102 images),[3] although the proposed method has been designed for video text recognition.

The proposed text binarization method is evaluated on the binarization test set (430 words and 2178 characters), which consists of text line images collected from

[2] *Mediaglobe* is a SME project of the THESEUS research program, supported by the German Federal Ministry of Economics and Technology on the basis of a decision by the German Bundestag, cf. http://www.projekt-mediaglobe.de/ (last access: 14/09/2012).

[3] Text localization is the first task of “reading text in born-digital images (web and email)” challenge.

Table 3 Comparison results on Microsoft common test set

Method	Recall	Precision	F_1 measure
Zhao et al. [37]	0.94	0.98	0.96
Thillou et al. [33]	0.91	0.936	0.92
Lienhard et al. [20]	0.94	0.91	0.92
Shivakumara et al. [29]	0.92	0.9	0.91
Gllavata et al. [7]	0.9	0.87	0.88
Our method	0.93	0.94	0.93

video text images. Furthermore, ICDAR 2011 database "text segmentation" (102 images) and "word recognition" (918 words) [12] have also been applied to evaluate the binarization as well as our entire recognition framework.

Due to copyright restrictions, the videos of generated test sets are not available for public use on the web. However, all test frames including manual annotation used for this evaluation and detailed experimental results are online available.[4] In order to enable a fair comparison of processing time we have used similar machine configuration with [1] for the performance evaluation (Intel single core, Pentium 4, 3.2 GHz).

5.1 Evaluation of text detection

For the evaluation of our text detection method, we distinguish pixel-based evaluation, where the percentage of overlapping pixels in the ground truth and the experimental results determines recall and precision, and text bounding box-based evaluation as proposed by Zhao et al. [37].

In order to enable a comparison to other existing methods, we applied the evaluation method that has been proposed for the Microsoft test set, together with evaluation results reported in [37]. Table 3 shows the achieved results for the Microsoft test set. Although our proposed method is not able to outperform the method of [37] for this test set, it surpasses all remaining methods.

The Microsoft test set has been published almost a decade ago, and video quality as well as resolution has improved dramatically ever since. Therefore, we have extended the evaluation on 3 additionally compiled video test sets, which demonstrate a wide variety of video resolution format and quality to avoid the over-fitting effect and to enable a more general evaluation. Pixel-based evaluation has been performed for text detection with and without the SVM verifier. The results of the standalone analytical method for all 4 test sets are illustrated in Table 4.

$Entropy_{\text{high}}$ and $Entropy_{\text{low}}$ of our analytical method, which indicate the appropriate parameter set for projection profile refinement have been learned from our training dataset. Figure 11b and a demonstrate the evaluation results, in which the influences of value changes to the detection performance are depicted.

In the stroke width verification stage, we have applied the parameter configurations according to [6]. In addition, we have optimized the *color distance* parameter by using our training dataset. Figure 12 shows the evaluation results.

[4]http://www.yovisto.com/labs/VideoOCR/ (last access: 14/09/2012)

Table 4 Text detection results using pixel-based evaluation without machine learning-based verification

Test set	Recall	Precision	F_1 measure
Microsoft common test set	0.92	0.90	0.91
German TV news	0.84	0.87	0.85
YouTube video	0.85	0.78	0.81
Mediaglobe test set	0.77	0.69	0.73

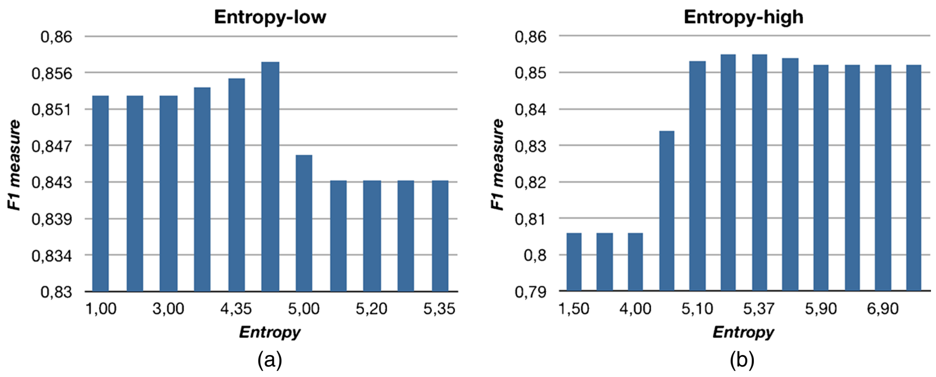


Fig. 11 **a** Evaluation results of parameter *Entropy*$_{\text{low}}$. **b** evaluation results of parameter *Entropy*$_{\text{high}}$

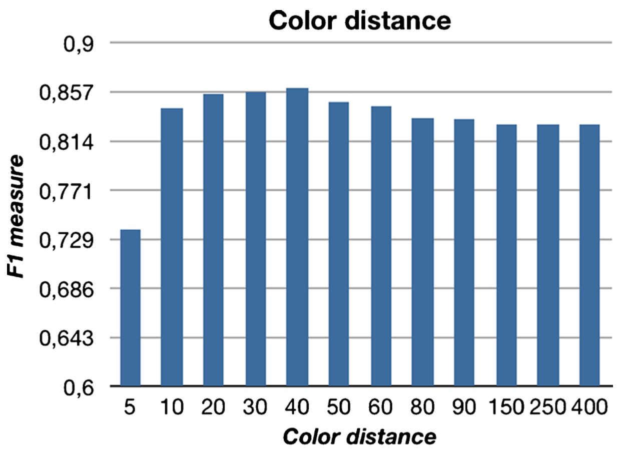


Fig. 12 Evaluation results of parameter *color distance* in the stroke width verification

Table 5 Text detection results using pixel-based evaluation with the SVM verifier

	Recall	Precision	F_1 measure
YouTube test set (overlay)			
Detection result without SVM verification	0.85	0.78	0.81
Detection result with SVM verification	0.84	0.86	0.85
YouTube test set (overlay + scene)			
Detection result without SVM verification	0.82	0.79	0.80
Detection result with SVM verification	0.80	0.85	0.82
Microsoft common test set			
Detection result without SVM verification	0.92	0.90	0.91
Detection result with SVM verification	0.92	0.92	0.92

The detection accuracy has been further improved by applying a subsequent SVM-based verifier. We only extracted features for machine learning procedures from previously computed bounding boxes, which have been divided in different blocks. In our experiments, the ratio of block width and height has been set 2:1. We linearly combined the HOG features with 9 bins, eLBP with 32 features and the image entropy for each block resulting in a 42-dimensional feature vector f_{vec} (cf. (9)).

$$f_{\text{vec}} = \{hog_1, hog_2, .., hog_9, elbp_1, ..., elbp_{32}, entropy\} \tag{9}$$

Text or non-text classification for each bounding box is determined by majority vote. 4343 text and 6787 non-text bounding box samples have been compiled for training. Text training samples originate from TRECVID,[5] YouTube, or have been created artificially. For the SVM classifier we have used *Radial Basis Function* (RBF) as kernel. Parameters have been determined by grid search.

Since the compiled training data has been designed for TV news video, The evaluation has been performed on Microsoft and YouTube test set. In order to set up two distinct SVM classifiers for overlay text and for scene text, we have compiled YouTube test set into two sets accordingly. Table 5 shows the achieved results. As a result the machine learning verifier generally improves the system accuracy (cf. Fig. 13). Due to the content of the training data, the SVM classifier obtains better results for overlay text.

In order to investigate the generality of the proposed text detection method, we have also performed the evaluation on ICDAR 2011 databased "text localization". We have achieved the second place in the current ranking list (cf. Table 6).

Evaluation results for system performance of several stat-of-the-art video text detection methods [1, 4, 17, 19] have been reported in [1]. Due to differences in resolution and aspect ratio we are not able to give an exact comparison of runtime (cf. Table 7).

The most time consuming tasks of other SVM-based systems are the prediction calls to the classifier (as e.g., [1, 4, 17]). Since in our system, the SVM classifier is designed only to verify a few non-text patterns, the processing time of our SVM classification process is only about 0.04 s per frame. In our method, the adaptive

[5]http://trecvid.nist.gov/ (last access: 14/09/2012)

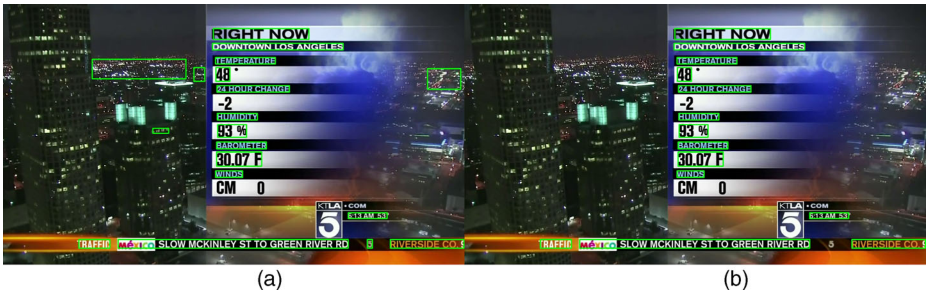


Fig. 13 **a** Is an exemplary text detection result of standalone analytical method, **b** is the result after the machine learning verification. Some repeating patterns such as e.g., windows, tiles, and light sources from the analytical detection result have been removed by machine learning verifier

Table 6 Text localization results of ICDAR 2011 challenge 1: "text localization" competition [12] (last check: 14/09/2012)

Participant	Recall (%)	Precision (%)	F_1 measure (%)
Mario Anthimopoulos	81.88	87.35	84.53
Our method	80.66	83.37	82
Alvaro Gonzalez	70.08	89.23	78.51
TDM_IACAS	69.7	85.83	76.93
TH-TextLoc	73.06	80.39	76.55
Textorter	69.08	85.54	76.43
Baseline	69.94	83.92	76.3
OTCYMIST	75.65	63.85	69.25
SASA	64.91	67.38	66.12
Text Hunter	58.43	75.52	65.88
Zhang Yang	73.21	46.72	57.04

Table 7 Average processing time per frame of several text detection methods

Method	Test frame resolution	Average processing time/frame (s)
Kim et al. [17]	768×576	8
Chen et al. [4]	768×576	3.35
Li et al. [19]	768×576	1.5
Anthimopoulos et al. [1]	768×576	2
Our method	960×544	1.1

refinement process and the SWT computation are the most time consuming tasks. Figure 14 illustrates some detection results of our method.

5.2 Evaluation of text binarization

The experimental results of our text detection method have been shown in the previous section. Next, we will discuss the evaluation of the proposed skeleton-based binarization method.

5.2.1 Parameter optimization

We have optimized the seed factor σ for the global and adaptive seed selection method by using our training dataset. The correctly recognized character rate has

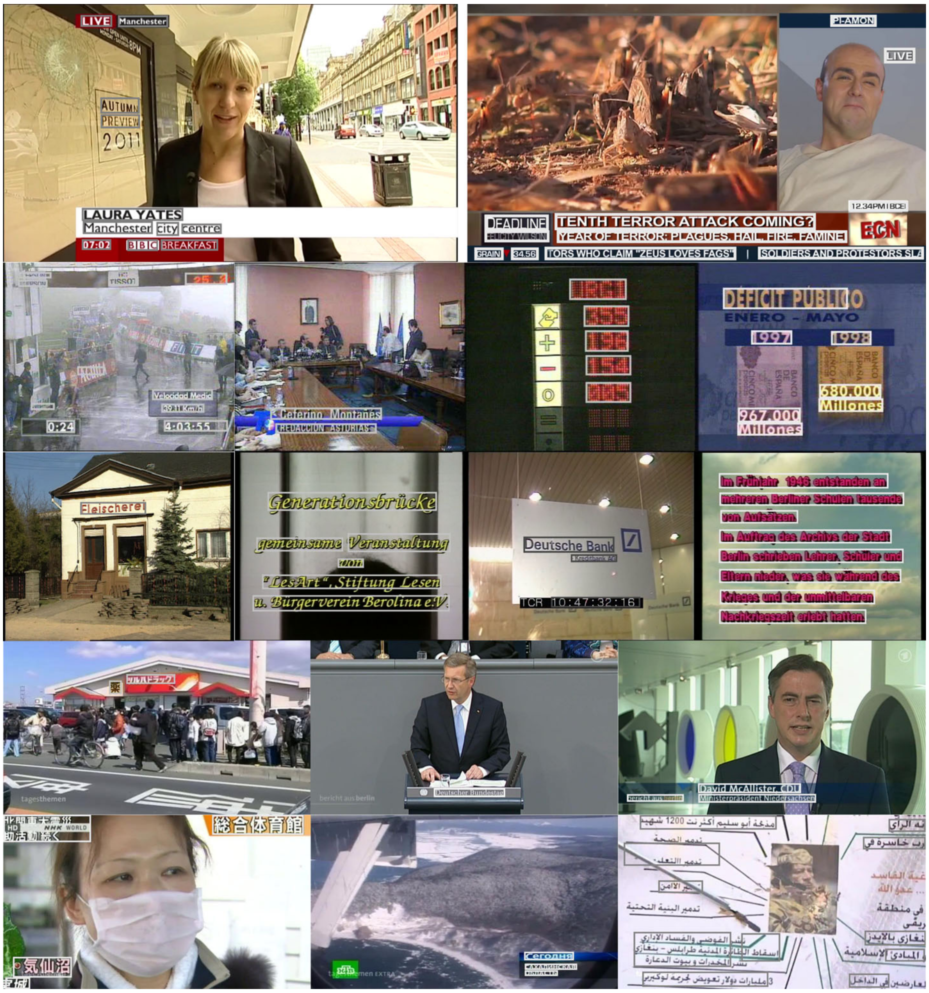

LAURA YATES
Manchester city centre
BBC BREAKFAST
DEADLINE
TENTH TERROR ATTACK COMING?
YEAR OF TERROR: PLAGUES, HAIL, FIRE, FAMINE
ECN
LIVE
DEFICIT PUBLICO
ENERO - MAYO
1997
1998
967.000 Millones
680.000 Millones
Ceferino Montañes
Fleischerei
Generationsbrücke
gemeinsame Veranstaltung
2011
"LesArt", Stiftung Lesen
u. Bürgerverein Berolina e.V.
Deutsche Bank
Im Frühjahr 1946 entstanden an
mehreren Berliner Schulen tausende
von Aufsätzen.
Im Auftrag des Archivs der Stadt
Berlin schrieben Lehrer, Schüler und
Eltern nieder, was sie während des
Krieges und der unmittelbaren
Nachkriegszeit erlebt hatten.
David McAllister, CDU
総合体育館
気仙沼
Сегодня

Fig. 14 Example text detection results from various test sets, detection results are indicated by bounding boxes

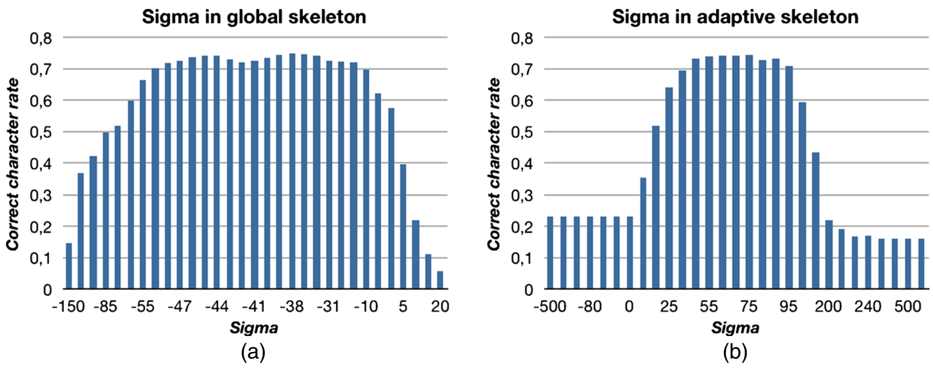


Fig. 15 **a** Evaluation results of seed factor σ for global seed selection method, **b** evaluation results of seed factor σ for adaptive seed selection method

been applied to indicate the recognition performance. Figure 15a and b show the evaluation results.

5.2.2 Video text images

The evaluation for video text images is performed on the binarization test set. Since text binarization usually follows the text detection process, the test set has been compiled with collected text line images. For text recognition, we have applied the open-source OCR engine *tesseract-ocr*.[6]

In order to provide a comparison to other existing text binarization methods, we have also evaluated 5 different reference methods on the same test set. In particular, we have applied the C++ implementation of *Niblack*, *Sauvola*, *wolf2001* and *wolf2007* from.[7] In oder to achieve the best result for each method, the optimal parameter k[8] has been empirically determined. We applied *Correctly Recognized Word Rate* (W_{rate}) and *Correctly Recognized Character Rate* (C_{rate}) to measure the efficiency of the text binarization methods as defined in (10) and (11), respectively:

$$W_{\text{rate}} = \frac{\text{number of correctly recognized words}}{\text{number of words in ground truth}} \tag{10}$$

$$C_{\text{rate}} = \frac{\text{number of correctly recognized characters}}{\text{number of characters in ground truth}} \tag{11}$$

The evaluation results are illustrated in Table 8. The proposed skeleton-based method outperforms the results of all reference algorithms. Figure 16 shows some exemplary binarization results of this method.

[6]http://code.google.com/p/tesseract-ocr/ (last access: 14/09/2012)

[7]http://liris.cnrs.fr/christian.wolf/software/binarize/index.html (last access:14/09/2012)

[8]k serves as a constant parameter used to determine the local threshold in [21, 27, 34].

Table 8 Results of text binarization methods on the binarization test set

Method	Correct characters	Correct words	C_{rate}	W_{rate}
Otsu [23]	1391	229	0.64	0.53
Niblack [21] ($k = 0.3$)	1329	199	0.61	0.46
Sauvola et. al [27] ($k = -0.01$)	1324	189	0.61	0.44
Wolf 2001 ($k = -0.2$) [34]	1297	194	0.59	0.45
Wolf 2007 ($k = -0.2$) [34]	1022	139	0.47	0.32
Global skeleton ($\sigma = -38$)	1588	263	0.73	0.61
Adaptive skeleton ($\sigma = 75$)	1655	274	0.76	0.64

The overall average processing time of the text detection and recognition process for the German TV news test set is 1.43 s per frame.

5.2.3 Born-digital images

In addition, we have performed the evaluation on ICDAR 2011 database "text segmentation" and "word recognition" which are the second and third task of "reading text in born-digital images (web and email)" challenge. These test datasets have been compiled to evaluate the text segmentation and recognition algorithms for born-digital text images. By applying those datasets we are able to investigate the generality, although the proposed method has been designed for artificial video text recognition.

In order to achieve the ICDAR 2011 compatible segmentation result, we first apply the text localization method on the input image. Then, we binarize the detected bounding box images and subsequently merge them into a black mask which has the same size as the input image according to their original positions. As shown in Table 9, our algorithm outperforms other submitted methods.

Our recognition method consists of a preprocessing algorithm and a multi-hypotheses recognition engine. In the preprocessing step, we have scaled the image size to two or three times larger by using cubic interpolation, when its height is between 20 and 30 pixels or smaller than 20 pixels, respectively. Subsequently, we remove the rand noises by applying horizontal-vertical projection profile refinement method. Then, the processed images have been sent to the multi-hypotheses engine, which is based on the adaptive skeleton binarization, otsu thresholding, and

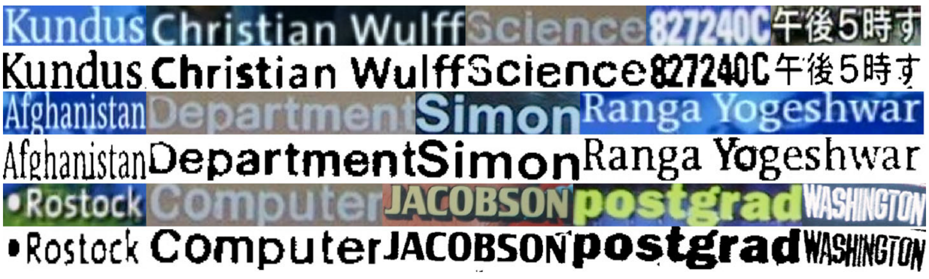


Fig. 16 Some exemplary binarization results

Table 9 Results of ICDAR 2011 challenge 1: "text segmentation" competition [12] (last check: 14/09/2012)

Participant	Recall (%)	Precision (%)	F_1 measure (%)
Our method	84.14	77.54	80.71
Marios Anthimopoulos	86.15	71.19	77.96
OTCYMIST	80.62	72.06	76.1
Textorter	65.23	63.63	64.42
SASA	71.68	55.44	62.52

tesseract-ocr engine. Three OCR-hypotheses have been created from two binarization images and the sharpened original image for each test image, respectively. The open source spell checker *Hunspell*[9] has been applied in spell checking process.

We have submitted our recognition results to the ICDAR 2011 online evaluation framework, in which the editing distance and the percentage of correctly recognized words in the ground truth and the achieved experimental results are used to perform the recognition accuracy [14]. Table 10 shows the comparison results to other submitted methods. Our method achieves the first and third place in the actual ranking list by applying commercial OCR engine *ABBYY FineReader*[10] and the open-source OCR engine *tesseract-ocr*, respectively.

5.3 Discussion

As shown in Fig. 14, the proposed text detection method is applicable to both Western and Eastern writing systems as well. Furthermore, it is likewise applicable for overlay text and scene text in video images. Our text recognition system currently supports German and English language. However, to recognize other languages we only need to extend the corresponding OCR libraries.

Although the combined edge- and SWT-based text detection approach works well for identifying most text and non-text blocks, some repeating patterns, such as windows (cf. Fig. 6) can hardly be distinguished with it. We solve this problem by adding a machine learning-based verifier. By using this approach the system robustness are further improved.

The proposed text detection method may not perform well on detection of skewed text because of using projection profile analysis. In addition, the skeleton-based binarization method might not be suitable for video text of rather small stroke width ($\leq$ 3px), since this might affect the seed-region growing process. However, most artificial texts in video are aligned in horizontal- or vertical-lines. Furthermore, as already mentioned in Section 3.1, video texts with a height smaller than 5 pixels are hard to read for both, human and OCR engines. Therefore, these weak points almost do not affect the applicability and efficiency of our system.

[9]http://hunspell.sourceforge.net/ (last access: 14/09/2012)

[10]http://finereader.abbyy.com/

Table 10 Text recognition results of ICDAR 2011 challenge 1: "word recognition" competition [12] (last check: 14/09/2012)

Participant	Total edit distance	Correctly recognized words (%)
Our method 1	106.6	82.03
Deepak Kuma	108.7	82.9
Our method 2	145.4	74.95
Bolan Su	188.2	72.77
Chen Yang	189.1	61.98
Alvaro	226.8	66.88
Baseline	231.2	63.94

6 Conclusion

In this paper, we have proposed an entire workflow for text detection and recognition in video images. For text detection, we have developed a localization-verification scheme, which consists of a fast edge-based text detector and an adaptive refinement procedure intended to reduce false alarms. In the verification stage, we have applied the SWT- and SVM-based verifiers in the adaptive refinement workflow. In order to achieve a better text recognition rate, we have proposed a novel video text binarization algorithm. This algorithm consists of three main steps: text gradient direction analysis, seed pixel selection and seed-region growing. After seed region-growing, the video text image are converted into a standard OCR engine readable format. Experimental results show that the proposed text detection method is able to compete with other best existing methods. Moreover, our proposed workflow also performed very well on up-to-date video test sets that we have compiled and published. The skeleton-based binarization approach outperforms the other reference methods for recognizing video text images.

As future work, a context- and dictionary-based post processing could additionally improve the text recognition rate. By applying text tracking algorithms, we might be able to further improve the text detection result.

Acknowledgement This work has been supported by the *Mediaglobe* project. *Mediaglobe* is a SME project of the THESEUS research program, supported by the German Federal Ministry of Economics and Technology on the basis of a decision by the German Bundestag (FKZ: 01MQ09031).

References

1. Anthimopoulos M, Gatos B, Pratikakis I (2010) A two-stage scheme for text detection in video images. J Image Vis Comput 28:1413–1426
2. Bhaskar H, Mihaylova L (2010) Combined feature-level video indexing using block-based motion estimation. In: Proc. of 13th conference on information fusion (FUSION). Edinburgh, pp 1–8
3. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698
4. Chen D, Odobez JM, Bourlard H (2004) Text detection and recognition in images and video frames. J Pattern Recogn Soc 37(3):595–608
5. Deza MM, Deza E (2009) Encyclopedia of distances. Springer
6. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. In: Proc. of international conference on computer vision and pattern recognition, pp 2963–2970

7. Gllavata J, Ewerth R, Freisleben B (2004) Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In: Proceedings of 17th international conference on (ICPR'04), vol 1, pp 425–428
8. Gllavata J, Qeli E, Freisleben B (2006) Detecting text in videos using fuzzy clustering ensembles. In: Proceedings of the 8th IEEE international symposium on multimedia, ISM '06. IEEE Computer Society. Washington, DC, pp 283–290
9. Hanif SM, Prevost L (2009) Text detection and localization in complex scene images using constrained adaboost algorithm. In: Proceedings of the 2009 10th international conference on document analysis and recognition, ICDAR '09. IEEE Computer Society. Washington, DC, pp 1–5
10. Hua XS, Chen XY, Zhang HJ (2001) Automatic location of text in video frames. In: Proc. of ACM multimedia 2001 workshops: multimedia information retrieval, pp 24–27
11. Hua XS, Liu WY, Zhang HJ (2004) An automatic performance evaluation protocol for video text detection algorithms. IEEE Trans Circuits Syst Video Technol 14(4):498–507
12. ICDAR RWR (2011) http://www.cvc.uab.es/icdar2011competition/?com=results (last access: 10/07/2012)
13. Jung K, Kim KI, Jain AK (2004) Text information extraction in images and video: a survey. Pattern Recogn 37(5):977–997
14. Karatzas D, Mestre SR, Mas J, Nourbakhsh F, Roy PP (2011) Icdar 2011 robust reading competition: challenge 1: reading text in born-digital images (web and email). In: Proc. international conference on document analysis and recognition (ICDAR). Beijing, pp 1485–1490
15. Keysers D (2006) Comparison and combination of state-of-the-art techniques for handwritten character recognition: topping the mnist benchmark
16. Kim HH (2011) Toward video semantic search based on a structured folksonomy. J Am Soc Inf Sci Technol 62(3):478–492
17. Kim KI, Jung K, Park SH, Kim HJ (2001) Support vector machine-based text detection in digital video. Pattern Recogn 34(2):527–529
18. Li H, Kia O, Doermann D (1999) Text emhancement in digital video. In: Proc. of SPIE, document recognition IV, pp 1–8
19. Li H, Doermann DS, Kia OE (2000) Automatic text detection and tracking in digital video. IEEE Trans Image Process 9(1):147–156
20. Lienhart R, Wernicke A (2002) Localizing and segmenting text in images and videos. IEEE Trans Circuits Syst Video Technol 12(4):256–268
21. Niblack W (1986) An introduction to digital image processing. Prentice-Hall, Englewood Cliffs
22. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
23. Otsu N (1978) A threshold selection method from gray level histogram. IEEE Trans Syst Man Cybern 19(1):62–66
24. Pan YF, Hou X, Liu CL (2008) A robust system to detect and localize texts in natural scene images. In: Proceedings of the 2008 the eighth IAPR international workshop on document analysis systems, DAS '08. IEEE Computer Society. Washington, DC, pp 35–42
25. Qian X, Liu G, Wang H, Su R (2007) Text detection, localization and tracking in compressed video. In: Proc. of international conference on signal processing: image communication, pp 752–768
26. Sato T, Kanade T, Hughes EK, Smith MA, Satoh S (1999) Video OCR: indexing digital new libraries by recognition of superimposed captions. Multimedia Syst 7(5):385–395
27. Sauvola J, Pietikainen M (2000) Adaptive document image binarization. Pattern Recogn 33(2):225–236
28. Serra J (1983) Image analysis and mathematical morphology. Academic Press, Orlando
29. Shivakumara P, Phan TQ, Tan CL (2009) Video text detection based on filters and edge features. In: Proc. of the 2009 international conference on multimedia and expo. IEEE, pp 1–4
30. Sobel I (1990) An isotropic 3×3 image gradient operator. In: Machine version for three-dimentional scenes, pp 376–379
31. Sobottka K, Bunke H, Kronenberg H (1999) Identification of terxt on colored book and journal covers. In: Proc. of international conference on document analysis and recognition, pp 57–63
32. Sonnenburg S, Rätsch G, Henschel S, Widmer C, Behr J, Zien A, Bona, FD, Binder A, Gehl C, Franc V (2010) The shogun machine learning toolbox. J Mach Learn Res 11:1799–1802
33. Thillou CM, Gosselin B (2007) Color text extraction with selective metric-based clustering. Comput Vis Image Underst 107:1–2

34. Wolf C, Jolion JM, Chassaing F (2002) Text localization, enhancement and binarization in multimedia documents. In: Proc. of the international conference on pattern recognition, vol 2, pp 1037–1040
35. Yang H, Siebert M, Lühner P, Sack H, Meinel C (2011) Automatic lecture video indexing using video OCR technology. In: Proc. of international symposium on multimedia (ISM), pp 111–116
36. Zeng C, Ma H (2010) Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In: Proceedings of the 2010 20th international conference on pattern recognition, ICPR '10. IEEE Computer Society. Washington, DC, pp 2069–2072
37. Zhao M, Li S, Kwok J (2010) Text detection in images using sparse representation with discriminative dictionaries. J Image Vis Comput 28:1590–1599
38. Zhong Y, Zhang HJ, Jain A (2000) Automatic caption localization in compressed video. IEEE Trans Pattern Anal Mach Intell 22(4):385–392
39. Zhou Z, Li L, Tan CL (2010) Edge based binarization for video text images. In: Proc. of 20th international conference on pattern recognition. Singapore, pp 133–136

Haojin Yang received the Diploma engineering degree at the Technical University Ilmenau, Germany, in 2008. In 2010, he is research assistant and PhD student at the Hasso Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam. His current research interests revolve around multimedia analysis, information retrieval, semantic web technology, content based search technology. He is involved in the development of the Web-University project space enhancement of the tele-TASK video recording system. He is also involved in the Mediaglobe project of HPI. Mediaglobe is a SME project of the THESEUS research program, supported by the German Federal Ministry of Economics and Technology on the basis of a decision by the German Bundestag, cf. http://www.projekt-mediaglobe.de/.

Bernhard Quehl received the Diploma Informatiker degree from the Brandenburg University of Applied Sciences, Germany. Now he works as research assistant at the Hasso Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam.

Harald Sack is senior researcher at the Hasso Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam. After graduating in computer science at the University of the Federal Forces Munich Campus in 1990, he worked as systems/network engineer and project manager in the Signal Intelligence Corps of the German Federal Forces from 1990 to 1997. In 1997 he became an associated member of the graduate program “mathematical optimization” at the University of Trier and graduated with a PhD thesis on formal verification in 2002. From 2002 to 2008 he did research and teaching as a postdoc at the Friedrich-Schiller-University in Jena and since 2007 he has a visiting position at the HPI. His areas of research include multimedia retrieval, semantic web, knowledge representations and semantic enabled retrieval. Since 2008 he is speaker of the special interest group ‘multimedia- and hypermediasystems’ of the German Computer Science Society (Gesellschaft für Informatik) and general secretary of the german IPv6 council.