A Text Detection, Localization and Segmentation System for OCR in Images

Julinda Gllavata[1], Ralph Ewerth[1] and Bernd Freisleben[1,2]
[1]*SFB/FK 615, University of Siegen, D-57068 Siegen, Germany*
[2]*Dept. of Math. and Computer Science, University of Marburg, D-35032 Marburg, Germany*
{gllavata, ewerth}@fk615.uni-siegen.de; freisleb@informatik.uni-marburg.de

Abstract

One way to include semantic knowledge into the process of indexing databases of digital images is to use caption text, since it provides important information about the image content and is a very good entity for queries based on keywords. In this paper, we propose an approach to automatically localize, segment and binarize text appearing in complex images. First, an unsupervised method based on a wavelet transform is used to efficiently detect text regions. Second, connected components are generated, and the exact text positions are found via a refinement algorithm. Third, an unsupervised learning method for text segmentation and binarization is applied using a color quantizer and a wavelet transform. Comparative experimental results demonstrate the performance of our approach for the main processing steps: text localization and segmentation, and in particular their combination.

1. Introduction

The area of content-based image indexing and retrieval of digital multimedia databases has drawn a lot of attention in recent years. Extensive research is being done in order to develop efficient algorithms for identifying semantic content such as faces, human gestures, various objects, genres, etc. Text appearing in images can provide very useful semantic information and may be a good key to describe the image content.

In general, text appearing in images can be classified into two groups: scene text and artificial text [10]. Scene text is part of the image, and usually does not represent information about the image content, (traffic signs in an outdoor scene, etc.), whereas artificial text is laid over the image in a later stage (e.g. the name of somebody during an interview). Artificial text is often a good key for successful indexing and retrieval of images (or videos). At first, extraction of text may seem to be a trivial application for existing optical character recognition (OCR) tools. However, compared to OCR for document images, extracting text from real images faces numerous challenges due to lower resolution, unknown text color, size and position, or complex backgrounds.

Text extraction and recognition, which includes text detection, localization, binarization and recognition, is a useful process regarding text-based image indexing. The first three processing stages are important to achieve high-quality text recognition results when applying an OCR system. There are two classes of methods for text localization: connected component (CC)-based and texture-based methods. The first class of methods [1, 2, 3] employs connected component analysis, which is based on the analysis of the geometrical arrangement of edges or homogeneous color and grayscale components that belong to characters. They are simple to implement, but they are not very robust for text localization in images with complex background. The second class of methods [4, 6, 8, 10, 13] considers texts as regions with distinct textural properties. Methods of texture analysis like Gabor filtering and the wavelet transform are used to analyze text regions.

In this paper, we propose a hybrid approach to automatically localize, segment and binarize text which is superimposed over complex images. Texture-based methods and CC-based methods are combined. First, an unsupervised texture-based method is used to efficiently detect text regions. Texture features are extracted from the high-frequency wavelet coefficients. Then, the text candidates are localized via CC-based filtering using predefined geometric text properties, and some false alarms are discarded. An unsupervised learning method is applied to achieve accurate character segmentation and binarization. Here, the text and background color are determined using a color vector quantizer and line histograms. The estimated text color and the standard deviation of the

Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04)
0-7695-2217-3/04 $20.00 © 2004 **IEEE**

wavelet transformed image are used to classify the pixels into text and non-text pixels. The performance of text localization and its combination with the proposed text segmentation algorithm is evaluated using a test set of complex images. Finally, our re-implementation of an alternative high-quality approach [2] is compared with the proposed approach.

The remainder of the paper is organized as follows. Section 2 provides a brief overview of related work in the field. Section 3 presents the individual steps of our approach to text localization, segmentation and binarization in detail. Section 4 describes the comparative experimental results obtained for a set of images. Section 5 concludes the paper and outlines areas for future research.

2. Related work

Cai et al. [2] have presented a text detection approach which uses character features like edge strength, edge density and horizontal alignment. First, they apply a color edge detection algorithm in YUV color space and filter out non-text edges using a low threshold. Then, a local thresholding technique is employed in order to keep low-contrast text and further simplify the background. An image enhancement process using two different convolution kernels follows. Finally, projection profiles are analyzed to localize text regions.

Jain and Yu [7] first perform color reduction by bit dropping and color clustering quantization, and afterwards a multi-value image decomposition algorithm is applied to decompose the input image into multiple foreground and background images. Then, CC analysis is performed on each of them to localize text candidates. This method extracts only horizontal texts of large sizes.

Agnihotri and Dimitrova [1] have presented an algorithm which operates directly on the red frame of the RGB color, with the aim of obtaining high contrast edges for the frequent text colors. By means of the convolution process with different masks, they first enhance the image and then detect edges. Then, neighboring edge pixels are grouped to single CC structures. At the end, a simple binarization process takes place.

Gllavata et al. [3] have presented a method to localize and extract text automatically from color images. First, they transform the color image to a grayscale image and only the Y channel is used further. The text candidates are found by analyzing the projection profile of the edge image. Finally, a binarized text image is generated using a simple binarization algorithm based on a global threshold.

Hao et al. [6] use at first a color image edge detector to segment the image. Then, an artificial neural network is used for further classification of text blocks and non-text blocks, using features obtained by Gabor filtering. To improve the precision of the system, the neural network is trained with every block which is falsely classified as a text block, until the desired results are obtained.

Li et al. [8] scan the image using a small window of 16x16 pixels, classifying each of the windows as text or non-text using a three-layer neural network, based on the local features extracted from the high-frequency wavelet coefficients. To detect various text sizes, they propose a three-level pyramid approach. At the end a projection profile analysis is used to extract text elements from text blocks.

Wu et al. [13] have proposed an automatic text extraction system which first uses distinctive characteristics of texts, such as the fact that text possesses certain frequency and orientation information or that text shows spatial cohesion (characters of the same text string are of similar height, orientation and spacing) to identify the possible text regions in an image. As a second step, bottom-up methods are applied to extract connected components. A simple histogram-based algorithm is proposed to automatically find the threshold value for each text region, making the text segmentation process more efficient.

Lienhart and Wernicke [10] have proposed an approach to detect mainly horizontally aligned texts. Text lines are identified by using a multi-layer feed-forward network trained to detect text at a fixed scale. The gradient image of the input RGB image serves as their feature for text localization. A multi-scale schema is used to detect texts of different sizes. Localized text lines are scaled to a fixed height and segmented into a binary image. First, the color of the text and background is determined. Second, the background is reduced using a connected component based analysis. At the end, the binarization is performed using the intensity value halfway between the intensity of the text colors and the background color as a threshold.

3. Our approach

The proposed approach for text extraction and recognition in images can be divided into several elementary tasks:

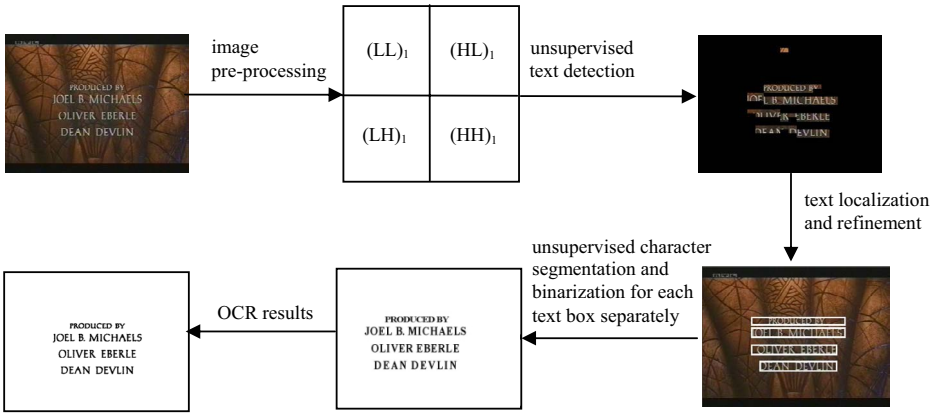


Figure 1. The main processing steps of the proposed system.

- *Text detection* is aimed at identifying image parts containing text.
- *Text localization* merges text regions which belong to the same text candidate and determines the exact text positions.
- *Character segmentation and binarization* include the separation of the text from the image background. The output of this step is a binary image where black text characters appear on a white background.
- *Text recognition* performs OCR on the binarized image and converts the binarized image to ASCII text.

In this paper, we present solutions for the first three tasks in order to allow text recognition by a subsequently used OCR system. The proposed system consists of two main modules: (1) Text Detection and Localization (TDL); and (2) Character Segmentation and Binarization (CSB).

The TDL module is aimed at detecting and localizing the possible text blocks in the image. As a result, this module creates rectangular boxes that surround the detected text areas.

The CSB module employs an unsupervised clustering process which aims at classifying each pixel from the text rectangle into "text" or "background" pixels. The output of this module is the binarized text image. No assumptions are made about the nature (polarity, color, font, etc) of text appearing in images during the text detection step. The text localization and binarization processes work for texts exceeding a minimum height and with a horizontal alignment. However, they may be easily adapted to texts with a vertical alignment. The main steps of the entire system are shown in Figure 1.

Both modules apply a wavelet transform. One important purpose of the wavelet transform is to decompose a signal into sub-bands at various scales and frequencies. In case of images, the wavelet transform is e.g. useful to detect edges with different orientations. The wavelet transform can be implemented using filter banks consisting of high-pass and low-pass filters. The application to an image consists of a filtering process in horizontal direction and a subsequent filtering process in vertical direction. For example, when applying a 2-channel filter bank (L: low pass filter, H: high-pass filter), four sub-bands are obtained after filtering: LL, HL, LH and HH. The three high-frequency sub-bands (HL, LH, HH) enhance edges in horizontal, vertical or diagonal direction, respectively. Since text areas are commonly characterized by having high contrast edges, high valued coefficients can be found in the high-frequency sub-bands.

3.1 Text detection and localization

At first, the local texture features are analyzed to determine the areas of the image that probably are text areas. Then, a refinement algorithm is applied to further improve the localization of the text candidates. Given an image, the main algorithmic processing steps are as follows (see pseudo code in Figure 2).

Unsupervised Text Detection. First, in case of an image with RGB color space, the image is converted into a gray-scale image using an appropriate transform from RGB to YUV color space. During further processing, only the Y channel is used, since the luminance information is sufficient for text detection. Then, a wavelet transform is applied to the image using a two-channel 5/3 filter bank (evaluated in [12]).

1. Convert the color image to a grayscale image.
2. Apply the wavelet transform to the grayscale image.
3. For each pixel block$_i$ of size MxN from the transformed image (e.g. in the HL-subband) do:
3.1 Create a feature vector $f_i(x_1, x_2)$, where x_1 is estimated using formula (1) and x_2 is estimated using formula (2).
4. Initialize the three clusters ("text", "background" and "complex background") with the pixel block whose feature vectors have the minimal Euclidian distance to the ideal feature vectors.
5. Run the clustering algorithm k-means to classify the image pixel blocks in three clusters.
6. Estimate the connected components (CC) in the "text" cluster to build bounding boxes.
7. Refine the rectangles that surround the text components and analyze them geometrically.

Figure 2. The pseudo code for the proposed text localization algorithm.

Texture analysis is performed block-wise on the HL wavelet coefficients of the transformed image (the high-pass horizontally filtered image and the low-pass vertically filtered image). For each block of size M*N (e.g. 16*8), two features are derived from the corresponding wavelet coefficients: (i) the standard deviation of the histogram H; (ii) the standard deviation of the high frequency coefficients. The first feature is computed using formula (1):

$$stdevHisto_{window}(H) = \sqrt{\frac{\sum_{i=1}^{k}(f(i) - mean_{f(i)})^2}{k}}, \begin{cases} if\ H_i > 0\ then \\ \quad f(i) = i \\ elseif\ H_i = 0: \\ \quad f(i) = 0 \end{cases}, \quad (1)$$

where H_i is the value of histogram bin i, k is the number of histogram bins. This feature choice is based on the observation that wavelet coefficients in the LH, HL and the HH sub-bands tend to follow a Laplacian distribution for the non-text areas, while the coefficients for text areas are dispersed and concentrated on a few discrete values [9]. It is assumed that the text blocks are characterized by a higher standard deviation than other blocks. The standard deviation of wavelet coefficients for the second feature is defined as follows:

$$stdev_{window_{MxN}}(x, y) = \sqrt{\frac{1}{M*N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(x+i, y+j) - mean_{window_{MxN}})^2}, \quad (2)$$

where I(x+i, y+j) is the wavelet coefficient at pixel position (x+i, y+j), and $mean_{window}(x, y)$ is defined as:

$$mean_{window_{MxN}}(x, y) = \frac{1}{M*N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(x+i, y+j), \quad (3)$$

The features in the vector are normalized for each component separately using the corresponding maximum computed from the image under consideration. Finally, the classical k-means algorithm is applied to obtain clusters whose members have the minimum Euclidian distance to the respective cluster mean feature vector. In our case, there are two clearly distinguishable clusters of pixel blocks: "text" and "background". In order to deal with complex background, an intermediate category has been defined named "complex background". Figure 3/b illustrates the result of this process. More details about this unsupervised text detection approach are described in [4].

Text Localization. First, a so called marked image is created based on the classified pixel blocks in the "text" cluster. This marked image is modeled as an undirected graph, where nodes represent text blocks and edges represent the fact that these text blocks are neighbors (eight-neighborhood). Then, in order to find the connected components present in the image, a depth-first-search (DFS) algorithm is performed, which marks the nodes (i.e. the respective text blocks) during the DFS traversal as belonging to a certain CC. Components which do not fulfill the minimum geometrical restrictions and appear alone, are discarded as "background" (see Figure 3/c). Long vertical components (regions) are also eliminated.

The left-most, the right-most, and the top and bottom coordinates of each component are taken to create the initial bounding rectangle (see Figure 3/d). Then, text box refinement follows. The standard deviation $stDev_k$ of the distances between sharp edges in the HL sub-band image is analyzed for each line k. It is assumed that the distances between sharp edges will be more regular in case of a text line than in the other case. As a result, we expect that the text lines will be characterized by a lower standard deviation of edge distances than the others. Each text box is divided in two classes of smaller text boxes $textBox_j$, where $stDev_k$ is either lower or higher than a predefined threshold. Text boxes which belong to the second class are considered as false alarms. At the end, a geometrical analysis is applied to eliminate the text boxes which do not fulfill the following geometric restrictions: a text box must exceed a minimum height (e.g. 8 pixels) and the text box width must be greater than its height (see Figure 3/e).

(a) (b) (c) (d) (e)

Figure 3. Intermediate results of the TDL module. (a) the original image; (b) the result of the unsupervised TD; (c) elimination of two isolated text blocks; (d) the result of TL; (e) the result of the TL + refinement.

3.2 Character segmentation and binarization

Once the positions of text bounding boxes are determined via the previous processing steps, character segmentation follows. The proposed method is designed to horizontally segment aligned text characters in images of arbitrary font, size and color. No assumptions are made about the text polarity. The input is the original image and the coordinates of the text bounding boxes in the image. In contrast to many other approaches, the use of global thresholds [1, 3] or local thresholds [10, 13] is avoided in our approach by applying unsupervised clustering. For each localized text, the main processing steps of the proposed CSB algorithm are as follows (see pseudo code in Figure 4).

First, the bounding boxes are increased such that no text pixels fall outside the boxes. Each text box is extended by 4 pixels in each direction (vertically and horizontally). Second, the image resolution is enhanced up to 300 dpi using a cubic interpolation. It has been shown in [5] that the segmentation and the subsequent OCR processes perform better on a higher resolution than on the original video frame resolution of 72 dpi. The dominating text and background colors are estimated for each text box following the approach suggested in [10]. The problem of character segmentation is considered as a classification problem of pixels into text pixels and background pixels. For this purpose, a slightly modified k-means algorithm is used. Basically, the feature vector consists of the RGB color values, scaled to the range [0, 1]. Furthermore, the feature vector is extended with the standard deviation (see Formula (2)) of wavelet coefficients in the three high frequency sub-bands in order to consider local text-specific characteristics. To achieve this, a small sliding window$_{MxN}$ (e.g. 3*3 pixel) is moved over the text box in order to consider local image properties. This technique is motivated by two observations. First, characters usually have a unique texture. Second, the border of superimposed characters results in high-contrast edges. Consequently, we apply the wavelet transform to the image to consider these properties and pass them to the subsequent clustering algorithm. The standard deviation of wavelet coefficients in a small environment should be low on a character's texture, but high at its boundary [5]. The two clusters "text" and "background" are initialized with the text color and the background color, respectively, and 1 (high frequencies for "text") and 0 for the wavelet features. The k-means algorithm is applied to the feature vectors representing all bounding box pixels to classify them in two clusters. Finally, the binarization is done where all pixels in the "text" cluster are painted black and those from the "background" cluster are painted white (see Figure 6).

4. Experimental results

The proposed text localization and segmentation approach was tested on several types of images. The quantitative evaluation of text localization and segmentation is an open research issue due to several reasons: (i) the lack of common image data bases and (ii) the use of different measures by different researchers. In this paper, two types of evaluations will be presented. The first one reports the accuracy of the text localization module in terms of recall and precision. The second evaluation reports the performance of the whole system (TDL + CSB + OCR). This evaluation is done on the OCR level in terms of character (word) recognition rate. However, such an evaluation depends on the quality of the OCR software. Comparative experimental results are reported for both cases. To evaluate the performance of the proposed system, an alternative high-quality approach [2] has been re-implemented to allow an objective comparison with our approach. Several parameters have to be set in the approach of [2] and, unfortunately, its localization performance depends noticeably on the parameter settings. We have conducted several experiments to estimate the parameters which gave the best results. Furthermore, this algorithm employs a coarse to fine projection analysis at the end to localize the text candidates which

1. *Increase the text bounding box (e.g. 4 pixels in each direction).*
2. *Increase the text image resolution to 300 dpi and rescale the text boxes.*
3. *Estimate the possible text and background color.*
4. *Apply the wavelet transform to the text boxes.*
5. *For each pixel$_i$ in a text bounding box do:*
  5.1 *Create the feature vector $f_i(r, g, b, x_1, x_2, x_3)$, where r, g ,b are the values of each of the channels R, G and B, and x_1, x_2, x_3 are the standard deviations of the wavelet (LH, HL, and HH-subbands) coefficients in the 8-neighborhood of pixel$_i$.*
6. *Initialize the two clusters "text" and "background" with the feature vectors which have the minimal Euclidian distance to the ideal feature vectors.*
7. *Run the k-means clustering algorithm to classify the pixels into the "text" and "background" cluster.*
8. *Binarize the text image so that pixels which are assigned to the "text" cluster are marked as black.*

Figure 4: The pseudo code for the proposed text segmentation and binarization algorithm.

is not described in detail in [2]. We have implemented such a refinement following a description which was kindly provided by the authors of [2]. In this step, two thresholds are used: (a) the number of edges in a line (numberEdgesLine) and (b) the number of edges in a column (numberEdgesColumn). The algorithm was tested with the following parameters: numberEdgeLine = 10 and numberEdgeColumn = 5. In contrast to [2], a threshold of 0.3 was used (instead of 0.6) to binarize the enhanced edge image. The remaining parameters were set as described in [2].

4.1 Text detection and localization results

The proposed text detection and localization system and the alternative approach were tested on two test sets containing various types of images. The first test set (TS1) consists of 51 images and covers a wide variety of background complexity and text types. These images (video frames) were captured from commercial televisions broadcasts, news videos, movie sequences and web pages. In total, there are 438 words in those 51 images. The second test set (TS2) consists of 44 video frames taken from the MPEG-7 video test set containing 241 words.

Table 1. The detection and localization performance in terms of pixel-based recall and precision of the proposed approach and of the approach in [2] for two test sets.

Test Set	Re-Impl. of alg. in [2]		Proposed TD+TL alg.	
	recall	precision	recall	precision
TS1	73.43 %	53.42 %	82.56 %	74.28 %
TS2	76.60 %	54.75 %	88.88 %	68.27 %
TS1+TS2	*74.36 %*	*53.81 %*	*84.43 %*	*72.31 %*

Table 2. The detection and localization performance in terms of word-based recall and precision for the proposed approach and the re-implementation of [2] for two test sets.

Test Set	Re-Impl. of alg. in [2]		Proposed TD+TL alg.	
	recall	precision	recall	precision
TS1	63.55 %	87.74 %	95.21 %	96.30 %
TS2	80.99 %	91.16 %	95.44 %	88.80 %
TS1+TS2	*69.36 %*	*89.06 %*	*95.27 %*	*93.47 %*

The localization performance was evaluated on both text box level and pixel level in terms of recall and precision. Recall and precision are commonly used performance in the field of information indexing and retrieval. Recall is defined as:

$$\text{Recall} = \frac{\text{Correct localized words (text pixels)}}{\text{(Correct localized words (text pixels) + Missed words (text pixels))}}, \quad (4)$$

whereas precision is defined as the number of correctly localized text words (pixels) divided by the number of all localized words (text pixels), including false alarms:

$$\text{Precision} = \frac{\text{Correct localized words (text pixels)}}{\text{(Correct localized words (text pixels) + False alarms)}}, \quad (5)$$

The performance was measured on two levels. The first one describes the word-based localization performance. In this case, correctness was determined manually by checking the localization results. A text word is considered as localized correctly, if the word is completely surrounded by a box (at least 90%), while a text box is considered as a false alarm, if there is not any text in that box. The second one evaluates the performance of the approaches on the pixel level. First, the ground truth text boxes were drawn manually. Then, the evaluation is done automatically, comparing the ground truth data with the localization results of the algorithm. The results are shown in Table 1 (pixel-level) and in Table 2 (word level). The experimental results demonstrate the effectiveness of the proposed TDL module. The system has achieved a pixel-based recall of 84.43% and a precision of 72.31% for all 95 images, while on the word-level 644 out of 676 ground truth words were localized correctly, whereas 45 image

Figure 5. Some exemplary results of the proposed text detection and localization module.

Table 3. The binarization and recognition performance is shown in both cases for test set (TS3), where the output of the respective localization algorithm is used as input for the text segmentation & binarization.

Test Set	Re-Impl. of the alg. in [2]		The proposed TDL alg.		Re-Impl. of alg. in [2] + Otsu's binarization method (System-B) + OCR		The proposed system + OCR	
	Pixel-based recall	Pixel-based precision	Pixel-based recall	Pixel-based precision	Word-based recognition rate	Char.-based recognition rate	Word-based recognition rate	Char.-based recognition rate
TS3	75.34 %	58.63 %	83.06 %	74.85 %	45.23 %	68.60 %	55.83 %	77.71 %

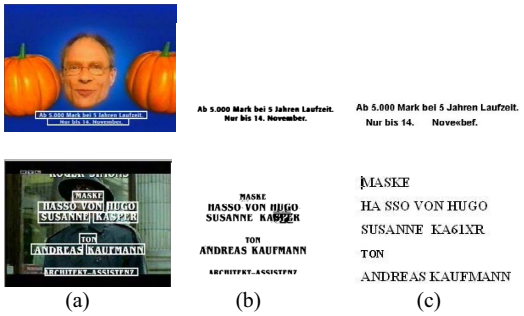
areas were falsely localized as text areas (see Table 2). In Figure 5, some exemplary results are shown for our text detection and localization algorithm. Our re-implementation of the alternative approach [2] was also tested for both test sets (see also Table 1 and 2). The re-implemented algorithm of [2] has achieved an overall pixel-based recall of 74.36% and a precision of 53.81%, while a recall of 69.36% and a precision of 89.06% was achieved on the word level. The experimental results show that the proposed localization method clearly outperforms the re-implementation of [2] in terms of both pixel level and word level. Table 1 and 2 show that the recall of the re-implemented approach [2] is lower at the word level compared to the pixel level, whereas the TDL approach achieves even a higher recall at the word level. This occurs because one word was counted as localized correctly only if the bounding box surrounded at least approximately 90% of the word under consideration which was achieved more often while using the TDL approach.

4.2 Text segmentation and binarization results

In order to evaluate the performance of the proposed character segmentation and binarization algorithm, OCR experiments were conducted with a set of 38 images (TS3). These images were selected from the two test sets (section 4.1). In total, there are 325 words and 2156 characters in those 38 images. The recognition rate is used as an objective measure of the system performance. We have used a demo version of the commercial OCR software ABBYY FineReader 7.0 Professional for recognition purposes. After binarization, the binary text image was fed manually to the OCR software. Some exemplary results for all three processing stages are shown in Figure 6.

The text localization algorithm in [2] is combined with the classical binarization method from Otsu [11] in order to allow some comparative experiments. This system is called System-B from now on. The OCR experimental results and their comparison as well as the localization performance for test set 3 are displayed in Table 3. If the text bounding boxes are computed using the proposed localization approach and binarization is performed using the proposed text segmentation algorithm, a character recognition performance of 77.7% is achieved, while the word recognition rate is 55.8%. Using the output of System-B, the character recognition rate is 68.6% and the word recognition rate is 45.2%. We would like to point out that the usage of an appropriate dictionary could lead to a noticeably higher word recognition rate.

Actually, a word is not recognized correctly if only



(a) (b) (c)

Figure 6. Some results after: (a) localization; (b) binarization; (c) recognition process.

one character is not recognized. For example, consider the last example in figure 6: The character recognition rate is about 94% (46 of 49 characters recognized correctly) while the word recognition is about 89%. Assume the worst case that the 3 errors are distributed over 3 words, the word recognition rate would decrease even down to 67%.

Overall, the proposed localization, segmentation and recognition system demonstrates a robust performance and achieves a character recognition rate of 77.71%. The alternative approach (System-B) achieves a worse character recognition rate of 68.60%.

5. Conclusions

In this paper, we have proposed an approach to automatically localize and segment text appearing in images with complex backgrounds. First, k-means clustering based on wavelet features was used to efficiently detect the "text" blocks. These blocks were connected using local neighborhood and geometrical properties. Then, the exact positions of text bounding boxes were determined via a simple refinement process. For each box, the text color was estimated and some local image properties were exploited in the subsequent text binarization process, which utilized k-means clustering, too. Experimental localization and OCR results for a set of images were presented to demonstrate the good performance of the entire system. The proposed system outperformed our re-implementation of an alternative high-quality approach.

There are several areas for further research. The integration of a freely available OCR system will be investigated to support the whole processing chain from the input image to the ASCII text in the end. Finally, the implementation of an advanced system for automatic indexing of images and videos and their content-based retrieval is envisaged in the near future.

6. Acknowledgments

This work is financially supported by the Deutsche Forschungsgemeinschaft (SFB/FK 615, Teilprojekt MT) and by Deutscher Akademischer Austausch Dienst (DAAD, Stability Pact for South Eastern Europe). The authors would like to thank M. Grube for his implementation work and M. Gollnick, M. Grauer, F. Mansouri, E. Papalilo, R. Sennert and J. Wagner for their valuable support.

7. References

[1] L. Agnihotri, and N. Dimitrova, "Text Detection for Video Analysis", *Proc. of Int'l Conf. on Multimedia Computing and Systems,* Florence, 1999, pp. 109-113.

[2] M. Cai, J. Song, and M. R. Lyu, "A New Approach for Video Text Detection", *Proc. of IEEE Int'l Conference on Image Processing*, Rochester, New York, USA , 2002, pp. 117-120.

[3] J. Gllavata, R. Ewerth, and B. Freisleben, "A Robust Algorithm for Text Detection in Images", *Proc. of 3rd Int'l Symposium on Image and Signal Processing and Analysis*, Rome, 2003, pp. 611-616.

[4] J. Gllavata, R. Ewerth, and B. Freisleben, "Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients", *Proc. of Int'l Conf. on Pattern Recognition*, Cambridge, UK, 2004, pp. 425-428.

[5] J. Gllavata, R. Ewerth, T. Stefi, and B. Freisleben, "Unsupervised Text Segmentation Using Color and Wavelet Features", *Proc. of 3rd Int'l Conf. on Image and Video Retrieval,* Dublin, Ireland, 2004, pp. 216-224.

[6] Y. Hao, Z. Yi, H. Zengguang, and T. Min, "Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network and CED", *Journal of WSCG Vol. 11, No.1*, Plzen, Czech Republic, 2003, ISSN 1213-6972.

[7] A. K. Jain, and B. Yu, "Automatic Text Location in Images and Video Frames", *Pattern Recognition 31(12),* 1998, pp. 2055-2076.

[8] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Videos", *IEEE Transact. on Image Processing*, Vol. 9, Nr. 1, 2000, pp. 147-156.

[9] J. Lia and R. M. Gray, "Text and Pictures Segmentation by the Distribution Analysis of Wavelet Coefficients", IEEE *Int'l Conf. on Image Proc.*, Chicago, 1998, pp. 790-794.

[10] R. Lienhart, and A. Wernicke, "Localizing and Segmenting Text in Images and Videos", *IEEE Transact. on Circuits and Systems for Video Technology*, Vol. 12, Nr. 4, 2002, pp. 256-268.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems*, *Man and Cybernetics*, 9 (1), 1979, pp. 62–66.

[12] J. Villasenor, B. Belzer, and J. Liao, "Wavelet Filter Evaluation for Efficient Image Compression", *IEEE Transact. on Image Processing*, Vol. 4, 1995, pp. 1053-1060.

[13] V. Wu, R. Manmatha, and E.M. Riseman, "Textfinder: An Automatic System to Detect and Recognize Text in Images", *IEEE Transact. on Pattern Analysis and Machine Intelligence*, Vol. 21, Issue 11, 1999, pp. 1224-1229.