Automatic Lecture Video Indexing Using Video OCR Technology

Haojin Yang, Maria Siebert, Patrick Lühne, Harald Sack, Christoph Meinel
Hasso Plattner Institut (HPI), University of Potsdam
P.O. Box 900460, D-14440 Potsdam
e-mail: {Haojin.Yang, Maria.Siebert, Harald.Sack, Meinel}@hpi.uni-potsdam.de; Patrick.Luehne@student.hpi.uni-potsdam.de

***Abstract*—During the last years, digital lecture libraries and lecture video portals have become more and more popular. However, finding efficient methods for indexing multimedia still remains a challenging task. Since the text displayed in a lecture video is closely related to the lecture content, it provides a valuable source for indexing and retrieving lecture contents. In this paper, we present an approach for automatic lecture video indexing based on video OCR technology. We have developed a novel video segmenter for automated slide video structure analysis and a weighted DCT (*discrete cosines transformation*) based text detector. A dynamic image constrast/brightness adaption serves the purpose of enhancing the text image quality to make it processible by existing common OCR software. Time-based text occurence information as well as the analyzed text content are further used for indexing. We prove the accuracy of the proposed approach by evaluation.**

***Keywords*-video OCR; video segmentation; e-learning; multimedia retrieval; recorded lecture videos.**

I. Introduction

In the last decade tele-teaching and lecture video portals have become more and more popular. The amount of lecture video data available on the World Wide Web (WWW) is constantly growing. Thus, the challenge of finding lecture video data on the WWW or within lecture video libraries has become a very important and challenging task.

We focus our research on such lecture recordings having been produced by state-of-the-art lecture recording systems like *tele-TASK* [6]. The development of tele-TASK began in 2001; today, more than 3200 lectures have already been recorded using tele-TASK systems which are available on [10]. With this kind of a system, we are able to record the lecure video such that we combine two video streams: the main scene of lecturers which is recorded by using a video camera, and the second which captures the images projected onto the screen during the lecture through a frame grabber. We synchronize the frame grabber output with the video camera so that each slide can be linked to a video recording segment. In this way, indexing two-part lecture videos can be performed by indexing the slide videos only.

Content-based retrieval within video data can be performed by automated extraction of textual metadata. Techniques from standard OCR, which focus on high resolution scans of printed (text) documents, have to be improved and adapted to be also applicable for video OCR. In video

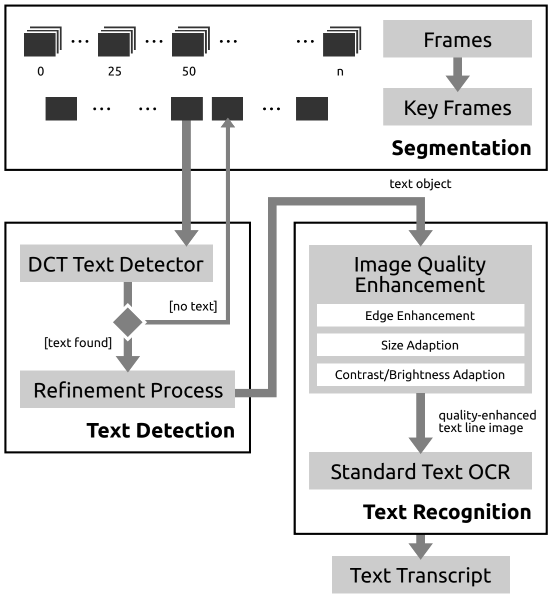


Figure 1. The entire system workflow. After having segmented the frame sequence, text detection is performed on every single key frame in order to find text occurences. The resulting text objects are then used for text recognition; this includes enhancing the text images and passing them to an OCR engine which eventually returns the desired text transcript.

OCR, the text within video frame has to be automated localized and separated from its background and the image quality enhancement have to be applied before standard OCR procedures can process the text successfully.

In this paper, we propose an entire workflow for both the structural segmentation of lecture videos and the video OCR analysis, as illustrated in Fig. 1. We developed a video segmenter based on fast connected component (CC) analysis and slide transition recognition. For text localization, we realized a weighted DCT-based text detector–this kind of text detector is invariant to font size and style while it provides suitable runtime efficiency. An edge-filter-based refinement algorithm completes the text detection process by supplying text bounding boxes of high accuracy. Usually, text within the bounding boxes has different resolutions and comes with a heterogeneous background and a difficult contrast ratio that often prohibits valid OCR results. In order to resolve this

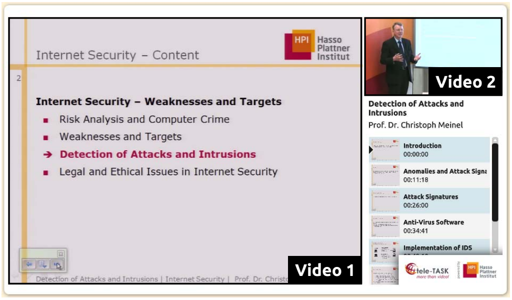

Internet Security – Content
HPI Hasso Plattner Institut
Internet Security – Weaknesses and Targets
Risk Analysis and Computer Crime
Weaknesses and Targets
Detection of Attacks and Intrusions
Legal and Ethical Issues in Internet Security
Video 1
Video 2
Detection of Attacks and Intrusions
Prof. Dr. Christoph Meinel
Introduction 00:00:00
Anomalies and Attack Signa 00:11:18
Attack Signatures 00:26:00
Anti-Virus Software 00:34:41
Implementation of IDS
tele-TASK

Figure 2. An example lecture video from *tele-TASK* [10]. Video 2 shows the speaker giving his lecture, whereas his presentation is played in video 1.

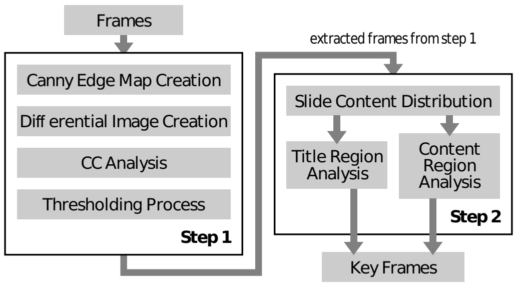


Figure 3. Segmentation workflow. (Step 1) Adjacent frames are compared with each other by applying *Connected Components Analysis* on their differential edge maps. (Step 2) Title and content region analysis ensure that only actual slide transitions are captured.

issue, the processing continues with image size adaption and dynamic contrast/brightness adaption to improve the image quality. Finally, we process all quality-enhanced text lines by using common OCR engine.

The paper is organized as follows: Section 2 presents related work, whereas the sections 3, 4, 5 and 6 describe our proposed framework in detail. Experimental results are provided in section 7. Section 8 concludes the paper with an outlook on future work.

II. Related Work

[3] propose an automated system for authoring hypermedia lecture recordings. They use a VGA frame grabber to extract the slide frames and a common OCR tool on the extracted frames for text recognition. Instead of applying the video segmentation process, OCR is executed on every extracted slide frame. This is much less efficient concerning computation time and implies a large amount of redundancies. Furthermore, they don't make use of a text localization process to identify text regions. Therefore, the recognition accuracy of their approach is much lower than our system's.

[9] use the automatic speech recognition (*ASR*) output for lecture video retrieval. However, since ASR for lecture videos is still an active research area, most of those recognition systems have poor recognition rates degrading the indexing and the retrieval yet.

[4] experiment with differently weighted DCT coefficients in order to improve the accuracy of their text detection. Their test images have a fixed size of 256×256 pixels only, and the weight functions are not suitable for frames with larger resolutions.

[2] present a text location method using image corner points and edge maps to reduce false alarms. The edge refining methods increase the detection accuracy significantly, however, their statical thresholding methods are not suitable for videos of different genres.

III. Video Segmentation

By decomposing the video into a set of representative key frames, we are able to index the video. These selected frames will be processed in other steps later, since they capture and encapsulate the entire video content.

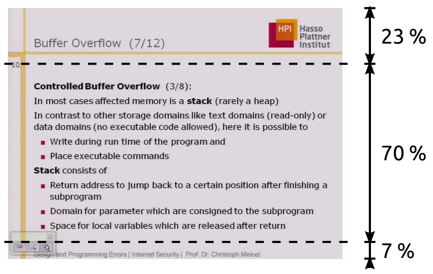
Regarding the characteristics of lecture videos, we can see that the retrieval of video contents can be achieved through the extraction of text information from slide videos: Fig. 2 shows an example lecture video that consists of two video streams showing the speaker and the current slide respectively—in such kind of videos, each part of the video can be associated to a corresponding slide. Hence, we carry out the analysis of the slide video stream in our study.

The segmentation process can be achieved by using frame differencing metrics which take both image structure and color distribution into account. The global pixel difference metric has commonly been used for determining slide segments. One deficiency of this method is that the high-frequency image noise within video frames can still degrade the segmentation accuracy.

Observing the slide video stream, we can see that video contents such as texts, figures, tables etc. can all be considered as a collection of connected components (*CC*s). Therefore, the difference between two frames can be determined by calculating the difference of the amount of CCs. In this way, we are able to control the valid size of CCs so that the image noise can be removed from the comparison. We have investigated several CC algorithms and finally chose the most efficient one according to [1].

Our segmentation algorithm consists of two steps (cf. Fig. 3): In the first step, the entire slide video is analyzed. For reasons of efficiency, we do not perform the analysis on every video frame; instead, we established a time interval of one second. Therefore, our video segmenter considers only one frame per second (fps), i. e. for a 25-fps-encoded video, only one frame of 25 is considered.

Subsequently, we create the differential edge map of two adjacent frames and perform the CC analysis on this map. In order to ensure that each newly emerging knowledge point or newly added figure within a slide can be detected, we have identified the segmentation threshold value T_{s1}. This means that a new segment is captured if the number of CCs



Buffer Overflow (7/12)

Hasso Plattner Institut

Controlled Buffer Overflow (3/8):

In most cases affected memory is a **stack** (rarely a heap)

In contrast to other storage domains like text domains (read-only) or data domains (no executable code allowed), here it is possible to

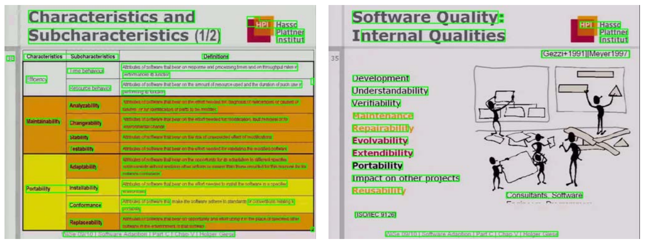
- Write during run time of the program and
- Place executable commands

Stack consists of

- Return address to jump back to a certain position after finishing a subprogram
- Domain for parameter which are consigned to the subprogram
- Space for local variables which are released after return

Design and Programming Errors | Internet Security | Prof. Dr. Christoph Meinel

Figure 4. Slide content distribution. Title and content act as indicators for actual slide transitions and are thus analyzed in the second step.



Characteristics and Subcharacteristics (1/2)

Hasso Plattner Institut

Characteristics	Subcharacteristics	Definitions
Efficiency	Time behaviour	Attributes of software that bear on response and processing times and on throughput rates in performances its function
	Resource behavior	Attributes of software that bear on the amount of resource used and the duration of such use in performing its function
Maintainability	Analyzability	[illegible]
	Changeability	[illegible]
	Stability	[illegible]
	Testability	[illegible]
Portability	Adaptability	[illegible]
	Installability	Attributes of software that bear on the effort needed to install the software in a specified environment
	Conformance	Attributes of software that make the software adhere to standards or conventions relating to portability
	Replaceability	[illegible]

WiSe 09/10 | Software Adaption | Part C | Chap V | Holger Giese

Software Quality: Internal Qualities

Hasso Plattner Institut

[Gezzi+1991][Meyer1997]

Development
Understandability
Verifiability
Maintenance
Repairability
Evolvability
Extendibility
Portability
Impact on other projects
Reusability

Consultants, Software

[ISO/IEC 9126]

WiSe 09/10 | Software Adaption | Part C | Chap V | Holger Giese

Figure 5. Refined bounding boxes from text detection. The results are automatically verified in order to remove false alarms.

of a differential image exceeds T_{s1}.

The results of the first segmentation step is too redundant for indexing, since they contain the progressive build-up of a complete final slide over sequence of partial versions. Hence, the segmentation process continues with the second step that aims to find the actual slide page transition based on the frames detected in the first step.

After a statistical analysis of large amounts of slide videos, we defined the content distribution of the most commonly used slide style (cf. Fig. 4). Let R_t and R_c denote the title and the content regions in Fig. 4 which account for $23\,\%$ and $70\,\%$ of the entire frame height respectively. If the amount of CCs in the differential edge image for R_t exceeds the threshold T_{s2}, a slide page transition is captured. Otherwise, in R_c, we detect the first text lines top-down and bottom-up respectively, and perform the corresponding CC differencing analysis on these text line bounding sequences from two adjacent frames. If both of the two difference values exceed T_{s2}, a slide page transition is also captured. In our study, T_{s1} and T_{s2} have been set to 20 and 5 respectively. All thresholds were learned on our training data.

IV. Text Detection

Text detection is the first task performed for video OCR. Our approach is not executed on every video frame due to reasons of efficiency. But on the key frames we obtained during the video segmentation process. For managing detected text lines, a *text line object* class is defined with the following properties: bounding box location (the top-left corner position) and size, start time, end time, and text content. After the coarse detection phase, a text region refinement procedure ensures the validity of our results.

A. DCT Text Detector

We have implemented a weighted DCT-based text detector, subsequently referred to as *DCT detector*. The advantage of the DCT detector is its computational efficiency compared to the classifier-based detector, because it does not require a training period. However, for visually different video sequences, a parameter adaption is necessary. In order to determine the most suitable parameter combination for all video types of our test set, we have developed an automated exhaustive test.

In image processing, the DCT exhibits excellent energy compaction for highly correlated images. "An uncorrelated image has its energy spread out, whereas the energy of a correlated image is packed into the low frequency region" [4]. Regarding text detection, text is usually uncorrelated with the image background (e.g. overlay text, slide text). Therefore, we are able to distinguish text blocks from the rest of the image by using DCT.

The DCT processing workflow for a single frame is depicted in Fig. 6: First, DCT is executed block by block on a video frame (cf. Fig. 6 (b)). The original DCT block size used in [4] is 8×8 pixels which has proven to be insufficient for our purpose. Therefore we decided to use 16×16 pixel blocks for DCT with various overlaps. Since texts usually occur in the mid-frequency region, we apply weight functions to reduce the noise in the uninteresting regions while enhancing the text regions.

We apply the linear and quadratic weight functions proposed by [4]. In addition, we define a weight function WE for 16×16 DCT:

$$\mathrm{WE}(x,y) = \begin{cases} \mathrm{C}(x,y)^2 & \text{for } \mathrm{T}_1 \leq \mathrm{x}+\mathrm{y} \leq \mathrm{T}_2, \\ & 1 \leq x,y \leq 16, \\ 0 & \text{otherwise} \end{cases}$$

where $\mathrm{WE}(x,y)$ is the weighted block energy, $\mathrm{C}(x,y)$ is the DCT coefficient, and T_1, T_2 are thresholds (cf. Fig. 6 (c)).

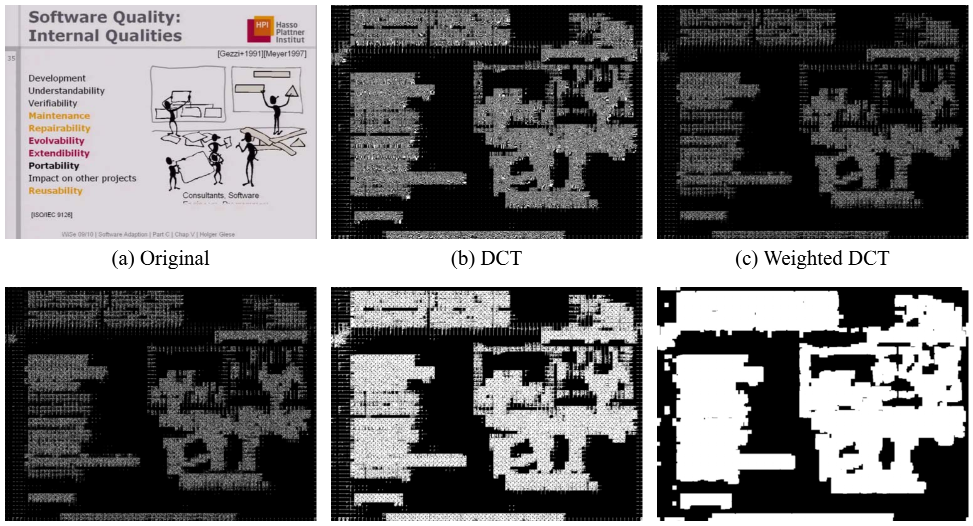
A subsequent normalization method NE converts the weighted DCT frames into greyscale images:

$$\mathrm{NE}(x,y) = 255 \times \frac{\mathrm{WE}(x,y) - \min(\mathrm{WE})}{\max(\mathrm{WE}) - \min(\mathrm{WE})},$$

where $\mathrm{NE}(x,y)$ is the normalized value (cf. Fig. 6 (d)).

Then, we apply the Otsu thresholding method [5] to binarize the normalized image (cf. Fig. 6 (e)). Ultimately, morphological erosion and delation are used to create a binary mask where text candidate regions are indicated by white pixels (cf. Fig. 6 (f)).

To determine the best combination of parameters, we have performed an automated exhaustive test for various parameter combinations that resulted in the parameter set given in Table I.



(a) Original (b) DCT (c) Weighted DCT

(d) Normalized (e) Binarized (f) Mask after erosion and dilation

Figure 6. Workflow of the DCT-based text detection. (b) First, DCT processes the image in blocks of size 16×16 (with overlaps). (c) Weight functions help to reduce noise. (d) The frame is converted into a normalized grayscale image. (e) Otsu thresholding is applied to binarize the image. (f) Finally, a binary mask indicating candidate regions with white color is created through morphological erosion and delation.

B. Text Region Refinement

To extract proper text lines from the candidate text regions being delivered by the DCT detector, we apply a refinement procedure. [2] propose a vertical and a horizontal image decomposition procedure using Sobel edge maps that we had to extend: [2] were able to apply a statical edge threshold, because a single video genre (*CNN* news) had been considered only—we have applied our algorithm able to consider different video genres by using a dynamic thresholding algorithm.

First, we calculate the image entropy value. If it exceeds a value of 4.81, the edge threshold T_e is set to 140 (i. e., pixels exceeding this value are identified as edge points). In our test runs, the entropy values were in the range of 2.0 and 5.7. We found out that if the image entropy exceeds a value of 4.81, a statical T_e value of 140 is sufficient for the refinement. If the entropy value is lower than 4.81, the edge threshold can be computed as follows:

$$\begin{aligned} E_p &= 0.307 - 0.048 \times \mathit{entropy} \\ E_t &\overset{T_e}{\to} E_{afterthresholding} = E_p \times E_t, \end{aligned}$$

with E_p being the edge percentage after thresholding and E_t the sum of horizontal and vertical edge points. All fixed parameter values were taken from our test runs.

After the text region decomposition, we perform a text box verification procedure to remove false alarms. We have implemented the size constraint, horizontal-vertical-aspect-ratio constraint and fill-factor constraint described in [2]. Fig. 5 shows the results after verification, which is the final output of the text detection.

V. Text Recognition

Texts in video frames have often poor resolution and heterogeneous background. To enable an efficient way of processing the detected text line objects, we apply an image quality enhancement process. To resolve the problem of having low-resolution images, we enlarge text line images with fonts smaller than 40 pixels text height using bicubic interpolation. Then, we remove the background by a subsequent adaptive image contrast and brightness adjustment.

Dynamic Contrast/Brightness Adaption

By applying the dynamic contrast and brightness adjustment, the grayscale text line images are converted to an image showing black font on almost plain white background. The workflow of the algorithm is depicted in Fig. 7, where m and sd denote the median and standard deviation of the image histogram, with T_s as the threshold value for sd.

First, the image contrast is adapted until m equals 255 or 0. Subsequently, sd is adapted until it reaches its threshold T_s. If sd exceeds T_s, the image's brightness is gradually increased—otherwise, its contrast is increased until sd reaches T_s. When the image contrast reaches its maximum while sd is still smaller than T_s, we increase the image brightness until sd reaches its maximum. For our purpose, we chose a fixed value of 100 for the standard deviation threshold T_s. Fig. 8 shows several result images from our trials.

VI. Result Discussion

We integrated the results gained from segmentation and OCR processing into our web portal. The tele-TASK system

Table I
RESULTS OF EXHAUSTIVE TESTS FOR PARAMETER COMBINATIONS OF DCT PROCESSING METHODS

DCT overlap	Weight function	Binarization	Erosion/dilation times	T_1	T_2
3	weight function for 16×16 blockwise DCT	Otsu thresholding	3	9	21

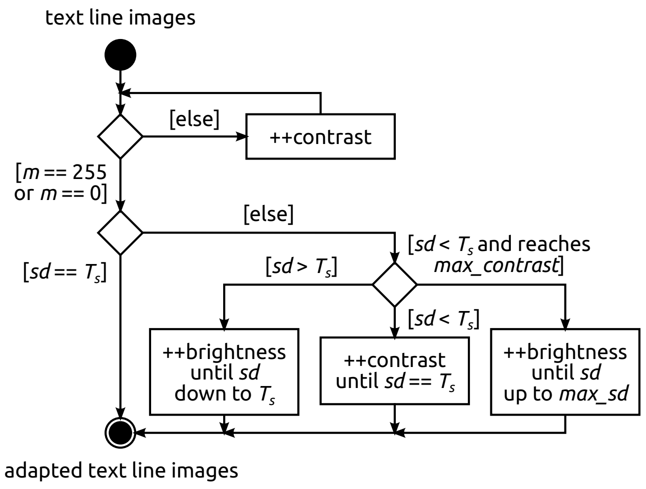
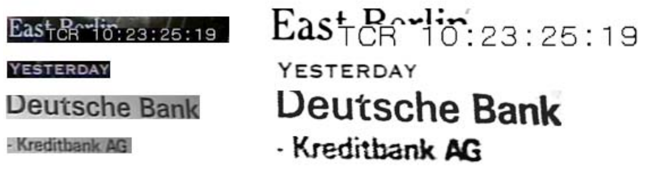

text line images
[else]
++contrast
[m == 255 or m == 0]
[else]
[sd < Ts and reaches max_contrast]
[sd == Ts]
[sd > Ts]
[sd < Ts]
++brightness until sd down to Ts
++contrast until sd == Ts
++brightness until sd up to max_sd
adapted text line images

Figure 7. Workflow of the dynamic adaptive contrast and brightness adjustment. After applying the algorithm, the text appears black on a plain white background. m and sd denote the median and standard deviation of the image histogram, T_s is the threshold value for sd.



(a) Text line images (b) Adapted text line images

Figure 8. Example resulting images of dynamic contrast/brightness adaption. Contrary to the originals, the revised images are suitable for being processed by OCR engines, as the background has been removed.

makes all recorded videos available to the public via its web interface which is implemented using *Django*, a python web framework[1]. Based on a plug-in architecture formerly written for our project [7], we are able to embed new sections into existing pages or even expand features like the search function easily. In the following, we describe enhancements of the web portal using the extracted data.

For every lecture, tele-TASK offers a page showing all information concerning it. Furthermore, the users are presented with the tags and ratings corresponding to the lecture—signed-in users have the possibility to add new tags or create playlists of lectures. Aside from that, there are pages providing the videos in different formats like *Flash* and *Real*. These two kinds of pages are thus favorable for adding the extracted information.

[1]https://www.djangoproject.com/

We extended the overview page first. On this page, there is no video to combine the data with; instead, the first suggestion was to display all slides belonging to a lecture, giving users a brief impression of the presentation. One issue about this option is the varying number of slides between different lectures—most consist of 10 to 20 slides whereas some contain more than hundred. Displaying all slides on the page would result in a big loading effort and huge space consumption thus. Therefore, we implemented an image gallery for the website allowing users to browse through a large number of slides. Using the same slide gallery, we also refined the video page, yet extending it slightly: By scripting additional functionality to the video player, we cause the video to jump to the corresponding position if a user clicks on a slide. As a result, this provides an easy method to navigate inside the video, even to parts which are not loaded yet.

In addition to the direct presentation, it is possible to embed the obtained data into the website's search function by using it as metadata the search engine interprets. This approach expands the search space with new keywords and probably helps to find a larger amount of more accurate results. The existing plug-in interfaces used in our web portal [8] allow an easy integration of new data for the search by adding the OCR-generated data in the same way as data from user-created tags.

The same information could also be used to detect similarities between different lectures. Therefore, we have to filter it to find suitable keywords for comparing the lectures. This method requires a higher amount of data preparation than just adding the data to the search function, because filtering is an important step to lower the number of false positive results when comparing two objects. Thus, the use of stop word lists and content-related dictionaries can help achieving better results. This part is still in development in order to further improve lecture interconnections and the search function.

VII. EVALUATION AND EXPERIMENTAL RESULTS

We restricted on testing the efficiency of the video segmentation, text detection and text recognition algorithms on test videos and single video frames. Our test data set consists of lecture videos and video frames from the *tele-TASK* project [10]. The test data is available at [11]. We randomly chose 20 lecture videos from different lecturers with varying layouts and font styles for the estimation of our segmentation algorithm; in order to evaluate the text detection and recognition procedures, we used 180 video

Table II
TEXT DETECTION RESULTS FOR THE DCT DETECTOR

	Recall	Precision	F_1 Measure
Pixel-based	0.95	0.85	0.90
Bounding-box-based	0.88	0.82	0.85

test frames including respective manual text annotations. Overall, our test data set contains 2997 text lines, 12533 words, and 76099 characters. The video frame sizes vary between 640×480 pixels and 1024×768 pixels.

A. Evaluation of Video Segmentation

Our slide video segmenter achieves segmentation recall and precision: 98% and 95%, respectively. The detailed evaluation data for each video is available at [11]. Since our algorithm is defined for slide videos (like presentations), videos embedded within slides are not considered in our evaluation.

B. Evaluation of Text Detection

In order to evaluate our text detection methods, we selected 180 key frames from the segmentation results. Then, we applied the following, commonly used evaluation methodology: We distinguish between a pixel-based evaluation, where the percentage of overlapping pixels in the ground truth and the experimental results are used to determine recall and precision, and an evaluation based on text bounding boxes. For the latter one, a bounding box is deemed to be a correct match if the overlap with the ground truth exceeds 80 %. Table II shows the text detection results of our video test data set.

C. Evaluation of Text Recognition

We applied the open-source OCR engine *Tesseract*[2] for the text recognition. The recognition evaluation is based on the results of our preceding text detection, using test frames containing English and German texts. Overall, we achieve recognizing 85 % of all characters and 74 % of all words correctly. To some extent this is due to some text passages are set in special graphical and artistic fonts and are therefore difficult to read by standard OCR libraries. We also evaluated three popular print OCR engines Gocr 0.48 [3], Tesseract and Ocropus [4] by using our test frames.The results show that the achieved word recognition rate of our system is two times better than the best one of the three.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have presented an effective, robust system for lecture video segmentation, text detection and recognition. A novel slide video segmenter and a DCT-based text detector have been developed and evaluated. To remove heterogeneous image backgrounds we have developed a novel adaptive contrast and brightness adjustment. The results of our video segmentation, text detection and recognition algorithms are suitable for content-based video indexing and retrieval:

As the further work, the analysis of a lecture slides' structuring will take place by using text objects. Implementing context- and dictionary-based post-processing will improve the text recognition rate further.

[2]http://code.google.com/p/tesseract-ocr/
[3]http://jocr.sourceforge.net/
[4]http://code.google.com/p/ocropus/

REFERENCES

[1] F. Chang, C-J. Chen, and C-J. Lu. "A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique," in *Computer Vision and Image Understanding*, vol. 93, no. 2, January 2004, pp. 206–220.

[2] X. S. Hua, X. R. Chen, L. W. Yin, H. J. Zhang. "Automatic Location of Text in Video Frames," in *Proc. of ACM Multimedia 2001 Workshops: Multimedia Information Retrieval*, 2001, pp. 24–27.

[3] M. Liška, V. Rusňák, E. Hladká. "Automated Hypermedia Authoring for Individualized Learning," in *8th International Conference on Information Technology Based Higher Education and Training*. Kumamoto, Japonsko: Kumamoto University, Kumamoto, Japan, 2007, p. 4.

[4] S. Lu, K. E. Barner. "Weighted DCT coefficient based text detection," in *Proc. of Acoustics, Speech and Signal Processing, 2008. ICASSP*, 2008, pp. 1341–1344.

[5] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. SCM-9, no. 1, 1979, pp. 62–66.

[6] V. Schillings, C. Meinel. "tele-TASK—Teleteaching Anywhere Solution Kit," in *Proc. of ACM SIGUCCS 2002*, Providence (Rhode Island), USA, 2002, pp. 130–133.

[7] M. Siebert, F. Moritz, C. Meinel. "Establishing an Expandable Architecture for a tele-Teaching Platform," in *Proceeding 2010 Ninth IEEE/ACIS International Conference on Computer and Information Science Article*, Yamagata, Japan, 2010.

[8] M. Siebert, C. Meinel. "Realization of an Expandable Search Function for an E-Learning Web Portal," in *Proceeding Workshop on e-Activity at the Ninth IEEE/ACIS International Conference on Computer and Information Science Article*, Yamagata, Japan, 2010.

[9] H-J. Yang, C. Oehlke, and C. Meinel. "A Solution for German Speech Recognition for Analysis and Processing of Lecture Videos," in *10th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2011)*, Sanya, Heinan Island, China, May 2011, pp. 201–206.

[10] "tele-TASK". Internet: http://www.tele-task.de [08.09.2011].

[11] Internet: http://www.yanghaojin.com/research/videoOCR.html [08.09.2011].