AUTOMATED EXTRACTION OF LECTURE OUTLINES FROM LECTURE VIDEOS
A Hybrid Solution for Lecture Video Indexing

Haojin Yang, Franka Gruenewald and Christoph Meinel
Hasso Plattner Institute (HPI), University of Potsdam
P.O. Box 900460, D-14440 Potsdam, Germany

Keywords: Recorded Lecture Videos, Tele-teaching, Video Indexing, Multimedia Retrieval.

Abstract: Multimedia-based tele-teaching and lecture video portals have become more and more popular in the last few years. The amount of multimedia data available on the WWW (*World Wide Web*) is rapidly growing. Thus, finding lecture video data on the web or within a lecture video portal has become a significant and challenging task. In this paper, we present an approach for lecture video indexing based on automated video segmentation and extracted lecture outlines. First, we developed a novel video segmenter intended to extract the unique slide frames from the lecture video. Then we adopted video OCR (*Optical Character Recognition*) technology to recognize texts in video. Finally, we developed a novel method for extracting of lecture outlines from OCR-transcripts. Both video segments and extracted lecture outlines are further utilized for the video indexing. The accuracy of the proposed approach is proven by evaluation.

1 INTRODUCTION

In the last decade, more and more universities and research-institutions recorded their lectures and published them on the WWW, which is intended to make them accessible for students around the world (S. Trahasch, 2009). Hence, the development of a process for automated indexing of multimedia lecture content is highly desirable and would be especially useful for e-learning and tele-teaching.

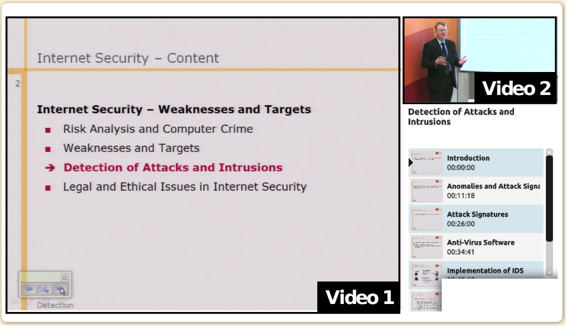
Since most lecturers in colleges use slide-presentations instead of blackboard writing today, the state-of-the-art lecture recording systems usually record two video streams parallelly: the main scene of lecturers which is recorded by using a video camera, and the second which captures the computer screen during the lecture through a frame grabber. The second stream may include a presentation of slides as well as demonstrations, videos and other material. The frame grabber output stream can be synchronized with the camera stream automatically during the recording process.

Fig. 1 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 with a corresponding slide. Thus, the indexing task can be performed by indexing the slide video only.

Content-based video gathering is usually achieved by using textual metadata which is provided by users (e.g, manual video tagging (F. Moritz, 2011)) or has to be extracted by automated analysis. For this purpose, we apply video OCR technology to obtain texts from the lecture videos. In video OCR, video frames containing visible textual information have to be identified first. Then, the text location within the video frames has to be detected, and the text pixels have to be separated from their background. Finally, the common OCR algorithms are applied to recognize the characters.

For lecture video indexing, most of the existing OCR-based approaches only make use of the pure textual information (F. Wang, 2008; J. Waitelonis, 2010), although the structure of lecture slides could provide relevant information for the indexing task. In this paper, we present a hybrid approach for the lecture video indexing. First, an improved slide video segmentation method is developed. The segmented unique video frames can be used immediately for the video browsing. Second, we applied video OCR technology to recognize the texts from the segmented frames. The subsequent lecture outline extraction procedure can extract the timed lecture outlines by using geometrical information of detected text lines from the video

Yang H., Gruenewald F. and Meinel C..
AUTOMATED EXTRACTION OF LECTURE OUTLINES FROM LECTURE VIDEOS - A Hybrid Solution for Lecture Video Indexing.
DOI: 10.5220/0003903700130022
In *Proceedings of the 4th International Conference on Computer Supported Education* (CSEDU-2012), pages 13-22
ISBN: 978-989-8565-06-8



Internet Security – Content

2

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

Detection
Video 1
Video 2
Detection of Attacks and Intrusions
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

Figure 1: An example lecture video from (Yang et al., 2011). Video 2 shows the speaker giving his lecture, whereas his presentation is played in video 1.

OCR stage.

The accuracy of proposed methods have been evaluated by using test data.

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

2 RELATED WORK

Hunter et al. proposed a SMIL (*Synchronized Multimedia Integration Language*) based framework to build a digital archive for multimedia presentations (J. Hunter, 2001). Their recording system consists of two video cameras, and they record both the lecturer scene and the slide scene during the presentation. However, they do not apply text recognition process on the slide video. Each speaker has to prepare a slide file in *PDF* format and is requested to upload it to a multimedia server after the presentation. Then the uploaded slide files have to be synchronized with the recorded videos. Finally, they perform the video indexing based on the text-resources of the corresponding slide files. In contrast to their approach, since our system directly analyzes the videos, we do not need to take care of the slide format. Synchronization between the slide file and the video is also not required. In a further contrast to our approach, the synchronization method employed by Hunter et. al is simply based on the pixel-based differencing metric of the binary image. It might not work robustly when animated content appears in the presentation slide. Although the OCR based approach may introduce a few errors, the recognized texts are still robust enough for the indexing task by performing a dictionary based filtering.

Some other approaches e.g. Moritz et al. (F. Moritz, 2011) and Waitelonis et al. (J. Waitelonis, 2010) make use of manual video tagging to generate the indexing data. The user tagging-based approaches work well for the recently recorded lecture videos. However, for videos which were created more than 10 years ago, there is no manually generated metadata available. And it is hard to attract the user's interest to annotate the outdated videos. Furthermore, regarding the quantitative aspect, the automated analysis is more suitable than the user annotation for large amounts of video data.

Wang et al. proposed a method for lecture video browsing by video text analysis (F. Wang, 2008). However, their approach does not consider the structure information of slides. Making use of the structured text lines (e.g, title, subtitle etc.) for building a more efficient search procedure is therefore impossible.

Yang et al. (H-J. Yang, 2011) and Leeuwis et al. (E.Leeuwis et al., 2003) use the ASR (*Automatic Speech Recognition*) 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 in contrast with OCR results. This will lead to a degrading of the indexing performance.

In our previous work (Yang et al., 2011), we developed a novel approach for lecture video indexing by using video OCR technology. We develop a novel slide segmenter that is intended to capture the slide transitions. A two-stage (text localization and text verification) approach is developed for the text detection within the video frames; and a multi-hypotheses framework is adopted for the text recognition. However, the proposed video segmenter in (Yang et al., 2011) is only defined for slide videos. It is therefore not suitable for videos, which are embedded in a slide with a different genre. Thus, we improve the segmentation method by a using machine learning classifier. The slide frames and the embedded video frames could be classified correctly by using an image intensity histogram feature and SVM (*Support Vector Machine*) classifier. For the embedded videos, we apply a common video segmenter. The proposed lecture outline extraction algorithm in this work is also based on the video OCR results from (Yang et al., 2011).

3 VIDEO SEGMENTATION AND VIDEO OCR

Fig. 2 shows the entire system workflow. The video segmentation method is first applied on the slide

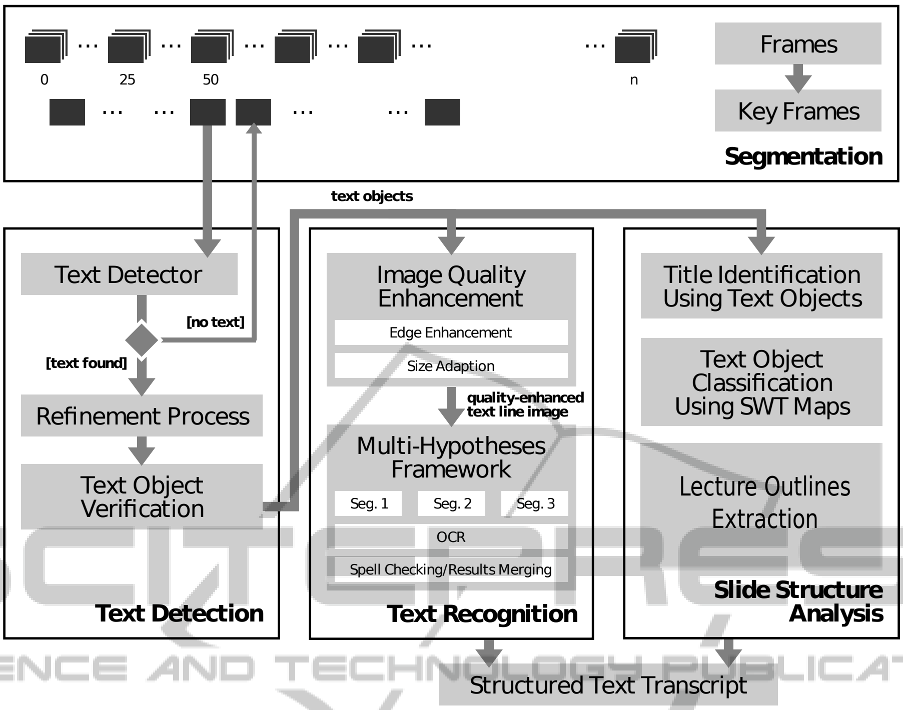

0
25
50
n
Frames
Key Frames
Segmentation
text objects
Text Detector
[no text]
[text found]
Refinement Process
Text Object Verification
Text Detection
Image Quality Enhancement
Edge Enhancement
Size Adaption
quality-enhanced text line image
Multi-Hypotheses Framework
Seg. 1
Seg. 2
Seg. 3
OCR
Spell Checking/Results Merging
Text Recognition
Title Identification Using Text Objects
Text Object Classification Using SWT Maps
Lecture Outlines Extraction
Slide Structure Analysis
Structured Text Transcript

Figure 2: The entire system workflow. After having segmented frame sequences, text detection is performed on every single key frame in order to find text occurrences. The resulting text objects are then used for video OCR and slide structure analysis.

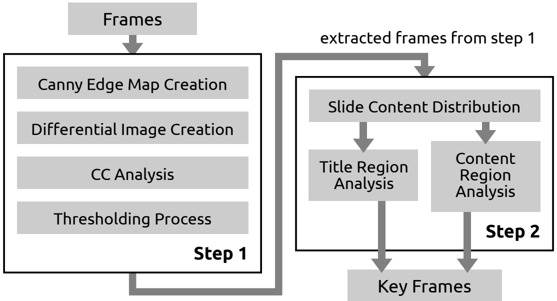


Figure 3: Segmentation workflow (Yang et al., 2011). (Step 1) Adjacent frames are compared with each other by applying *Connected Components Analysis* on their differential edge maps. (Step 2) Title and content region analysis ensure that only actual slide transitions are captured.

videos. For reasons of efficiency, we do not perform the analysis on every video frame. Instead, we established a time interval of one second. Subsequently, we adopt the text detection process on the unique frames we obtained during the video segmentation process. The occurrence duration of each detected text line is determined by reading the time information of the corresponding segment. The text recognition is applied on detected text line objects from the text detection stage. We also apply the slide structure analysis and the lecture outline extraction procedures on the detected text line objects.

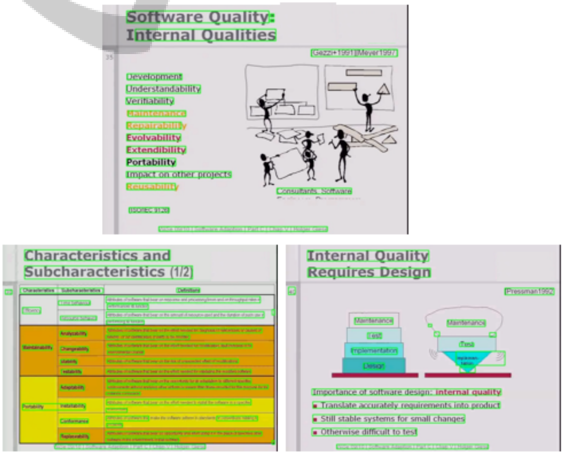

Software Quality: Internal Qualities
Development
Understandability
Verifiability
Maintenance
Repairability
Evolvability
Extendibility
Portability
Impact on other projects
Reusability
Characteristics and Subcharacteristics (1/2)
Internal Quality Requires Design
Importance of software design: internal quality
Translate accurately requirements into product
Still stable systems for small changes
Otherwise difficult to test

Figure 4: Text detection results. All detected text regions are identified by bounding boxes.

3.1 Video Segmentation

The video indexing process can be performed by decomposing the video into a set of representative unique frames. We call this process *video segmentation*. In order to build a suitable segmenter for slide videos, we have implemented a two-step analysis approach (cf. Fig 3): In the first step, we try to capture all changes from the adjacent frames by using CCs (*Connected Component*) based differencing-

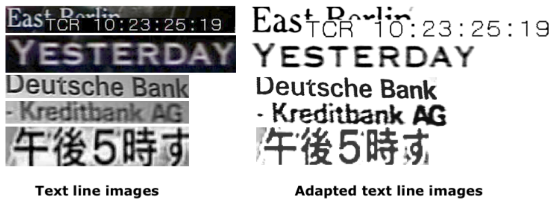


Figure 5: Text binarization results of our dynamic contrast-brightness adaption method.

metric. In this way, the high frequency noises in the video frames can be removed from the comparison by enlarging the size of valid CCs. The segmentation results from the first step are too redundant for indexing, since they may contain the progressive build-up of a complete final slide over sequence of partial versions (cf. Fig. 6). Therefore, the segmentation process continues with the second step: we first perform a statistical analysis on large amount of slide videos, intending to find the commonly used slide content distribution in lecture videos. Then, in the title region of the slide we perform the same differencing-metric as in the first step to capture the slide transition. Any change in the title region may cause the slide transition. While in the content region, we detect regions of the first and the last text lines. Checking the same regions in two adjacent frames. When the same text lines can not be found in both of the adjacent frames, a slide transition is then captured. More details about the algorithm can be found in (Yang et al., 2011).

Since the proposed method in (Yang et al., 2011) is defined for slide frames, it might be not suitable, when videos with varying genres were embedded in the slides and/or are played during the presentation. To fix this issue, we have developed a SVM classifier to distinguish the slide frames and the other video frames. In order to train an efficient classifier we evaluate two features: HOG (*Histogram of Oriented Gradient*) feature and image intensity histogram feature. The detailed evaluation for both features is discussed in Section 6.

Histogram of oriented gradient feature has been widely used in object detection (N. Dala, 2005) and optical character recognition fields, due to its efficiency for description of the local appearance and the shape variation of image objects. To calculate the HOG feature, the gradient vector of each image pixel within a predefined local region is calculated by using Sobel operator (Sobel, 1990). All gradient vectors are then decomposed into n directions. A histogram is subsequently created by using accumulated gradient vectors, and each histogram-bin is set to correspond to a gradient direction. In order to avoid sensitivity of the HOG feature to illumination, the feature values are often normalized.

Observing slide frames, the major slide content, like texts and tables, consists of many horizontal and vertical edges. The distribution of gradients would be distinguishable from other video genres.

We have also adopted an image intensity histogram feature to train the SVM classifier. Since the complexity and the intensity distribution of slide frames are strongly different from other video genres, we decided to use this simple and efficient feature to make a comparison to the HOG feature.

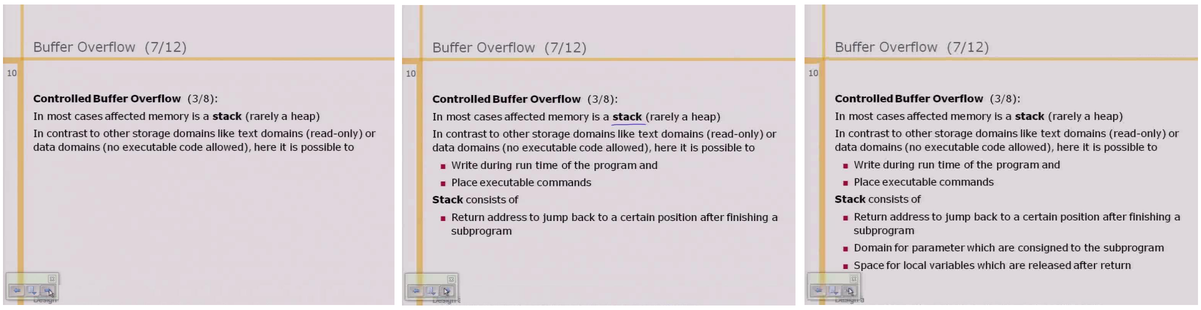
3.2 Video OCR

The texts in video provide a valuable source for indexing. Regarding lecture videos, the texts displayed on slides are strongly related to the lecture content. Thus, we have developed an approach for video text recognition by using video OCR technology. The commonly used video OCR framework consists of three steps.

Text detection is the first step of video OCR: this process determines, whether a single frame of a video file contains text lines, for which a tight bounding box is returned (cf. Fig. 4). Text extraction: in this step, the text pixels are separated from their background. This work is normally done by using a binarization algorithm. Fig. 5 shows the text binarization results of our dynamic contrast-brightness adaption method (Yang et al., 2011). The adapted text line images are converted to an acceptable format for a standard print-ocr engine. Text recognition: we applied a multi-hypotheses framework to recognize texts from extracted text line images. The subsequent spell-checking process will further filter out incorrect words from the recognition results.

The video OCR process is applied on segmented key frames from the segmentation stage. The occurrence duration of each detected text line is determined by reading the time information from the corresponding segment. Our text detection method consists of two stages: we develop an edge-based fast text detector for coarsely detection, and a SWT (*Stroke Width Transform*) (B. Epshtein, 2010) based text verification procedure is adopted to remove the false alarms from the detection stage. The output of our text detection method is an XML-encoded list serving as input for the text recognition and the outline extraction.

For the text recognition, we develop a multi-hypotheses framework. Three segmentation methods are applied for the text extraction: Otsu thresholding algorithm (Otsu, 1979), Gaussian based adaptive thresholding (R. C. Gonzalez, 2002) and our dynamic image contrast-brightness adaption method. Then, we apply the OCR engine on each segmentation result.



Buffer Overflow (7/12)

10

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

(a)

Buffer Overflow (7/12)

10

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

(b)

Buffer Overflow (7/12)

10

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

(c)

Figure 6: Segmentation result from the first step. Adjacent frames are compared with each other by applying *Connected Components Analysis* on their differential edge maps. The content build-up within the same frame can therefore not be identified.

Thus, three text results are generated for each text line image. Subsequently, the best OCR result is obtained by performing spell-checking and result-merging processes.

After the text recognition process, the detected text line objects and their texts are further utilized in the outline extraction process—which is described in the next section.

4 LECTURE OUTLINE EXTRACTION

In this section we will describe our text line extraction method, in detail. Regarding lecture slides, we can realize that the content of title, subtitle and key point have more significance than the content of the body, because they summarize each slide. Unlike other approaches, we observe characteristics of detected text line objects, and use geometrical information to classify the recognized OCR text resource. In this way, a more flexible search algorithm can be implemented based on the classified OCR texts. Meanwhile, by using these classified text lines, we have developed a method for automatic extraction of lecture outlines. On the one hand, the extracted outline can provide an overview about the lecture for the students. On the other hand, each text line has a timestamp, therefore, they can also be used to explore the video.

4.1 Slide Structure Analysis

The process begins with a slide structure analysis procedure. We identify the title text line by applying the following conditions:

- the text line is in the upper third part of the frame,
- the text line has more than three characters,
- the horizontal start position of the text line is not larger than the half of the frame width,
- it is one of three highest text lines and has the uppermost vertical position.

A title is detected when it satisfies all of the above conditions. Then, we label this text line object as a title line, and repeat the processes on the remaining text line objects in order to detect the next potential title line. The further detected title lines must have a similar height and stroke width as the first one. We have set the tolerance value of the height difference to 10 pixels, and the tolerance value of the stroke width difference to 5 pixels, respectively. For our purpose, we allow up to three title lines for each slide frame.

All of the none-title text line objects are further classified into three classes: *content text*, *subtitle-key point* and *footline*. The classification algorithm is based on detected text line height and the average stroke width value of the text line object. The algorithm can be described as follows:

$$\begin{array}{ll} \textit{subtitle/key point} & \text{if } \mathrm{s_t} > \mathrm{s_{mean}} \wedge \mathrm{h_t} > \mathrm{h_{mean}} \\ \textit{footline} & \text{if } \mathrm{s_t} < \mathrm{s_{mean}} \wedge \mathrm{h_t} < \mathrm{h_{mean}} \wedge \mathrm{y} = \mathrm{y_{max}} \\ \textit{normal content} & \text{otherwise} \end{array}$$

where s_{mean} and h_{mean} denote the average stroke width value and the average height of text line objects within a slide frame. And y_{max} denotes the maximal vertical position of a text line object.

4.2 Lecture Outline Extraction

The workflow of the lecture outline extraction algorithm is described in Fig. 7. In this step, we only consider text line objects obtained from the previous processing stage—which have a text line type of *title* or *subtitle-key point*.

First, we perform a spell checking process to filter out text line objects that do not satisfy the following

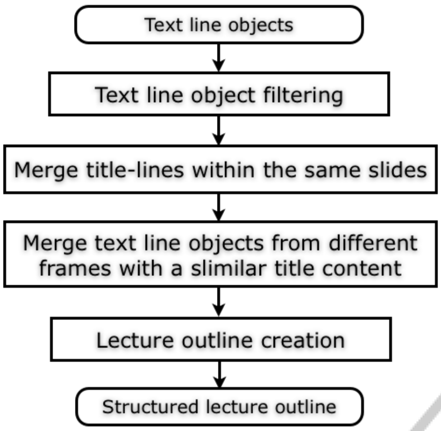

Text line objects
Text line object filtering
Merge title-lines within the same slides
Merge text line objects from different frames with a slimilar title content
Lecture outline creation
Structured lecture outline

Figure 7: Workflow of outline extraction process.

conditions. A valid text line object will be considered as an outline object.

- a valid text line object must have more than three characters,
- a valid text line object must contain at least one noun,
- the textual character count of a valid text line object must be more than 50% of the entire string length.

Then, we merge all title lines within the same slide according to their positions. All the other text line objects from this slide will be considered as the subitems of the title line. Subsequently, we merge all text line objects from different frames which have a similar title content. The similarity is determined by calculating the amount of the same characters and the same words. During the merging process, all duplicated *subtitle-key points* will be removed, since we want to avoid the redundancy that might affect the further indexing process. The final lecture outline is created by assigning all valid text line objects into a tree structure according to their occurrences.

The visualization methodology and the utility of our analysis results will be discussed in the next section.

5 VISUALISATION AND UTILITY OF THE RESULTS

The use case for our video slide segmenter and lecture outline generator was implemented and evaluated in the context of a video lecture portal used at a university institute working with computer sciences. This portal was developed in Python with the web framework Django. The video player was developed in OpenLaszlo. About 400 students and quite a number of external people are using the portal on a regular basis.

Before going into detail about the implementation and interface design of the visualization of the segmentation and the outline, the utility of those two features for the users of the portal shall be elaborated.

5.1 Utility of Lecture Video Outline and Extracted Slides

One of the most important functionalities of a teleteaching portal is the search for data. Since recording technology has become more and more inexpensive and easier to use, the amout of content produced for these portals has become huge. Therefore it is nearly impossible for students to find the required content without a search function.

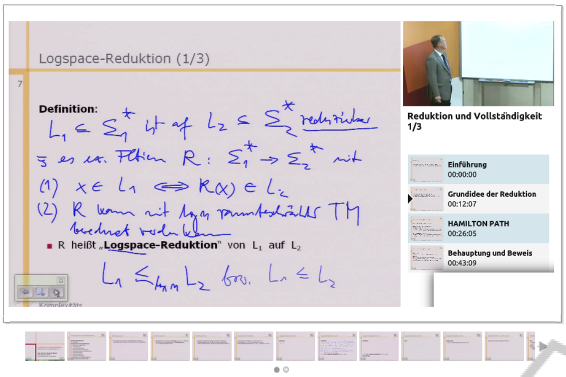
But even when the user has found the right content, he still needs to find the piece of information he requires within the 90 minutes of a lecture. This is where the lecture video outline and the extracted slides come in to play. Without an outline the user will not have any orientation about which time frame within the video he is interested in. The outline will thus help to guide the user with navigation within the video. This can be both utilized by manual navigation or with the help of a local search function that only searches the current video.

The video slides extracted from the segementer have a similar function. They help visually oriented users to fulfill this task in an easier and quicker way. Especially students repeating a lecture will benefit from the slide visualisation as they can jump to slides they can remember as important from the first time they visited the lecture.

Also the slides can help for learning and repetition tasks when used in connection with user-generated annotations in the form of a manuscript, a function we will further describe in the following paragraphs. This is crucial, because it was proven that more than 80 percent of all learning tasks take place with the help of the eye and visual memory (Addie, 2002). This means that vision is an essential aspect of learning, which can be supported by using the lecture slides when repeating and studying for an exam.

5.2 Visualising the Video Outline in Interaction with the Lecture Video Player

There are two options to visualise the video outline within the tele-teaching portal. So far chapters that



Logspace-Reduktion (1/3)

7

Definition:

$L_1 \subseteq \Sigma_1^*$ ist auf $L_2 \subseteq \Sigma_2^*$ reduzierbar

$\exists$ es ex. Fktion $R: \Sigma_1^* \to \Sigma_2^*$ mit

(1) $x \in L_1 \Longleftrightarrow R(x) \in L_2$

(2) R kann mit log. raumbeschränkter TM berechnet werden

- R heißt „**Logspace-Reduktion**" von L_1 auf L_2

$L_1 \leq_{log\,m} L_2$ bzw. $L_1 \leq L_2$

Komplexitäts

Reduktion und Vollständigkeit 1/3

Einführung 00:00:00

Grundidee der Reduktion 00:12:07

HAMILTON PATH 00:26:05

Behauptung und Beweis 00:43:09

Figure 8: Visualisation of the slides extracted from the segmentation process underneath the video player.

have been added manually from technical personnel were visualised within the OpenLaszlo video player. This was done to keep all related functionality in one spot and thereby ensure a short and easy access from the player to the navigation.

The extracted video outline is more detailed and provides a more finegrained navigation option, which is, however, more space consuming. Therefore, a separate django plugin was developed that shows all headlines and subheadlines with their corresponding timestamp (cf. Fig. 9). Timestamps serve as links to jump to the associated position in the lecture video.

5.3 Using Extracted Video Slides for Navigation within Lecture Videos

As explained in section 5.1, the slides extracted from the segmenter serve as visual guideline for the user to navigate through a certain lecture video.

In order to fulfil that function, the slides are visualised in the form of a time bar underneath the video (cf. Fig. 8).

When sliding a mouse over the small preview pictures, they are scaled up to a larger preview of the slide. When clicking one of the preview slides in the time bar, the video player will jump to the correct timestamp in the lecture video that corresponds with the timestamp stored in the database for that slide. Those functions are realized with Javascript triggering the action within the OpenLaszlo player.

5.4 Visualising Extracted Slides in Connection with User-generated Lecture Video Annotations

An important aid for students in their learning process are manuscripts. Traditionally, those were handwritten notes the student wrote during the lecture. With the rise of tele-teaching, digital manuscript functionalities have become available, where notes can be written digitally in conjunction with a timestamp that references at what time in the video the note was written (Zupancic, 2006).

For some people, however, there is still the need to have printed notes available. To make both available with a minimum effort for the students, a PDF printout of the manuscript can be provided. Single notes without reference to the lecture itself can be confusing for the students though. Therefore the PDF should be enriched with the lecture slides etracted from the segmenter. With the help of the timestamp of both the slide and the notes, the slides can be printed with simultaneously written notes.

6 EVALUATION AND EXPERIMENTAL RESULTS

We have evaluated our slide video segmentation algorithm (without a consideration of embedded videos), text detection and text recognition methods by using 20 selected lecture videos with varying layouts and font styles (cf. Fig. 10) in (Yang et al., 2011). In this paper, we extended our evaluation in the following manner: a SVM classifier has been trained and evaluated; the outline extraction algorithm has also been evaluated by using our test videos. The test data and some evaluation details are available at (Yang et al., 2011).

For analyzing a 90 minutes lecture video, the entire frame extraction, video segmentation, video OCR and lecture outline extraction processes need about 10–15 minutes.

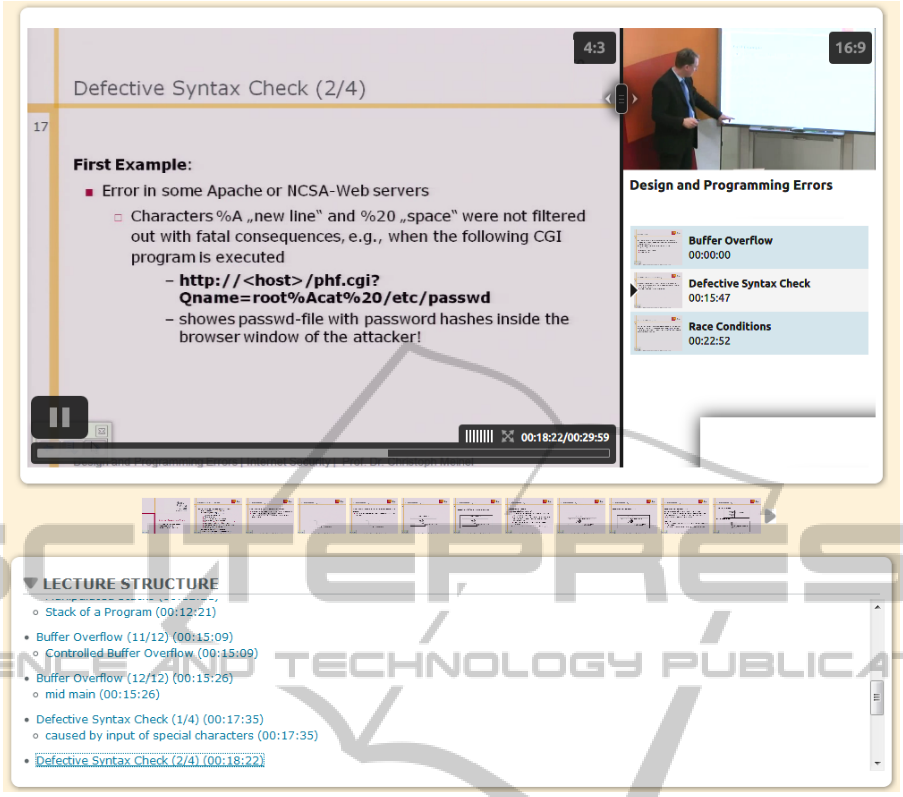
The experiments were performed on a Mac OS X, 2.4 GHz platform. Our system is implemented in C++.

6.1 Evaluation of SVM Classifiers

We have used 2597 slide frames and 5224 none-slide frames to train the SVM classifier. All slide frames are selected from large amount of lecture videos. In order to build a none-slide frame training set which should have varying image genres, we collect about 3000 images from (flickr, 2011), and other as well as over 2000 none-slide video frames from our lecture video database.

Our test set consists of 240 slide frames and 233 none-slide frames that completely differ from the training set.

The evaluation results for HOG and image intensity histogram features are shown in Table 1. We



Defective Syntax Check (2/4)

17

First Example:

- Error in some Apache or NCSA-Web servers
  - Characters %A „new line" and %20 „space" were not filtered out with fatal consequences, e.g., when the following CGI program is executed
    - **http://<host>/phf.cgi? Qname=root%Acat%20/etc/passwd**
    - showes passwd-file with password hashes inside the browser window of the attacker!

Design and Programming Errors

Buffer Overflow 00:00:00

Defective Syntax Check 00:15:47

Race Conditions 00:22:52

LECTURE STRUCTURE

- Stack of a Program (00:12:21)
- Buffer Overflow (11/12) (00:15:09)
  - Controlled Buffer Overflow (00:15:09)
- Buffer Overflow (12/12) (00:15:26)
  - mid main (00:15:26)
- Defective Syntax Check (1/4) (00:17:35)
  - caused by input of special characters (00:17:35)
- Defective Syntax Check (2/4) (00:18:22)

Figure 9: Visualization of the extracted outline of the lecture video underneath the video player.

Table 1: Slide Frame Classification Results.

	Recall	**Precision**	F_1 **Measure**
HOG feature	0.996	0.648	0.785
Image intensity histogram feature	0.98	0.85	**0.91**

use HOG feature with 8 gradient directions, which have been extracted from each 64x64 local image block. The image intensity histogram features consist of 256 histogram bins that correspond to 256 image grayscale values. The classification accuracy for slide frames of both features have achieved close to 100%. However, the HOG feature has a much worse performance than intensity histogram feature for distinguishing images—that have varying genres—in contrast to slide frames. Although the recall of intensity histogram feature is slightly lower than HOG feature, it could achieve a more significant improvement in precision and processing speed (about 10 times faster).

6.2 Evaluation of Lecture Outline Extraction

We evaluated our lecture outline extraction method by randomly selecting 180 segmented unique slide frames of 20 test videos. The achieved word recognition rate of our text recognition algorithm is about 85%, which has been reported in (Yang et al., 2011). To evaluate the outline extraction method, we determine not only whether the outline type is correctly distinguished by slide structure analysis. We also calculate the word accuracy for each detected outline object. Therefore, the accuracy of our outline extraction algorithm is based on the OCR recognition rate.

We have defined recall and precision metrics for the evaluation as follows:

$$Recall = \frac{number\ of\ correctly\ retrieved\ outline\ words}{number\ of\ all\ outline\ words\ in\ ground\ truth}$$

$$Precision = \frac{number\ of\ correctly\ retrieved\ outline\ words}{number\ of\ retrieved\ outline\ words}$$

Table 2: Evaluation Results of Lecture Outline Extraction.

	Recall	**Precision**	F_1 **Measure**
Title	0.86	0.95	**0.90**
Subtitle-key point	0.61	0.77	0.68

Table 2 shows the evaluation results for the *title* and *subtitle-key point* extraction, respectively. Although the extraction rate of *subtitle-key point* still has

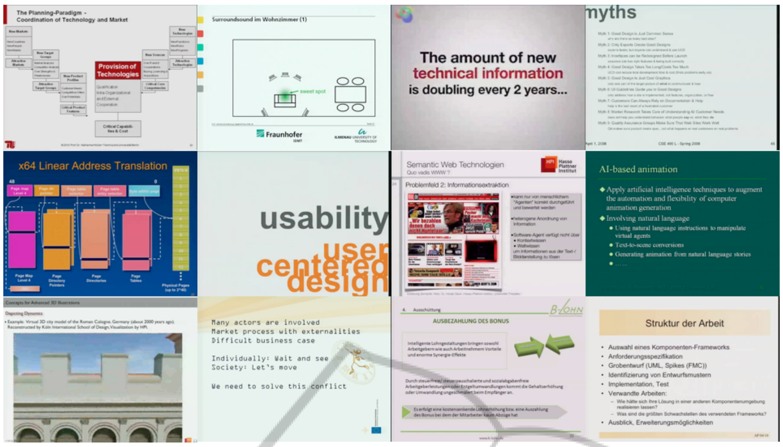

The Planning-Paradigm – Coordination of Technology and Market
Provision of Technologies
Surroundsound im Wohnzimmer (1)
Fraunhofer
The amount of new technical information is doubling every 2 years...
myths
x64 Linear Address Translation
usability
user centered design
Semantic Web Technologien
AI-based animation
Many actors are involved
Market process with externalities
Difficult business case
Individually: Wait and see
Society: Let's move
We need to solve this conflict
AUSBEZAHLUNG DES BONUS
Struktur der Arbeit

Figure 10: Example frames of our test videos which have varying layouts and font styles for the estimation of our segmentation algorithm (Yang et al., 2011).

a improvement space, the extracted titles are already robust for the indexing task. Furthermore, since the outline extraction rate depends on the OCR accuracy, it can be further improved by achieving a better text recognition rate.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a hybrid solution for lecture video indexing: An improved slide video segmentation algorithm has been developed by using CCs analysis and SVM classifier. We have further implemented a novel method for the extraction of timed lecture outlines using geometrical information and stroke width values. The indexing can be performed by using both, unique slide frames (visual information) and extracted lecture outlines (textual information).

As an upcoming improvement, implementing context- and dictionary-based post-processing will improve the text recognition rate further.

The extracted lecture outline as well as the OCR transcripts we obtained can be used for the development of intelligent search algorithms and new recommendation methods in the future.

Therefore we plan to further extend the pluggable search function (Siebert and Meinel, 2010) within our sample tele-teaching portal. An extended plugin will be built that enables the search among the lecture outline globally. A marking of the searched term or the right video sequence, where the search term can be found within the lecture outline, is a further step.

The features involving visualisation of the results have been built as proof of concept for the segmentation and OCR processes so far. Since it is desired that those features become frequently used functionalities within the tele-teaching portal, their usability and utility need to be evaluated within a user study.

REFERENCES

Addie, C. (2002). *Learning Disabilities: There is a Cure–A Guide for Parents, Educators and Physicians*. Achieve Pubns.

B. Epshtein, E. Ofek, Y. W. (2010). Detecting text in natural scene with stroke width transform. In *Proc. of Computer Vision and Pattern Recognition*, pages 2963–2970.

E.Leeuwis, M.Federico, and Cettolo, M. (2003). Language modeling and transcription of the ted corpus lectures. In *Proc. of the IEEE ICASSP*.

F. Moritz, M. Siebert, C. M. (2011). Community tagging in tele-teaching environments. In *Proc. of 2nd International Conference on e-Education, e-Business, e-Management and E-Learning*.

F. Wang, C-W. Ngo, T.-C. P. (2008). Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis. *Journal of Pattern Recognition*, 41(10):3257–3269.

flickr (2011). http://www.flickr.com.

H-J. Yang, C. Oehlke, C. M. (2011). A solution for german speech recognition for analysis and processing of lecture videos. In *Proc. of 10th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2011)*, pages 201–206, Sanya, Heinan Island, China. IEEE/ACIS.

J. Hunter, S. L. (2001). Building and indexing a distributed multimedia presentation archive using smil. In *Proc. of ECDL '01 Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 415–428, London, UK.

J. Waitelonis, H. S. (2010). Exploratory video search with yovisto. In *Proc. of 4th IEEE International Confer-*

ence on Semantic Computing (ICSC 2010), Pittsburg, USA.

N. Dala, B. T. (2005). Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 886–893.

Otsu (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, SCM-9(1):62–66.

R. C. Gonzalez, R. E. W. (2002). *Digital Image Processing*. Englewood Cliffs.

S. Trahasch, S. Linckels, W. H. (2009). Vorlesungsaufzeichnungen – Anwendungen, Erfahrungen und Forschungsperspektiven. Beobachtungen vom GI-Workshop eLectures 2009“. *i-com*, 8(3 Social Semantic Web):62–64.

Siebert, M. and Meinel, C. (2010). Realization of an expandable search function for an e-learningweb portal. In *Workshop on e-Activity at the Ninth IEEE/ACIS International Conference on Computer and Information Science Article*, page 6, Yamagata/Japan.

Sobel, I. (1990). An isotropic 3 3 image gradient operator. *Machine Version for Three-Dimensional Scenes*, (376–379).

Yang, H., Siebert, M., Lühne, P., Sack, H., and Meinel, C. (2011). Lecture video indexing and analysis using video ocr technology. In *Proc. of 7th International Conference on Signal Image Technology and Internet Based Systems (SITIS 2011)*, Dijon, France.

Zupancic, B. (2006). *Vorlesungsaufzeichnungen und digitale Annotationen Einsatz und Nutzen in der Lehre*. Dissertation, Albert-Ludwigs-Universität Freiburg.