Content Based Lecture Video Retrieval Using Speech and Video Text Information

Haojin Yang and Christoph Meinel, *Member, IEEE*

Abstract— In the last decade e-lecturing has become more and more popular. The amount of lecture video data on the *World Wide Web* (WWW) is growing rapidly. Therefore, a more efficient method for video retrieval in WWW or within large lecture video archives is urgently needed. This paper presents an approach for automated video indexing and video search in large lecture video archives. First of all, we apply automatic video segmentation and key-frame detection to offer a visual guideline for the video content navigation. Subsequently, we extract textual metadata by applying video *Optical Character Recognition* (OCR) technology on key-frames and *Automatic Speech Recognition* (ASR) on lecture audio tracks. The OCR and ASR transcript as well as detected slide text line types are adopted for keyword extraction, by which both video- and segment-level keywords are extracted for content-based video browsing and search. The performance and the effectiveness of proposed indexing functionalities is proven by evaluation.

Index Terms—Lecture videos, automatic video indexing, content-based video search, lecture video archives

1 Introduction

Digital video has become a popular storage and exchange medium due to the rapid development in recording technology, improved video compression techniques and high-speed networks in the last few years. Therefore audiovisual recordings are used more and more frequently in e-lecturing systems. A number of universities and research institutions are taking the opportunity to record their lectures and publish them online for students to access independent of time and location. As a result, there has been a huge increase in the amount of multimedia data on the Web. Therefore, for a user it is nearly impossible to find desired videos without a search function within a video archive. Even when the user has found related video data, it is still difficult most of the time for him to judge whether a video is useful by only glancing at the title and other global metadata which are often brief and high level. Moreover, the requested information may be covered in only a few minutes, the user might thus want to find the piece of information he requires without viewing the complete video. The problem becomes how to retrieve the appropriate information in a large lecture video archive more efficiently. Most of the video retrieval and video search systems such as YouTube, Bing and Vimeo reply based on available textual metadata such as title, genre, person, and brief description etc. Generally, this kind of metadata has to be created by a human to ensure a high quality, but the creation step is rather time and cost consuming. Furthermore, the manually provided metadata is typically brief, high level and subjective. Therefore, beyond the current approaches, the next generation of video retrieval systems apply automatically generated metadata by using video analysis technologies. Much more content-based metadata can thus be generated which will lead to two research questions in the e-lecturing context:

- Can those metadata assist the learner in searching required lecture content more efficiently?
- If so, how can we extract the important metadata from lecture videos and provide hints to the user?

According to the questions, we postulated the following hypothesis:

Hypothesis 1 *The relevant metadata can be automatically gathered from lecture videos by using appropriate analysis techniques. They can help a user to find and to understand lecture contents more efficiently, and the learning effectiveness can thus be improved.*

Traditional video retrieval based on visual feature extraction cannot be simply applied to lecture recordings because of the homogeneous scene composition of lecture videos. Figure 1(a) shows an exemplary lecture video recorded using an outdated format produced by a single video camera. Varying factors may lower the quality of this format. For example, motion changes of the camera may affect the size, shape and the brightness of the slide; the slide can be partially obstructed when the speaker moves in front of the slide; any changes of camera focus (switching between the speaker view and the slide view) may also affect the further slide detection process.

Nowadays people tend to produce lecture videos by using multi-scenes format (cf. Figure 1(b)), by which the speaker and his presentation are displayed synchronously. This can be achieved either by displaying a single video of the speaker and a synchronized slide file, or by applying a state of the art lecture recording system

• *The authors are at Hasso-Plattner-Institute for Software Systems Engineering GmbH (HPI), PO Box 900460, D-14440 Potsdam, Germany E-mail: {haojin.yang, meinel}@hpi.uni-potsdam.de*

(a) (b)

Fig. 1. (a) An example of outdated lecture video format.
(b) An exemplary lecture video. Video 1 shows the professor giving his lecture, whereas his presentation is played in video 2.

such as tele-TASK (*tele-Teaching Anywhere Solution Kit*)[1].

Figure 1(b) illustrates an example of such a system which delivers two main parts of the lecture: the main scene of lecturers which is recorded by using a video camera and the second which captures the desktop of the speaker's computer (his presentations) during the lecture through a frame grabber tool. The key benefits of the latter one for a lecturer is the flexibility. For the indexing, no extra synchronization between video and slide files is required, and we do not need to take care of the slide format. The main drawback is that the video analysis methods may introduce errors. Our research work mainly focus on those lecture videos produced by using the screen grabbing method. Since two videos are synchronized automatically during the recording process. Therefore, the temporal scope of a complete unique slide can be considered as a lecture segment. This way, segmenting two-scenes lecture videos can be achieved by only processing slide video streams, which contain most of the visual text metadata. The extracted slide frames can provide a visual guideline for video content navigation.

Text is a high-level semantic feature which has often been used for content-based information retrieval. In lecture videos, texts from lecture slides serve as an outline for the lecture and are very important for understanding. Therefore after segmenting a video file into a set of key frames (all the unique slides with complete contents), the text detection procedure will be executed on each key frame, and the extracted text objects will be further used in text recognition and slide structure analysis processes. Especially, the extracted structural metadata can enable more flexible video browsing and video search functions.

Speech is one of the most important carriers of information in video lectures. Therefore, it is of distinct advantage that this information can be applied for automatic lecture video indexing. Unfortunately, most of the existing lecture speech recognition systems in the reviewed work cannot achieve a sufficient recognition result, the *Word Error Rates* (WERs) having been reported from [1], [2], [3], [4], [5] and [6] are approximately 40%–85%. The poor recognition results not only limit the usability of speech transcript, but also affect the efficiency of the further indexing process. In our research, we intended to continuously improve the ASR result for German lectures by building new speech training data based on the open-source ASR tool. However, in the open-source context, it lacks method for generating the German phonetic dictionary automatically, which is the one of the most important part of an ASR software. Therefore, we developed an automated procedure in order to fill this gap.

A large amount of textual metadata will be created by using OCR und ASR method, which opens up the content of lecture videos. To enable a reasonable access for the user, the representative keywords are further extracted from the OCR and ASR results. For content-based video search, the search indices are created from different information resources, including manual annotations, OCR and ASR keywords, global metadata etc. Here the varying recognition accuracy of different analysis engines might result in solidity and consistency problems, which have not been considered in the most related work (cf. Section 2.2). Therefore, we propose a new method for ranking keywords extracted from various information resources by using the extended *Term Frequency Inverse Document Frequency* (TFIDF) score [7]. The ranked keywords from both segment- and video-level can directly be used for video content browsing and video search. Furthermore, the video similarity can be calculated by using the *Cosine Similarity Measure* [8] based on extracted keywords.

In summary, the major contributions of this paper are the following:

- We extract metadata from visual as well as audio resources of lecture videos automatically by apply-

1. tele-TASK system was initially designed in 2002 at the university of Trier. Today, weekly 2000 people (unique visits) around the world visit the tele-TASK lecture video portal (www.tele-task.de) with more than 4800 lectures and 14000 podcasts free of charge via internet.

ing appropriate analysis techniques. For evaluation purposes we developed several automatic indexing functionalities in a large lecture video portal, which can guide both visually- and text-oriented users to navigate within lecture video. We conducted a user study intended to verify the research hypothesis and to investigate the usability and the effectiveness of proposed video indexing features.
- For visual analysis, we propose a new method for slide video segmentation and apply video OCR to gather text metadata. Furthermore, lecture outline is extracted from OCR transcripts by using stroke width and geometric information. A more flexible search function has been developed based on the structured video text.
- We propose a solution for automatic German phonetic dictionary generation, which fills the gap in open-source ASR domain. The dictionary software and compiled speech corpus are provided for the further research use.
- In order to overcome the solidity and consistency problems of a content-based video search system, we propose a keyword ranking method for multimodal information resources. In order to evaluate the usability, we implemented this approach in a large lecture video portal.
- The developed video analysis methods have been evaluated by using compiled test datasets as well as opened benchmarks. All compiled test sets are publicly available from our website for the further research use.

The rest of the paper is organized as follows: section 2 reviews related work in lecture video retrieval and content-based video search domain. Section 3 describes our automatic video indexing methods, while the user study is presented in section 5. A content-based lecture video search engine using multimodal information resources is introduced in section 4. Finally, section 6 concludes the paper with an outlook on future work.

2 Related Work

Information retrieval in the multimedia-based learning domain is an active and multidisciplinary research area. Video texts, spoken language, community tagging, manual annotations, video actions, or gestures of speakers can act as the source to open up the content of lectures.

2.1 Lecture Video Retrieval

Wang et al. proposed an approach for lecture video indexing based on automated video segmentation and OCR analysis [9]. The proposed segmentation algorithm in their work is based on the differential ratio of text and background regions. Using thresholds they attempt to capture the slide transition. The final segmentation results are determined by synchronizing detected slide key-frames and related text books, where the text similarity between them was calculated as indicator. Grcar et al. introduced *VideoLectures.net* in [10] which is a digital archive for multimedia presentations. Similar to [9], the authors also apply a synchronization process between the recorded lecture video and the slide file, which has to be provided by presenters. Our system contrasts to these two approaches since it directly analyzes the video, which is thus independent of any hardware or presentation technology. The constrained slide format and the synchronization with an external document are not required. Furthermore, since the animated content evolvement is often applied in the slide, but has not been considered in [9] and [10], their system might not work robustly when those effects occur in the lecture video. In [9], the final segmentation result is strongly dependent on the quality of the OCR result. It might be less efficient and imply redundancies, when only poor OCR result is obtained.

Tuna et al. presented their approach for lecture video indexing and search [11]. They segment lecture videos into key frames by using global frame differencing metrics. Then standard OCR software is applied for gathering textual metadata from slide streams, in which they utilize some image transformation techniques to improve the OCR result. They developed a new video player, in which the indexing, search and captioning processes are integrated. Similar to [9], the used global differencing metrics cannot give a sufficient segmentation result when animations or content build-ups are used in the slides. In that case, many redundant segments will be created. Moreover, the used image transformations might be still not efficient enough for recognizing frames with complex content and background distributions. Making use of text detection and segmentation procedures could achieve much better results rather than applying image transformations.

Jeong et al. proposed a lecture video segmentation method using *Scale Invariant Feature Transform* (SIFT) feature and the adaptive threshold in [12]. In their work SIFT feature is applied to measure slides with similar content. An adaptive threshold selection algorithm is used to detect slide transitions. In their evaluation, this approach achieved promising results for processing one-scene lecture videos as illustrated in Figure 1(a).

Recently, collaborative tagging has become a popular functionality in lecture video portals. Sack et al. [13] and Moritz et al. [14] apply tagging data for lecture video retrieval and video search. Beyond the keyword-based tagging, Yu et al. proposed an approach to annotate lecture video resources by using Linked Data. Their framework enables users to semantically annotate videos using vocabularies defined in the Linked Data cloud. Then those semantically linked educational resources are further adopted in the video browsing and video recommendation procedures. However, the effort and cost needed by the user annotation-based approach cannot satisfy the requirements for processing large amounts of web video data with a rapid increasing speed. Here, the automatic analysis is no doubt much more suitable.

Nevertheless, using Linked Data to further automatically annotate the extracted textual metadata opens a future research direction.

ASR provides speech-to-text information on spoken languages, which is thus well suited for content-based lecture video retrieval. The studies described in [5] and [15] are based on out-of-the-box commercial speech recognition software. Concerning such commercial software, to achieve satisfying results for a special working domain an adaption process is often required, but the custom extension is rarely possible. [1] and [6] focus on English speech recognition for *Technology Entertainment and Design* (TED) lecture videos and webcasts. In their system, the training dictionary is created manually, which is thus hard to be extended or optimized periodically. Glass et al. proposed a solution for improving ASR results of English lectures by collecting new speech data from the rough lecture audio data [3]. Inspired by their work, we developed an approach for creating speech data from German lecture videos. Haubold et al. focus on multi-speaker presentation videos. In their work speaker changes can be detected by applying a speech analysis method. A topic phrases extraction algorithm from highly imperfect ASR results (WER$\approx 75\%$) has been proposed by using lecture course-related sources such as text books and slide files [4].

Overall, most of those lecture speech recognition systems have low recognition rate, the WERs of audio lectures are approximately 40%-85%. The poor recognition results limit the further indexing efficiency. Therefore, how to continuously improve ASR accuracy for lecture videos is still an unsolved problem.

The speaker-gestures-based information retrieval for lecture videos has been studied in [16]. The author equipped the lecture speaker with special gloves that enable the automatic detection and evaluation of gestures. The experimental results show that 12% of the lecture topic boundaries were correctly detected using speaker-gestures. However, those gesture features are highly dependent on the characteristics of speakers and topics. It might have limited use in large lecture video archives with massive amounts of speakers.

2.2 Content-Based Video Search

Several content-based video search engines have been proposed recently. Adcock et al. proposed a lecture webcast search system [17], in which they applied a slide frame segmenter to extract lecture slide images. The system retrieved more than 37.000 lecture videos from different resources such as YouTube, Berkeley Webcast etc. The search indices are created based on the global metadata obtained from the video hosting website and texts extracted from slide videos by using a standard OCR engine. Since they do not apply text detection and text segmentation process, the OCR recognition accuracy of their approach is therefore lower than our system's. Furthermore, by applying the text detection process we are able to extract the structured text line such as title, subtitle, key-point etc. that enables a more flexible search function. In the CONTENTUS [18] project, a content-based semantic multimedia retrieval system has been developed. After the digitization of media data, several analysis techniques e.g., OCR, ASR, video segmentation, automated speaker recognition etc. have been applied for metadata generation. An entity recognition algorithm and an open knowledge base are used to extract entities from the textual metadata. As mentioned before, searching through the recognition results with a degree of confidence, we have to deal with the solidity and the consistency problem. However the reviewed content-based video search systems did not consider this issue.

3 Automated Lecture Video Indexing

In this chapter we will present four analysis processes for retrieving relevant metadata from the two main parts of lecture video, namely the visual screen and audio tracks. From the visual screen we firstly detect the slide transitions and extract each unique slide frame with its temporal scope considered as the video segment. Then the video OCR analysis is performed for retrieving textual metadata from slide frames. Based on OCR results, we propose a novel solution for lecture outline extraction by using stroke width and geometric information of detected text lines. In speech-to-text analysis we applied the open-source ASR software CMU Sphinx[2]. To build the acoustic and language model, we collected speech training data from open-source corpora and our lecture videos. As already mentioned, it lacks method in open-source context for creating German phonetic dictionary automatically. We thus developed a solution to fill this gap and made it available for the further research use.

3.1 Slide Video Segmentation

Video browsing can be achieved by segmenting video into representative key frames. The selected key frames can provide a visual guideline for navigation in the lecture video portal. Moreover, video segmentation and key-frame selection is also often adopted as a preprocessing for other analysis tasks such as video OCR, visual concept detection etc.

Choosing a sufficient segmentation method is based on the definition of "video segment" and usually depends on the genre of the video. In the lecture video domain, the video sequence of an individual lecture topic or subtopic is often considered as a video segment. This can be roughly determined by analyzing the temporal scope of lecture slides. Many approaches (as e.g., [17], [11]) make use of global *pixel-level-differencing* metrics for capturing slide transitions. A drawback of this kind of approach is that the salt and pepper noise of video signal can affect the segmentation accuracy. After observing the content of lecture slides, we realize

2. http://cmusphinx.sourceforge.net/

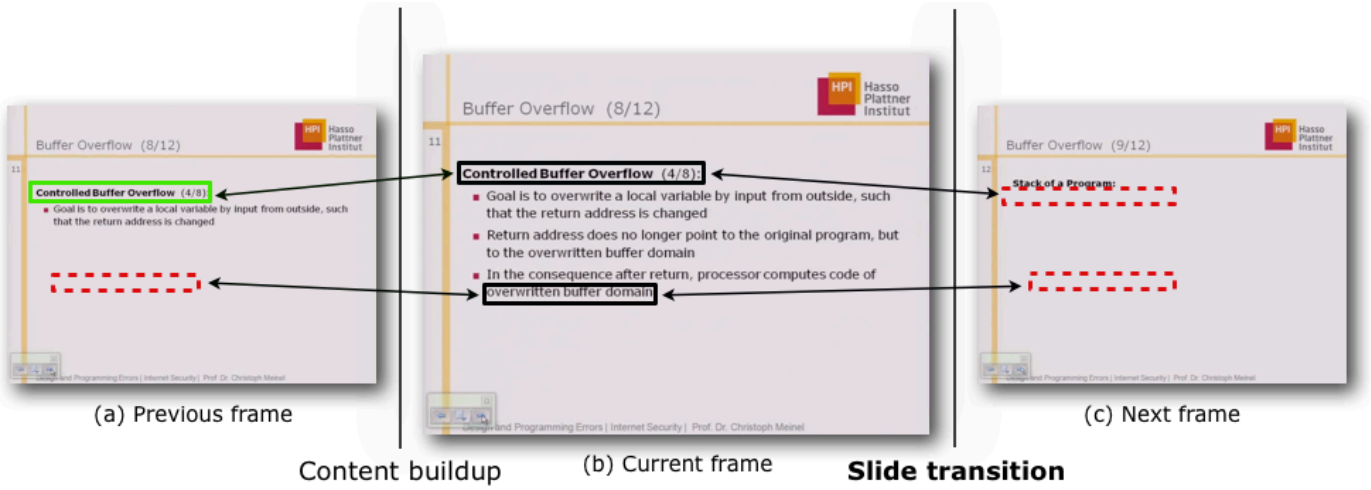

Fig. 2. We detect the first and the last object line in R_c vertically and perform the CC differencing metric on those line sequences from adjacent frames. In frame (a) and (b), a same text line (top) can be found; whereas in frame (b) and (c), the CC-based differences of both text lines exceed the threshold T_{s2}. A slide transition is thus found between frame (b) and (c).

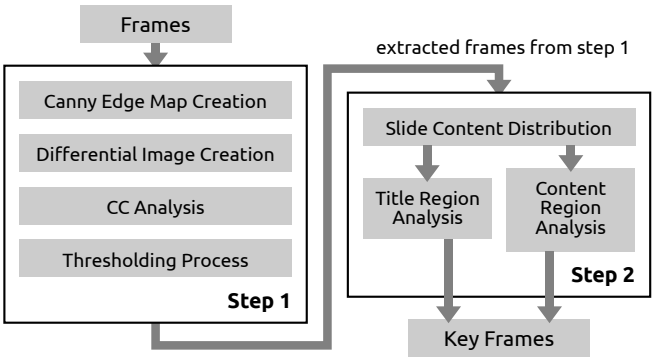

Fig. 3. Lecture video segmentation workflow. Step 1: adjacent frames are compared with each other by applying the CC analysis on their differential edge maps. Step 2: slide transitions are captured by performing title and content region analysis.

that the major content as e.g., text lines, figures, tables etc. can be considered as *Connected Components* (CCs). We therefore propose to use CC instead of pixel as the basis element for the differencing analysis. We call it *component-level-differencing* metric. This way we are able to control the valid size of the CC, so that the salt and pepper noise can be rejected from the differencing process. For creating CCs from binary images, we apply the algorithm according to [19] which demonstrated an excellent performance advantage. Another benefit of our segmentation method is its robustness to animated content progressive build-ups used within lecture slides. Only the most complete unique slides are captured as video segments. Those effects affect the most lecture video segmentation methods mentioned in chapter 2.

Our segmentation method consists of two steps (cf. Figure 3):

- In the first step, the entire slide video is analyzed. We try to capture every knowledge change between adjacent frames, for which we established an analysis interval of three seconds by taking both accuracy and efficiency into account. This means that segments with a duration smaller than three seconds may be discarded in our system. Since there are very few topic segments shorter than three seconds, this setting is therefore not critical. Then we create canny edge maps for adjacent frames and build the pixel differential image from the edge maps. The CC analysis is subsequently performed on this differential image and the number of CCs is then used as a threshold for the segmentation. A new segment is captured if the number exceeds T_{s1}. Here we establish a relatively small T_{s1} to ensure that each newly emerging knowledge point will be captured. Obviously, the first segmentation step is sensitive to animations and progressive build-ups. The result is thus too redundant for video indexing. Hence, the process continues with the second segmentation step based on the frames in the first step.

- In the second segmentation step the real slide transitions will be captured. The title and content region of a slide frame is first defined. We established the content distribution of commonly used slide styles by analyzing a large amount of lecture videos in our database. Let R_t and R_c denote the title and content region which account for 23% and 70% of the entire frame height respectively. In R_t we apply CC-based differencing as described above with a small threshold value of 1 for capturing the slide transitions. Here any small changes within the title region may cause a slide transition. For instance, two slides often differ

from each other in a single chapter number. If there is no difference found in R_t, then we try to detect the first and the last bounding box object in R_c vertically and perform the CC-based differencing within the object regions of two adjacent frames (cf. Figure 2). In case that the difference value of both object regions between adjacent frames exceed the threshold T_{s2}, a slide transition is then captured. For detecting the content progressive build-up horizontally, the process could be repeated in a similar manner. In our experiment, T_{s1} = 20 and T_{s2} = 5 have proven to serve best for our training data. Exact setting of these parameters is not critical.

- Since the proposed method is designed for segmenting slide videos, it might be not suitable when videos with varying genres have been embedded in the slides and are played during the presentation. To solve this problem we have extended the original algorithm by using a *Support Vector Machine* (SVM) classifier and image intensity histogram features. We use the *Radial Basis Function* (RBF) as kernel. In order to make a comparison, we also applied the *Histogram of Oriented Gradients* (HOG) feature [20] in the experiment. We adapted the HOG feature with 8 gradient directions, whereas the local region size was set to 64×64. Moreover, in using this approach our video segmentation method is also suitable for processing such one-screen lecture videos with a frequent switch between slide- and speaker-scene.

3.1.1 Experimental Result

To evaluate our slide-video segmentation method, we have compiled a lecture video test dataset, which consist of 20 randomly selected videos from different speakers with various layouts and font styles. Each unique slide with complete contents is regarded as a correct match. The overall number of detected segments is 860, of which 816 are correct. The achieved segmentation recall and precision are $98\,\%$ and $95\,\%$ for the test dataset, respectively. Since this experiment is designed to evaluate the slide segmentation algorithm, videos with various genres embedded within the slides are not considered. The detailed evaluation result as well as the url for each test video can be found at [21].

To evaluate the slide-image classifier, 2597 slide frames and 5224 non-slide frames have been collected in total for the training. All slide frames are collected from our lecture video database. To create a non-slide frame set with varying image genres, we have collected additional 3000 images from *flickr* [3]. Moreover, about 2000 non-slide video frames have been collected from the lecture video database. The test set consists of 240 slide frames and 233 non-slide frames, which differ from the training set.

In the classifier training the SVM-parameters were determined by using the grid-search function. To calculate the image intensity histogram, 256 histogram bins were initially created corresponding to the 256 image grayscale values. Then the normalized histogram values were applied to train the SVM classifier. We have evaluated the normalization factor (cf. Figure 5), which has proven to serve best when set to 1000[4].

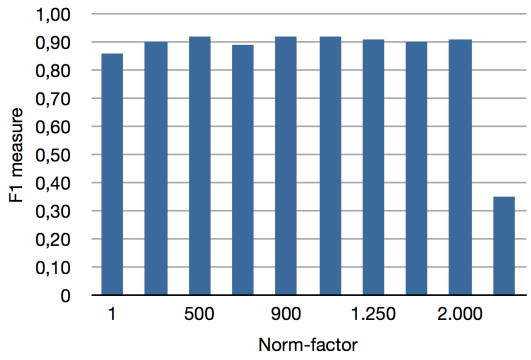

Fig. 5. Evaluation results of the normalization factor of the image intensity histogram feature.

TABLE 1
Feature comparison result

	Recall	Precision	F_1 Measure
HOG feature	0.996	0.648	0.785
Image intensity histogram feature	0.91	0.93	0.92

The comparison results of two features are illustrated in Table 1. Both features achieved a good recall rate for recognizing slide frames. However, compared with the HOG feature the intensity histogram feature showed a considerable improvement in precision and F_1 measure.

3.2 Video OCR for Lecture Videos

Texts in the lecture slides are closely related to the lecture content, can thus provide important information for the retrieval task. In our framework, we developed a novel video OCR system for gathering video text.

For text detection, we developed a new localization-verification scheme. In the detection stage, an edge-based multi-scale text detector is used to quickly localize candidate text regions with a low rejection rate. For the subsequent text area verification, an image entropy-based adaptive refinement algorithm not only serves to reject false positives that expose low edge density, but also further splits the most text- and non-text-regions into separate blocks. Then *Stroke Width Transform* (SWT) [22]-based verification procedures are applied to remove the non-text blocks. Since the SWT verifier is not able to correctly identify special non-text patterns such as sphere, window-blocks, garden fence, we adopted

3. www.flickr.com

4. The norm-function normalizes the histogram bins, by which the sum of the bins is scaled to equal the factor.

(a) (b)

Fig. 4. (a) Exemplary text detection results (b) Text binarization results.

an additional SVM classifier to sort out these non-text patterns in order to further improve the detection accuracy. For text segmentation and recognition, we developed a novel binarization approach, in which we utilize image skeleton and edge maps to identify the text pixels. The proposed method consists of three main steps: text gradient direction analysis, seed pixel selection, and seed-region growing. After the seed-region growing process, the video text images are converted into a suitable format for standard OCR engines. The subsequent spell-checking process will further sort out incorrect words from the recognition results. Our video OCR system has been evaluated by applying several test datasets. Especially by using the online evaluation of the opened benchmark ICDAR 2011 competition test sets for born-digital images [23], our text detection method achieved the second place, and our text binarization and word recognition method achieved the first place in the corresponding ranking list on their website (last check 08/2013). An in-depth discussion of the developed video OCR approach and the detailed evaluation results can be found in [24]. By applying the open-source print OCR engine *tesseract-ocr*[5], we achieved recognition of 92% of all characters and 85% of all words correctly for lecture video images. The compiled test dataset including 180 lecture video frames and respective manual annotations is available at [21]. Figure 4(a) and Figure 4(b) demonstrate some exemplary text detection and binarization results of our methods.

3.3 Slide Structure Analysis

Generally, in the lecture slide the content of title, subtitle and key point have more significance than the normal slide text, as they summarize each slide. Due to this fact, we classify the type of text lines recognized from slide frames by using geometrical information and stroke width feature. The benefits of the structure analysis method can be summarized as follows:

- The lecture outline can be extracted using classified text lines, it can provide a fast overview of a lecture video and each outline item with the timestamp can in turn be adopted for video browsing (e.g. Figure 7). The usability of the outline feature has been evaluated by a user study provided in chapter 5.
- The structure of text lines can reflect their different significance. This information is valuable for a search/indexing engine. Similar to web search engines which make use of the explicitly pre-defined HTML structure (HTML-tags) for calculating the weight of texts in web pages, our method further opens up the video content and enables the search engine to give more accurate and flexible search results based on the structured video text.

The process begins with a title line identification procedure. A text line will be considered as a candidate title line when it localizes in the upper third part of the frame, it has more than three characters, it is one of three highest text lines and has the uppermost vertical position. Then the corresponding text line objects will be labeled as the title objects and we repeat the process on the remaining text objects in the same manner. The further detected title lines must have a similar height (the tolerance is up to 10px) and stroke width value (the tolerance is up to 5px) as the first one. For our purposes, we allow up to three title lines to be detected for each slide frame. All non-title text line objects are further classified into three classes: *content text*, *key-point* and *footline*. The classification is based on the height and the average stroke width of the text line object, which is described as follows:

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

where s_{mean} and h_{mean} denote the average stroke width and the average text line height of a slide frame, and y_{max} denotes the maximum vertical position of a text line object (starts from the top-left corner of the image).

To further extract the lecture outline, we firstly apply a spell checker to sort out text line objects which do not satisfy the following conditions:

- a valid text line object must have more than three characters,
- a valid text line object must contain at least one noun,

5. https://code.google.com/p/tesseract-ocr

- the textual character count of a valid text line object must be more than 50% of the entire string length.

The remaining objects will be labeled as the outline object. Subsequently, we merge all title lines within the same slide according to their position. Other text line objects from this slide will be considered as the subitem of the title line. Then we merge text line objects from adjacent frames with the similar title content (with 90% content overlap). The similarity is measured by calculating the amount of same characters and same words. After a merging process all duplicated text lines will be removed. Finally, the lecture outline is created by assigning all valid outline objects into a consecutive structure according to their occurrences.

3.3.1 Experimental Result

We evaluated the lecture outline extraction method by selecting 180 segmented unique slide frames from the test videos used in chapter 3.1.1. In our experiment, both outline type classification accuracy and word recognition accuracy were considered. Therefore, the accuracy of outline extraction is based on the OCR recognition result. We have defined the recall and precision metric 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 shows the evaluation results for the *title* and *key-point* extraction, respectively. Although the extraction

TABLE 2
Evaluation results of lecture outline extraction

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

rate of *key-point* still has a certain improvement space, the extracted titles are already well suited for the automatic video indexing. Furthermore, the outline extraction accuracy can be further improved by providing better OCR results.

3.4 ASR for German Lecture Videos

In addition to video OCR, ASR can provide speech-to-text information from lecture videos, which offers the chance to improve the quantity of automatically generated metadata dramatically. However, as mentioned, most lecture speech recognition systems cannot achieve a sufficient recognition rate. A model-adaption process is often required. Furthermore, in the open-source context, it lacks method for generating the German phonetic dictionary automatically. Therefore, we developed a solution intended to fill this gap. We decided to build acoustic models for our special use case by applying the

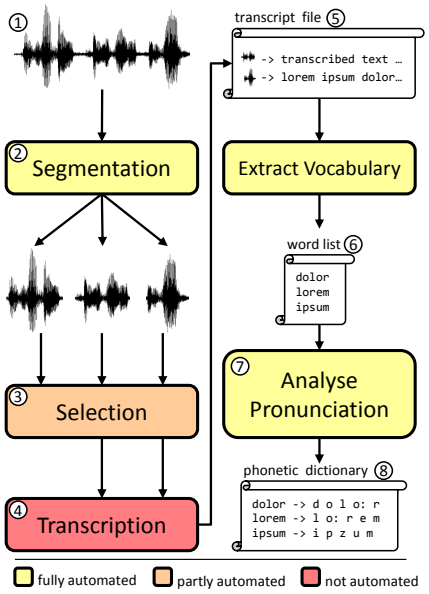

Fig. 6. Workflow for extending our speech corpus

CMU Sphinx Toolkit[6] and the German Speech Corpus by Voxforge[7] as a baseline. Unlike other approaches that collect speech data by applying dictation in a quiet environment, we gathered hours of speech data from real lecture videos and created corresponding transcripts. This way the real teaching environment can be involved in the training process. For the language model training we applied the collected text corpora from German daily news corpus (radio programs, 1996-2000), Wortschatz-Leipzig[8] and the audio transcripts of the collected speech corpora.

Figure 6 depicts the workflow for creating the speech training data. First of all, the recorded audio file is segmented into smaller pieces and improper segments are sorted out. For each remaining segment the spoken text is transcribed manually, and added to the transcript file automatically. As an intermediate step, a list of all used words in the transcript file is created. In order to obtain the phonetic dictionary, the pronunciation of each word has to be represented phonetically.

A recorded lecture audio stream yields approximately 90 minutes of speech data, which is far too long to be processed by the ASR trainer or the speech decoder at once. Shorter speech segments are thus required. Manually collecting appropriate speech segments and transcripts is rather time consuming and costly. There are a number of steps to be performed to acquire high quality input data for a ASR trainer tool. Our current approach is to fully automate segmentation (Figure 6 (2)) and partly automate selection (Figure 6 (3)) without suffering from quality drawbacks like dropped word endings. The fundamental steps can be described as follows: we first

6. cmusphinx.sourceforge.net/
7. www.voxforge.org/
8. corpora.informatik.uni-leipzig.de/download.html

compute the absolute values of input samples to get a loudness curve, which is then downsampled (e.g., by factor 100). Then a blur filter (e.g., radius=3) is applied to eliminate outliers and a loudness threshold determines which areas are considered *quiet*. *Non-quiet* areas with a specified maximum length (5 seconds has proven to serve best for our training data) will serve as a potential speech utterance. Finally, we retrieve speech utterances from the original audio stream and save them into files.

From the experiments we have learned that all speech segments used for the acoustic model training must meet certain quality requirements. Experience has shown that approximately 50% of the generated audio segments have to be sorted out due to one of the following reasons:

- the segment contains acoustical noise created by objects and humans in the environment around the speaker, e.g., doors closing, chairs moving, students talking,
- the lecturer mispronounces some words, so that they are completely invalid from an objective point of view,
- the speaker's language is clear, but the segmentation algorithm cuts off parts of a spoken word so that it becomes invalid.

Therefore, the classification of audio segments in good or bad quality is not yet solvable automatically, as the term "good quality" is very subjective and strongly depends on one's personal perception. Nevertheless, the proposed segmentation method can perform a preselection that speeds up the manual transcription process significantly. The experimental results show that the WER decreased by about 19%, when adding 7.2 hours of speech data from our lecture videos to the training set (cf. Table 3).

TABLE 3
Current progress of our acoustic model training. The WER results have been measured through a random set of 708 test segments from 7 lecturers.

Training Corpus	WER
Voxforge (23.3h)	82.5%
Voxforge (23.3h) + Transcribed Lectures (7.2h)	62.6%

3.4.1 Creation of The German Phonetic Dictionary

The phonetic dictionary is an essential part of every ASR software. For each word that appears in the transcripts, it defines one or more phonetic representations. As our speech corpus is growing continuously, the extension and maintenance of the dictionary becomes a common task. An automatic generator is highly desired. Unfortunately it lacks such a tool in the open-source context. Therefore, we have built a phonetics generator by using a customized phonetic alphabet, which contains 45 phonemes used in German pronunciation. We provide this tool and the compiled speech training data for the further research use.

The generation algorithm operates on three layers: On *word level*, we have defined a so-called *exemption dictionary* which contains foreign words in particular and whose pronunciation does not follow German rules, e.g., `byte`, `phänomen`. First of all, a check is made as to whether the input word can be found in the *exemption dictionary*. If this is the case, the phonetic representation is completely read from this dictionary and no further steps have to be performed. Otherwise, we scale down to *syllable level* by applying an extern hyphenation algorithm[9].

On *syllable level*, we examine the result of the hyphenation algorithm, as e.g., `com-pu-ter` for the input word `computer`. For many single syllables (and also pairs of syllables), the *syllable mapping* describes the corresponding phonetic representation, as e.g., `au f` for the German prefix `auf` and `n i: d er` for the disyllabic German prefix `nieder`. If there is such a representation for our input syllable, then it is added to the phonetic result immediately. Otherwise, we have to split the syllable further into its characters and proceed on *character level*.

On *character level*, a set of German pronunciation rules are applied to determine how the current single character is pronounced, including character type (consonant or vowel), neighboring characters, relative position inside the containing syllable, absolute position inside the whole word etc. First and foremost, a heuristic checks if the current and the next 1–2 characters can be pronounced natively. If this is not the case, or the word is only one character long, the characters are pronounced as if they were spelled letter by letter, as e.g., the abbreviations `abc` (for alphabet) and `ZDF` (a German TV channel). In the next step, we determine the character type (consonant or vowel) in order to apply the correct pronunciation rules. The conditions of each of these rules are verified, until one is *true* or all conditions are verified and proven to be *false*. If the latter is the case, the standard pronunciation is applied, which assumes *closed* vowels.

An evaluation with 20000 words from transcripts shows that 98.1% of all input words were processed correctly without any manual amendment. The 1.9% incorrect words mostly have an English pronunciation. They are corrected manually and added to the *exemption dictionary* subsequently.

Besides the striking advantage of saving a lot of time and effort needed for dictionary maintenance, Table 4 shows that our automatically created dictionary does not result in worse WER compared with the Voxforge dictionary. The used speech corpus for the evaluation is a smaller version of the German Voxforge Corpus

TABLE 4
Comparison result with the Voxforge dictionary

Phonetic Dictionary	WER
Voxforge	22.2%
Our dictionary with optimized phone-set	21.4%

9. http://swolter.sdf1.org/software/libhyphenate.html

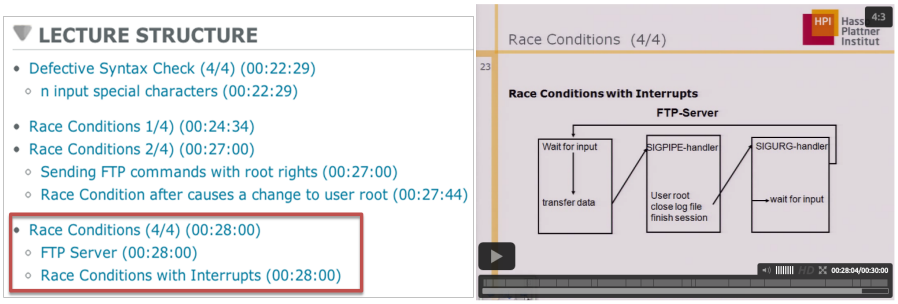

Fig. 7. Video browsing using extracted titles and key-points from OCR transcripts, where the highlighted key-points with *Bold* font style were extracted by using their stroke width feature.

which contains 4.5 hours of audio data from 13 different speakers. It is directly available from the Voxforge website including a ready-to-use dictionary. Replacing this dictionary with our automatically generated one results in a slightly better recognition result.

4 Video Content Browsing and Video Search

In this chapter we will demonstrate several video content browsing features and discuss our video search method in a large lecture video portal.

4.1 Browsing With Lecture Keyframes and Extracted Lecture Outline

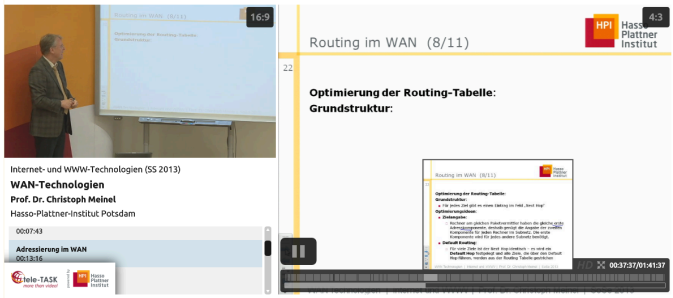

Fig. 8. Visualization of segmented lecture slides in video player

Figure 8 shows the visualization of slide key frames in our lecture video portal. The segments are visualized in the form of a timeline within the video player. This feature is intended to give a fast hint for navigation. If the user wants to read the slide content clearly, a slide gallery has been provided underneath the video player. By clicking on the thumbnails or on the timeline-units, the video will navigate to the begin of the segment.

Figure 7 demonstrates an exemplary lecture outline (structure), where the title "Race Conditions (4/4)", and the highlighted key-points "FTP Server", "Race Conditions with Interrupts" with *Bold* font style have been successfully detected by using their geometric information and the stroke width feature respectively. By clicking on the outline items the video will jump to the corresponding time position. Furthermore, the user can use the standard text search function of a web browser to search through the outline content.

4.2 Keyword Extraction and Video Search

The lecture content-based metadata can be gathered by using OCR and ASR tools. However the recognition results of automatic analysis engines are often error prone and a large amount of irrelevant words are also generated. Therefore we extract keywords from the raw recognition results. Keywords can summarize a document and are widely used for information retrieval in digital libraries. In this work, only nouns and numbers are considered as keyword candidate. The top n words from them will be regarded as keyword. Segment-level as well as video-level keywords are extracted from different information resources such as OCR and ASR transcripts respectively. For extracting segment-level keywords, we consider each individual lecture video as a document corpus and each video segment as a single document, whereas for obtaining video-level keywords, all lecture videos in the database are processed, and each video is considered as a single document.

To extract segment-level keywords, we first arrange each ASR and OCR word to an appropriate video segment according to the timestamp. Then we extract nouns from the transcripts by using the stanford part-of-speech tagger [25] and a stemming algorithm is subsequently utilized to capture nouns with variant forms. To remove the spelling mistakes resulted by the OCR engine, we perform a dictionary-based filtering process.

We calculate the weighting factor for each remaining keyword by extending the standard TFIDF (*Term Frequency Inverse Document Frequency*) score [26]. In general, the TFIDF algorithm calculates keywords only according to their statistical frequencies. It cannot represent the location information of keywords, that might be important for ranking keywords extracted from web pages or lecture slides. Therefore, we defined a new formula for calculating TFIDF score, as shown by Eq. 1:

$$tfidf_{seg-internal}(kw) = \frac{1}{N}(tfidf_{ocr} \cdot \frac{1}{n_{type}} \sum_{i=1}^{n_{type}} w_i + tfidf_{asr} \cdot w_{asr}) \quad (1)$$

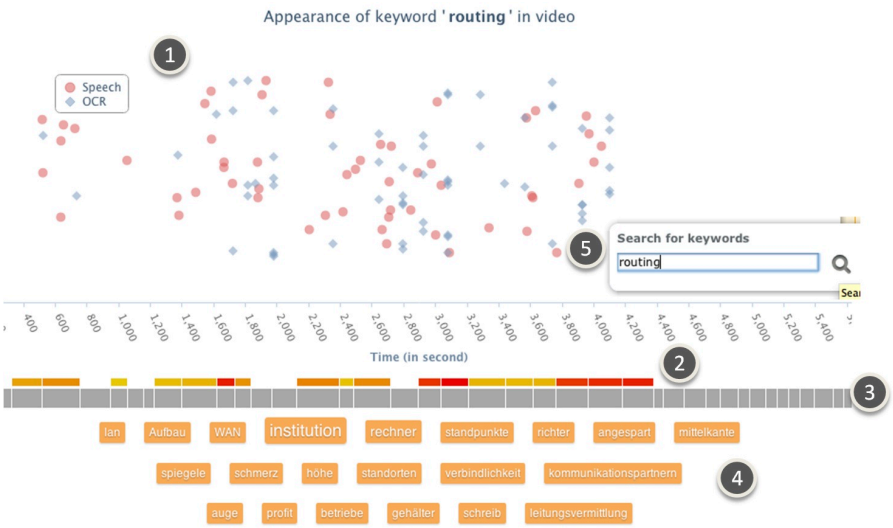

Fig. 9. Segment-level keyword browsing and keyword search function in our lecture video portal.

where kw is the current keyword, $tfidf_{ocr}$ and $tfidf_{asr}$ denote its TFIDF score computed from OCR and ASR resource respectively, w is the weighting factor for various resources, n_{type} denotes the number of various OCR text line types. N is the number of available information resources, in which the current keyword can be found, namely the corresponding TFIDF score does not equal 0.

Since OCR text lines are classified into four types in our system (cf. Chapter 3.3), we can calculate the corresponding weighting factor for each type and for each information resource by using their confidence score. Eq. 2 depicts the formula:

$$w_i = \frac{\mu}{\sigma_i} \ (i = 1 \dots n) \tag{2}$$

where the parameter μ is set to equal 1 in our system, and σ can be calculated by using the corresponding recognition accuracy of the analysis engine, as shown by Eq. 3:

$$\sigma_i = 1 - Accuracy_i \ (i = 1 \dots n) \tag{3}$$

Figure 9 demonstrates the web GUI of the segment-level keyword browsing and search function in our lecture video portal. The scatter chart diagram has been used for visualizing the keywords, as shown in Figure 9 (1), in which the horizontal axis of the scatter chart represents the video time, the red and light-blue plots represent the keywords extracted from ASR and OCR results, respectively. Underneath the chart a timeline serves for representing the linear-structure of the video (Figure 9 (3)), whereas the weight information of the actual keyword for corresponding segments is indicated by another colored timeline (Figure 9 (2)), where stronger color gradients refer to higher keyword weight of the corresponding segment. The user can zoom in/zoom out to the appropriate segment interval by clicking on the colored timeline-unit or via mouse selection in the diagram area. When mouse-hovering on the gray timeline, the recommended keywords are displayed for each segment (Figure 9 (4)). By clicking on the keyword its appearance will be highlighted in the scatter chart. The user can further click on the plot-point in the chart, the video will then navigate to the position where the word has been spoken or appears in the slide.

As already mentioned, to build a content-based video search engine by using multimodal information resources, we have to deal with solidity and consistency problems. Those information resources might be generated either by a human or by an analysis engine. For the latter case, different analysis engines may have various confidence scores. Therefore, during the ranking process we should consider both the statistical feature of the keywords and their confidence scores. We have thus defined a formula for computing the video-level TFIDF score, as shown by Eq. 4:

$$tfidf_{vid-level}(kw) = \frac{1}{N} \sum_{i=1}^{n} tfidf_i \cdot w_i \tag{4}$$

where $tfidf_i$ and w_i denote the TFIDF score and the corresponding weighting factor for each information resource. N is the number of available information resources, in which the current keyword can be found.

In our case, video search indices can be built from three information resources currently, including manually created global video metadata (lecturer name, course description etc.), ASR and OCR words. As described in Eq. 1, the OCR text lines were in turn classified into several types, and the formula extended as:

$$tfidf_{vid-level}(kw) = tfidf_g \cdot w_g + \frac{1}{N}(tfidf_{ocr} \cdot \frac{1}{n_{type}} \sum_{i=1}^{n_{type}} w_i + tfidf_{asr} \cdot w_{asr}) \tag{5}$$

where $tfidf_g$ and w_g denote the TFIDF score and the weighting factor for global video metadata.

The ranked video-level keywords are used for the content-based video search in our lecture video por-

Fig. 10. Content-based video search in the lecture video portal.

TABLE 5
keyword-video matrix *A*

	v_1	v_2	$\cdots$	v_n
kw_1	a_{11}	a_{12}	$\cdots$	a_{1n}
kw_2	a_{21}	a_{22}	$\cdots$	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
kw_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}

tal. Figure 10 shows some exemplary search results, in which segmented lecture slides, search hits and keyword weights are additionally provided to the user when hovering on the corresponding timeline-unit. Clicking on the unit the video segment will be played by a pop-up player.

The video similarity can further be computed by using a vector space model and the *cosine similarity measure*. Table 5 shows an exemplary keyword-video matrix $A_{kw \times v}$, its columns and rows correspond to video and keyword indices respectively. The value of each matrix element is the calculated $tfidf_{vid-level}$ score of the keyword kw_i in the video v_j.

Let us assume that each column of A denotes a vector d_j which corresponds to a video v_j. Here, the dimension of d_j is the number of selected keywords. Let q denote the query vector which corresponds to another video, the similarity between two videos can then be calculated by using cosine similarity measure according to Eq. 6:

$$sim(d_j, q) = \frac{\sum_{i=1}^{m}(a_{ij}q_i)}{\sqrt{\sum_{i=1}^{m}(a_{ij})^2}\sqrt{\sum_{i=1}^{m}(q_i)^2}} \qquad (6)$$

Furthermore, the TFIDF score for the *inter-video* segment comparison can be derived by combining $tfidf_{seg-internal}$ and $tfidf_{vid-level}$. Using this score we are able to implement a segment-based lecture fragment search/recommendation system.

5 Initial User Study

We have evaluated the proposed indexing features in the form of a user study intended to verify the research hypothesis 1 with the following derived questions:

- Which video indexing tools are helpful for finding a specific lecture topic within a lecture video, and
- how quickly and how accurately can this be achieved?
- Can learning effectiveness be improved by using video indexing tools in a lecture video portal?

The involved indexing tools in this evaluation include automatically extracted slides, that are provided in a timeline format, lecture outline extracted from the slides enhanced with direct links into the video as well as the keyword browsing/search functionality.

12 single participants were recruited for the user study. They were bachelor or first semester master students at our institute in the field of "IT Systems Engineering". In order to fulfill the goal, two tasks have been conducted:

- **Task 1** search for the temporal scope of a specific lecture topic within a 1 hour lecture video by using one of the following setups:
  - video with a seekable player
  - key-frames and video
  - lecture outline and video
  - keywords and video
  - all available indexing tools and video
- **Task 2** was to watch a complete topic segment of 10 minutes. One time the participants were allowed to use all of the indexing tools on the video and for the second video they were only allowed to use the video player without additional tools. After watching a video the students had to take a short exam so we could measure learning effectiveness.

After a preliminary introduction, each student had 1 hour time to fulfill the requested tasks. We applied a within-subject-design for the experiment intended to compare the outcomes in speed and accuracy of various setups. Therefore, each participant was asked to perform all tasks. For each of the setups we prepared different videos. All the videos were chosen carefully with similarity in complexity of the topic and findability of the required information in mind. The videos are part of computer science studies and they were made in the subjects' native language. We randomly selected the order

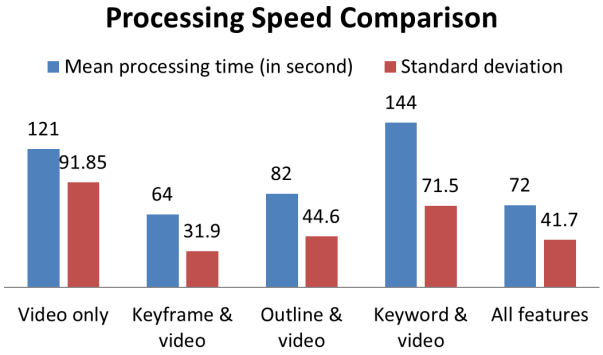

Fig. 11. Processing speed evaluation results of task 1. The bar chart depicts the mean processing time and standard deviation of each setup.

TABLE 6
Processing accuracy evaluation results of task 1

Setup	Recall	Precision	F_1 Measure
Keyframe & Video	0.99	1.0	0.99
Keyword & Video	0.99	1.0	0.99
All features & Video	0.96	0.99	0.97
Outline & Video	0.87	0.95	0.91
Video only	0.81	0.83	0.82

TABLE 7
Mean and standard deviation results of task 2

	Video Only			Indexing Tools & Video		
	W1	W2	norm-W2	W1	W2	norm-W2
Mean	1.3	5.6	6.6	3.6	6.8	8.1
Std-dev.	4.1	2.7	2.6	2.8	2.4	2.2

of the tasks and used a counterbalanced order to assign videos to each setup. This was intended to prevent the subjects from becoming tired and their learning being thereby effected over time.

All participants had never worked with the video indexing tools before the user study. But even though the tools were new to them, every single subject was able to use them to their advantage after a small introduction. We found that the requested information was found faster and more accurate than searching without the video indexing tools whenever the students used keyframes, lecture outline or all of the available features (cf. Figure 11 and Table 6). The only tool showing a slower processing speed were the keywords. In our understanding this is caused by the necessity to scroll the website down from the video player to the keywords GUI. Nevertheless, it achieved the best accuracy results as shown in Table 6.

In the second task, we prepared three multiple choice and a free-text question for each test video. The total score for one video is around 12 points. We designed three methods for the result scoring:

- *W1* – for multiple choice question, +1 for each correct answer, -2 for each incorrect/missed answer; +2 for each correct point in free-text answer.

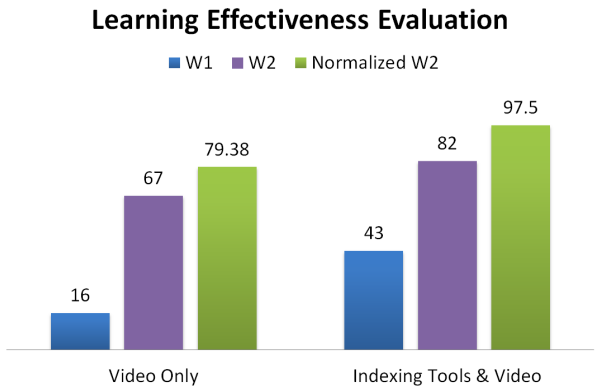

Fig. 12. Results of the learning effectiveness evaluation (task 2), where the bar chart presents the accumulated score for each setup of the exam.

- *W2* – for multiple choice question, +1 for each correct answer, -1 for each incorrect answer, 0 for missed answers; +2 for each correct point in free-text answer.
- *Normalized W2* – since the maximal number of correct answers for the multiple choice questions is different for each test video, we thus additionally built the normalized result of W2.

Figure 12 shows that learning effectiveness can be improved measurably by using video indexing tools, where the mean and standard deviation of scoring results are depicted in Table 7. Almost all the participants have used the keyframe and outline feature in this task for reviewing knowledge points, while the keyword feature was rarely used during the watching process. The results of the post-test-questionnaire indicate that all participants would like to see the indexing tools be used in lecture video archives.

The preliminary results of the user study can provide a trend which corroborates the hypothesis 1. The results are not statistically significant though and need a higher number of test persons to ensure reliability.

6 Conclusion and Future Work

In this paper, we presented an approach for content-based lecture video indexing and retrieval in large lecture video archives. In order to verify the research hypothesis we apply visual as well as audio resource of lecture videos for extracting content-based metadata automatically. Several novel indexing features have been developed in a large lecture video portal by using those metadata and a user study has been conducted.

As the future work, the usability and utility study for the video search function in our lecture video portal will be conducted. Automated annotation for OCR and ASR results using Linked Open Data resources offers the opportunity to enhance the amount of linked educational resources significantly. Therefore more efficient search and recommendation method could be developed in lecture video archives.

REFERENCES

[1] E. Leeuwis, M. Federico, and M. Cettolo, "Language modeling and transcription of the ted corpus lectures," in *Proc. of the IEEE ICASSP*. IEEE, 2003, pp. 232–235.

[2] D. Lee and G. G. Lee, "A korean spoken document retrieval system for lecture search," in *Proc. of the SSCS speech search workshop at SIGIR*, 2008.

[3] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio data: Preliminary investigations," in *Proc. of the HLT-NAACL Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, 2004.

[4] A. Haubold and J. R. Kender, "Augmented segmentation and visualization for presentation videos," in *Proc. of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 51–60.

[5] W. Hürst, T. Kreuzer, and M. Wiesenhütter, "A qualitative study towards using large vocabulary automatic speech recognition to index recorded presentations for search and access over the web," in *Proc. of IADIS WWW / Internet (ICWI)*, 2002, pp. 135–143.

[6] C. Munteanu, G. Penn, R. Baecker, and Y. C. Zhang, "Automatic speech recognition for webcasts: How good is good enough and what to do when it isn't," in *Proc. of the 8th international conference on Multimodal interfaces*, 2006.

[7] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513 – 523, 1988.

[8] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975. [Online]. Available: http://doi.acm.org/10.1145/361219.361220

[9] T.-C. P. F. Wang, C-W. Ngo, "Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis," *Journal of Pattern Recognition*, vol. 41, no. 10, pp. 3257–3269, 2008.

[10] M. Grcar, D. Mladenic, and P. Kese, "Semi-automatic categorization of videos on videolectures.net," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II*, ser. Lecture Notes in Computer Science, W. L. Buntine, M. Grobelnik, D. Mladenic, and J. Shawe-Taylor, Eds., vol. 5782. Springer, 2009, pp. 730–733.

[11] T. Tuna, J. Subhlok, L. Barker, V. Varghese, O. Johnson, and S. Shah, "Development and evaluation of indexed captioned searchable videos for stem coursework," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, ser. SIGCSE '12. New York, NY, USA: ACM, 2012, pp. 129–134. [Online]. Available: http://doi.acm.org/10.1145/2157136.2157177

[12] H. J. Jeong, T.-E. Kim, and M. H. Kim, "An accurate lecture video segmentation method by using sift and adaptive threshold," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, ser. MoMM '12. New York, NY, USA: ACM, 2012, pp. 285–288. [Online]. Available: http://doi.acm.org/10.1145/2428955.2429011

[13] H. Sack and J. Waitelonis, "Integrating social tagging and document annotation for content-based search in multimedia data," in *Proc. of the 1st Semantic Authoring and Annotation Workshop*. Athens , USA: Springer, 2006.

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

[15] S. Repp, A. Gross, and C. Meinel, "Browsing within lecture videos based on the chain index of speech transcription," *IEEE Trans. Learn. Technol.*, vol. 1, no. 3, pp. 145–156, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1109/TLT.2008.22

[16] J. Eisenstein, R. Barzilay, and R. Davis, "Turning lectures into comic books using linguistically salient gestures," in *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, ser. AAAI'07. AAAI Press, 2007, pp. 877–882. [Online]. Available: http://dl.acm.org/citation.cfm?id=1619645.1619786

[17] J. Adcock, M. Cooper, L. Denoue, and H. Pirsiavash, "Talkminer: A lecture webcast search engine," in *Proc. of the ACM international conference on Multimedia*, ser. MM '10. Firenze, Italy: ACM, 2010, pp. 241–250.

[18] J. Nandzik, B. Litz, N. Flores-Herr, A. Löhden, I. Konya, D. Baum, A. Bergholz, D. Schönfuß, C. Fey, J. Osterhoff, J. Waitelonis, H. Sack, R. Köhler, and P. Ndjiki-Nya, "Contentus technologies for next generation multimedia libraries," *Multimedia Tools and Applications*, pp. 1–43, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11042-011-0971-2

[19] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, vol. 93, no. 2, pp. 206–220, January 2004.

[20] B. T. N. Dala, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[21] "Ground truth data," 2013. [Online]. Available: http://www.yanghaojin.com/research/videoOCR.html

[22] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2963–2970.

[23] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, "Icdar 2011 robust reading competition: Challenge 1: Reading text in born-digital images (web and email)," in *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, Beijing, September 2011, pp. 1485–1490.

[24] H. Yang, B. Quehl, and H. Sack, "A framework for improved video text detection and recognition," *Multimedia Tools and Applications*, pp. 1–29, 2012, 10.1007/s11042-012-1250-6. [Online]. Available: http://dx.doi.org/10.1007/s11042-012-1250-6

[25] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *In Proc. of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL 2003)*, 2003, pp. 252–259.

[26] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *INFORMATION PROCESSING AND MANAGEMENT*, 1988, pp. 513–523.

Haojin Yang received the Diploma Engineering degree at the Technical University Ilmenau, in Germany 2008. In 2013, he received the doctorate degree at the Hasso-Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam. His current research interests revolve around multimedia analysis, information retrieval, semantic technology, content based video search technology. He is involved in the development of the Web-University project space enhancement of the tele-TASK video recording system.

Christoph Meinel studied mathematics and computer science at Humboldt University in Berlin. He received the doctorate degree in 1981 and was habilitated in 1988. After visiting positions at the University of Paderborn and the Max-Planck-Institute for computer science in Saarbrücken, he became a full professor of computer science at the University of Trier. He is now the president and CEO of the Hasso-Plattner-Institute for IT-Systems Engineering at the University of Potsdam. He is a full professor of computer science with a chair in Internet technology and systems. He is a member of acatech, the German "National Academy of Science and Engineering," and numerous scientific committees and supervisory boards. His research focuses on IT-security engineering, teleteaching, and telemedicine. He has published more than 400 papers in high-profile scientific journals and at international conferences. He is a member of the IEEE.