Electures-Wiki—Toward Engaging Students to Actively Work with Lecture Recordings

Christoph Hermann and Thomas Ottmann, *Member*, *IEEE Computer Society*

Abstract—In this paper, we present the integration of a Wiki with lecture recordings using a tool called *aofconvert*, enabling the students to visually reference lecture recordings in the Wiki at a precise moment in time of the lecture. This tight integration between a Wiki and lecture materials allows the students to elaborate on the topics they learned in class as well as thoroughly discuss their own aspects of those topics. This technology can enable students to get actively involved in a collaborative learning process. One prerequisite for facilitating this consists in a reliable method for detecting slide transitions in lecture recordings. We describe an improved technique for slide transition detection in video-based/screen-grabbed lecture recordings when the object-based representation is not available. Our experiments demonstrate the accuracy of this new technique. A survey conducted with our students after using the Wiki in class completes this article and demonstrates which technical features are most important for such a Wiki.

Index Terms—Electures, wiki, lecture recordings, enabling technology.

1 Introduction

With the increasing use of lecture recordings (they have been used over 10 years and are nowadays widely accepted) more and more lecturers tend to think that recording their lectures is sufficient and that it is no longer necessary to distribute a script of the lecture contents. Zupancic and Horz [1] even state that "the time students spent with [...] recordings was comparable to [using] scripts and books during the semester." This might be true; however, this leads to the fact that students consume the lecture recordings passively at home, since no real interaction with the learning material is needed. At many universities lecture recordings are used as a simple replacement for scripts. They are recorded during the lecture or in a studio and then uploaded to the web or some other distribution channel for time-independent delivery. In addition, the content providers tend to create online portals (like Academic Earth[1] or our Electures-Portal[2]) for the distribution of these *e-lectures* (or *Electures* as we call them) that enables them to categorize and organize the materials to make them more easily accessible.

Often no didactic scenario for using additional materials or tools is developed at the universities employing lecture recordings. The lecturers are not given any training on how to reuse those materials nor are they at all taught how to use lecture recordings or other new media. They also do not provide the students with precise guidelines on how to use these tools in their learning processes.

Research in the direction of students' interaction with learning materials [2], [3] shows that the level of engagement with the given materials (e.g., animations) has an effect on the students' learning results.

Since there is no direct contact with other students or the teacher while learning with lecture recordings at home—compared to a live lecture—it would be useful to explore whether using a tool to support the collaborative creation of learning materials in a Wiki while maintaining a direct relation to the employed lecture recordings helps the students. To this end, investigations on the level of engagement of students learning with lecture recordings have to be conducted. More precisely, it has to be verified if the students are motivated to collaboratively create Wiki pages to enhance knowledge acquisition.

Before such studies are possible, enabling technology needs to be developed. Therefore, in a first step, we created a special Wiki and tested the technical aspects of these applications in a computer science course with first year students. Future work will comprise a learner centered study to compare the learning outcome of students learning with such a tool with the learning outcome of students using a regular wiki or no wiki at all.

Our main contribution and technical developments described in the following sections consists of the following parts:

1. Using a newly developed tool (*aofconvert*), students are able to directly and visually reference lecture recordings in a Wiki at a specific point in time.
2. With this tool, we can convert object-based lecture recordings to PDF and other output formats. Since we access the vector representation of the elements in the recording, this conversion is lossless and scalable and, therefore, fits all possible output sizes.
3. If the vector-based representation is not accessible, we use an improved technique for slide transition detection in video-based/screen-grabbed lecture recordings for further postprocessing.

[1] http://www.academicearth.org.
[2] http://electures.informatik.uni-freiburg.de.

• *The authors are with the Department of Computer Science, Institut für Informatik, Georges-Köhler-Allee 51, 79110 Freiburg, Germany. E-mail: {hermann, ottmann}@informatik.uni-freiburg.de.*

Manuscript received 3 May 2010; revised 27 Sept. 2010; accepted 25 Jan. 2011; published online 31 Mar. 2011.
For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2010-05-0074.
Digital Object Identifier no. 10.1109/TLT.2011.18.

1939-1382/11/$26.00 © 2011 IEEE Published by the IEEE CS & ES

4. We show that using the object-based representation for lecture recording has significant advantages over screen-grabbing since the interaction of the lecturer with the recording application itself is not recorded. This facilitates the postprocessing of such materials a lot.

2 Related Work

More than 15 years ago the University of Freiburg established a base for today's lecture recordings. At that time our group started with the development of a system called *AOF* [4] which allows to record and transfer a presentation (speech and the slides as well as the annotations of the author) to several computers.

Of course over time those systems evolved a lot and nowadays are very easy to use (i.e., commercial products like Camtasia Studio[3] or Lecturnity[4]). Therefore, more and more lecturers want (or are pushed by the collective voice of the students) to create lecture recordings of their everyday presentations. Since almost every lecture from our computer science and microsystem technology departments (as well as lectures from other disciplines like psychology, etc.) are recorded, we get vast amounts of data that have to be efficiently stored and organized.

Therefore, we created an advanced archive called the Electures-Portal [5] where all of the Electures are archived and organized for the students to easily access all materials. They can download or stream the recordings to their computers and search through them [6], [7] to easily find what they need to revise or prepare for a course.

Wikis are getting more and more commonly used at schools and universities in different settings. Parker and Chao [8] showed how broadly Wikis are used, investigated their contribution to several learning paradigms and suggest different scenarios for an educational use of Wikis.

A few major points in using Wikis and other so-called "Web 2.0" or "social software" are: they are easily and rapidly deployable as well as easy to use. They allow the users to focus on information exchange and collaborative tasks instead of hindering them with the use of a difficult technical environment.

Parker and Chao reflected that mostly two paradigms in learning are supported by using a Wiki: the cooperative/collaborative paradigm and the constructivist paradigm. Both of these paradigms are interesting for us and with our technological development we try to address both in our learning scenario described in Section 5.

They also pointed out that using Wikis facilitates group interaction and enables the students to create a set of documents that reflect the shared knowledge of the learning group.

By introducing the possibility to directly reference lecture materials in the Wiki and even to further visually cite what has been taught in class, we tighten the relation of the created contents to the presented learning contents.

Using the *slides2Wiki* tool, O'Neill [9] described a setting similar to ours, where students can create Wiki pages starting from the contents of lecture slides. O'Neill put a skeleton of the lecture materials onto a Wikipage and encouraged the students to write them out.

O'Neill stated that there are basically three different possibilities whether or not to distribute a script or the slides to the students: 1) Do not give out the slides or materials at all and, therefore, force the students to take notes by themselves. 2) Give out the slides after the course. 3) Give out the materials to the students before the course so that they can add their own notes.

She stated that, basically, all three approaches are problematic since none of them satisfies all students, especially when additional materials are used in class. She tried to involve students to work on a Wiki by automatically transferring the slides content into the Wiki and let the students flesh out the created skeleton.

We consider this low-tech approach insufficient for the creation of high-quality lecture notes. The main disadvantage is that it imposes restrictions on the format of the lecture materials.

The difference in our approach is that we keep the original contents intact and do not copy them into the Wiki, but allow the students to visually reference them and, therefore, support the students to use a lot of different materials (also from other courses) to create an article about a certain topic.

We discovered that very often the slides do not follow a structure that can be extended in the way O'Neill described it. Therefore, it is more useful to pick out the lecture contents that explain certain topics (i.e., an algorithm) or show a diagram or an interesting object and let the students discuss and develop an article in the Wiki that completely depicts all the contents related to that topic.

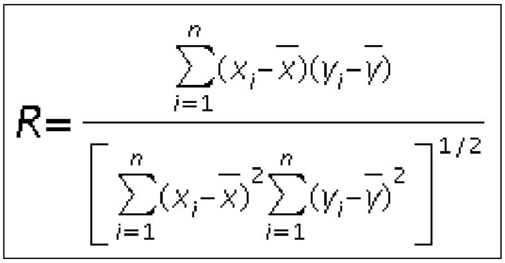
There are other synchronous systems that allow students to work more closely together on the topics of a lecture, such as shared whiteboard systems or note-taking systems such as "livenotes" [10]. Those systems are completely different from ours in the way that they only support the collaborative aspect of the students working together or help the students to communicate between each other during a lecture.

Sack and Waitelonis [11] described that they added Wiki pages to their academic video search engine *Yovisto* which can be supplemented either by the lecturer and other academic staff members or the students by adding more content to the lecture recordings. Using their engine, parts of videos can also be referenced directly using a simple hyperlink. The content of these Wiki pages can then be used to improve the results of their search engine. They also employed additional student- or user-generated content such as tags to further improve those results.

Compared to our work, these Wiki pages simply supplement the videos listed by the search engine rather than giving students the possibility to reference certain parts of the lecture recordings when editing the Wiki pages. Yovistos Wiki pages can merely be used to explain or discuss the contents of the video. In our work, we want to employ the Wiki pages for collaborative work on a certain topic. We offer students the possiblity to reuse and directly reference all available lecture materials (such as lecture

[3] http://www.techsmith.de/camtasia.asp.

[4] http://www.lecturnity.de/.



$$R=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\left[\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2\right]^{1/2}}$$

Fig. 1. Screenshot of the output of the new math Wiki plugin supporting very complex math formulae.

recordings) within the pages of our Wiki. For example, in the introductory part of a Wikipage, they could include videos or explanations from earlier materials of a certain mathematical proposition and then build further explanations upon this knowledge.

Lauer and Trahasch [12] proposed a model for directly anchoring user's discussions in learning contents. They described how this model can be used in an application to visualize students' discussions on lecture recordings directly within those contents. Other authors mainly reused the lecture recordings for distribution of the lectures contents to other students or target groups over the web or other mobile devices [13].

3 The Electures-Wiki

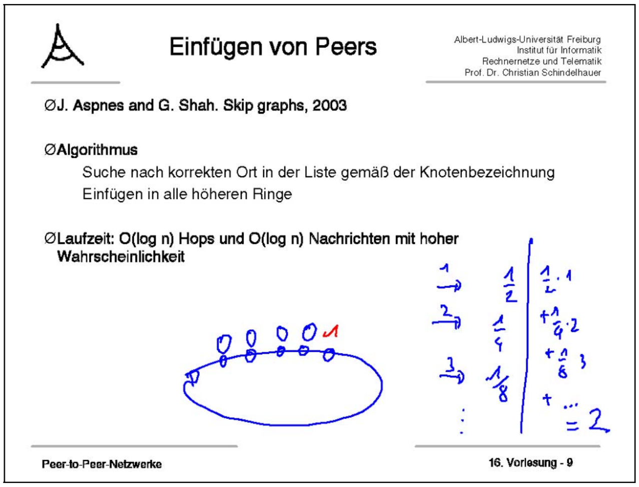
Wikis are widely employed at schools and universities [14], [15], [16] as well as in companies [17], [18] for creating intranets and other collections of knowledge. As stated by Notary [19] and others, collaboration and negotiation with other students leads to what is often referred to as "self-explanation effect." (We use the term collaboration in the sense of a group of students working on the same clearly defined exercise with different roles: the students have to work together in small groups and are supervised by tutors [20].) This effect can be increased by not only using text, but also diagrams [21] and other materials (e.g., algorithm visualisations).

Based on this, we were searching for possibilities to combine the lecture recordings with a Wiki. We wanted the students to be able to directly reference the contents of the lectures by embedding visual references to the lecture recordings, just as hyperlinks are commonly employed in Wikis to reference other sources of knowledge.

Of course students could simply link to a certain lecture recording, but we wanted to push this one step further and allow a much more fine grained mode of reference: using our Wiki, students can directly create a visual reference of a certain point in the recording enabling them to directly access the explanation the lecturer gave at the precise referenced moment while learning with those Wiki pages.

Therefore, we implemented additional features on top of an open source Wiki to fulfill our requirements as well as other needed functions in a scientific environment (for example advanced math formulae support, see Fig. 1).

In addition to visually reference lecture recordings and other materials, we want the students to be able to continue to learn anywhere using their created contents, independent from having access to the internet or even a computer. Using the Wiki, they can create a printable script, which is often not available when lecture recordings are employed.



Einfügen von Peers

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

ØJ. Aspnes and G. Shah. Skip graphs, 2003

ØAlgorithmus

Suche nach korrekten Ort in der Liste gemäß der Knotenbezeichnung

Einfügen in alle höheren Ringe

ØLaufzeit: O(log n) Hops und O(log n) Nachrichten mit hoher Wahrscheinlichkeit

$1 \rightarrow \frac{1}{2}$ $\frac{1}{2}\cdot 1$
$2 \rightarrow \frac{1}{4}$ $+\frac{1}{4}\cdot 2$
$3 \rightarrow \frac{1}{8}$ $+\frac{1}{8}\cdot 3$
$\vdots$ $+ \dots = 2$

Peer-to-Peer-Netzwerke

16. Vorlesung - 9

Fig. 2. Scalable graphic of a lecture recording on peer-to-peer networks at an arbitrary point in time.

The results of our evaluation (see Section 5) show that the Wiki and the additional features we implemented are accepted and esteemed as helpful by most of the students. Of course these subjective results have to be verified by a more elaborate evaluation involving the real learning outcome of the students, which is beyond the scope of this paper.

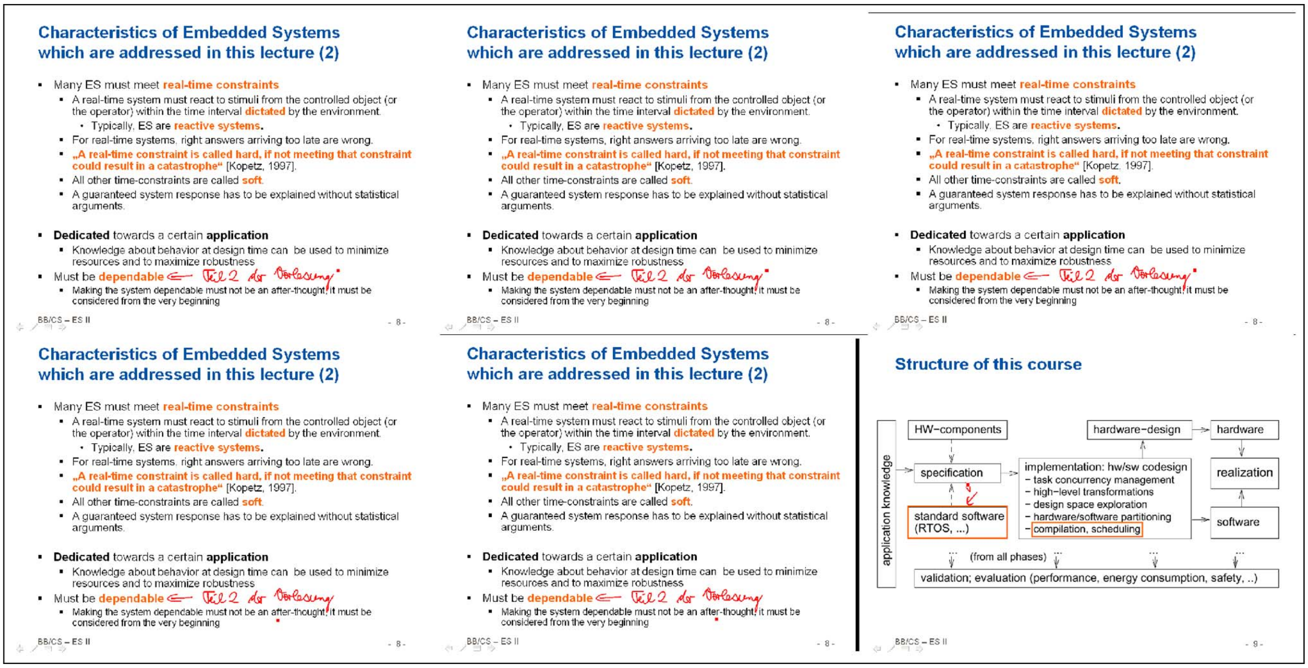
Here, we concentrate on creating enabling technologies for the reuse and postprocessing of lecture recordings.

4 Technical Background

Visually referencing lecture recordings means that we had to develop a tool (we call it *aofconvert*) which allows us to include screenshots of learning materials (see Fig. 2) into the Wiki. Since those screenshots are employed at several sizes, this has to be done in a scalable way and in the best possible quality (using lossless scaling/vector graphics whenever possible).

To explain the conversion process for different media types utilized in the Wiki, we introduce different classes of learning materials. Lecture materials can be classified into *page-based* media (e.g., PowerPoint or PDF documents) and *time-based* media (as lecture recordings). Page-based contain no notion of time. Navigation in these files is only possible by navigating forward and backward in the pages of the document. For every page, the content of this specific page is clearly defined.

Contrarily to this, time-based media can display different contents at different points in time on the same page. Most of those documents allow random access based on a time slider or other means of time-based navigation. During a time interval, while displaying a specific page from time-based media, the content of the currently displayed page might change. Therefore, it is difficult to specify which timestamp of the whole document represents a specific page of this document.



Characteristics of Embedded Systems which are addressed in this lecture (2)

- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
    - Typically, ES are **reactive systems.**
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments.
- **Dedicated** towards a certain **application**
  - Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Must be **dependable** ⇐ Teil 2 der Vorlesung
  - Making the system dependable must not be an after-thought, it must be considered from the very beginning

BB/CS – ES II - 8 -

Characteristics of Embedded Systems which are addressed in this lecture (2)

- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
    - Typically, ES are **reactive systems.**
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments.
- **Dedicated** towards a certain **application**
  - Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Must be **dependable** ⇐ Teil 2 der Vorlesung
  - Making the system dependable must not be an after-thought, it must be considered from the very beginning

BB/CS – ES II - 8 -

Characteristics of Embedded Systems which are addressed in this lecture (2)

- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
    - Typically, ES are **reactive systems.**
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments.
- **Dedicated** towards a certain **application**
  - Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Must be **dependable** ⇐ Teil 2 der Vorlesung
  - Making the system dependable must not be an after-thought, it must be considered from the very beginning

BB/CS – ES II - 8 -

Characteristics of Embedded Systems which are addressed in this lecture (2)

- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
    - Typically, ES are **reactive systems.**
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments.
- **Dedicated** towards a certain **application**
  - Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Must be **dependable** ⇐ Teil 2 der Vorlesung
  - Making the system dependable must not be an after-thought, it must be considered from the very beginning

BB/CS – ES II - 8 -

Characteristics of Embedded Systems which are addressed in this lecture (2)

- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
    - Typically, ES are **reactive systems.**
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments.
- **Dedicated** towards a certain **application**
  - Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Must be **dependable** ⇐ Teil 2 der Vorlesung
  - Making the system dependable must not be an after-thought, it must be considered from the very beginning

BB/CS – ES II - 8 -

Structure of this course

application knowledge
HW−components
specification
standard software (RTOS, ...)
implementation: hw/sw codesign
− task concurrency management
− high−level transformations
− design space exploration
− hardware/software partitioning
− compilation, scheduling
hardware−design
hardware
realization
software
(from all phases)
validation; evaluation (performance, energy consumption, safety, ..)

BB/CS – ES II - 9 -

Fig. 3. Example showing the slide transition detection performed by our tool *aofconvert*. The last image in this series is detected as a new slide.

Additionally, for time-based media, two subclasses exist: time-based media with slide transition metadata (e.g., Lecturnity lecture recordings) and time-based media without slide transition metadata (e.g., AVI movies like Techsmith Camtasia encoded lecture recordings). Slide transition metadata consists at the very least of a list of timestamps with corresponding page numbers denoting the start time for displaying this specific page in the time-based document.

Our tool *aofconvert* is able to process page-based as well as time-based media and is able to create screenshots for each slide or page that is shown in the lecture recording. For time-based media, we have to detect the slide transitions before we are able to create the screenshots in order to reduce the vast amount of movie frames contained in a lecture recording.

This problem is known as *shot detection* or *temporal video segmentation* and there has been a lot of research on digital video in this field. The survey conducted by Koprinska and Carrato [22] gives an excellent overview about different techniques in the area of digital video. Note, however, that videos obtained by screen-grabbing have different characteristics from digital videos such as movies. We will explain this further in Section 4.2 where we will exploit the characteristics of the histogram of such recordings.

Videos can be seen as a collection of images (video frames) that are displayed one after the other. Since storing all those images that often only differ in small parts of the image would be very inefficient, most videos codecs employ some sort of compression technique to only store the changes needed to compute the next image based on the previous frame.

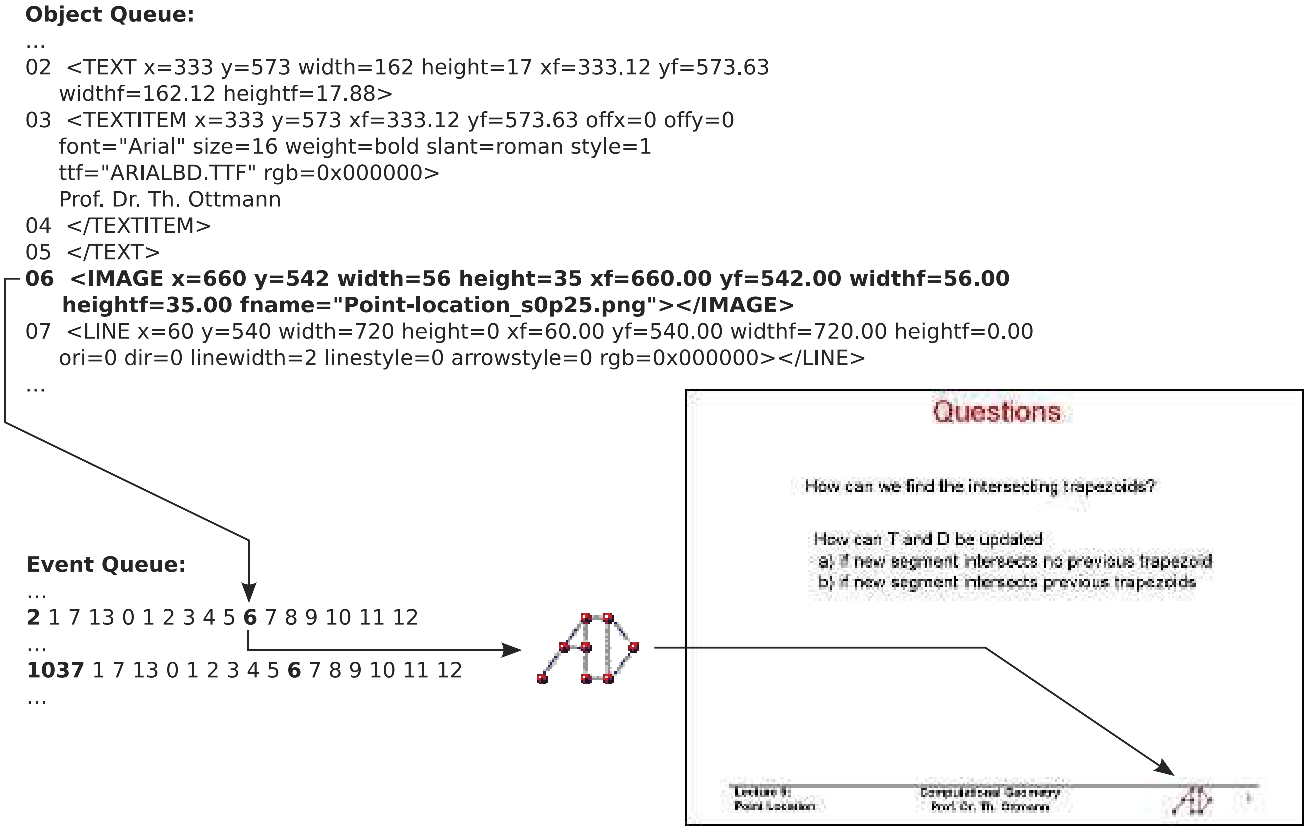
Basically, all of the above algorithms measure the difference between two consecutive video frames. There are techniques for uncompressed video frames like the local or global pixel difference (global = based on the whole picture, local = only parts of the image are compared), histogram-based comparisons (global and local), feature-based techniques (i.e., edge detection) or more complex variants like model-based techniques.

For compressed video (like MPEG), other techniques are employed. An MPEG video, for example, mainly consists of three different types of frames: I (intra), P (predicted), and B (bi-directional) frames. For processing this format, one can only use direct information from one frame (I-frames) or more than one frame (P and B-frames). Additionally, precomputed information, like motion vectors and block averages can be used. A scene transition is detected if a certain treshold (which is dependent on the employed technique) for the difference of two consecutive frames is met. If the similarity of two frames exceeds this threshold, the image is completely decoded and the techniques of uncompressed video are applied.

In our setting, the slide transitions of lecture recordings are easier to identify than scene changes in digital videos/movies. Additionally, videos created with an object-based recording system (like Lecturnity) record additional slide transition metadata that can be used to easily identify slide transitions. When using screen-grabbing this metadata has to be recorded by capturing keyboard or mouse input; otherwise, it will have to be recomputed afterward with complex postprocessing steps.

For other material belonging to the class without slide transition metadata such as screen-grabbing videos (i.e., Camtasia encoded AVI videos), we developed a new and sufficiently precise method to detect slide transitions.

Fig. 3 depicts the problem: we want to create only the necessary amount of screenshots from a lecture recording. The first five images look very similar, although they contain small differences since the author moved the mouse



Object Queue:

```
...
02  <TEXT x=333 y=573 width=162 height=17 xf=333.12 yf=573.63
    widthf=162.12 heightf=17.88>
03  <TEXTITEM x=333 y=573 xf=333.12 yf=573.63 offx=0 offy=0
    font="Arial" size=16 weight=bold slant=roman style=1
    ttf="ARIALBD.TTF" rgb=0x000000>
    Prof. Dr. Th. Ottmann
04  </TEXTITEM>
05  </TEXT>
06  <IMAGE x=660 y=542 width=56 height=35 xf=660.00 yf=542.00 widthf=56.00
    heightf=35.00 fname="Point-location_s0p25.png"></IMAGE>
07  <LINE x=60 y=540 width=720 height=0 xf=60.00 yf=540.00 widthf=720.00 heightf=0.00
    ori=0 dir=0 linewidth=2 linestyle=0 arrowstyle=0 rgb=0x000000></LINE>
...
```

Event Queue:

```
...
2 1 7 13 0 1 2 3 4 5 6 7 8 9 10 11 12
...
1037 1 7 13 0 1 2 3 4 5 6 7 8 9 10 11 12
...
```

Questions

How can we find the intersecting trapezoids?

How can T and D be updated
a) if new segment intersects no previous trapezoid
b) if new segment intersects previous trapezoids

Lecture 9:
Point Location

Computational Geometry
Prof. Dr. Th. Ottmann

Fig. 4. Structure of the Lecturnity Format.

after adding the handwritten annotation. The next image shows a new page and our task is to identify the first five images as the same page, and to recognize the slide transition from image five to six.

There has already been some research tailored to screen-grabbing videos done by Welte et al. [23], Kopf et al. [24], Ziewer et al. [25] and others, of which the work of Ziewer et al. is probably most similar to ours.

4.1 The Lecturnity Format

The Lecturnity Format (which still uses the format originally developed for AOF [4], [26]) allows us to access the vector-based representation of the recordings. It basically consists of two main files: the *event queue* and the *object queue*. The *object queue* defines all the objects that appear in the lecture recording and the *event queue* defines which of these objects appears at any point in time of the recording. Fig. 4 shows the relation between these data structures and the resulting slides. Line 6 of the *object queue* (highlighted in bold in Fig. 4) describes an object with the ID 6 (an image) which shall be displayed at timestamp 2 and timestamp 1,037 of the lecture recording (the first column of the *event queue* is the timestamp in milliseconds since the beginning of the playback).

For each slide (the second column in the *event queue* denotes the slide number), our application draws the contents of the lecture for the given timestamp as vector objects on a panel which then can be scaled to any size according to our preferences. Using a Java ImageWriter, this panel can then be easily exported to a bitmap picture.

4.2 The Camtasia TSCC Format

To process the slides from a Camtasia encoded AVI video, we implemented a full AVI demuxer and TSCC decoder in the Java programming language. This enables us to access every video frame and to decode it to create an image from the frame data.

The Camtasia Format is not documented publicly. It can only implicitly be derived from the implementation. Because we are using some specialties of this format in our tool, we will briefly describe how video frames are encoded in the TSCC Format.

One AVI chunk (see [27] for a complete description of the AVI format) in TSCC encoded videos consists of zlib compressed data. After uncompressing this data, the bitmap image can be decoded using an algorithm similar to the run length encoding [28] with the difference that one color pixel of the resulting image can be encoded in 8, 16, 24, or 32 bits. The color depth of the encoding can be read from the metadata stored in the AVI header. Decreasing the color depth allows higher compression of the files, since a fewer number of colors can then be displayed in the resulting files. Algorithm 1 shows the main procedure needed to decode the final image.

Algorithm 1. Pseudocode of the decompression of a TSCC encoded video frame
(AVI chunk after decompression with zlib).

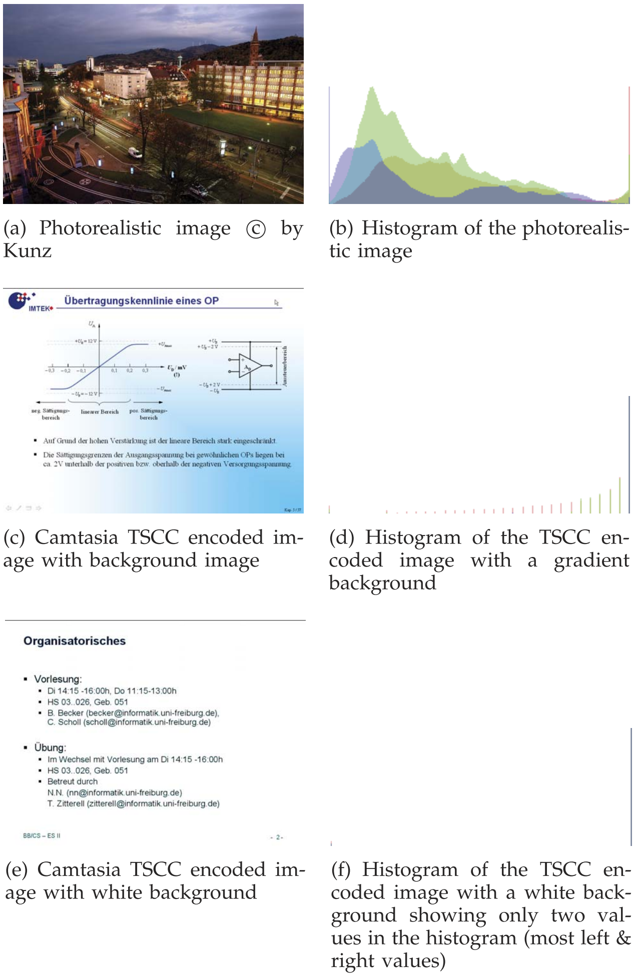
```
 Input: An RLE compressed byte sequence
Output: A decompressed picture
```

```
1:  $b0 = getNextByte(aviFile)$
2:  if $b0 == 0$ then
3:      $b1 = getNextByte(aviFile)$
4:      if $b1 == 0$ then
5:          return END_OF_LINE
6:      else if $b1 == 1$ then
7:          return END_OF_IMAGE
8:      else if $b1 == 2$ then
9:          {* change cursor position *}
10:         $p1 = getNextByte(aviFile)$
11:         $p2 = getNextByte(aviFile)$
12:         $currentPos_Y += p2$
13:         $currentPos_X += p1$
14:     else
15:         {* depending on color depth (8,16,24,32)
            copy color pixels to output *}
16:         for $i = 0; i < b1; i++$ do
17:             $pixel = getNextByte(aviFile, depth);$
18:             $draw(pixel);$
19:         end for
20:     end if
21: else
22:     {* $b0! = 0$; read one pixel and draw it b0 times *}
23:     $pixel = getNextByte(aviFile, depth);$
24:     for $i = 0; i < b0; i++$ do
25:         $draw(pixel);$
26:     end for
27: end if
```

The `draw()` method of this algorithm (see Lines 18 and 25) calculates the color from the `pixel` value and draws this pixel in the final image at the specified position ($currentPos_X$, $currentPos_Y$). The RGB color to be drawn is compressed in either 1, 2, or 3 bytes (depending on the color depth) and can be obtained by expanding these bytes to three integer values in the range of 0..255. A color histogram usually consists of an array of size 256 for each band (red, green, blue). During the color decoding process, we can easily compute the three-band color histogram of the image without adding to the overall complexity of the decoding algorithm (increasing one integer for every color pixel is done in constant time), since the values for each band of a color pixel are already known. This histogram can then be used to detect the slide transitions, we need to identify the start point for each page of the document.

As we can see in Fig. 5, the histogram for TSCC encoded images (see Figs. 5c and 5d) is very special compared to a histogram from a photorealistic image (see Figs. 5a and 5b). Fig. 5d shows a histogram with very nice curves for the three-color bands. The histograms of the TSCC encoded images are very different and look more like a discrete signal (which results from the color compression). Fig. 5f shows the histogram of a TSCC encoded image with black text on a white background. Here, only two values in the histogram are left (black and white). Using this information in the histograms page transitions in TSCC encoded videos can be detected more easily.

When changing from one slide to another, either the amount of text on this slide differs from the amount of the previous slide or additional images/objects are displayed on it. The intuitive approach of our slide detection algorithm is, that whenever a slide transition occurs, then the value of the most frequent color in the histogram changes significantly. According to this assumption, it is sufficient to evaluate this value to decide whether a slide transition happens between two consecutive frames or not. In case the author is using a background with a gradient (see Fig. 5c) then the font color is usually the most frequent color. If the background is filled with a solid color (i.e., white, see Fig. 5e), then this color is the most frequent color. When changing from one slide to another one either increases/decreases the number of pixels with the font color or increases/decreases the number of pixels with the background color. The case where a slide change occurs without changing the value of the most frequent color in the histogram is considered so rare that it can be neglected (in our experiments it only happened when artificially forced).


(a) Photorealistic image © by Kunz

(b) Histogram of the photorealistic image

Übertragungskennlinie eines OP

- Auf Grund der hohen Verstärkung ist der lineare Bereich stark eingeschränkt.
- Die Sättigungsgrenzen der Ausgangsspannung bei gewöhnlichen OPs liegen bei ca. 2V unterhalb der positiven bzw. oberhalb der negativen Versorgungsspannung.

(c) Camtasia TSCC encoded image with background image

(d) Histogram of the TSCC encoded image with a gradient background

Organisatorisches

- Vorlesung:
  - Di 14:15 -16:00h, Do 11:15-13:00h
  - HS 03..026, Geb. 051
  - B. Becker (becker@informatik.uni-freiburg.de),
    C. Scholl (scholl@informatik.uni-freiburg.de)
- Übung:
  - Im Wechsel mit Vorlesung am Di 14:15 -16:00h
  - HS 03..026, Geb. 051
  - Betreut durch
    N.N. (nn@informatik.uni-freiburg.de)
    T. Zitterell (zitterell@informatik.uni-freiburg.de)

BB/CS – ES II - 2 -

(e) Camtasia TSCC encoded image with white background

(f) Histogram of the TSCC encoded image with a white background showing only two values in the histogram (most left & right values)

Fig. 5. Difference between the histogram of normal images and TSCC encoded images.

As the author can interact with the slides and the presentation tool during the recording, annotations (such as handwritten annotations, mouse pointers, or popup menus) appear on the slides and have to be removed to be able to precisely identify slide transitions. Fig. 6 depicts some of the *noise* our tool removes/ignores during the detection process.

Handwritten annotations are detected by analyzing the color of pixels and comparing it with a previously trained

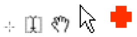


Fig. 6. Mouse pointers appearing in the recorded slides.

model of annotations and the chosen color for the annotation pen. Manually training this model for every author improves the results of this detection process. Our tool then removes the pixels of the handwritten annotations from the computed histogram and then decides, based on the most frequent color of the resulting histogram, if a slide transition has occured.

4.3 Effectiveness of the Slide Transition Detection Algorithm

Experiments have shown that this simplistic approach is sufficiently precise for detecting slide transitions in normal lecture recordings which do not have slide transition metadata.

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of n documents. Let $d_i \in D$ be made up of m slides, $\{s_{i1}, s_{i2}, \ldots, s_{im}\} = S_i$.

From the complete set D of 1,800 TSCC encoded videos, we have in the archive of the Electures-Portal (each with a duration between 10 and 90 minutes), we processed 185 with our tool. We call this set of videos $D_{185} \subset D$. The videos in D_{185} have an average of 138 keyframes (full image frames) and our tool selects an average of 32 slides from the keyframes of the corresponding video.

Of course the original media used can be composed of more slides (i.e., if the lecturer is using one set of presentation slides for several recordings) or fewer slides (if the lecturer is jumping forward and backward in the presentation during recording) than we have in the actual recording.

From D_{185}, we chose 40 videos where we manually decided which frames should be selected by our tool if it were working optimally. We call this set of videos $D_{40} \subset D_{185}$. Let S_r, $S_s \subseteq S_i$ be the set of retrieved and relevant slides, respectively, where S_i are the slides in document $d_i \in D_{40}$. We then calculate the precision p_i as shown in (1) and the recall r_i from (2) as follows:

$$p_i = \frac{|S_r \cap S_s|}{|S_r|}, \tag{1}$$

$$r_i = \frac{|S_r \cap S_s|}{|S_s|}. \tag{2}$$

The precision is $p = 0.86\%$ on average and recall is $r = 1.00$, on average, which means that our tool behaves very closely to the manual selection of slide transitions (this means that 86 percent of the slides are correctly identified).

Table 1 in the Appendix, which can be found on the Computer Society Digital Library at http://doi. ieee computersociety.org/10.1109/TLT.2011.18, shows the detailed results of the experiments with D_{40}.

Fig. 7 shows one of the reasons for errors. During recording, the lecturer interacts with the presentation software, which means that some of the used tools show up in the recorded image when using screen-grabbing. Of course the appearance of toolbars, menus, popups, etc., in the recording is unwanted and can be avoided by employing object-based recording or specifying a specific region to be screen-grabbed where these popups do not occur. Whenever such a popup is shown in the lecture recording our tool detects this as a slide transition since the contents of the slide significantly change.

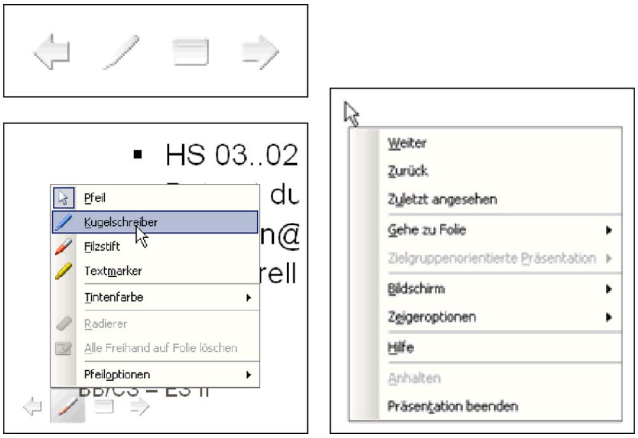


Fig. 7. Tools/popups appearing in the recorded slides.

4.4 Using the Wiki

The process to insert a visual reference (called screenshot) for a specific timestamp works as follows (see Fig. 8):

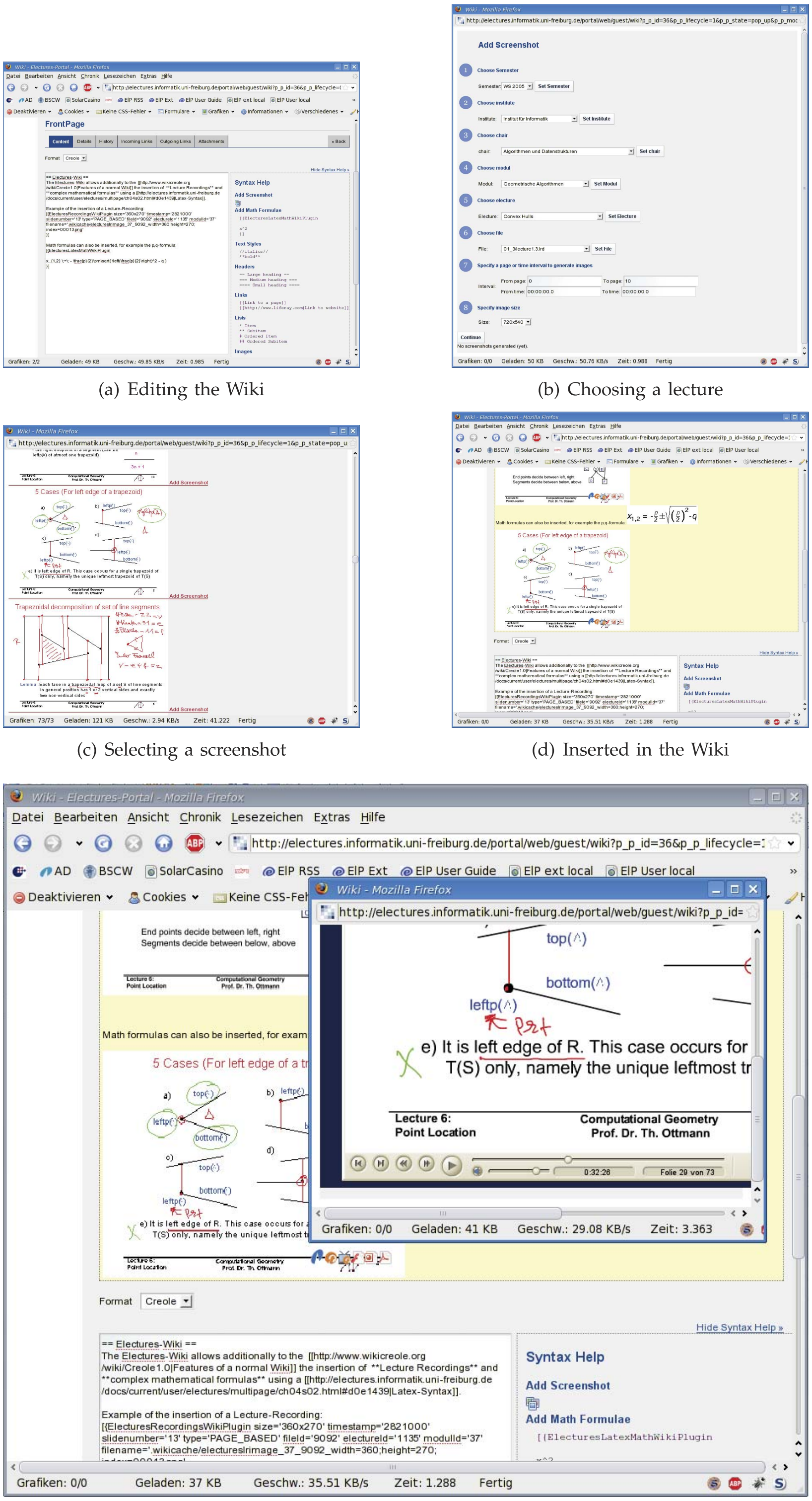
1. The user edits the Wikipage and clicks on *add screenshot*.
2. Then, he chooses the desired lecture by picking the appropriate values from the select boxes in the displayed form.
3. The screenshots are then automatically generated by our tool. Of course our tools carefully selects the displayed images (by detecting the slide transitions with the method described in Sections 4.1 and 4.2) to display only the minimal amount of images and therefore ease the selection process. The user then selects one of these screenshots.
4. The selected screenshot is inserted in the Wiki and displayed with a set of icons which point to the original files.
5. Clicking on one of these icons (i.e., the Flash icon) directly opens the requested recording (in Flash format) at the specific point in time (see Fig. 8e).

By reducing the number of slides created by our tool, we simplify the process of inserting the correct screenshot in the Wiki.

The conversion, we implemented for object-based media, gave us another very useful feature as a bonus: we are now able to generate a scalable version of the annotated slides (e.g., in PDF format) which can be distributed to our students and printed. This solves the problem that, so far, lecture recordings could not be printed in an efficient manner (only by taking screenshots of each slide and printing those manually).

Nowadays, more and more services can be accessed online without requiring a special software to be installed on the users' client.[5] We also follow this approach with our

[5] Google for example is a pioneer in this field with their Browser Chrome and Google Chrome OS: http://en.wikipedia.org/w/index.php?title=Google_Chrome_OS&oldid=331630477.


Wiki - Electures-Portal - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://electures.informatik.uni-freiburg.de/portal/web/guest/wiki?p_p_id=36&p_p_lifecycle=
AD BSCW SolarCasino EIP RSS EIP Ext EIP User Guide EIP ext local EIP User local
Deaktivieren Cookies Keine CSS-Fehler Formulare Grafiken Informationen Verschiedenes
FrontPage
Content Details History Incoming Links Outgoing Links Attachments
Format Creole
Hide Syntax Help »
Syntax Help
Add Screenshot
Add Math Formulae
Text Styles
Headers
Links
Lists
Images
Grafiken: 0/0 Geladen: 37 KB Geschw.: 35.51 KB/s Zeit: 1.288 Fertig

(a) Editing the Wiki

Add Screenshot

1. Choose Semester — Semester: WS 2005 — Set Semester
2. Choose institute — Institute: Institut für Informatik — Set Institute
3. Choose chair — chair: Algorithmen und Datenstrukturen — Set chair
4. Choose modul — Modul: Geometrische Algorithmen — Set Modul
5. Choose electure — Electure: Convex Hulls — Set Electure
6. Choose file — File: 01_3lecture1.3.lrd — Set File
7. Specify a page or time interval to generate images — Interval: From page: 0 To page: 10 From time: 00:00:00.0 To time: 00:00:00.0
8. Specify image size — Size: 720x540

Continue

No screenshots generated (yet)

(b) Choosing a lecture

5 Cases (For left edge of a trapezoid)
Trapezoidal decomposition of set of line segments
Add Screenshot

(c) Selecting a screenshot

End points decide between left, right
Segments decide between below, above
Math formulas can also be inserted, for example the p,q-formula:
$x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$

(d) Inserted in the Wiki

Wiki - Mozilla Firefox
top(Δ)
bottom(Δ)
leftp(Δ)
e) It is left edge of R. This case occurs for T(S) only, namely the unique leftmost tr
Lecture 6: Point Location
Computational Geometry Prof. Dr. Th. Ottmann
0:32:26
Folie 29 von 73
Grafiken: 0/0 Geladen: 41 KB Geschw.: 29.08 KB/s Zeit: 3.363

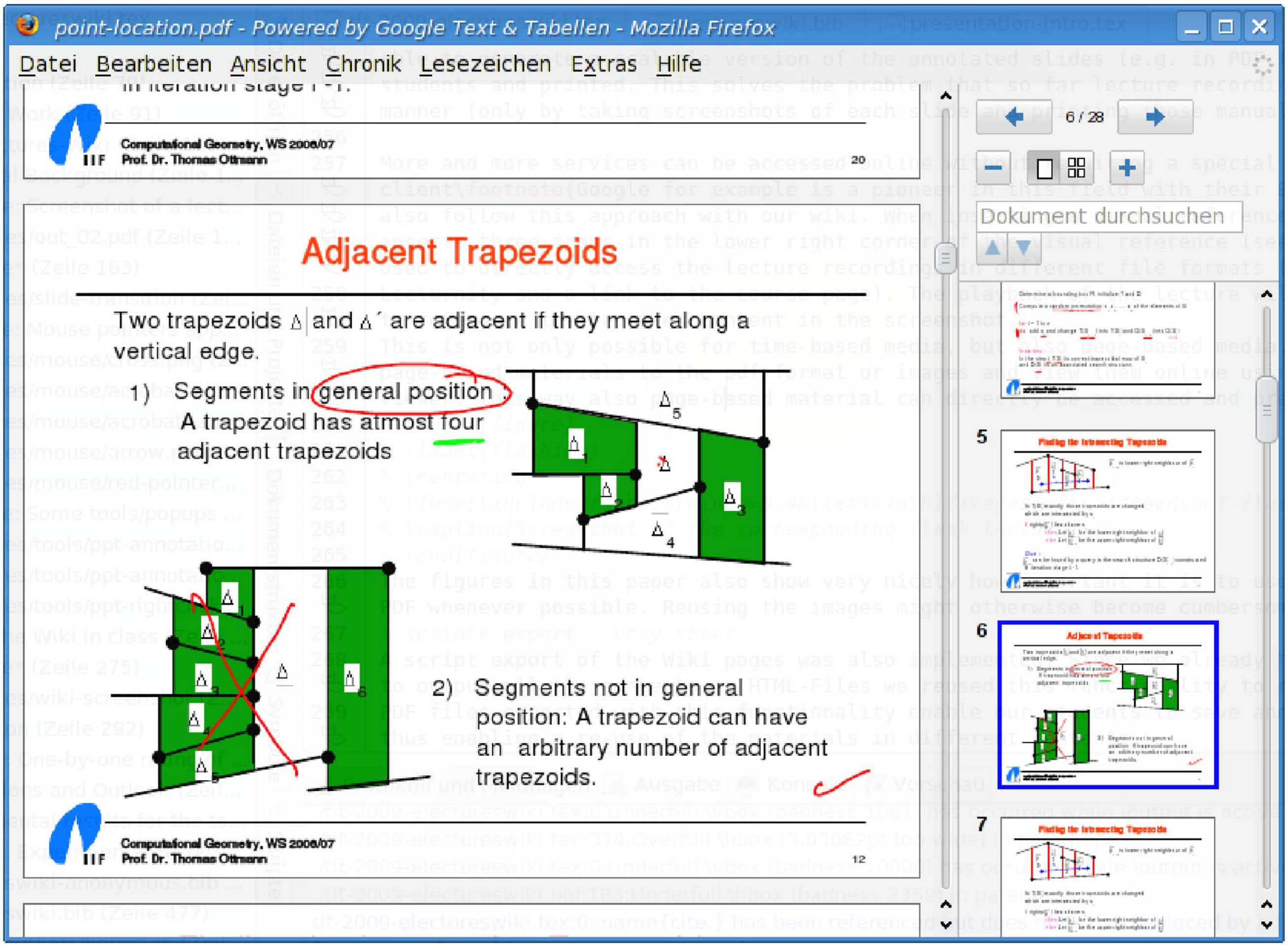
```
== Electures-Wiki ==
The Electures-Wiki allows additionally to the [[http://www.wikicreole.org
/wiki/Creole1.0|Features of a normal Wiki]] the insertion of **Lecture Recordings** and
**complex mathematical formulas** using a [[http://electures.informatik.uni-freiburg.de
/docs/current/user/electures/multipage/ch04s02.html#d0e1439|Latex-Syntax]].

Example of the insertion of a Lecture-Recording:
[{ElecturesRecordingsWikiPlugin size='360x270' timestamp='2821000'
slidenumber='13' type='PAGE_BASED' fileId='9092' electureId='1135' modulId='37'
filename='wikicache/electureslrimage_37_9092_width=360;height=270;
```

(e) Play flash

Fig. 8. Process to insert a screenshot in the Wiki.

Wiki. When inserting a visual reference into the Wiki, it also automatically inserts several icons in the lower right corner of the visual reference (see Fig. 10 on page 11), which can be used to directly access the lecture recordings in different file formats (i.e., Flash, Lecturnity, and a link to the course page). When selecting one of these links, the playback of the lecture will start at the exact time which was used to reference the lecture content in the screenshot.

This is not only possible for time-based media, but also for page-based media. Using our tool *aofconvert*, we can also convert the page-based materials to the PDF format or into a set of images (including annotations) and view them online using an online viewer like the *Google Docs Viewer*. This enables us to directly access and precisely reference page-based material as well (see Fig. 9).


point-location.pdf - Powered by Google Text & Tabellen - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
6 / 28
Dokument durchsuchen
5
6
7
Finding the Intersecting Trapezoids
Adjacent Trapezoids
IIF
Δ1
Δ2
Δ3
Δ4
Δ5
Δ6

Computational Geometry, WS 2006/07
Prof. Dr. Thomas Ottmann
20

Adjacent Trapezoids

Two trapezoids Δ and Δ´ are adjacent if they meet along a vertical edge.

1) Segments in general position : A trapezoid has atmost four adjacent trapezoids

2) Segments not in general position: A trapezoid can have an arbitrary number of adjacent trapezoids.

Computational Geometry, WS 2006/07
Prof. Dr. Thomas Ottmann
12

Fig. 9. Online-viewing of a page-based document.

The figures in this paper also show very nicely how important it is to use scalable file formats such as Flash or PDF whenever possible. Reusing nonscalable images might become cumbersome. Fig. 2 is a scalable vector graphics created with our tool and can perfectly be displayed in arbitrary size. Figs. 3, 4, or 10 are not scalable (screenshots) and contain artifacts resulting from the image compression algorithm.

A script export of the Wiki pages was also implemented. Since we already had a renderer in the Wiki which was able to output all the contents as HTML-Files, we reused this functionality to create a direct PDF export of the pages. PDF files exported with this functionality enable our students to save and print their work for “offline use,” thus enabling a reuse of the materials in different contexts.

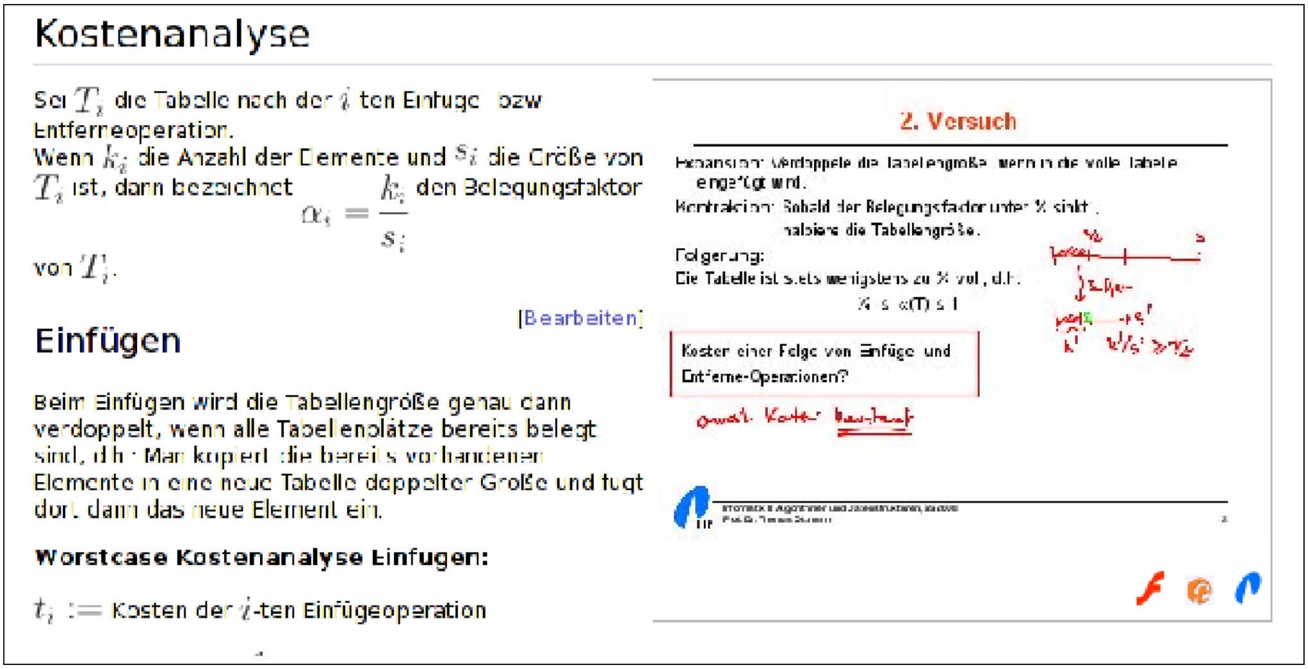
5 Use and Evaluation of the Wiki

Before testing the level of engagement of students using the Wiki, we first wanted to improve the quality of our tool. We, therefore, presented an early prototype of the tool to a group of e-learning experts to get their opinion about it and further enhance our technology. After receiving their feedback and implementing changes, we then decided to find out if our tool would be usable in a course and if so, which of the technical features we implemented were the most important. To this end, we used the Wiki in a computer science course by asking groups of students to collaboratively create Wiki pages on certain topics as described in Section 5.2. We then describe the technical evaluation of our tool in Section 5.3.

5.1 Prototype Poll among E-Learning Experts

Before the Wiki was used by our students in class, we presented the idea of the Wiki as a prototype demo to an audience of e-learning users and professionals. The prototype was presented to the audience (employees from companies working in the e-learning field and also employees from universities) and they were asked five questions and could answer the given choices with a remote control. The admittedly subjective feedback we got from this short live evaluation was very positive ($n \approx 40$):

- Question 1) How did you like the presented idea? very good: 52.40 percent, good: 38.10 percent, satisfactory: 9.50 percent, fair: 0 percent, bad: 0 percent.
- Question 2) Does the presented prototype have practical relevance? yes, unrestricted: 40.90 percent, yes, if certain improvements are made: 50.00 percent, rather pertaining to research: 9.10 percent, no, currently not: 0 percent, no not even in the future: 0 percent.
- Question 3) How do you judge the technical implementation? very good: 8.30 percent, good:



Kostenanalyse

Sei T_i die Tabelle nach der i ten Einfüge bzw Entferneoperation.
Wenn k_i die Anzahl der Elemente und s_i die Größe von T_i ist, dann bezeichnet $\alpha_i = \frac{k_i}{s_i}$ den Belegungsfaktor von T_i.

[Bearbeiten]

Einfügen

Beim Einfügen wird die Tabellengröße genau dann verdoppelt, wenn alle Tabellenplätze bereits belegt sind, d.h.: Man kopiert die bereits vorhandenen Elemente in eine neue Tabelle doppelter Größe und fügt dort dann das neue Element ein.

Worstcase Kostenanalyse Einfügen:

t_i := Kosten der i-ten Einfügeoperation

2. Versuch

Expansion: Verdoppele die Tabellengröße, wenn in die volle Tabelle eingefügt wird.

Kontraktion: Sobald der Belegungsfaktor unter ¼ sinkt, halbiere die Tabellengröße.

Folgerung:
Die Tabelle ist stets wenigstens zu ¼ voll, d.h.
¼ ≤ α(T) ≤ 1

Kosten einer Folge von Einfüge- und Entferne-Operationen?

amort. Kosten konstant

Fig. 10. Screenshot of a part of one of the Wiki pages created by students using visual references.

66.70 percent, satisfactory: 20.80 percent, fair: 4.20 percent, bad: 0 percent.

- Question 4) How do you judge the overall presented functionality? very good: 8.70 percent, good: 65.20 percent, satisfactory: 21.70 percent, fair: 0 percent, bad: 4.30 percent.
- Question 5) Would you consider employing the system at your organisation when it is ready for production use? yes, unrestricted: 28.60 percent, yes, if certain improvements are made: 57.10 percent, no, currently not: 9.50 percent, no since it does not fit our needs: 4.80 percent.

This feedback inspired us to improve the system further and to employ it in a real course in class.

5.2 Use of the Wiki in Class

During the 2008 summer semester, we made our first experiments with our enhanced Wiki in a computer science course on Algorithms and Data Structures.

Seventy seven students were enrolled in this course and apart from the main course (where a professor taught the topics), they were enrolled in tutorials and had to solve exercises by themselves. Those exercises were partly programming exercises, while others had the goal to create Wiki articles about the more theoretical and conceptual parts of the course. We used this second kind of exercises to get feedback on the following points: the technical features of our Wiki, the practicality of the current implementaion, and if we succeeded in creating enabling technology that supports students working more actively with lecture recordings.

The students were split into eight groups of more or less the same size, depending on the complexity of the topic they had to work on in the exercise.

Tutors helped the students in further splitting their group into the various parts of the topics contributing to the overall group article.

The first topics given were “Linear Linked Lists,” “Stacks and Queues,” “Skip Lists,” “Bubble Sort, Selection Sort, and Insertion Sort,” “Heap Sort,” “Merge and Distribution Sort,” “Search Algorithms,” and “Quick Sort.”

The article created by one group of students in the Wiki was then assessed by instructors as follows:

- correctness: 40 percent;
- contents: 30 percent;
- clarity of article: 15 percent;
- style of presentation: 15 percent.

This exercise was repeated with a second list of topics and the same assessment scheme: “Hashing,” “Balanced Trees,” “Amortisation,” “Fibonacci-Heaps,” “Natural Search trees,” “Dynamic tables,” “Graphs,” and “Self-organizing Lists.”

In addition to the topics, the students were given a list of contents to include, for example for the topic “Linear linked lists” this was

1. Clearly define linear linked lists.
2. Describe the linked list data structure; node elements, pointers, accessibility.
3. Explain the insertion and deletion operations and the run-time analyses.
4. Describe the concepts of the double linked list; node elements, pointers, accessibility.
5. Explain the insertion and deletion operation of the double linked list, and the run-time analyses.
6. Provide examples.

The tutors helped the students regarding the use of the Wiki and were able to give further hints on improving the articles.

In the first round, the students were given an incentive to create excellent articles, as each student of the winning group should get a 1 GB USB Stick. The topics of the

TABLE 1
Rating and Ranking Results of the Wiki Functionalities

Functionality	$\mu1$	$m1$	$\mu2$	$m2$
Including pictures in the Wiki	1.66	2	2.18	2
Visually referencing lecture recordings	1.31	2	3.33	3
Syntax highlighting	1.36	1	3.85	4
Including Flash animations	1.14	1	4.44	5
Including Java Applets	1.00	1	4.12	5
Generating a PDF-Script from the Wiki	0.65	1	4.79	5
Creating GIF-like animations	0.68	0	5.18	6

$\mu1$ Arithmetic mean of the *rating* of the functionalities
$m1$ Median of the *rating* of the functionalities
$\mu2$ Arithmetic mean of the *ranking* of the functionalities
$m2$ Median of the *ranking* of the functionalities

second round were a lot more difficult for first-year students than those in the first round and also the incentive for the best contribution was not given anymore. The quality of the created articles differed for both rounds: in the first round the articles were rated with an average grade of 92.01 percent (contents: 26.00 percent, correctness: 40.00 percent, clarity of article: 13.13 percent, style of presentation: 12.88 percent).

The second round still yielded good articles, but not as clear as the first round, with 75.16 percent, on average. This can either be assigned to the missing incentive or to the increased difficulty of the given exercise and should be checked in further experiments.

5.3 Evaluation of Technical Features

In order to evaluate how the students were working with the Wiki and what technical features were the most important, we asked them a set of questions (77 students were asked to fill in the questionnaire and we received 33 full answers).

Most of the questions were Likert-scale questions with five items ranging from $+2$ to -2, including 0, where $+2$ represents a positive rating and -2 a negative one.

From the 33 students who completely filled the questionnaire, 88 percent had German as their first language. We asked this to check whether people with another native language had more problems with the Wiki than others, but that was not the case.

First, we asked the students general questions regarding the use of the Wiki: most of the students found it useful to use a Wiki in class, with an arithmetic mean (μ) of 0.97 and median (m) of 1. The students were indifferent in recommending to reuse the Wiki in future classes: $\mu = 1$, $m = 0$. The pre-partitioning in groups that had to work on different topics in the Wiki was also not rated as especially useful: $\mu = 0.55$, $m = 0$. On the other hand, the given article structure was rated as useful: $\mu = 1.09$, $m = 1$. The help from the tutors does not seem to be as important to the students participating in the survey: $\mu = 0.18$, $m = 0$. Also, the students did not use the Wiki very intensively to prepare for the exams: $\mu = -0.48$, $m = -1$. This is probably due to a wrong design of the integration of the wiki in our course and could be improved by making it a part of the final assessment.

After those introductory questions, we also asked the students some more technical questions and how easy it was for them to use the Wiki. It seems that using the Wiki did not pose the students many problems, only two students answered that it was difficult for them to use it: $\mu = 1.03$, $m = 1$.

We also wanted to know which of the special features we implemented were the most important for the students and how they rated them one by one.

We found that all of the features we implemented (export as a PDF, syntax highlighting for programming languages, insertion of applets and Flash as well as visually referencing lecture recordings) were regarded as useful. Table 1 shows the results of the students rating of the functionalities (columns $\mu1$ and $m1$).

To be able to directly compare the usefulness of the functionalities in the Wiki, we also asked the students to rank the features from 1 (best place, most useful functionality) to 7 (last place, least useful functionality).

This yielded almost the same results (only including Java applets and including Flash switched their positions), see columns $\mu2$ and $m2$ in Table 1.

The additional feedback we got from the students was also very important to us. The comments ranged from "Using the Wiki should be integrated more into the classes" to "the whole thing seems to be completely superfluous." Fortunately, only one of the students stated such a negative opinion regarding the use of our Wiki.

Most of the comments pointed us in directions on how to improve the usefulness of the Wiki. "Better LATEX support is needed," "please include the possibility to directly insert an OpenOffice math formula," or "please make this offer a sustainable one."

6 Conclusions and Outlook

Mostly due to the positive feedback we decided to further improve the original prototype of the Wiki: the main features (like visually referencing lecture recordings and advanced math formulae support) had to be included in the Electures-Portal to be accessible for everyone. The most important features have already been implemented (like the math formulae plugin) and the tool to visually reference lecture recordings (*aofconvert*) is now supporting various filetypes (like PDF, OpenDocument Presentations, PowerPoint, TSCC, and Lecturnity videos and so on).

Regarding the Wiki, the user interface could be improved; especially selecting the material to link to is a tedious process if one has to fill in the values for all eight steps as seen in Fig. 8b. This could be improved by analyzing the current text of the edited section of the Wiki and automatically recommending matching lecture materials to the user.

Most importantly this work has settled the basis for our further work: testing if the learning outcome of students can be improved using our tool. Based on the findings of our experiments and the feedback we got from our students we will further evaluate the use of the Wiki in class with different experimental groups.

Our plans also include to ensure that the Wiki will always be available and can be used by all the teachers at our faculty. We will offer the possibility to create a Wiki for a specific course (where the access permissions can be restricted to the students of the course) and we will also offer one central Wiki that can be used by everyone.

But of course just implementing a lot of different tools does not improve the quality of the lectures. This would also involve training the lecturers to make more use of multimedia as well as didactical training for instructors to avoid the completely passive consumption of lectures toward integrated projects and a more active learning.

Acknowledgments

The authors would like to thank Andreas Janzen for the very first prototype of the Electures-Wiki. We would also like to thank Mike Melanson for his tips regarding the TSCC codec and Jan-Michael Brummer for moral support while implementing the decoding of the color information. We also want to thank Cristina Amparo Hagman for proof reading this article and adding several missing commas and especially Khaireel Mohamed and Tobias Lauer as well as the reviewers for all their valuable suggestions.

References

[1] B. Zupancic and H. Horz, "Lecture Recording and Its Use in a Traditional University Course," *ACM SIGCSE Bull.*, vol. 34, no. 3, pp. 24-28, 2002.

[2] N. Myller, M. Laakso, and A. Korhonen, "Analyzing Engagement Taxonomy in Collaborative Algorithm Visualization," *Proc. 12th Ann. SIGCSE Conf. Innovation and Technology in Computer Science Education (ITiCSE '07)*, pp. 251-255, 2007.

[3] T. Lauer, "When Does Algorithm Visualization Improve Algorithm Learning?—Reviewing and Refining an Evaluation Framework," *Proc. Informatics Education Europe III,* Dec. 2008.

[4] W. Hürst, G. Maass, R. Müller, and T. Ottmann, "The Authoring on the Fly System for Automatic Presentation Recording," *Proc. Extended Abstract Conf. Human Factors in Computing Systems (CHI '01),* pp. 5-6, Apr. 2001.

[5] C. Hermann, W. Hürst, and M. Welte, "The eLecture-Portal: An Advanced Archive for Lecture Recordings," *Proc. Informatics Education Europe,* Oct. 2006.

[6] W. Hürst, "Suche in Aufgezeichneten Vorträgen und Vorlesungen," *Tagungsband der 1. e-Learning Fachtagung Informatik (DeLFI '03),* vol. 37, pp. 27-36, Sept. 2003.

[7] W. Hürst, "Indexing, Searching, and Skimming of Multimedia Documents Containing Recorded Lectures and Live Presentations," *Proc. 11th Ann. ACM Int'l Conf. Multimedia,* pp. 450-451, Nov. 2003.

[8] K.R. Parker and J.T. Chao, "Wiki as a Teaching Tool," *Interdisciplinary J. Knowledge and Learning Objects,* vol. 3, pp. 57-72, 2007.

[9] M.E. O'Neill, "Automated Use of a Wiki for Collaborative Lecture Notes," *Proc. 36th SIGCSE Technical Symp. Computer Science Education,* vol. 37, no. 1, pp. 267-271, Feb. 2005.

[10] M. Kam, J. Wang, A. Iles, E. Tse, J. Chiu, D. Glaser, O. Tarshish, and J. Canny, "Livenotes: A System for Cooperative and Augmented Note-Taking in Lectures," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI '05),* pp. 531-540, 2005.

[11] H. Sack and J. Waitelonis, "Zeitbezogene Kollaborative Annotation zur Verbesserung der Inhaltsbasierten Videosuche," *Proc. Good Tags and Bad Tags—Workshop Social Tagging in der Wissensorganisation,* pp. 107-118, 2008.

[12] T. Lauer and S. Trahasch, "Scripted Anchored Discussion of Multimedia Lecture Recordings," *Proc. Scripting Computer-Supported Collaborative Learning,* Feb. 2007.

[13] M. Ketterl, R. Mertens, and K. Morisse, "Alternative Content Distribution Channels for Mobile Devices," *Proc. Int'l Conf. Micromedia and eLearning 2.0: Getting the Big Picture,* pp. 119-130, June 2006.

[14] A. Bruns and S. Humphreys, "Wikis in Teaching and Assessment: The M/Cyclopedia Project," *Proc. Int'l Symp. Wikis (WikiSym '05),* pp. 25-32, 2005.

[15] B.D. Honegger, "Wikis: A Rapidly Growing Phenomenon in the German-Speaking School Community," *Proc. Int'l Symp. Wikis (WikiSym '05),* pp. 113-116, 2005.

[16] A. Klampfer, "Wikis in der Schule—Eine Analyse der Potentiale im Lehr-/Lernprozess," bachelor's thesis, FernUniversität in Hagen, http://teaching.eduhi.at/alfredklampfer/bachelor-wikis-schule.pdf, July 2005.

[17] T. Bartel, "Nutzung von Wikis in Unternehmen—Kollaboratives Arbeiten mit Wikis im Unternehmensumfeld," *Proc. Mit Wissensmanagement besser im Wettbewerb (KnowTech '06),* pp. 161-168, 2006.

[18] A. Majchrzak, C. Wagner, and D. Yates, "Corporate Wiki Users: Results of a Survey," *Proc. Int'l Symp. Wikis (WikiSym '06),* pp. 99-104, 2006.

[19] M. Notary, "How to Use a Wiki in Education: Wiki Based Effective Constructive Learning," *Proc. Int'l Symp. Wikis (WikiSym '06),* pp. 131-132, 2006.

[20] P. Dillenbourg, "What Do You Mean by Collaborative Learning?," *Collaborative-Learning: Cognitive and Computational Approaches,* P. Dillenbourg, ed., pp. 1-19, Elsevier, 1999.

[21] S. Ainsworth and A.T. Loizou, "The Effects of Self-Explaining when Learning with Text or Diagrams," *Cognitive Science: A Multidisciplinary J.,* vol. 27, pp. 669-681, 2003.

[22] I. Koprinska and S. Carrato, "Temporal Video Segmentation: A Survey," *Proc. Signal Processing: Image Comm.,* pp. 477-500, 2001.

[23] M. Welte, T. Eschbach, and B. Becker, "Automated Text Extraction and Indexing of Video Presentation Recordings for Keyword Search via a Web Interface," *Proc. World Conf. E-Learning in Corporate, Govt., Healthcare, and Higher Education,* pp. 3200-3205, Oct. 2005.

[24] S. Kopf, F. Lampi, T. King, M. Probst, and W. Effelsberg, "EDL-Editor: Eine Anwendung zur Automatischen Aufbereitung von Vorlesungsvideos," *Proc. e-Learning Fachtagung Informatik (DeLFI '07),* pp. 33-44, 2007.

[25] P. Ziewer, M. Ebner, C. Safran, and W. Slany, "Searching for Classes of Visual Content in Electronic Lectures," *Proc. IEEE Ninth Int'l Symp. Multimedia Workshops (ISMW '07),* pp. 377-382, 2007.

[26] W. Hürst and R. Müller, "A Synchronization Model for Recorded Presentations and Its Relevance for Information Retrieval," *Proc. Seventh ACM Int'l Conf. Multimedia (MULTIMEDIA '99),* pp. 333-342, 1999.

[27] A. Noé, "Avi File Format," http://www.alexander-noe.com/video/documentation/avi.pdf, Jan. 2006.

[28] Wikipedia, "Run-Length Encoding," http://en.wikipedia.org/w/index.php?title=Run-length_encoding&oldid=328008157, Nov. 2009.

Christoph Hermann graduated in business informatics from the Technical University of Clausthal-Zellerfeld, Germany, in 2005. He is a researcher at the University of Freiburg and project leader for the master online study programme "Intelligent Embedded Microsystems." His research interests are e-learning, search in lecture recordings, recommender systems for lecture archives and improving teaching with multimedia in general.

Thomas Ottmann studied mathematics, physics, and mathematical logic at the University of Münster. There he received the PhD (Dr rer nat) degree in mathematical logic in 1971. In 1975, he obtained the Facultas Docendi in informatics from the University of Karlsruhe. From 1976 until 1987, he was a professor for computer science at the University of Karlsruhe. In 1987, he became the founder of the Department for Computer Science at the University of Freiburg where he served until his retirement in 2008. During his academic career he had guest positions at many universities all over the world (University of Waterloo (Canada), Dartmouth College (New Hampshire), ETH Zürich (Switzerland), University of Western Australia, Perth). Since 2007, he is the head of the technical committee for Informatics of the German Accreditation agency ASIIN. He is the coauthor and editor of seven books, author and coauthor of more than 150 papers in scientific journals and conference proceedings. His research interests include algorithms and data structures, computational geometry, multimedia systems, and the use of computers for educational purposes. He is a member of the IEEE Computer Society.