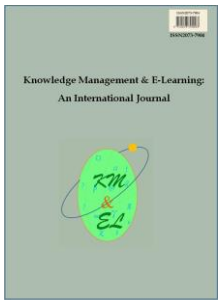
Knowledge Management & E-Learning



ISSN 2073-7904

Automatic annotation of lecture videos for multimedia driven pedagogical platforms

Ali Shariq Imran
Faouzi Alaya Cheikh
Stewart James Kowalski
Gjøvik University College, Norway

Recommended citation:
Imran, A. S., Cheikh, F. A., & Kowalski, S. J. (2016). Automatic annotation of lecture videos for multimedia driven pedagogical platforms. *Knowledge Management & E-Learning*, *8*(4), 550–580.

Knowledge Management & E-Learning, 8(4), 550–580

Automatic annotation of lecture videos for multimedia driven pedagogical platforms

Ali Shariq Imran*

Faculty of Computer Science and Media Technology
Gjøvik University College, Norway
E-mail: ali.imran@hig.no

Faouzi Alaya Cheikh

Faculty of Computer Science and Media Technology
Gjøvik University College, Norway
E-mail: faouzi.cheikh@hig.no

Stewart James Kowalski

Faculty of Computer Science and Media Technology
Gjøvik University College, Norway
E-mail: stewart.kowalski@hig.no

*Corresponding author

Abstract: Today's eLearning websites are heavily loaded with multimedia contents, which are often unstructured, unedited, unsynchronized, and lack inter-links among different multimedia components. Hyperlinking different media modality may provide a solution for quick navigation and easy retrieval of pedagogical content in media driven eLearning websites. In addition, finding meta-data information to describe and annotate media content in eLearning platforms is challenging, laborious, prone to errors, and time-consuming task. Thus annotations for multimedia especially of lecture videos became an important part of video learning objects. To address this issue, this paper proposes three major contributions namely, automated video annotation, the 3-Dimensional (3D) tag clouds, and the hyper interactive presenter (HIP) eLearning platform. Combining existing state-of-the-art SIFT together with tag cloud, a novel approach for automatic lecture video annotation for the HIP is proposed. New video annotations are implemented automatically providing the needed random access in lecture videos within the platform, and a 3D tag cloud is proposed as a new way of user interaction mechanism. A preliminary study of the usefulness of the system has been carried out, and the initial results suggest that 70% of the students opted for using HIP as their preferred eLearning platform at Gjøvik University College (GUC).

Keywords: Multimedia/Hypermedia systems; Intelligent tutoring systems; Media in education; Interactive learning environment

Biographical notes: Dr. Ali Shariq Imran obtained his Ph.D. from University of Oslo (UiO), Norway in computer science and a Masters in Software Engineering and Computing from National University of Science & Technology (NUST), Pakistan. His current research interests lie in the areas of image and video processing, semantic web, eLearning, and online social

network (OSN) analysis. He is currently associated with Faculty of Computer Science and Media Technology at Gjøvik University College (GUC) as an associate professor. Dr. Ali Shariq Imran is a technical committee member and an expert reviewer to a number of scientific journals and conferences related to the field of his research. He is also a member of IEEE, Norway section and has co-authored more than 35 papers in international journal and conferences.

Dr. Alaya Cheikh, received his Ph.D. in Information Technology from Tampere Univ. of Technology, in Tampere, Finland in April 2004; where he worked as a researcher in the Signal Processing Algorithm Group since 1994. From 2006, he has been affiliated with the Department of Computer Science and Media Technology at Gjovik University College in Norway, at the rank of Associate Professor. He teaches post-graduate level courses on image and video processing and analysis and media security. His research interests include e-Learning, 3D imaging, image and video processing and analysis, video-based navigation, Video Surveillance, biometrics, pattern recognition and content-based image retrieval. In these areas, he has published over 80 peer-reviewed journal and conference papers, and supervised PhD and MSc thesis projects.

Dr. Alaya Cheikh is currently the co-supervisor of four PhD students. He has been involved in several European and national projects among them: ESPRIT, NOBLESS, COST 211Quat, HyPerCept and IQ-MED. He is a member of: the steering committee of the European Workshop on Visual Information Processing (EUVIP), the editorial board of the IET Image Processing Journal and the editorial board of the Journal of Advanced Robotics & Automation and the technical committees of several international conferences. Dr. Alaya Cheikh is an expert reviewer to a number of scientific journals and conferences related to the field of his research. He is a senior member of IEEE, member of NOBIM and Forskerforbundet (The Norwegian Association of Researchers - NAR).

1. Introduction

For some years now, students study using distance learning. This concept was not always easy, as the only method of communication between the students and the university originally was by mails. In the last decade, with the world shifting towards what is considered now, digital age, the concept of distance learning took a different form. With the evolution of the Internet, distance learning evolved and became more accessible, introducing a new concept called eLearning. New means of communication and studying were introduced and new frameworks were created in order to simplify the whole process (Imran & Cheikh, 2012).

Simultaneously, numerous eLearning platforms, educational tools, learning management systems (LMS), and open educational video resources have emerged in last decade with rapid development in eLearning technology. These include *Fronter* (http://www.com.fronter.info), *ATutor* (http://atutor.ca), *Moodle* (http://moodle.org), *Khan academy* (http://www.khanacademy.org), *Coursea* (http://www.coursera.org), *edX* (http://www.edx.org) etc. These eLearning platforms and tools provide useful mechanism of delivering educational resources for distance and blended education. The resources normally comprise of recorded lecture videos, presentation slides, audio transcripts, and related documents. They are stored on a server or in a learning object repository (LOR) such as MERLOT (http://www.merlot.org), either centrally located or distributed.

Learning objectives are defined and meta-data is associated with these resources before they are distributed to masses as learning objects (LO) (Northrup, 2007), via eLearning platforms.

The purpose of these instructional websites is to provide as much information as possible to students, in order to help them during their classes and exams. This made today's eLearning websites rely heavily on lecture videos, including accompanying material such as lecture notes, presentation slides, audio transcripts, quizzes etc., and thus these websites became heavily loaded with multimedia contents. However, the choice of multimedia alone can't help improve the learning process. Theories like learning styles should also be taken into consideration. Learning styles is a theory developed based on the fact that the ability of every individual to process information differs during the learning process. In other words, every individual learns in a different way (Tuan, 2011). Although studies showed no concrete evidence that learning styles can improve the knowledge acquisition process of students in classroom environment, learning theories nevertheless remained significant and resulted in different models in order to categorize learning style (De Bello, 1990).

Over the years, more studies were conducted in order to see if the learning styles affect the quality of learning through eLearning platforms and if there is any difference between the way of learning through the classroom and the eLearning platforms. The findings of studies like the one performed by Manochehr (2006), showed that learning styles although they are irrelevant when the students are in a classroom, they had statistically significant value according to the knowledge performance in a web-based eLearning environment.

There are possibly many ways to transfer knowledge to individuals based on their learning style (Felder & Silverman, 1998). The success of a learning process is depended upon two factors: users' learning style or preferences and the way knowledge is presented to the user. Fleming's VARK model (Fleming, 2014) has grouped learners into four categories: visual, audio, read/write and kinesthetic. To aid the learning process, we need to deliver the educational resources adhering to users' preferences based on their learning style (Franzoni, Assar, Defude, & Rojas, 2008). Existing eLearning platforms mostly rely only on lecture videos, which tend to be targeted better suited for visual learners. Additionally, lecture videos are often quite large, lack interactivity, and are normally non-structured. This makes it difficult for the learners to keep their interest level high. For that, video annotations become necessary and they cannot be considered optional any more. Lecture videos available online mostly lack the necessary supporting information and meta-data. This makes it extremely difficult for the interested students to easily and rapidly find relevant information. Having this in mind, another problem arises. Finding meta-data information to represent hyperlinks in order to connect different components available in an eLearning platform is a challenging, laborious, and time-consuming task. To address these problems, in this paper, we propose the use of an automated video annotation method, the 3D tag cloud, and a HIP eLearning platform utilizing video annotations and the 3D tag cloud presented in this paper.

Tags are words that are weighted by factors such as frequency, time, appearance, etc., depending on the content they are used in. Usually the importance of a tag in a tag cloud is specified by its font size or the color of the word. For instance, the bigger the font size of a word the more important it is in a given context. According to research made by Halvey and Keane (2007), a number of interesting observations were made concerning tag representation. Firstly, it was found that alphabetization aided users to find the information they were interested in, easier and faster. The font size and the

position of the tags were also found to be very important factors in the information finding process. Finally, it was found that users usually scan through the lists or clouds instead of reading them thoroughly.

As tag clouds became very popular more studies were conducted in order to evaluate their effectiveness. An example is the study conducted by Rivadeneira, Gruen, Muller, and Millen (2007), in which differently constructed tag clouds were evaluated. It was stated that tag clouds can assist in navigation as table of content and they can provide a way to get a first impression of the content presented in the current paper, the book or the website.

The proposed 3D tag clouds are used for random access of and navigation through multimedia rich educational material, by automatically extracting candidate keywords from presentation slides and lecture videos. A tag cloud is used to navigate through a set of presentation slides and its associated lecture video. While the video annotation method is to link the presentation slides and the lecture videos, and a HIP platform is to test the presented methods.

The rest of the paper is organized as follows. In section 2, we present the HIP platform that uses both lecture videos and presentation slides to present educational content, in a structured and synchronized manner. Section 3 describes the proposed annotation methods for content-based linking of presentation slides to lecture videos. Section 4 presents the accuracy results of the proposed content-based linking approach. In section 5, we show how the proposed method can be used to create annotations and 3D tag clouds for eLearning platforms. Section 6 presents usability evaluation results of the proposed HIP system while section 7 concludes our paper.

2. Hyper interactive presenter

HIP is an eLearning platform that provides technology-rich pedagogical media for continuous education and connected learning (Imran & Kowalski, 2014). It brings together different types of media elements to deliver the learning objects (LOs). These include text documents such as wiki pages and PDF documents, presentation slides, lecture videos, an intelligent pedagogical agent along with navigational links, tagged keywords, and frequently asked questions (FAQ). HIP supports nano-learning (Masie, 2005) by creating smaller chunks of video learning objects (VLOs), and hyperlinking similar LOs across different media.

HIP comprises of many media elements, which are assembled in different components, and are bundled (*interlinked*) together to form a HIP web page. These components are designed to support different types of learning styles. Fig. 1 shows an example of a HIP page layout with different components.

2.1. HIP components

HIP comprises of five main components: a) hyper-video, b) slide viewer, c) PDF/ Wiki page, d) frequently asked questions (FAQ), and e) a pedagogical agent. The components are designed to present the knowledge in many ways by utilizing all the available media modalities.

a. ***Hyper-video***: A video of the presented subject. It focuses mainly on the visual and auditory learners.

b. ***Slide viewer****:* It contains images of presented slides that were used during the lecture. It mainly focuses on visual learners.

c. ***PDF/Wiki page***: A page containing information relevant to presented subject. It is intended towards read/write learners.

d. ***Frequently Asked Questions*** *(FAQ):* Focuses on the creation of a conversational agent to provide to the users a way to ask questions.

e. ***Pedagogical agent*** *(chat bot):* Intended for a variety of different learning styles. It can benefit learners that like to write and read, and also auditory learners that learn better through discussion.

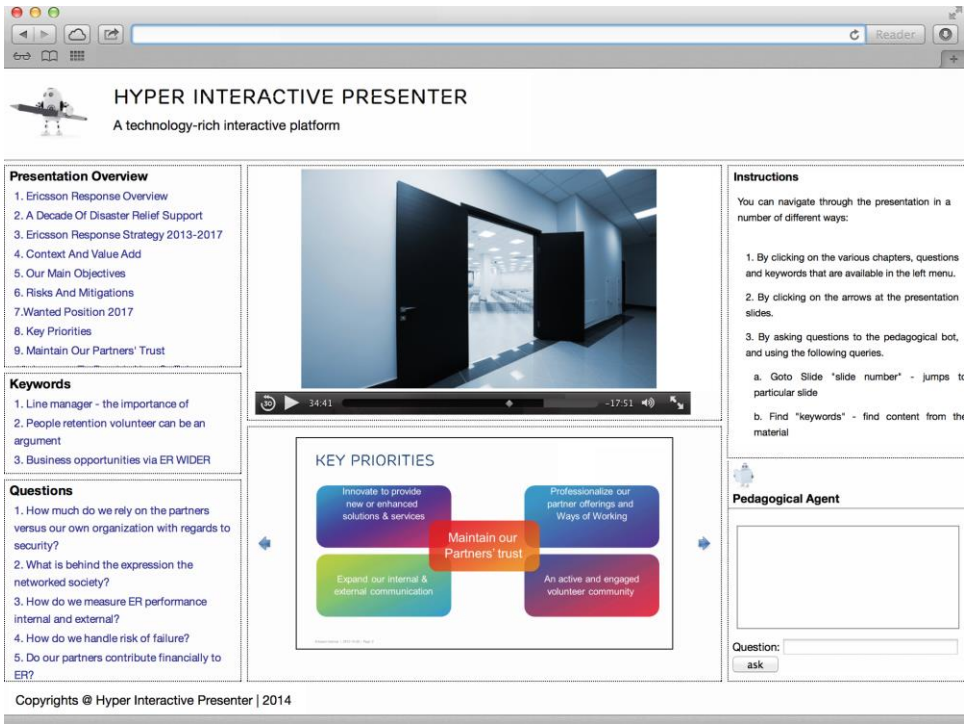


Fig. 1. A sample screen shot of hyper interactive presenter (HIP)

HIP use these different media components to map the VARK model, in order to support variety of learning styles. It is a well-established fact that about approximately 65% of the population is visual learners while others are textual learners (Jonassen, Carr, & Yueh, 1998). Additionally, 90% of information that comes to the brain is visual (Hyerle, 2009). HIP therefore, supports different learning styles by combining visual information with the text and by providing users with an intelligent pedagogical chat bot to engage them in discussions. The pedagogical chat bot not only provides a two-way communication but also keeps user's interest level high. This is achieved by interlinking different media items together.

Let's look at the functionality of the two HIP components that are important for automatic content-based linking (synchronization) of lecture videos and presentation slides.

2.1.1. Hyper-video

The Recorded lecture videos are used as an educational resource to primarily assist visual and auditory learners. Lecture videos are usually rather long. A lecture video can often last for one to two hours and it can contain large amounts of pedagogical content covering one or more subjects. For these reasons lecture videos are very content rich media with high complexity. Even though numerous lecture videos are available on the web, most of the time they lack the necessary supporting information and metadata; they are usually unstructured, unedited, and non-scripted. This makes it extremely difficult for the interested student to find relevant information and reduces dramatically their pedagogical value. Taking bandwidth limitations into account the process becomes even more challenging.

Contrary to the existing eLearning platforms, HIP provides a hyper-video; segmented, structured and edited VLO, based on the concept of nano-learning (Masie, 2005). A lecture video undergoes a series of processing steps to identify the areas of interest (AOI). An AOI could be a start of a question, a new topic, or a pause during a lecture etc. The identified AOIs are used as index points to create a smaller segment of a video called VLO from the full-length instructional video. The index points are also used to create hyperlinks to jump to particular timestamps in the video for quick and nonlinear navigation.

2.1.2. Slide viewer

The second main component that defines HIP consists of presentation slides viewer. The use of slides caters to visual as well as textual learners. Presentation slides are processed independently to create images of the slides that are presented in the HIP slides viewer. The images are synchronized with corresponding lecture video based on the content present in video as well as in lecture slides, as explained in section 3.3. A ‘presentation overview’, ‘keywords’, ‘questions’, as can be seen in Fig. 1, all provide navigational links to jump directly to a desired slide and to corresponding timestamp in a video.

2.2. Content interaction

The HIP provides multi-way interaction between presentation slides, lecture videos, 3D word cloud, PDF/Wiki page and a pedagogical chat bot. For instance, if someone browses a presentation slide, the video automatically jumps to start of a segment in video that contains a particular slide and vice versa. At the same time, corresponding content from the PDF document or a Wiki page would appear in the document section. If it were a wiki document, the page containing the corresponding information would appear. Similarly, the presentation outline, extracted keywords and/or key phrases along with FAQ are all linked to their corresponding VLOs, presentation slides, and to accompanying documents/wiki pages.

For example, if someone clicks on a keyword about ‘eLearning’, appropriate video segment would appear which talks about the given topic i.e. eLearning in this case, and the corresponding slide would appear that was used during the talk, along with wiki page containing information about eLearning. In addition, it is possible to query the system via pedagogical agent to navigate to a particular topic simultaneously across different media.

2.3. Limitations

The process of annotating and synchronizing lecture videos and presentation slides requires a lot of manual processing. A variety of tools such as *Share stream*, *Kaltura*, *VIDIZMO* etc., are available in order to help with the annotation process. Synchronization can be added between the video and other supporting surrogate media items, and thus the pedagogical material information can be presented in different ways to learners depending on their preferred learning styles.

Even though these tools provide video annotations to some extent, there are few drawbacks:

- They require a lot of manual labor work in order to link content between videos and presentation slides.
- They do not provide support for PDF documents or Wiki pages.
- The available tools are not simple, and require a certain amount of experience and expertise in order to use them.

Therefore, we propose in this paper an approach to create automatic video annotations and interaction techniques by developing a framework for automatic feature extraction, annotation and user interaction with the lecture video and other supporting surrogates (*presentation slides, frequently asked questions (FAQ), presentation overview, keywords, PDF/Wiki page, 3D tag clouds, and pedagogical chat bot*).

3. Proposed method

The present work attempts to propose a novel approach to automatically create annotations for lecture videos to support a variety of eLearning platforms utilizing lecture videos and presentation slides – such as HIP. To our knowledge there are no previous studies on the use of intelligent pedagogical chat bot, automatic video annotations for 3D tag clouds, and the interaction between the media items. The main goal of this project is to automatically create lecture video annotations and to propose a new interaction and navigation tool i.e. the 3D tag cloud.

The proposed algorithm, according to the steps taken for its implementation, has been divided in three parts. The first part is about key-frame extraction, which involves shot detection, slide region detection and key frame selection. The second part is about presentation slides processing and the third and final part is about synchronization of the video and the presentation slides. Further details about these steps are described in the following subsections.

3.1. Key-frame extraction

The purpose of key-frame extraction is to automatically extract the best key-frame that could be used for synchronization purpose. Key-frame extraction consists of three sub-tasks. The first sub-task is to find the shots in a video, the second sub-task is to detect the slide region in each frame within a shot, and the third sub-task is to find the best key-frame. The sub-tasks are discussed in the following subsections.

3.1.1. Shot detection

For a given lecture video, a shot constitutes those frames in a video that have similar content visible in each frame. A change in the slide therefore means a start of a new shot and an end to a previous one. Different state-of-the-art shot boundary detection techniques including sum of absolute difference of histograms were examined (Wang, Kitayama, Lee, & Sumiya, 2009; Huang, Li, & Yao, 2008; Fan, Barnard, Amir, & Efrat, 2011), and a test was carried out in order to check their performances. It was found that the implementation based on SIFT (Lowe, 2004) provided much more accurate results in finding the shot boundaries. Using SIFT consecutive frames in a video were matched and by using a simple threshold shot boundaries were identified.

3.1.2. Slide region detection

The next step in the video-processing steps is the detection of the slide region in the video frames. To achieve this, the frames in a shot are read individually. For the detection of the slide region a number of image processing techniques are applied on every frame in each shot, where each frame is processed individually.

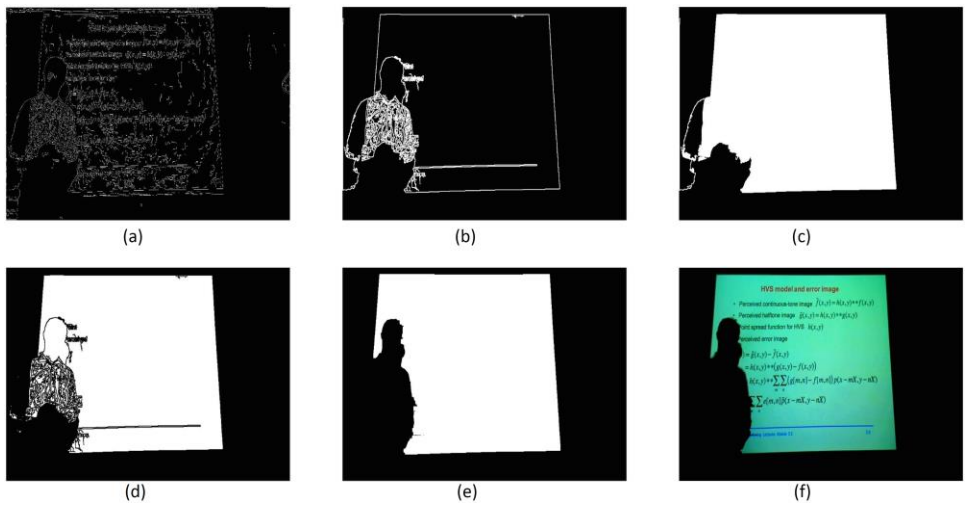


Fig. 2. Different steps of the slide region detection algorithm: (a) Canny Edge Detection, (b) "fill" the holes and select the biggest connected component, (c) "fill" the Component, (d) subtract images (b) and (c), (e) slide region selected in image and (f) the original Pixels are filtered

a) The first step is edge detection. Edges in the image are found using the Canny edge detection algorithm as shown in Fig. 2(a). It is necessary to apply this step in order to get a good approximation of where the slide is in the frame. By finding the edges only the contour of the different objects remains in the new image. This image is a binary image with every pixel containing the values of either '0' or '1', whether there is an edge or not in the specific pixel.

b) After obtaining the binary image the algorithm dilates the image in order to make sure that all the major shapes in the image are connected as shown in Fig. 2(b). Then the inside of the shapes (connected components) are "filled". This means that all the connected components in the image are filled with the value '1', as shown in Fig. 2(c).

c) From the resulting image the biggest connected component is selected (the connected component containing the maximum number of pixels). In a lecture video the biggest area is assumed to be the projected slide area. This is enough to find where the slides are in the presentation video. After selecting the biggest component again, it is filled with '1'. The new resulting binary image is an image containing '0' (black pixels) except at the potential position of the slide region.

This is enough to get a good estimation of the slide region if it were known beforehand that the slide region is completely uncovered and there are no obstacles moving in front of the projector screen. The slides are however, often occluded by the presenter, thus further processing is required to get an accurate estimation of slide region and to remove objects that can occlude part of the presentation slides in the video.

d) To take this into account, the two binary images mentioned before (the image of the contour of the biggest connected component and the image of the biggest connected component filled with '1'), shown in Fig. 2(b) and 2(c), are used. These two images are subtracted and a new image, illustrated in Fig. 2(d), is created that contains only the visible slide region in the frame and smaller areas that were mistakenly included earlier within the slide region are excluded.

e) From the resulting Fig. 2(d), the new biggest connected component is selected. Finally, the image is again dilated in order to compensate for information that might have been lost during the subtraction operation. The result is shown in Fig. 2(e).

f) Using only the slide region from Fig. 2(e) as a mask, a new image is created which contains the pixels from the original image in the slide region area as shown in Fig. 2(f). Results of the different steps of the algorithm are presented in Fig. 2.

3.1.3. Key-frame detection

The next step is to actually select the key-frames. A key-frame in this case is defined as a video frame that contains the maximum visible slide region having biggest amount of text information i.e. a frame with most information not occluded by any external objects.

That said a simple technique of counting the number of non-zero pixels in the image from the result obtained in Fig. 2(e) could be proposed. This method is not very efficient and does not always provide the best key-frame, as it doesn't take into account the actual text present in the slide region. Other state-of-the-art techniques proposing use of Hough transform (Wang, Kitayama, Lee, & Sumiya, 2009; Huang, Li, & Yao, 2008; Wang, Ramanathan, & Kankanhalli, 2009) and background modeling (Fan, Barnard, Amir, & Efrat, 2011; Ngo, Pong, & Huang, 2002; Ngo, Wang, & Pong, 2003) were also examined and a new approach is proposed using the energy percentage of the 2D wavelet decomposition (Daubechies, 1992; Mallat, 1989; Meyer, 1990) as the main factor for deciding which is the most appropriate frame in a shot to be used as a key-frame.

Wavelet energy (WE) is a method of finding energy in a signal for 1-D wavelet decomposition. The WE provides percentage of energy corresponding to the approximation and the vector containing the percentage of energy corresponding details (Thampi, Abraham, Pal, & Rodriguez, 2013). It is computed as follows:

$$WE = \frac{1}{2^{-m/2}} \sum_{i=1}^{N} S(I, i) \emptyset[2^{-m}(i - n)]$$

Where *S* is the signal and Ø is the basis function.

Results obtained in Fig. 2(f) are used to compute the wavelet energy percentage. The algorithm starts by creating two-dimensional wavelet decomposition of every image (only the images containing the slide region are used) in the shot. From the wavelet decomposition the wavelet energy percentage is calculated from the approximation coefficients and the percentages of energy corresponding to the horizontal, vertical, and diagonal details.

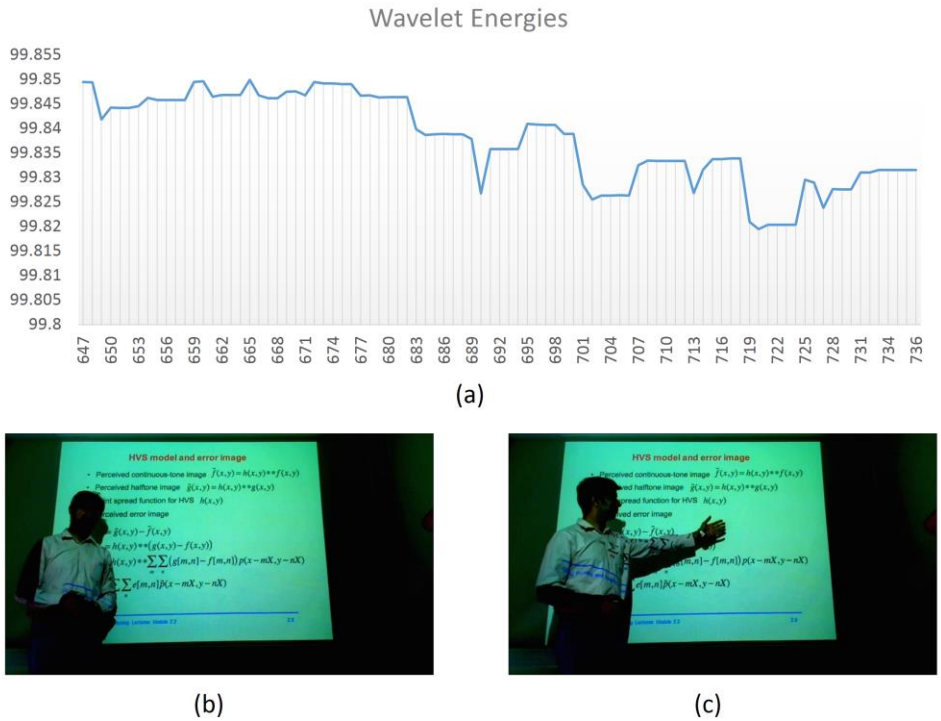


Fig. 3. (a) Wavelet Energy Percentage in graph. (b) Image with the highest energy is selected, (c) and the images with the lowest energy such as frame 720 are discarded

Finally, a graph like the one shown in Fig. 3(a) can be created for every shot. From the graph it is distinguishable which are frames with highest and lowest wavelet energy percentages.

In Fig. 3(b) and 3(c) it is shown which frames have highest and lowest wavelet energy percentage respectively. As it can be observed the image with the highest wavelet energy exhibits more information than the image with the lowest energy. Thus this image is the one selected as a key-frame for the current shot. The shot presented in the Fig. 3, was randomly selected from the set of more complicated shots in the video (shots that have occlusion of the slide content by the professor).

3.2. Slide processing

The second part of the algorithm, process the presentation slides to extract images and text for the synchronization process and for creating the video annotation. We use the MS office library in Matlab to extract the images, retrieve text from the PowerPoint slides

and other meta-data information such as font size, font type, bold face features, etc. The extracted information is recorded in various files as shown in Table 1.

Table 1
The meta-data information contained in the generated output files

File Type	Contents
Video Frames	Contains all the frame of the video
Slide Images	Contains image of every slide
Slide Text	a. Individual text files containing extracted text from individual slide. b. Complete text files containing extracted text from full presentation slides.
Word Information	Information about the font size and bold feature of extracted words.
Slide Video Sync Information	XML file containing slide-video synchronization information
Slide Duration	Text file containing the duration of each slide appearing in the video.
Chapter	WebVTT file containing slide-video synchronization time.

3.3. Video and slide synchronization

After obtaining the key-frames from the lecture videos and the images from the presentation slides, the synchronization process is carried out. For the synchronization process, it is proposed by a variety of implementations found in the literature, that a similar approach to the shot detection algorithm can be used using SIFT (Wang, Kitayama, Lee, & Sumiya, 2009; Fan, Barnard, Amir, & Efrat, 2011; Wang, Ramanathan, & Kankanhalli, 2008; Fan, Barnard, Amir, & Efrat, 2009). In this work, we have used the state-of-the-art SIFT algorithm to find the feature points for synchronization. Feature points on both the presentation slides images and the key-frames are compared and matched in order to find the images with most similarities and thus synchronize the video shots with the presentation slides, as depicted in Fig. 4.

Following information is extracted for synchronization. For each slide the starting point (starting time) and ending point (ending time) in the video are found. Next, the presentation time of each of the slides in the lecture video is calculated. To do this, the first and the last frames of each of the detected shots are found. The starting and ending time of each shot in the video is calculated using the starting and ending frames together with the frame rate of the video.

In addition, the time information needs to be in specific format for synchronization purpose and in order for different components to work correctly. Time information needs to be provided firstly in the form of seconds and secondly in time cues format, as specified in the WebVTT documentation and specifications for the creation of

recognizable time cues (Pfeiffer, Jägenstedt, & Hickson, 2014). A time cue has to have the format of hh:mm:s.sss, where hh for hours, mm for minutes and s.sss for milliseconds.

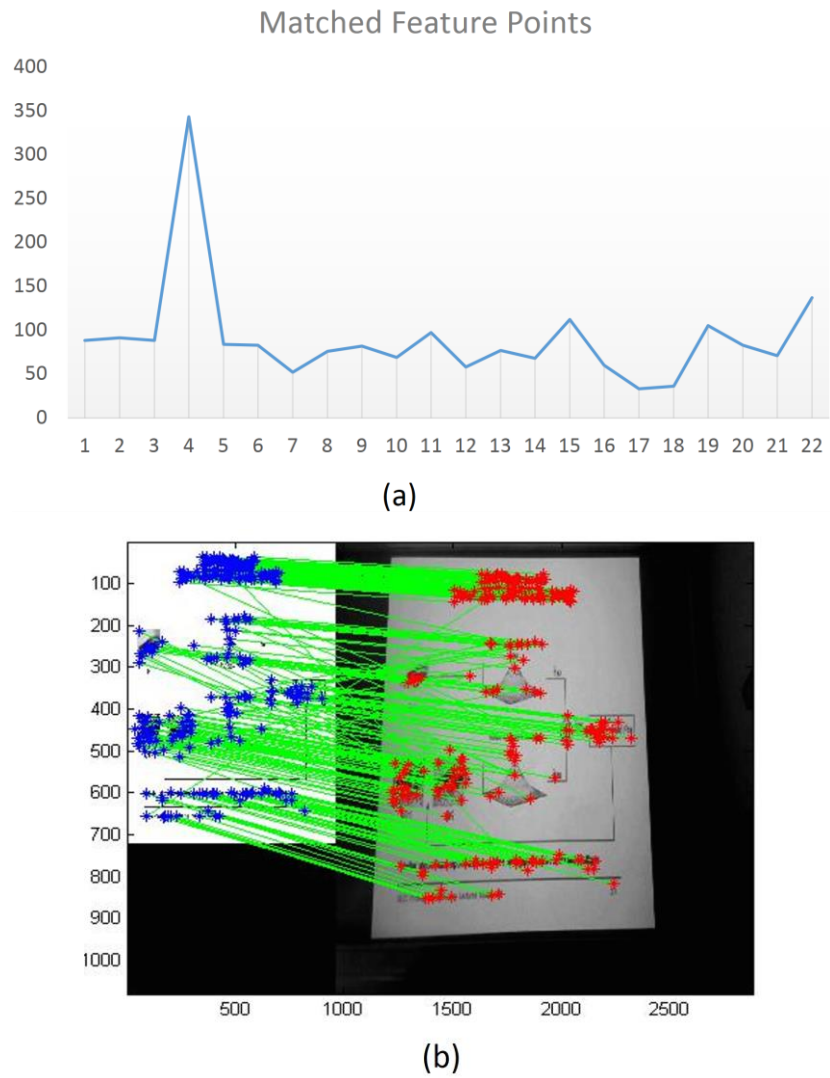


Fig. 4. SIFT for video and slide synchronization. (a) Graph showing total number of matched feature points between presentation slides and key-frame, (b) Feature points matching on presentation slides and key-frames

The algorithm generates several different files with information including meta-data for the annotations and the interaction mechanisms. The details of these different kinds of output files can be seen in Table 1. The first output is the set of video frames. Having the frames can be useful in case of visualizing some information about the video, such as the key-frames. The second output is the Slide Images. Images of every slide in the presentation are saved and are used for creating web slide shows or slide presentations in the slides viewer component of HIP. Together with the video frames, the slide images are used for the synchronization of the video shots and the slides. The third

and fourth outputs are a number of text files. First a text file containing all the text available in the presentation is created and then individual text files for every slide are created. Finally, a text file is created containing every word in the presentation along with its information about font size and bold face features. All this information from files is used in the creation of a variety of annotations and interaction mechanisms, described in Section 5.

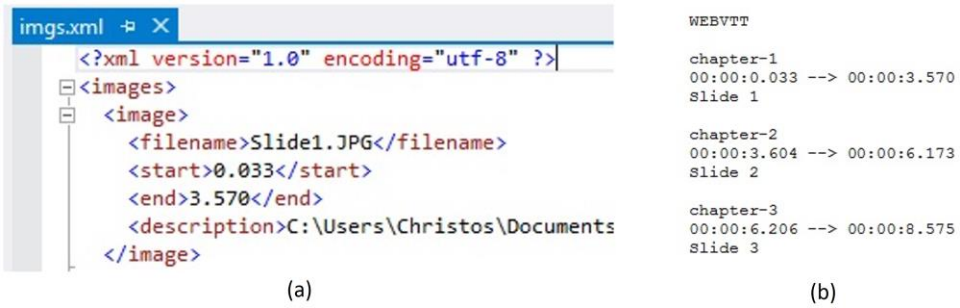


Fig. 5. Synchronization Output Structure in (a) XML File and (b) WebVTT File

The last three outputs files contain timing information. They are used in the video for slide synchronization to provide random access to the video segments. The structure of extensible markup language (XML) file and the WebVTT file are important. As shown in Fig. 5, the XML file has a standard XML structure containing fields that can be recognized from a web application, and can be used to extract useful information concerning each presentation slide individually. The WebVTT file also has a unique structure consisting of a time cue written is specific format. These files are later used for annotation and synchronization purpose.

4. Synchronization results

Seven recorded lecture videos were used to evaluate the proposed method as described in Section 3. The lecture videos were recorded with a normal built-in camera of a laptop. The camera was fixed at a distance of two meters from the projected PowerPoint presentation slides. The videos were grouped into three categories as shown in Table 2:

1. Video with no-occlusion
2. Video with partial-occlusion
3. Video with full-occlusion

In no-occlusion videos, a presenter never occludes the presentation slide. The projected presentation slides are visible all the time in the video. In partial-occlusion, the presenter walks freely in front of the projected screen, periodically occluding some of the content presented in some of the presentation slides. While in full-occlusion cases, a part of the presentation slides is always occluded by the presenter.

The majority of the videos are in the partial-occlusion category as it is the most common scenario for lecture videos. The presenter normally does not stand in front of the projector screen. He moves around and goes in front of the screen when he needs to show or explain something specific.

Table 2
Evaluation video category list

Video	Type of PowerPoint Presentation	Category
Video 1	No animation/no theme/white background	Partial-occlusion
Video 2	Animation (gif images)	Partial-occlusion
Video 3	No animation/ rich background	Full-occlusion
Video 4	Animation with simple theme/ transition effects	Partial-occlusion
Video 5	No animation/ rich background	Partial-occlusion
Video 6	Minimalistic theme, duplicate slides present	No-occlusion
Video 7	Minimalistic theme, duplicate slides present	Partial-occlusion

To create a ground truth, each video was manually split into shots. A shot is detected when a slide change occurs in a video. Each shot contained only one of the presentation slides. The manual slide shots are used as ground truth to evaluate the proposed algorithm. Table 3 shows the results of the automatic splitting vs. manual splitting on all of the seven lecture videos.

Table 3
Video comparison general results

Video #	Total slides in a video	Manual shots detected (Ground truth)	Automatic shots detected (Proposed method)	Success (%)	Error (%)
Video 1	12	12	12	100	0
Video 2	24	24	24	100	0
Video 3	27	27	26	96.3	3.7
Video 4	52	52	52	100	0
Video 5	67	67	65	97	3
Video 6	61	61	61	100	0
Video 7	22	22	22	100	0

As it can be seen in most of the cases (videos 1, 2, 4, 6 and 7), the automatic approach successfully matches the results of the manual approach with a success rate of 100%. In the other two cases (videos 3 and 5) it can be seen that small errors occur during the detection of the slides, resulting in an error rate of 3,7% and 2,9% respectively for the two videos. The errors that occurred are due to the fact that the two consecutive slides in the video contain almost identical information. This can be a problem because during the shot detection process the results obtained have to be filtered through a global threshold, while using SIFT. Thus if the two slides have almost identical content they can't be easily separated because of the high amount of matching feature points. Thus additional processing based on local threshold or adaptive threshold is required. This can be achieved using Otsu thresholding.

Table 4
Video 1 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	8.44
251				
254	2	254	8.47	15.08
453	3	453	15.11	18.05
542	4	542	18.08	23.12
694	5	694	23.15	27.16
813				
815	6	815	27.19	45.51
1365	7	1365	45.54	49.84
1495	8	1495	49.88	53.01
1590	9	1590	53.05	57.75
1730				
1732	10	1732	57.79	62.26
1867	11	1867	62.29	80.71
2420	12	2420	80.74	86.65

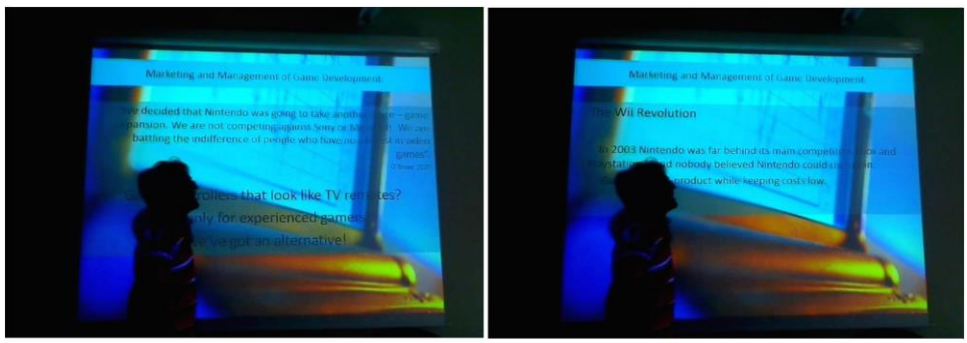
Video 1 is a partial-occlusion video that contains no animations and the presentation recorded in the video has a plain white background with no theme. As it can be observed from Table 4, the manual splits of the video correspond to the automatic segments splits. The values in rows 2, 7 and 12 of the table represent ‘half-slide’ frame regions that were manually detected in the video and they are highlighted with light grey. Usually when giving a presentation using presentation slides, the transition between slides can generate a transition frame, which the human eye cannot perceive. Even though the eye cannot see them, a camera can capture them. That is when a "half-slide" frame can occur. During the transition between two slides, for example slides one and two of the presentation, some frames can be recorded in-between the two slides that contain mixed information from both slides. In the final shot detection selection of the automatic algorithm, it can be seen that these false-positive shot detections have been discarded and merged to either one of the two shots (highlighted with dark grey) that contain one of the slides present in the current ‘half-slide’ frame.

The results obtained from Video 2 can be seen in Table 5, that all the slides are successfully recognized in this video. As mentioned before Video 2 belongs to the partial-occlusion category of videos. Also in this video animated images (.gif files) exist. From the results it can be seen that this approach can compensate in some extent this situation. This happens because feature points found from the SIFT algorithm can be matched even though they do not have the same position or scale. Thus if a presentation

contains simple gif images this does not appear be a problem but further investigation might be needed to be carried out in order to completely validate such an assumption.

Table 5
Video 2 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	15.28
459	2	459	15.31	22.75
680				
683	3	683	22.78	33.23
997	4	997	33.26	39.20
999				
1176	5	1176	39.23	44.27
…	…	…	…	…
3337	22	3337	111.34	113.41
3400	23	3400	113.44	115.68
3468	24	3468	115.71	118.98
3470				



(a) (b)

Fig. 6. Synchronization Error detected in Video 3. (a) Frame 4776, (b) frame 4777

In Table 6 the comparison results of Video 3 are presented. This is a full-occlusion video that contains the recording of a presentation with no animations but with very rich background. Different slides can have the same or different backgrounds consisting of complex images. As it can be observed in Table 6, the proposed solution manages to find all the slide transitions except one. At frame 4777 a mistake occurs. This happens because of the content of the frames. As it can be seen in Fig. 6, the two images

have a big amount of common points that can be detected as matching feature points by SIFT, which makes it extremely difficult to choose an optimal threshold value for it.

Table 6
Video 3 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	5.63
170	2	170	5.67	14.34
431	3	431	14.38	25.15
753				
755	4	755	25.19	35.03
1049				
1051	5	1051	35.06	43.00
1290	6	1290	43.04	48.74
1462	7	1462	48.78	58.75
1762	8	1762	58.79	67.90
1764				
2034				
2036	9	2036	67.93	74.10
2222	10	2222	74.14	78.34
2349	11	2349	78.37	84.45
2532	12	2532	84.48	92.69
2779	13	2779	92.72	101.43
3041	14	3041	101.46	115.44
3461	15	3461	115.48	128.82
3862	16	3862	128.86	133.5
3864				
4002	17	4002	133.53	137.73
4129	18	4129	137.77	142.07
4259	19	4259	142.10	146.88
4403	20	4403	146.91	150.28
4505	21	4505	150.31	154.1
4507				
4621	22	4621	154.18	162.92

4777				
4884	24	4884	162.69	166.39
4988	25	4988	166.43	173.07
5188	26	5188	173.10	179.07
5368	27	5368	179.11	182.14

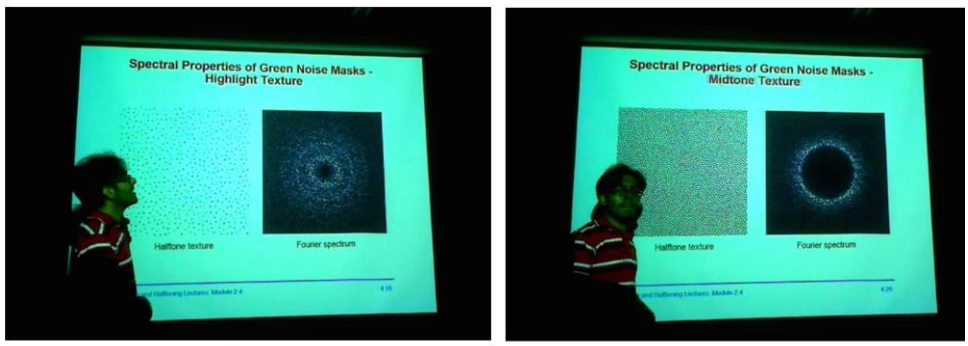
Video 4 is partial-occlusion video containing presentation slides with a simple theme and a white background. The video contains also small animations. These animations are placed on small arrows that do not cover a big amount of the slide space. Also a slide transition effect exists in this video. The results obtained by the processing of Video 4 can be seen in Table 7.

Table 7
Video 4 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	7.50
226	2	226	7.54	10.14
305	3	305	10.17	12.01
308				
361	4	361	12.04	13.54
363				
407	5	407	13.58	15.38
409				
462	6	462	15.41	17.75
…	…	…	…	…
3420	42	3420	114.11	114.88
3466	47	3466	115.64	115.81
3470				
3472	48	3472	115.84	116.14
3482	50	3482	116.18	116.28
3486	51	3486	116.31	116.41
3490	52	3490	116.44	123.62

Even though small animations and transition effects exist in the video, the algorithm can compensate for; the ‘half-slide’ frames (shown using light grey color), the small animations, and the slide transition effect by recognizing successfully the slide

transitions. But again as mentioned before this might need further investigation in order to be completely validated.



(a) (b)

Fig. 7. Error found in video 5. (a) Video 5 Frame 2077 and (b) video 5 Frame 2234

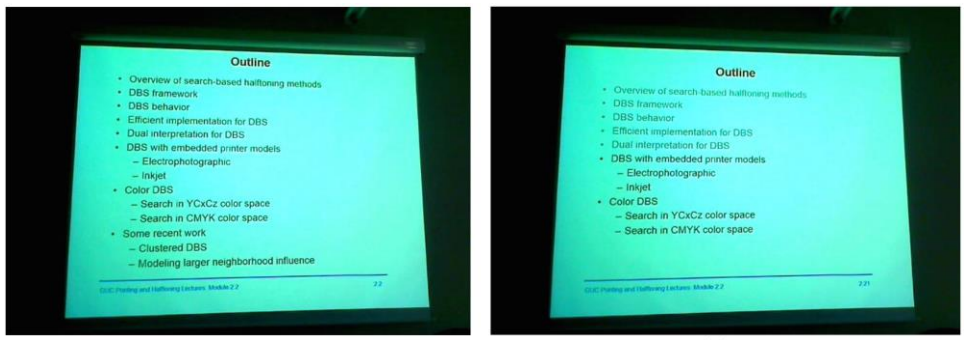
The results obtained from the processing of Video 5 can be found in Table 8. Video 5 is also a partial-occlusion video. From Table 8 it can be seen that even though the majority of the slides are successfully recognized, some errors exist. It can be seen that the mismatching appears between the frames 2277 and 2364. In this range of frames two slides exist and another split should have been detected at frame number 2234. As shown in Fig. 7, the two slides have very similar content. So the problem that occurs here is the exact same problem that occurred in Video 3. In this case the two slides were recognized as one and so they were put together in the same shot. Thus only one of the two slides was matched to this shot, resulting in losing information about the second slide.

Table 8
Video 5 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	7.34
221	2	221	7.37	9.80
223				
295	3	295	9.84	11.91
298				
358	4	358	11.94	13.74
413	5	413	13.78	15.01
451	6	541	15.04	16.61
453				
499	7	499	16.64	18.01
541	8	541	18.05	20.62
619	9	619	20.65	24.49

735	10	735	24.52	28.16
845	11	845	28.19	34.03
1021	12	1021	34.06	37.97
1139	13	1139	38.00	45.97
1379	14	1379	46.01	52.35
1570	15	1570	52.38	55.08
1652	16	1652	55.12	62.29
1868	17	1868	62.32	66.23
1986	18	1986	66.26	69.26
2077	20	2077	69.30	78.84
2234				
2364	21	2364	78.87	85.58
…	…	…	…	…
8613	66	8613	287.38	290.05
8694	67	8694	290.08	292.85
8697				

Video 6 is included in the category of the no-occlusion videos. This means that all the slides and all their content is always visible. In this specific presentation video, a minimalistic theme is used for the PowerPoint presentation and slides with the same content are included. These slides are 'outline' slides containing the exact same content but according to the position they are in the video, different kind of content is highlighted. An example of this kind of frames can be seen in Fig. 8.



(a) (b)

Fig. 8. Example of duplicate slides found in video 6. (a) Frame 127 and (b) frame 1664

In Table 9, a subset of results obtained from the comparison of Video 6 is presented. As it can be seen, manually detected shot boundaries and the shots detected automatically with our proposed application are identical.

Table 9
Video 6 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.03	4.20
127	2	127	4.23	8.20
247	3	247	8.24	11.14
333				
335	4	335	11.17	14.34
431	5	431	14.38	16.81
503				
505	6	505	16.85	19.41
583	7	583	19.45	22.75
585				
…	…	…	…	…
3559	53	3559	118.75	120.22
3604	54	3604	120.25	121.62
3605				
3646	55	3646	121.65	123.28
3696	56	3696	123.32	124.42
3730	57	3730	124.45	125.62
3766	58	376	125.65	127.32
3817	59	3817	127.36	128.69
3858	60	3858	128.72	130.19
3903	61	3903	130.23	138.13

The final tested video is a partial-occlusion video. Video 7 also contains the recording of a presentation in which the slides have a minimalistic theme. In this video like Video number 6 duplicate ‘outline’ slides exist.

As it can be seen from the results in Table 10, the results obtained from the manual and the automatic detections are consistent. With both methods, the different slides in the video are detected successfully and they are the same in both situations. Even though both occlusion and duplicate slides exist in the video, the algorithm managed to successfully recognize the different slides in the video.

Table 10
Video 7 results

Manual shot detection (Ground truth)	**Slide Number**	**Automatic shot detection** (Frame number)	**Start time in video**	**End time in video**
1	1	1	0.033	3.47
105	2	105	3.50	6.07
111				
183	3	183	6.10	8.44
248				
254	4	254	8.47	10.67
321	5	321	10.71	12.84
386	6	386	12.87	15.44
464	7	464	15.48	18.25
470				
…	…	…	…	…
1081	14	1081	36.06	39.47
1178				
1184	15	1184	39.50	42.64
1279	16	1279	42.67	45.44
1363	17	1363	45.47	48.24
1447	18	1447	48.28	51.25
1537	19	1537	51.28	54.05
1621	20	1621	54.08	57.25
1717	21	1717	57.29	59.85
1795	22	1795	59.89	61.52

In conclusion, it was found that the proposed algorithm has a high rate of success (99%) in detecting successfully the presentation slides in the lecture videos. However, the proposed method works best for linear slide changes in ascending order. During the presentation, if the presenter jumps back and forth between the slides or jumps from one slide to a faraway slide, then the algorithm would lose the actual slide count and its position in the video. This would then require manual validation. Having said that, the validation after automatic synchronization, will take far less time than the manual synchronization.

5. Annotations and 3D tag clouds

The main purpose of this section is to show how to use the generated output files in the proposed method to create the necessary annotations for different applications and eLearning platforms. Example of annotations can be seen in Fig. 9.

5.1. Two-way synchronization

A two-way synchronization between the lecture video and the presentation slides is implemented, which provides synchronized navigation of the presentation slides and video shots. Thus the presentation slides change automatically according to the content being shown in the video. Furthermore, when the student changes the slide manually, the video player automatically changes the video shot viewed in order to show the part of the video in which the selected slide is being shown.

This is implemented using the structured XML output file that was described in Section 3, and can be seen in Fig. 9(a). All the necessary information about every slide can be found in that file.

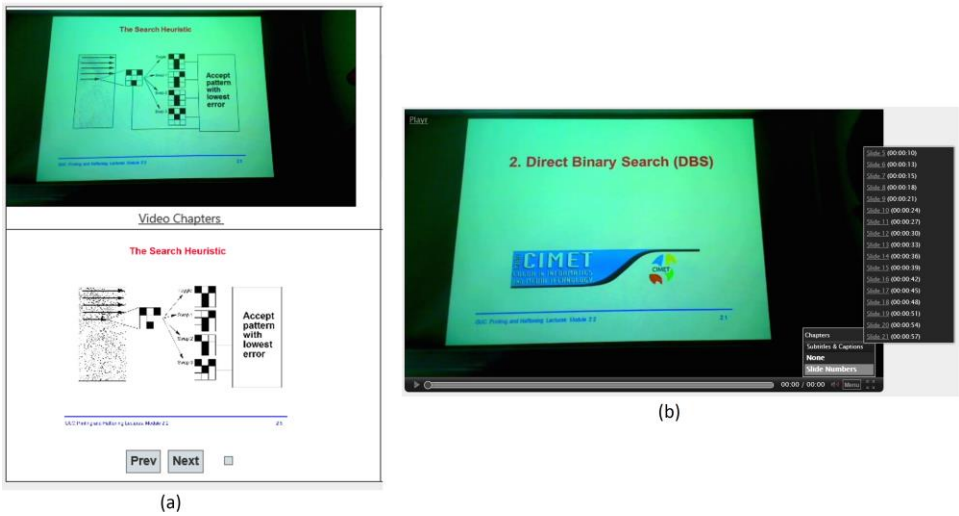


Fig. 9. Different video annotations: (a) Two-Way Synchronization, (b) Chapters Menu

5.2. Video chapters and subtitles

The second annotation that is proposed is the creation of video chapters and displaying them as subtitles. As shown in Fig. 9(b), video chapters are provided in the form of a drop-down menu, in which different menu items are added in order to allow the student to jump to specific positions in the video. Thus, the student will be granted random access to the video allowing him to move freely to specific parts depending on the contents he wants to view. The subtitles are displayed throughout the lecture video, specifying at any moment the position the video is at. This can provide relevant information about what is being displayed; like the slide number, title, etc. For the implementation of this functionality the WebVTT output file is required.

5.3. 3-Dimensional tag clouds

Next, we propose the use of 3D tag clouds. This will serve as an annotation and interaction mechanism. The 3D does not mean that the tag cloud will be presented as a 3D shape. It means that the information that will be shown through the tag cloud will have more than one dimension: time, font-size, font-type.

5.3.1. Tag cloud generation

To generate 3D tag clouds, we extract candidate keywords from the presentation slides and text documents (if applicable), and use the presentations and their corresponding lecture videos to extract certain features (attributes, weighting factors). These features are used as criteria to find the importance of a given word.

In the given context, the words are extracted from the presentation slides and are stored in the slide text file as described in Section 3. Each word is read independently and is processed twice. The first thing that is checked is the significance of the word. A list of common words containing stop words (Leskovec, Rajaraman, & Ullman, 2012) of the English language, such as in, and, or, has been created and every word is checked using this list. If the word is included in the list then it is discarded, because most probably it has no meaningful significance.

After every word has been checked and filtered, the remaining words (words that are meaningful) proceed for further processing. The second step is to apply stemming on these words. Stemming is a process used to reduce inflected or derived words to their stems, bases or roots forms. This process is applied to group words that have the same meaning but are used in different ways using different conjugations etc. For example, the words ‘stemmer’, ‘stemming’, ‘stemmed’, can all be reduced to the base form ‘stem’. With the stemming process finished, the words are not further processed; and are ready to be used in the creation of a variety of tag clouds.

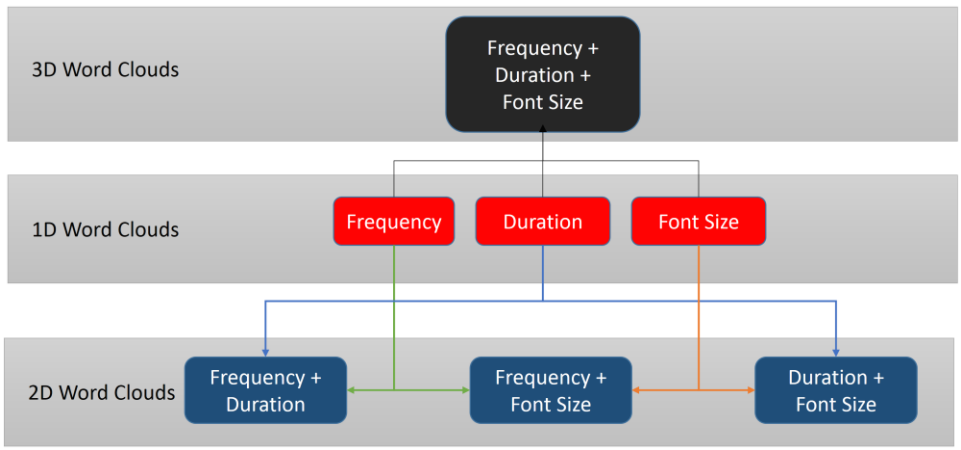


Fig. 10. 3D tag clouds

We generate three different types of tag clouds. They are grouped into three categories as shown in Fig. 10. The first category includes tag clouds that are created utilizing either one of the features - *frequency (f)*, *time duration (d)*, *font-size (Fs)* of every word. As the tag cloud is using only one of the features at a time, we therefore call

it a 1D tag cloud. The second category of tag clouds are created by combining any two of the above three features, whilst the third category combines all the three features to create a 3D tag cloud. These features are stored in word information file that was created during PowerPoint processing described in Section 3.

The 1D tag clouds can be considered the base of the other two categories; as combining two or three of the 1-D tag clouds create 2D and 3D tag clouds.

After the end of the word processing, three 1D tag clouds can be created. These tag clouds are created using the *frequency (f)*, *time duration (d)*, and *font-size (Fs)* information of every word.

From the text extracted from the presentation slides, a tag cloud can be created with the weights of every word being calculated according to the frequency of appearance, of every word in the whole document. This can be done by counting the instances of the stems of the words in the document. The second 1-D tag cloud that is created is the one in which every word has weight according to the time duration each word appears in the lecture video. This can be done by checking stemmed words in the presentation slide and the duration of appearance of every slide in the video. The duration of slide appearance in video is estimated by shot boundary detection (Boreczky & Rowe, 1996). A shot is detected in a video when a slide change event occurs. The last type of tag cloud that is created, calculates the weight of every word according to the font size and bold feature information of every word. These features are extracted directly from the presentation's meta-data information.

5.3.2. Tag cloud visualization

After the word processing is over, different types of tag clouds are created that can be presented visually using three different visual attributes, representing each of the extracted features.

In 1-D tag clouds the dimension of the information is presented using the 'size' of the word in the tag cloud. Therefore, the larger the font used for the word, the more significant it is; always according to the attributes used in the calculation of the weight. The representation of the words in a tag cloud changes if more than one attributes is used.

Having calculated different kinds of weights (features) for every word in the document instead of creating only 1-D tag clouds, it is possible to create tag clouds combining multiple features. In this case by combining two features (corresponding to two different attributes), 2-D tag clouds can be created. There are three possible combinations of these attributes:

I. Frequency & Duration
II. Frequency & Font Size
III. Duration & Font Size

To be able to present clearly the necessary information in the 2-D tag cloud, every word presented has to have two dimensions. In the first case where the *frequency (f)* and the *duration (d)* are used, the frequency is represented by the 'font size' of the word and the duration by its 'color'. The bigger the word is the more frequent it is. The more colorful the word is, the more it appears in the video.

For the combination of *frequency (f)* and *font-size (Fs)*, the 'size' of the word in the tag cloud represents the frequency and its 'color saturation' represents the font size in

the presentation slides. Finally, for the third case where *duration (d) and font-size (Fs)* are present, the 'size' of the word represents the duration and the 'saturation' represents the font size.

The last type of tag cloud is a 3-D tag cloud. In this case all three attributes are represented. So every word in the tag cloud has three dimensions in order to show clearly to the students the significance of every word according to each of the three attributes. In this case the *frequency (f)* is represented by the 'font size' of the word, the *duration (d)* by the 'color' and the *font-size (Fs)* by the 'saturation'.

5.3.3. Tag cloud examples

This section gives some examples of tag clouds according to different weighting factors i.e. *frequency (f), duration (d),* and *font-size (Fs)*. The examples are generated from hyperlinked lectures of computer science database courses in HIP. Fig. 11(a) shows a tag cloud according to the frequency of occurrence of each word. The more important the word is, the bigger it is in the cloud. As can be seen in Fig. 11(a), the word 'DBS' is the most occurring word for the given lecture, following 'search', 'model', and 'image'. Fig. 11(b) shows the tag cloud based on the duration of appearance of each word in the video. So, if a word is displayed for longer, it is an important word. While Fig. 11(c) shows a tag cloud based on word font size in the presentation. The bigger the word font is the more important it is. So, we see that some other words such as 'printer', 'model', 'halftone', and 'print', start to appear in the tag cloud. This is because these words may have been used either as headings or subheadings with larger font size in presentation slides.

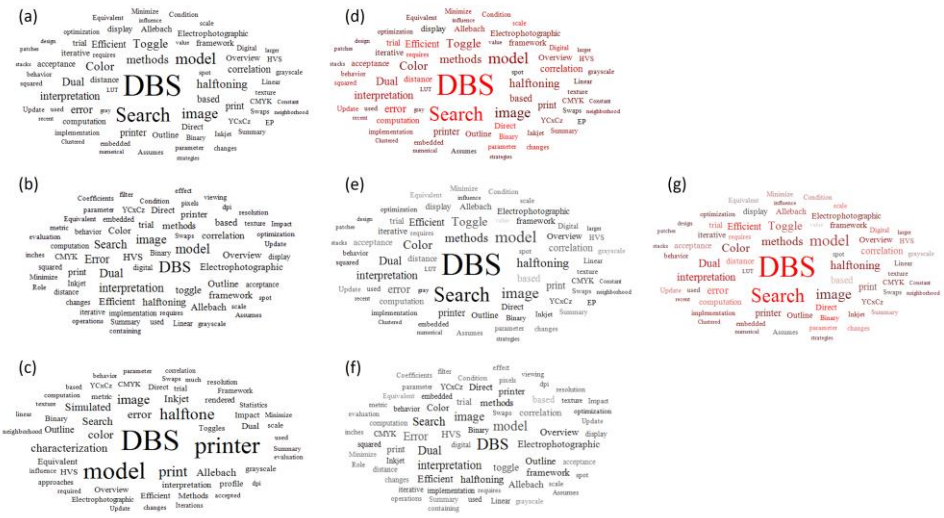


Fig. 11. Different types of tag clouds generated from frequency (f), duration (d), and font-size (Fs)

The 2D tag clouds are shown in Fig. 11(d), Fig. 11(e), and Fig. 11(f). Fig. 11(d) shows a tag cloud according to the *frequency (f)* and *duration (d)* of appearance of each word. In this case, the frequency is represented by the 'size', while the duration is represented by the 'color'. This means that the bigger the word is, the more frequent it is, and the more colorful the word is the more it appears in the video. Fig. 11(d) is similar to

Fig. 11(a) since both take into account the frequency of the word. Nevertheless, we see that the 'color' in the tag cloud representing the duration adds another dimension to it. Similarly, the tag clouds based on *frequency (f)* and *font-size (Fs)*, and *duration (d)* and *font-size (Fs)* are depicted in Fig. 11(e), and Fig. 11(f) respectively. In this case, the font size is represented by 'transparency', while the duration is represented by 'size'.

Finally, Fig. 11(g) shows a 3-D tag cloud. It takes into account all the three attributes i.e. *frequency (f)*, *duration (d)* and *font-size (Fs)*. In this case, the frequency is represented by the 'size' of the word, the duration by the 'color' and the font size by the 'transparency'. When we analyze this tag cloud and compare it with Fig. 11(e) and Fig. 11(f), we can see that although some words may have occurred the same number of times but they might have different duration of appearance in the video or font size in the presentations slides, and vice versa, such as 'model' and 'image', 'direct' and 'outline', 'toggle' and 'halftoning', etc.

6. System evaluation

An experiment to evaluate the usefulness of HIP with annotated lectures is conducted. The experiment is divided into two parts. The first part of the experiment group students into four categories based on their learning preferences as depicted in Fig. 12.

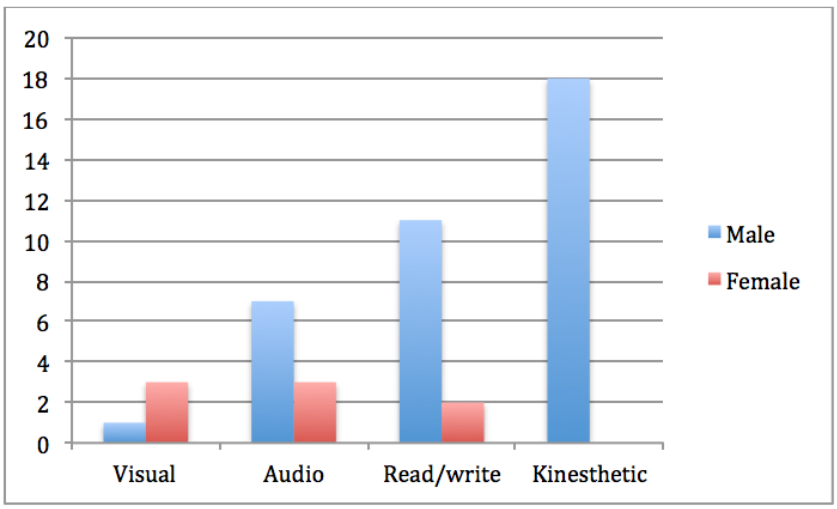


Fig. 12. Grouping of students based on VARK learning model

A standard VARK questionnaire was used to differentiate students' learning style and to group them as visual, audio, text and kinesthetic learners (Fleming, 2014). The questionnaire was distributed to 55 students comprising of bachelor and master level at Gjøvik University College (GUC). The students were grouped into their respective category based on their learning style, to have an equal distribution for the second part of the experiment. The distribution of students is shown in Fig. 12.

In the second part of the experiment, the students were asked to go through the HIP and non-HIP version of recorded lectures and to give their appreciation on a Likert scale from 1 to 5, where 1 corresponds to strongly disagree while 5 corresponds to strongly agree.

The first four questions were aimed towards usability study of HIP in comparison to existing system (Fronter) at GUC. The questions were:

1. Is covering material through annotated lecture videos including 3D tag clouds in HIP more useful?
2. Is it easy to cover material through HIP annotated videos and 3D tag clouds?
3. Is finding material in HIP less time consuming?
4. Is reviewing the material easier than existing system?

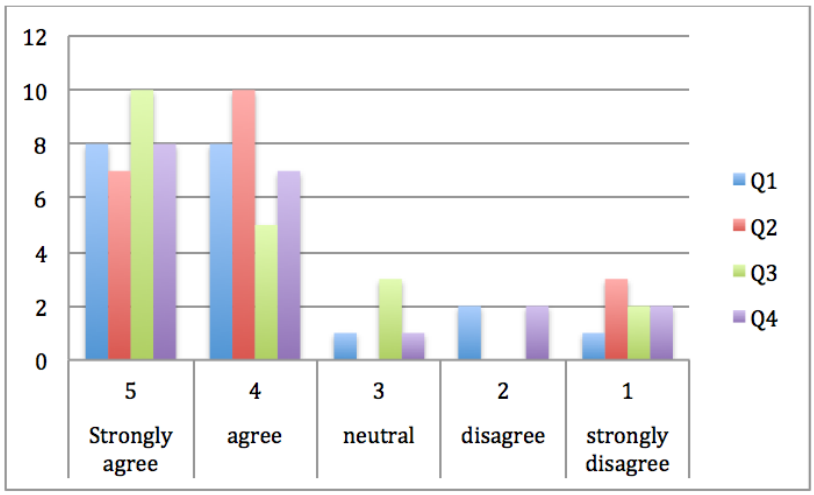


Fig. 13. Students' response on a Likert scale of 1-5 for first four questions

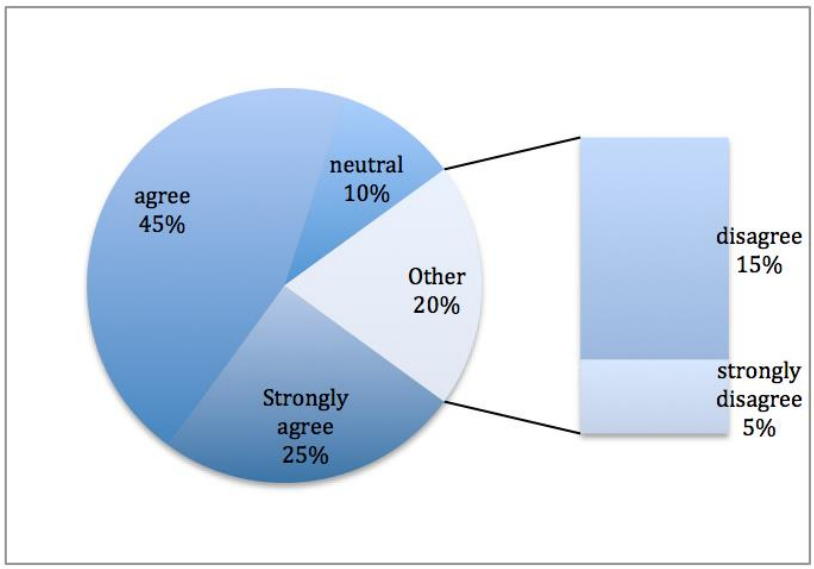


Fig. 14. Students' response to recommend and use HIP for preparation of exams

The initial feedback was encouraging as 80% of the participants agree with the first statement, whereas, 5% responded neutral and 15% against it. For the second statement, 85% students find it easy to cover the material through annotated lecture videos including 3D tag clouds in HIP, while 15% disagree. A similar trend was observed for other questions. The full set of this experiment results are shown in Fig. 13.

Mostly students responded in favor of HIP when asked if they would recommend a fellow student to use HIP, and if they will prefer to use such a system to prepare for the exams. The results are depicted in Fig. 14 showing that 70% of students' responses are in favor of HIP while 10% are neutral and 20% of them see no benefit of using HIP.

7. Conclusion

In this paper, an automatic annotation of lecture videos including a 3D tag cloud for a novel eLearning platform called HIP is proposed to address the challenges faced by today's LMS and massive open online courses (MOOCs). The proposed solution is targeted at eLearning platforms for robust and automatic annotation of lecture videos. The proposed solution can automatically detect slide transitions in the video, synchronize them with the presentation slides and provide outputs that can be used in an on-line eLearning platform like HIP to provide random access and interaction mechanisms to the students. A two-way synchronization is implemented between the presentation slides and the lecture video. The video element is embodied with subtitles presenting relevant information throughout the videos and a chapter menu that is always present to provide extra navigation into the video.

In addition, this paper proposes tag clouds targeted at eLearning platforms for robust and automatic annotation of hyperlinked pedagogical media. We propose three different types of tag clouds based on the extracted features i.e. frequency of occurrences, duration of appearance, and font-size of candidate tag words. These features are extracted automatically from the presentation slides and lecture videos. These tag clouds can benefit multimedia driven educational platforms by allowing, content structuring and synchronization, hyperlinking multimedia, and fast and easy retrieval of educational contents.

Concerning the teachers recording the lecture and using the system requires no prior knowledge and minimal effort on their side, as the only thing needed is to provide the video and the presentation files. The usefulness of the overall system is evaluated. However, in-depth study and analysis on the contribution and effectiveness of the 3D tag cloud is needed.

References

Boreczky, J. S., & Rowe, L. A. (1996). Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging, 5*(2), 122–128.

Daubechies, I. (1992). *Ten lectures on wavelets*. Philadelphia, PA: SIAM.

De Bello, T. C. (1990). Comparison of eleven major learning styles models: Variables, appropriate populations, validity of instrumentation, and the research behind them. *Journal of Reading, Writing, and Learning Disabilities International, 6*(3), 203–222.

Fan, Q., Barnard, K., Amir, A., & Efrat, A. (2009). Accurate alignment of presentation slides with educational video. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (pp. 1198–1201).

Fan, Q., Barnard, K., Amir, A., & Efrat, A. (2011). Robust spatiotemporal matching of electronic slides to presentation videos. *IEEE Transactions on Image Processing, 20*(8), 2315–2328.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education, 78*(7), 674–681.

Fleming, N. D. (2014). *VARK: A guide to learning styles*. Retrieved from http://vark-learn.com/the-vark-questionnaire/

Franzoni, A. L., Assar, S., Defude, B., & Rojas, J. (2008). Student learning styles adaptation method based on teaching strategies and electronic media. In *Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies (ICALT08)* (pp. 778–782).

Halvey, M. J., & Keane, M. T. (2007). An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 1313–1314). New York, NY, USA.

Huang, W., Li, X., & Yao, L. (2008). The multimedia educational resources editing system on synchronization of PPT presentation videos and electronic slides. In *Proceedings of the IEEE International Symposium on IT in Medicine and Education* (pp. 790–795).

Hyerle, D. (2009). *Visual tools for transforming information into knowledge* (2nd ed.). Corwin Press.

Imran, A. S., & Cheikh, F. A. (2012). Multimedia learning objects framework for e-learning. In *Proceedings of International Conference on e-Learning and e-Technologies in Education (ICEEE)* (pp. 105–109).

Imran, A. S., & Kowalski, S. J. (2014). HIP – A technology-rich and interactive multimedia pedagogical platform. In *Proceedings of 14th International Conference on Human Computer Interaction (HCII2014)* (pp.151–160). Crete, Greece.

Jonassen, D. H., Carr, C., & Yueh, H. P. (1998). Computers as mindtools for engaging learners in critical thinking. *TechTrends, 43*(2), 24–32.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2012). *Mining of massive datasets* (2nd ed.). Cambridge University Press.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

Mallat, S. G., (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 11*(7), 674–693.

Manochehr, N.-N. (2006). The influence of learning styles on learners in e-learning environments: An empirical study. *Computers in Higher Education Economic Review, 18*, 10–14.

Masie, E. (2005). *Nano-learning: Miniaturization of design*. Chief Learning Officer. Retrieved from http://www.clomedia.com/2005/12/28/nano-learning-miniaturization-of-design/

Meyer, Y. (1990). *Ondelettes et Opérateurs*. Tome I, Hermann, Paris.

Ngo, C. W., Pong, T. C., & Huang, T. S. (2002). Detection of slide transition for topic indexing. In *Proceedings of IEEE International Conference on Multimedia and Expo* (Vol. 2, pp. 533–536).

Ngo, C. W., Wang, F., & Pong, T. C. (2003). Structuring lecture videos for distance learning applications. In *Proceedings of the Fifth International Symposium on Multimedia Software Engineering* (pp. 215–222).

Northrup, P. T. (2007). *Learning objects for instruction: Design and evaluation*. IGI Global.

Pfeiffer, S., Jägenstedt, P., & Hickson, I. (2014). *WebVTT: The web video text tracks*

format. Draft Community Group Report, W3C. Retrieved from http://dev.w3.org/html5/webvtt/

Rivadeneira, A. W., Gruen, D. M., Muller, M. J., & Millen, D. R. (2007). Getting our head in the clouds: Toward evaluation studies of tag clouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 995–998).

Thampi, S. M., Abraham, A., Pal, S. K, & Rodriguez, J. M. C. (Eds.). (2013). *Recent advances in intelligent informatics: Proceedings of the Second International Symposium on Intelligent Informatics (ISI'13)*. Springer. ISBN: 978-3-319-01777-8

Tuan, L. T. (2011). Matching and stretching learners learning styles. *Journal of Language Teaching and Research*, *2*(2), 285–294.

Wang, X., Ramanathan, S., & Kankanhalli, M. (2009). A robust framework for aligning lecture slides with video. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP)* (pp. 249–252).

Wang, Y., Kitayama, D., Lee, R., & Sumiya, K. (2009). Automatic generation of learning channels by using semantic relations among lecture slides and recorded videos for self-learning systems. In *Proceedings of the 11th IEEE International Symposium on Multimedia* (pp.275–280).