MIGRATING LEGACY VIDEO LECTURES TO MULTIMEDIA LEARNING OBJECTS

Andrea De Lucia, Rita Francese, Massimiliano Giordano, Ignazio Passero, Genoveffa Tortora
Dipatimento di Matematica e Informatica, Università degli Studi di Salerno,via Ponte don Melillo 1,Fisciano (SA),Italy

Keywords: e-learning, learning object, video lectures, migration strategy.

Abstract: Video Lectures are an old distance learning approach which do not offer any feature of interaction and retrieval to the user. Thus, to follow the new learning paradigms we need to reengineer the e-learning processes while preserving the investments made in the past. In this paper we present a methodology for semi-automatically migrating traditional video lectures into multimedia Learning Objects. The process identifies the frames where a slide transition occurs and extracts from the PowerPoint Presentation information for structuring the Learning Object metadata. Similarly to scene detection approaches, we iteratively tune several parameters starting from a small portion of the video to reach the best results. Once a slide transition is correctly detected, the video sample is successively enlarged until satisfactory results are reached. The proposed approach has been validated in a case study.

1 INTRODUCTION

Oral expositions supported by slides are typical teaching and learning activities providing *Information/Knowledge Dissemination* that might usefully be transferred from the classroom to online mode (Pincas, 2003).

Video lectures are the first modality adopted to supply this kind of distance education. Often these lectures are taught by famous "gurus". Much of this material is broadcasted by satellite television or is available and is mainly sold through Web sites, in a videotape or CD format. Moreover, filming a teacher in the classroom while he/she is giving a traditional lecture, without special constraints to his/her movements or speaking, has still several advantages. Firstly, this approach does not require the teacher to change his/her didactical practice, as the lecture is located in the classroom where teaching is more natural then in studio; secondly it enables the Universities to obtain in a short time a rich repository of good quality learning content they can offer on the e-Learning market (Gerhard *et al.* 2002).

These old teaching approaches create a passive situation in which the user follows the classical lecture, but at distance. The learner receives the knowledge transmitted by the teacher which is at the centre of the learning process. The user cannot interact in any way with the material and he/she has to search the entire video to find a specific subject. Thus, there is a need for reengineering the e-learning processes to follow the new learning paradigms, such as blended e-learning (Bersin, 2003), while preserving the investments made in the past. As for legacy systems (Brodie and Stonebraker, 1995), it can be advantageous to migrate the video lectures into a more modern format that enables the learner to become an active subject.

It is common opinion that to embrace largely adopted standards and technologies augments the compatibility and enables to provide reusable contents online. Video lectures respect the format of traditional in presence ones, with a teacher that gives the lecture with the support of slides for one/two hours. On the contrary, the actual trend is to create short, at most twenty minutes (ADL, IEEE LTSC, IMS), online learning content including:

- text, graphics, and movies;
- a navigation scheme (easily a table of contents and/or buttons);
- assessments.

Learning contents should also enable to identify the learner and record information about the learning experience.

The new learning technologies are based on Learning Objects (IEE LTSC), which are

De Lucia A., Francese R., Giordano M., Passero I. and Tortora G. (2006).
MIGRATING LEGACY VIDEO LECTURES TO MULTIMEDIA LEARNING OBJECTS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - SAIC*, pages 51-58
DOI: 10.5220/0002447300510058

characterized by different granularity levels, are combined appropriately, depending on the learner profile, and deployed into an online course. Thus, to be able to reuse, in appropriate way, existing video materials on advanced learning systems as Learning Management Systems and Learning Content Management Systems we need to structure them as Learning Objects. In absence of any automatic support, this requires to manually fragment the video and to associate it to an index structure, a very tedious and time consuming activity.

In this paper we present a method for semi-automatically migrating traditional video lectures to multimedia Learning Objects. The proposed approach has been experimented at the Department of Mathematics and Informatics of the University of Salerno, where a lot of distance materials was available in terms of video lectures and the related PowerPoint presentations. The process both identifies the frames where a slide transition occurs and extracts information for structuring the Learning Object metadata from the PowerPoint Presentation. Similarly to scene detection approaches (Lienhart, 1999 and Yusoff *et al.,* 1998) we tune several parameters to reach the best results.

The proposed slide detection process first masks the frames of the video lecture to the aim of identifying the slide area, then confronts unmasked areas applying some similarity metrics. Two frames represent a slide transition if their similarity is lower then a given threshold. The parameters are iteratively tuned starting from a small portion of the video to reach the best results. Once tuned thresholds on the sample video, the complete lecture is processed. At the end of the detection, the user can interactively obtain best results discarding transitions incorrectly detected.

The learning objects produced by the tool have the following characteristics: a PowerPoint presentation is used as the main teaching resource and the flow of the presentation is synchronized with the audio of the lecture. A little window shows the original digital video clip of the lecture associated to the slide currently examined. A navigational schema enables to surf between the contents.

The method and the tool have been validated in a case study.

The rest of the paper is organized as follows: Section 2 discusses related work, while Section 3 presents the proposed approach. Section 4 discusses the results of a case study and Section 5 concludes.

2 RELATED WORK

Many approaches to the detection of scene changes, based on the analysis of entire images, have been proposed in the literature, see for example (Lienhart, 1999 and Yusoff *et al.,* 1998). The main methodology for detecting shot boundary concerns the extraction of one or more features from each frame of the video. In particular, difference metrics are often used to evaluate the changes between subsequent frames, whilst thresholds are used to determine whether changes take place (Smeaton *et al.,* 2003 and Robson *et al.,*1997).

Nagasaka and Tanaka experiment various frame similarity techniques, such as difference of grey-level sums, sum of grey-level differences, difference of grey-level histograms, coloured template matching, difference of colour histograms and χ^2 comparison of colour histograms (Nagasaka *et al.*, 1991). They concluded that the most robust methods is the χ^2 comparison of colour histograms.

Adams *et al.* shows that the detection of gradual transitions needs to perform frame to frame analysis considering great temporal distances, especially when dealing with low quality video materials (Adams *et al.*, 2003).

All these traditional shot boundary detection techniques have been applied for detecting slide transitions, yielding poor results because of the small changes in frames during a slide transitions. In fact, unlike shot transitions, a slide change does not present, in most cases, significant colour changes (Ngo *et al.,* 2002).

Several efforts have been also devolved to build structured hypermedia documents from lectures video and PowerPoint presentations (Ngo *et al.,* 2003, Deshpande *et al.,* 2001, Abowd *et al.,* 2000 and He *et al.,* 1999).

To automate structuring and indexing, major research issues prefer to investigate layout and content of video frames using various techniques (Ngo *et al.,* 2003, Ngo *et al.,* 2002 and Mukhopadhyay *et al.,* 1999), such as the detection of text regions in viewgraph, characters and words recognition, tracking of pointers and animations, gesture analysis and speech recognition.

Video Optical Character Recognition (OCR) is a recent area of intensive exploration, not only for detecting slide transitions, but also to facilitate the matching of videos and electronic slides (Ngo *et al.,* 2003). The process of video OCR mainly includes the detection, segmentation and recognition of video texts, not always balancing the greater computational efforts with better results. No user interaction is allowed to obtain best result because of the execution time.

To the best of our knowledge, no automatic support is provided to translate a video lecture to one or more multimedia learning objects. In particular, Learning Objects are created and edited with software tools called “metadata editors” or “metadata generators”. Several commercial, freeware and open-source tools have been developed in order to edit and manage Learning Object Metadata since the first publication of its specifications (KOM, RELOAD). As an example, the Learning Object Metadata Generator (LOMGen) automatically extracts the metadata with minimal user intervention from HTML pages. It also creates a keyword/key phrase database.

3 THE PROPOSED SOLUTION

In this section we present an overview of our approach for the migration of video lectures towards multimedia Learning Objects. Figure 1 illustrates the overall legacy lecture reengineering process, where the rounded rectangles represent process activities, whilst the rectangles represent the intermediate artefacts generated during the process phases. An actor symbol denotes that an activity is interactive.

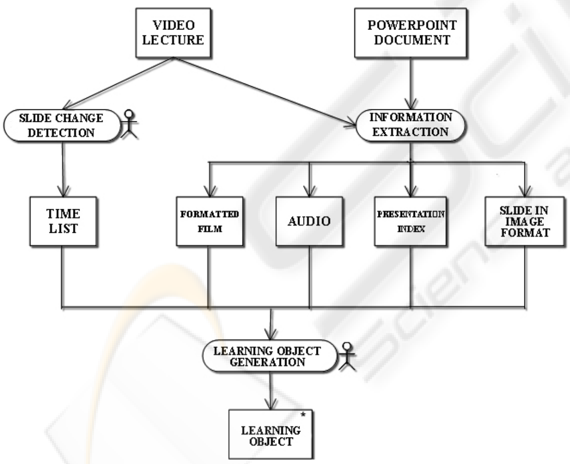

Figure 1: The video lecture reengineering process.

The materials provided as input to the process are a video lecture and the associated PowerPoint presentation. The slide change detection sub-process receives as input the video lecture and produces a list of frames where a change of slide occurs in the video. The information extraction sub-process translates the source information in a different format. In particular, it produces a smaller, resized version of the video lecture, extracts the audio track from the video and uses the PowerPoint presentation to extract the structure of the lecture. The structure can be further fragmented in order to create several learning objects to obtain the desired granularity. Concurrently, slides are converted in an image format. The *Learning Object Generation* activity, better detailed in Section 3.3, combines the outputs previously produced. In particular, this phase rearranges video and audio of the lecture according to the transitions previously detected, associating PowerPoint slides to the corresponding part of video/audio tracks. The symbol * on the generated *Learning Object* indicates that multiple occurrences of this object are generated, depending on the number of user required organizations of the lecture fragments.

3.1 Identifying Slide Transitions

The slide transition detection sub-process is organized in two sub-phases: parameters setting and slide detection. During the parameters setting phase a small portion of the video is examined and parameters are tuned to reach desired results. The slide detection phase is then applied to entire lecture. At the end of the process we require the user involvement to grant the correctness of the results.

3.1.1 Parameter Setting

To properly detect the slide transitions we need to set several parameters. Their values are iteratively tuned to reach the best results on a sample extracted from the lecture video. Let us define and detail the meaning of the parameters we need to tune to be able to detect slide transitions.

Sensitivity Threshold. The *Sensitivity Threshold* δ represents the minimum value of the difference between the intensity of two corresponding pixels in two subsequent frames to detect a variation.

Figure 2: Two frames on the same slides and their difference.

Applying a difference metric to two frames shows not null results only in the teacher’s moving area, if the lecture frames are depicting the same slide, as shown in Figure 2. Otherwise, if the difference is computed during a slide transition, as depicted in Figure 3, the result of the comparison shows both the teacher movements and slide differences.

Figure 3: Two frame with different slides and their difference.

Mask Threshold τ. The slide detection has to be performed by considering the transitions occurring only in the slide area. To be able to retail the slide from the remaining of the scene it is necessary to analyze the image pixel values. Figure 4(a) shows the result of the intensity analysis on the frame in Figure 4(b). In particular, we take the value of the pixels corresponding to the lines r_1 and r_2 in Figure 4(b) and analyze their brightness. The circled regions in Figure 4(a) correspond to the screen area and have the highest intensity values. As a consequence, to detect the portion of the image containing the slide we introduce a threshold τ on the minimum brightness value of pixels.

During the testing of the method we noticed that the application of the threshold τ had, as drawback, the exclusion of many pixels representing words, pointed out by local minimum in the circled area of Figure 4(a). This could cause to miss the detection of transitions. Thus, to avoid to loose interesting information, before applying the threshold τ to highlight the slide area, we have to blur the image. To this aim, we cut off letters by a *low pass averaging filter* (Gonzales *et al.,* 2002).

For every pixel *p(x,y)* let *blur(p(x,y))* be the mean of the pixels values in the square centred on it. If the pixel *p* is on the border line, the square is padded with value 127. At this point, we apply the threshold τ to the *blurred* frame and create a binary mask, hiding the pixels whose brightness value is lower than τ. As a result, the teacher and his movement are masked, while the other dark details, often representing text information on slides, are not masked. In particular, the pixel mask is defined as follows:

$$Mask\ (x,y)= \begin{cases} 1 & if\ blur(img(x,y)) \geq \tau \\ 0 & if\ blur(img(x,y)) < \tau \end{cases}$$

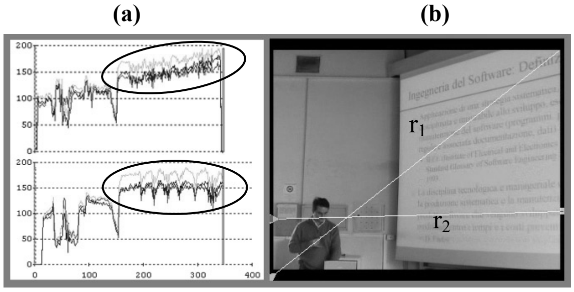

Figure 4: Frame brightness analysis.

Figure 5 shows some examples of results obtained by masking the same slide in different frames, characterized by a different teacher position.

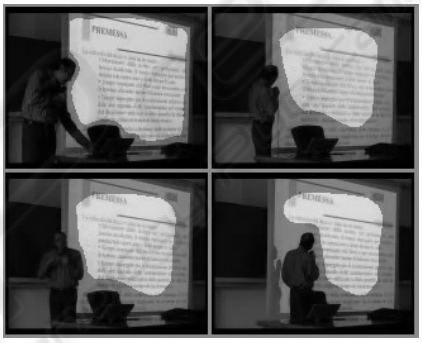

Figure 5: Examples of frame masking.

Time window. Examining video materials in our lecture archive, we noticed that some teachers prefer to adopt animated slide transitions, as Figure 6 shows. In this case a slide change takes up to fifty frames to occur, and, as a consequence, a slide transition spends about two seconds to occur.

Figure 6: Animated slide transition.

In this case, if we consider only adjacent shots we risk to loose some slide changes because the transition occurs in a gradual way. As a consequence, according to the duration of a slide transition, we set the *time window distance w* to select subsequent frames to be compared.

3.1.2 Slide Detections

Analyses and tests performed on learning materials show that some slide detection parameters have to be iteratively tuned to obtain the best results depending on the type of video shot or slide model. To this aim, we combine masking and user interaction in such a way to allow the operator to select the right execution parameters on the base of feedbacks he/she receives. These parameters clearly impact on both precision and recall, two well known metrics of the information retrieval and reverse engineering fields. In our case, the *recall* is the ratio between the number of slide transitions correctly identified by the tool and the total number of slide. The *precision* is the ratio between the number of slide transitions correctly identified by the tool and the total number of retrieved slide transitions.

The first goal is to train the process in order to maximize *recall* while reaching a good *precision.* In this phase the user starts setting thresholds with high values and analyzing a short video sample. He/she interactively examines the obtained results and greedily reduces the threshold values until all correct slide transitions are still detected. In this way the user tunes the working parameters to reach more restrictive values, still maintaining the 100% of *recall*. The slide detection sub-process produces a list of detected transitions validated by the user in the interactive final phase.

Several parameters are needed to reach the required results. Particularly useful is the *Detecting Ratio* ρ, representing the minimum amount of changed pixels with respect to the amount of unmasked ones to detect a slide transition.

$$\rho = \frac{\#\text{of detected changed pixels}}{\#\text{of unmasked pixels}}$$

By using the *difference metric*, a slide transition is detected as follows: first we examine the differences between homologous pixels composing two frames with distance *w*. Next, we determine σ, the number of different pixels, and *f*, the number of unmasked pixels. A slide change is detected if:

$$\sigma \geq \rho * f$$

We also consider that the great part of video materials provided by our case studies is realized with a mobile camera. As a consequence, a *difference metrics* could induce false positive detections (Yeo *et al.,* 1995). To overcome this problem, we decided to combine it with the *statistical metric* χ^2 (Ford *et al.*, 1997 and Sethi *et al.*, 1995). This metrics enables to represent for each frame an histogram depicting the number of pixels having a brightness value *x*, $x \in [0,255]$. The obtained histograms are used to better establish the variations between two frames, independently from the camera motion.

An erroneous slide detection can occur in two cases: a *false positive* detection is generated when we detect a transition which does not occur; a *missing* slide detection occurs if a slide transition is jumped. The slide detection sub-process includes an interactive final phase in which the user, starting from a preview of the slide and the associated frame corrects the erroneous transitions. In particular, if a false positive is detected the transition is discarded, while if the user identifies a missing transition the process is run again on the portion of video containing the transition.

3.2 The Information Extraction Sub-process

The input materials, a video lecture in MPEG format and the corresponding PowerPoint presentation, have to be manipulated to obtain the required Learning Objects. To this extent, we translate the educational contents into an XML based document. From this document we generate a representation compliant with a highly accepted standard, the Learning Object Metadata (Singh *et al.* 2004 and LOM).

In particular, the video lecture, audio and video tracks, have to be rearranged to obtain an efficient transmission in streaming modality. To this aim we need to obtain a low resolution video and high quality audio. Thus, we de-multiplex the lecture to separate the video and audio tracks. In particular, we extract the following information:

- teacher voice, describing the content of the slide;
- timing of the lecture, for synchronizing table of content with voice and slide;
- teacher video, for reducing the *loneliness sensation* of a remote student. We decide to resize the video and to show it in a little window, even if it has not a direct contribute to the understanding of the lecture. As a matter of fact, communications involves several aspects and one of them is the body language.

It is worth noting that the PowerPoint presentation is a source of descriptive information about the learning contents we are generating. In particular, to be able to create the navigation schema we extract the table of content from it. Other information like author name, title, date of creation, etc., can also be derived. Moreover, we get a snapshot of each slide in jpeg format, thus we are able to re-write the slides in a cross platform fashion. To obtain this information we exploited the Component Object Model framework (COM), defining how objects interact within a single Microsoft application or between applications.

3.3 Learning Object Generation

The standard SCORM (ADL) requires that, when a Learning Object is defined, additional descriptions called metadata should be provided. Metadata allows educators to find, reuse and evaluate learning resources matching their specific needs. The process of manually entering Metadata to describe a Learning Object is time-consuming. It also requires the Metadata administrator/author to be familiar with the Learning Object content. Thus, a semi-automated process which extracts information from the data sources can alleviate the difficulties associated with this time-consuming process. The information collected in the previous phase is used to partially fill in these descriptions.
By default, we assume that a single learning object is created. In any case, the user can access the extracted table of content and indicate the entries in this list he/she wants to include in each produced learning object.

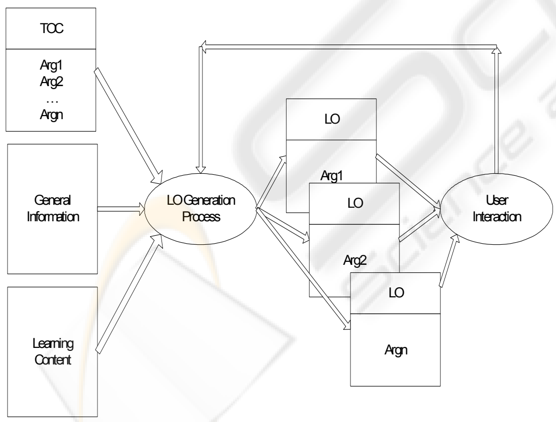

Figure 7: The Learning Object generation process.

The learning object is generated as follows: a video is produced by associating the audio and the teacher video to each slide snapshot. A slide change occurs depending on the time where the slide transition has been detected in the previous phase. A table of contents provides an easy way to navigate between the slides. Figure 8 shows the Web site automatically associated to the System Testing Learning Object extracted from a video lecture of the Software Engineering course.

4 CASE STUDY

The method presented in this paper has been validated on the video lectures of the Software Engineering (SE) and Operating Systems (OS) courses of the Computer Science Program at the University of Salerno.

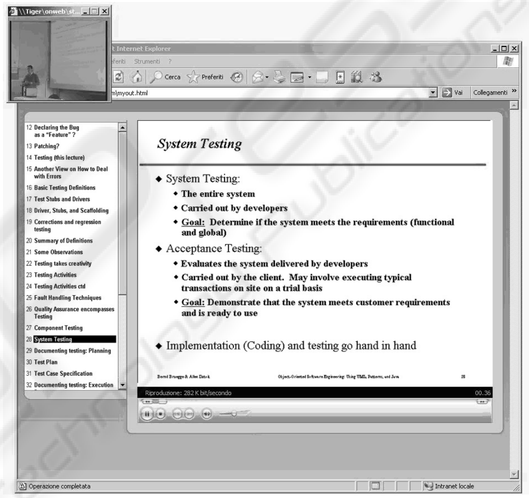

Figure 8: The generated Web Site.

To explain the adopted methodology and evaluate the results we examined a lecture of each course we processed. In particular, we received as input a lecture and the corresponding PowerPoint presentation consisting of *24* slides for the SE lecture and *21* slides for the OS lecture. The number of slides corresponds to the *Number of Transitions+1* in Table 1. As output, the tool provides, for the SE lecture, two Learning Objects which last *17* and *19* minutes, and three Learning Objects of *16*, *17* and *10* minutes for the OS one. The results obtained from the slide change detection algorithms have been evaluated in terms of precision, recall, and the number of user corrections required at the end of the automatic detection. For each lecture we experimented our approach using different threshold values for τ, w, δ and ρ. Table 1 shows the results achieved with the initial thresholds and the best ones. In both the cases, the slide detection process has been first applied to a short portion of the video containing a few transitions, as described in the column labelled "sample". The results obtained with the initial values of the parameters executed on the lecture of Software Engineering course, reached *1* as recall, but revealed an inadequate precision *0.5*. The reduction of the thresholds ρ and τ to *0.12* and *135*, respectively, produced best results on the sample, as reported in Table 1. Processing the complete lecture required a correction of the time window *w* to *5* to cope with the gradualism of some slide transition. The user intervention was finally required to correct five false positives due to unintentional vibrations of the camera.

Table 1: Slide detection results.

	Software Engineering Results			**Operating Systems** Results		
	Sample	Final Sample	Complete Lecture	Sample	Final Sample	Complete Lecture
τ	140	135	135	140	140	140
δ	7	7	7	7	10	10
w	2	2	5	2	1	1
ρ	0,16	0,12	0,12	0,14	0,1	0,12
Frame Number	4720		51771	4815		71272
Transition Number	2		23	2		20
Number of Slide Transition detected	4	2	28	2	3	19
Number of right Transition detected	2	2	23	1	2	19
Number of false positive	2	0	5	1	1	0
Precision	0,5	1	0,82	0,5	0,6	1
Recall	1	1	1	0,5	1	0,95
Number of user corrections	2	0	5	2	1	1

In the Operating Systems lecture we trained the slide detection tool to reduce the overall number of user interventions on the results, exploring the recall values in a suboptimal region of the domain. As Table 1 shows, the analysis of the sample with the initial parameters results in detecting a false positive and missing a transition. With the aim to exclude from the result set the false positive, we rised the *Sensitivity Threshold* δ to the value of *10*, and exploiting the absence of animated slide changes, we reduced the *Time window w* to *1*. Next, we observed the detection execution on the sample of about *5%* of the total length of lecture with the previous parameter and a *Detecting Ratio* ρ set to *0.1*. The detection of a false positive is still obtained Thus, we increased the value of ρ up to *0.12*. We decided to accept a reduction in recall performance, which is balanced by a smaller number of user interventions on the final result set. When we run the tool on the complete lecture with the selected parameters, as it is shown in Table 1, we reached *1* and *0.95* as precision and recall, respectively. It is worth nothing that this setting required only one correction on the result set.

5 CONCLUSION

In this paper we have presented an approach to migrate legacy video lectures into multimedia learning objects.

The method concurrently detects slide transitions and extracts information from a PowerPoint presentation both to get the slides images and to fill the Learning Object Metadata, as table of contents of the presentation.

The proposed solution is mainly based on the detection of slide transitions. To this aim, it first masks the frames of the video lecture to select the slide area. Two frames represent a slide transition if their similarity, deduced by some metrics applied to unmasked pixels, is lower then a given threshold. Parameters are tuned on a small portion of the video until reaching the best results. The detection is then applied to the remaining part of the lecture. The approach has been assessed in two case studies. The execution time required to process a lecture is linearly proportional to the length of the video, very good if compared to approaches based on OCR techniques.

Concerning the parameters tuning, the case studies showed that, after a brief training, low effort is required to achieve good values of precision and recall. This low effort is immediately compensated by the tool simplicity and the achieved results.

A final phase involves the user in the validation of the detected slide transitions. This is necessary to reach the maximum value for both precision and recall. In fact, a wrong slide transition produces the loss of the correspondence between the slide, the audio and video.

At the present we are refining the tool with several features contributing to obtain a better quality in terms of the synchronization of the audio track with the associated slide. The operator will interactively adjust the editing by anticipating (or delaying) the cut with respect to a slide transition to provide to student a complete phrase at the beginning and at the end of each slide. Moreover, to reduce the user involvement we will plan to investigate how the thresholds can be automatically tuned depending on the user interactions. Like in (LOMGen), we aim at automating the Metadata extraction. In addition to the Table of Content and the general information we are already able to extract, we will detect metadata information directly from the learning content by using semantic Web techniques.

REFERENCES

Abowd, G., Atkeson, C.,G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Savhney, N., Tani, M., 2000. Teaching and Learning as Multimedia Authoring: The Classroom 2000 Project. In *ACM Multimedia*. pp.187-198.

Adams, B., Iyengar, G., Neti, C., Nock, H., Amir, A., Permuter, H., Srinivasan, S., Dorai, C., Jaimes, A., Lang, C., Lin, C.,Y., Natsev, A., Naphade, M., Smith, J., Tseng, B., Ghosal, S., Singh, R., Ashwin, T., and Zhang, D., 2003. IBM research TREC 2002 video retrieval system. In *Voorhees, E. and Buckland, L., editors, Information Technology: The Eleventh Text Retrieval Conference.* TREC 2002, NIST Special Publication 500-251, pp. 289-298.

ADL, The Advanced Distributed Learning initiative. http://www.adlnet.org.

Bersin, J., 2003, What Works in Blended Learning, In *Learning Circuits, July.* Retrieved September 5, 2005 from http://www.learningcircuits.org/2003/jul2003/.

Brodie, M., L., Stonebraker, M., 1995. *Migrating Legacy Systems*, Morgan Kaufmann Publishers inc.

COM, Component Object Model, http://www.microsoft.com/com/default.mspx

Deshpande, S., G., Hwang, J., N., 2001. A Real-time Interactive Virtual Classroom Multimedia Distance Learning System. In *IEEE Trans. on Multimedia*, vol. 3, n. 4, pp. 432-444.

Ford, R. ,M., Robson, C., Temple, D., Gerlach, M., 1997. Metrics for scene change detection in digital video sequences. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*. pp. 610-611.

Gerhard, J., Mayr, P., 2002. Competing in the E-Learning Environment: Strategies for Universities. In *Proceedings of the 35th Hawaii International Conference on System Sciences.*

Gonzales, R., C.,Woods, R., E., 2002. *Digital Image Processing*. Prentice Hall, 2nd edition, pp. 75-141.

He, L., Sanocki, E., Gupta, A., Grudin, J., 1999. Auto-Summarization of Audio-Video Presentations. In *ACM Multimedia*, pp. 489-498.

IEEE LTSC, The IEEE Learning Technology Standards Committee. http://ltsc.ieee.org.

IMS Global Learning Consortium. http://www.imsproject.org.

Lienhart, R. 1999. Comparison of Automatic Shot Boundary Detection Algorithms. In *Proc. SPIE* Vol. 3656 *Storage and Retrieval for Image and Video Databases VII*, pp 290–301, San Jose, CA, USA.

KOM, 2001. LOM-Editor Version 1.0, Technische Universität Darmstadt. Retrieved September 19, 2005 from http://www.multibook.de/lom/en/index.html

LOMgen, 2005. Learning Object Metadata Generator. Retrieved September 6, from http://www.cs.unb.ca/agentmatcher/LOMGen.html

Mukhopadhyay, S., Smith, B, 1999. Passive Capture and Structuring of Lectures. In *ACM Multimedia*, pp. 477-487

Nagasaka A., Tanaka, Y., 1991. Automatic video indexing and fullmotion search for object appearances. In *Proc. IFIP TC2NG2.6, Second Working Conf. on Visual Database Syst.*, pp. 113-127.

Ngo, C., W., Wang, F., Ting-Chuen Pong, T., C., 2003. Structuring lecture videos for distance learning applications. In *Multimedia Software Engineering, Proceedings. Fifth International Symposium on*, Vol., Iss., 10-12 Dec. pp: 215-222

Ngo, C., W., Pong, T., C., Huang, T., S., 2002. Detection of Slide Transition for Topic Indexing. In *Int. Conf. on Multimedia Expo*.

Pincas, A., 2003. Gradual and Simple Changes to incorporate ICT into the classroom. Retrieved August 8, 2005 from *http://www.elearningeuropa.info/index.php?page=doc&doc_id=4519&doclng=5*

Robson C., Ford R. M., Temple D., Gerlach M., 1997. Metrics for scene change detection in digital video sequence. In *ICMCS '97: Proceedings of the 1997 International, Conference on Multimedia Computing and Systems*, pp 610 - 611, IEEE Computer Society, Washington, DC, USA,

RELOAD, Reusable Learning Object Authoring and Delivering. Retrieved September 7, 2005 from http://www.reload.ac.uk/

Sethi, I. K., Nilesh, V. Patel, 1995. A Statistical Approach to Scene Change Detection. In *IS&T SPIE Proceedings on Storage and Retrieval for Image and Video Databases III*, Vol. 2420, 1995.

Singh, A., Boley, H., Bhavsar, V.C., 2004. A learning object metadata generator applied to computer science terminology. In *Presentation in Learning Objects Summit*, Fredericton, CA.

Smeaton A, Kraaij W, and Over P., 2003. TRECVID 2003 Text REtrieval Conference In *TRECVID Workshop*, Gaithersburg, Maryland

Yeo, B., L., Liu, B., 1995. Rapid scene analysis on compressed video, In *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5, pp. 533-544

Yusoff, Y., Christmas, W., Kittler, J., 1998. A study on automatic shot change detection. In *Lecture Notes in Computer Science*.