Indexing an Archive of Streaming Media Educational Components

Lonnie Harvel, Monson H. Hayes, Yu-Xi Lim, Jialin Tian, and Sankeun Lee
School of Electrical and Computer Engineering
Center for Distributed Engineering Education
Georgia Institute of Technology
Atlanta, GA 30332-0250
+(1) 404.894.2958
mhh3@eedsp.gatech.edu

Abstract – In this paper we present some work on indexing streaming media content that is developed for educational delivery over the Internet or by CD-ROM. First, we describe a tool that we have developed for automatically indexing a set of PowerPoint slides. This paper also describes how, once the lecture information (keywords) from these slides is loaded into a database that contains information from all other courses, a user can perform a keyword search over that lecture, or over all lectures for the course.

INTRODUCTION

Georgia Tech, along with a number of other Universities, is witnessing an explosion in the delivery of courses to distance learning students via the Internet or, when bandwidth issues become a concern, by CD-ROM. For example, as a part of the Georgia Tech Regional Engineering Program (GTREP), Georgia Tech is delivering undergraduate electrical engineering courses to other campuses located in the State of Georgia. In addition, Georgia Tech is now offering on-line Masters Degree programs in Electrical and Computer Engineering and in Mechanical Engineering. As more and more faculty become involved in distance learning, and as an increased number of students are finding themselves accessing multimedia content over the Internet, the expectations and demands from both groups will increase. For example, instructors are finding an increasing need of tools that will assist them in the creation and assembly of multimedia content, and students are beginning to expect advanced search and query capabilities, interactivity with the multimedia content, communication with the instructor and other students, advanced hyper linking to related topics, course navigation that is tailored to different learning styles, and on-line assessment. In this paper, we describe some of our recent activities in building an automatic indexing program that allows the student to search over textual content for specific words and phrases, both within a given course, and across multiple courses. Specifically, we present our work on indexing streaming media content that is developed for educational delivery over the Internet or by CD-ROM. First, we describe a tool that we have developed for automatically indexing a set of PowerPoint slides, or slides that have been saved and stored in postscript or PDF

"Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition

format. This paper also describes how, once the lecture information (keywords) from these slides is loaded into a database that contains information from all other courses, a user can perform a keyword search over that lecture, or over all lectures for the course.

COURSE INDEXING

A typical course at Georgia Tech runs for fifteen weeks, and consists of 45 lecture hours. The model that we have been using for Internet delivery of courses is that they are to be "packaged" into short "lecture modules" that last no longer than fifteen minutes and, in some cases, last as short as five minutes. Thus, a typical semester course may consist of well over a hundred modules, each with its own set of slides, audio, video, and exercises. Using the Georgia Tech software tool, inFusion, over a thousand components have been created and stored. Each of these modules is between seven and twenty minutes long and are usually focused on a single topic. They are comprised of streaming video, audio, slides, and annotations. Through live capture of traditional courses, using the eClass system of Brotherton and Abowd, an additional 2900 lectures have been archived[4].

This amount of recorded or captured information can be difficult to search and manage. This situation is aggravated by the fact that students often take several courses each term and many courses over their entire academic experience. The capability of searching the content of many different courses may help a student find a relevant piece of information, or see how different courses are related along a common topic. Such a search can be used in different contexts, such as over a single course, or over several course modules over an extended period of time. A user interface for searching can be made available in several places, such as in the syllabus page of a course, or in the presentation interface of each lecture. Relevance based searches can also be used to assist faculty in the development of future lectures.

Automatic Indexing of the Archive

Though SQL database queries could be used to find matching files. Our goal is not only to locate slides in the archive, which contain a given word, but also to locate modules in the collection that are relevant to the word. Therefore, we have chosen to take an information retrieval approach to the problem. In this section of the paper, we briefly describe the various components of the indexing and retrieval engine.

Software Functionality

The Automatic Indexing program offers a combination of data processing and data management tools. The indexing feature provides a solution for extracting and sorting the contents of a presentation. This portion of the program will take a PowerPoint or postscript presentation file as its input, iterate through each slide, and print the formatted text to an output file. Each word in the output file will have a list of tags attached to it, where these tags contain information on slide number, course number, lecture number, date, and weighting factor associated with that particular word. Another input for the program is a "stop list", which is described as a list of words that are frequently used, but contain no significant information. By comparing with the contents of this list, matched patterns will be eliminated from the formatted output. The user can select or modify a pre-complied stop list, or choose a new set of words. Several extra functions are made available for a PowerPoint input file. For example, the user has the option to extract slide titles from a presentation, and save them into a separate output file. When a Realpix clip file is given, it is possible to obtain the animation time of each slide. The database function combines a number of formatted text files into a single database file that is suitable for further process. This is particularly useful for creating a search engine program. In addition, this feature allows the

user to build a new database, or to add files to an existing database. The program is also capable of detecting and handling repeated modules.

Software Structure

The AutoIndex program is composed of three major sections: the graphical user interface, the Office automation, and text processing. The graphical user interface section contains visual displays that serve as the primary communication tools between the user and the application. They are responsible for collecting information about a presentation, sending event notifications, getting users' responses and displaying error messages. For processing a PowerPoint presentation, the AutoIndex program takes advantages of the COM (Component Object Model) technology by adapting the pre-built PowerPoint components. Since the PowerPoint application is built on a component-based architecture, its object models and objects can be imported when creating a custom application. The availability of the **Slides** collection and **Slide** objects has made it feasible for the Indexing program to obtain slide information, animation timings, and raw text from a presentation. The core section of this program is the text processing section. It is responsible for pattern matching, sorting, and indexing of the raw text input that has been extracted from a presentation file. The Perl language was chosen to perform these tasks since it is known to be one of the most effective tools for manipulating text and files. The text processor can be a stand-alone application if a raw text file is supplied, however, it is not a windows-based program. Finally, the text processing procedure is linked to the graphic interface via Windows API (Application Peripheral Interface) calls.

Index Structure

In engineering a powerful and effective information retrieval system, we can consider accessing content that comes from a variety of different sources. Initially, however, our project is only concerned with extracting and accessing information, i.e., words, that appears in prepared slides (postscript). The Classroom 2000 Project (now known as eClass), on the other hand, considered a variety of other sources, including transcriptions of handwritten information, transcriptions of voice, titles of web pages that have been visited during a lecture, and annotation that is added by the instructor after a lecture [5,6,7]. Once the lecture information (keywords) has been loaded into a database that contains all of the information from all other courses, a user can initiate a search over that lecture, or over all lectures for the course.

A major goal in our work was to create an indexing system that would allow for the integration of distributed indices. To accomplish this, each module or component is indexed separately. The individual indices are merged to form a single index as modules and components are merged to form lectures or courses. This also allows for the creation of “on-the-fly” indices to meet special user requirements. In order to preserve the independence of the index, term weighting was done locally, without recourse to global data from the collection. That is, with the exception of the stop list, which is maintained as part of the overall collection. For the purpose of this work, the weighting scheme is explicit and contextual. The AutoIndex program applies a pre-determined significance weight to a term based on its location within the module. Weights are set between zero and one. Zero being "not present" and 1 being the “most important”. Based on the context of a term, we set the first slide title words to 1, the other slide title words to 0.8, and other words to 0.5. When we do weight comparisons, we assume that unlisted words have a weight of 0 for that slide.

```
* Course_Number: 6254
* Lecture_Number: 4200
* Date: 3/15/01
* Input_file:
C:\latex\Education\Courses_On_Line\Statistical_DSP\Animations\4200 -
Least Squares Direct Method Animation.ppt
* Input_RP_file: C:\latex\Education\Courses_On_Line\Statistical_DSP\CD
Files\4200\slides.rp
* Output_file:
C:\latex\Education\Courses_On_Line\Statistical_DSP\Animations\4200.txt
* Created_On: Thu Mar 22 08:42:35 2001
Least              1      1    6254     4200  3/15/01     0:0:0.0
Squares            1      1    6254     4200  3/15/01      0:0:0.0
Signal             1      1    6254     4200  3/15/01     0:0:0.0
Modeling           1      1    6254     4200  3/15/01     0:0:0.0
Direct             1      1    6254     4200  3/15/01     0:0:0.0
Method             1      1    6254     4200  3/1 5/01     0:0:0.0
Statistical        0.5    1    6254     4200  3/15/01     0:0:0.0
Digital            0.5    1    6254     4200  3/15/01     0:0:0.0
Signal             0.5    1    6254     4200  3/15/01     0:0:0.0
Processing         0.5    1    6254     4 200  3/15/01     0:0:0.0
ECE                0.5    1    6254     4200  3/15/01     0:0:0.0
Module             0.5    1    6254     4200  3/15/01     0:0:0.0
Approach           0.8    2    6254     4200  3/15/01     0:0:43.933
Signal             0.8    2     6254     4200  3/15/01     0:0:43.933
Modeling           0.8    2    6254     4200  3/15/01     0:0:43.933
Our                0.5    2    6254     4200  3/15/01     0:0:43.933
approach           0.5    2    6254     4200  3/15/01     0:0:43.933
approximate        0.5    2    6254     4200  3/15/01     0:0:43.933
signal             0.5    2    6254     4200  3/15/01     0:0:43.933
```

Figure 1: Part of an index file.

Since we have designed each module to teach one major concept, this form of weighting is an effective method of topic identification. At this point, phrases are not indexed. An aggregated result from multi-term retrieval is used to identify terms within the same slide.

Presenting the Search Results

The user interface is extremely important for effective retrieval. We currently provide two types of search interfaces over the collection. Both interfaces are designed with a dual search/browse mentality and they are combined into one application. In other words, they are designed to support individuals that know exactly what they are looking for as well as those that are hunting for the topic that they need. The first interface, shown in Figure 2, is text based and engineered for speed. Selecting the "Details" tab activates this interface. Users begin typing a word into the text input box. The system begins to show partial matches as the word is typed. Selecting one of the partial matches will show all of the relevant slides in the Details box. Weights are shown as an integral instead of a decimal, so all weights have been multiplied by 10. Once one of the relevant slides is selected, one can choose to launch either the "Large" lecture view or the "Medium" lecture view. Large views are designed for 1024 x 768 displays while Medium views are designed for 800 x 600 displays.

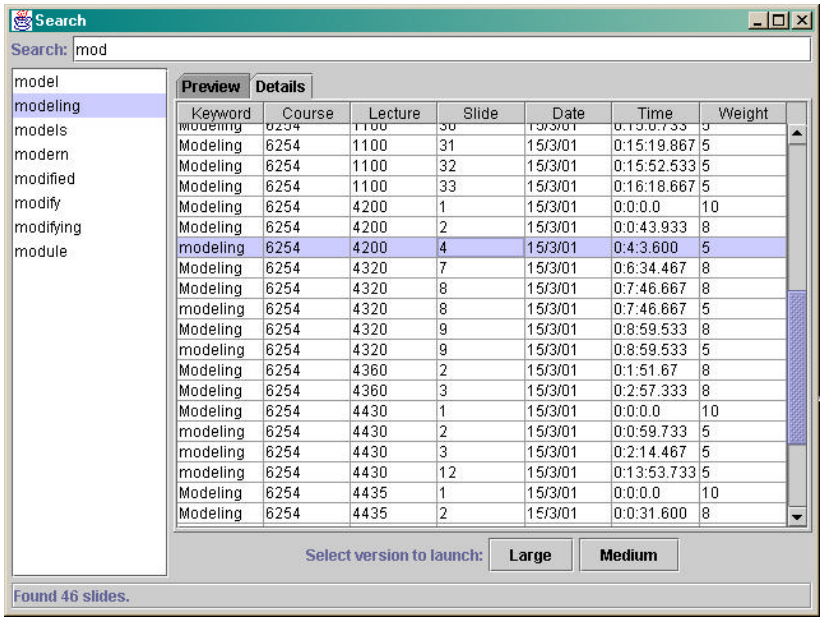

FIGURE 2: Text based search interface.

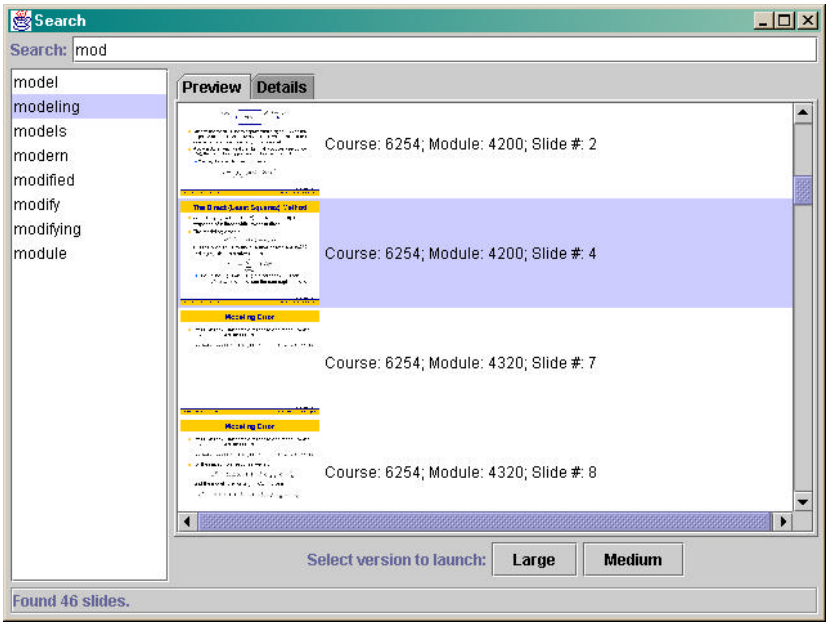

FIGURE 3: Thumbnail of candidate slides.

Deploying the System

During the Spring semester of 2002, our CD-ROM based courses will be delivered with our search capability. All courses, from this point forward, will include this capability. The index included with the CD-ROM will only present information related to the modules available on the CD-ROM. Access to the search feature in included in the opening page.

Also, the online archive has been constructed and is being populated with the newly constructed modules. A new version of the search engine is being developed that is more

appropriate for web-based access. The on-line archive will allow for searching over all of the modules in the archive, where permission has been granted. Faculty has the option to make their modules only available to their current students. So far, none of our faculty has chosen that option.

Future Work

As the archive of modules grows, many new project are possible. Currently, we are engaged in three new projects based on these indexed modules. These efforts are proceeding concurrently.

A major effort is the creation of a tool that allows for individuals to construct a course by using the modules currently available in the archive. Topic based searches would generate a candidate list of modules. These modules could then be arranged, using a timeline metaphor, into the final course. The system would allow for the inclusion of other media types to support the streaming module. Other media would include, Flash animation sequences, interactive exercises, and self tests. This system would be used for the construction of new courses, revising and updating existing courses, and the creation of short courses. There are obvious intellectual property issues inherent in this system, and they are being addressed on a faculty-by-faculty basis.

Another project combines the archive with online testing. Our goal is create a web-based system that will allow our new graduate students to test their knowledge in preparation for the Preliminary Exam. Based on their testing, they will be directed to modules that relate to their areas of weakness. This is intended to supplement their other study efforts in preparation for the exam. It will also make them aware of the existence and contents of the archive for their own exploration.

Finally, we hope to develop an interface that will allow for a user to visualize the "topical landscape" of an entire course or collection. The underlying data structure for this application is a topical bitmap built from high scoring terms in the collection. This is similar to the Hierarchical Axes work being done by Ben Shneiderman (et. al.) at the University of Maryland.[8] However, in our case, we are using automatically detected topics instead of structured hierarchies.

References

[1] M.H. Hayes, "Some approaches to Internet distance learning with streaming media", *IEEE Second Workshop on Multimedia Signal Processing*, pp. 514-519, Los Angeles, CA, Dec. 1998.

[2] M.H. Hayes and L.D. Harvel, "Distance learning into the 21st century", *Proc. ASEE Workshop*, Charlotte, NC, June 1999.

[3] J. Jackson, D. Anderson, and M. Hayes III, "Effective and Efficient Distance Learning Over the Internet: Tools and Techniques," *Proc. International Conference on Engineering Education*, Taiwan, August 2000.

[4] Abowd, G.D. Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal*. **38**(4):508–530, October 1999. See www.research.ibm.com/journal/sj/384/abowd.html.

[5] Brotherton, J.A, G.D. Abowd and J. Bhalodhia. Automated capture, integration and visualization of multiple media streams. *Proceedings of the IEEE Multimedia and Computing Systems '98 Conference*, July 1998 pp. 54–63.

[6] D. Anderson, L. Harvel, M. Hayes, J. Jackson, and M. Pimentel; Internet Course Delivery – Making it Easier and More Effective, *Proceedings of International Conference on Multimedia and Expo*, 2000.

[7] Gregory D. Abowd, Lonnie D. Harvel and Jason A. Brotherton; *Building a Digital Library of Captured Educational Experiences;* Invited paper for the 2000 International Conference on Digital LIbraries, Kyoto, Japan, November 13-16, 2000.

[8] B. Shneiderman, D. Feldman, A. Rose, X. F. Grau; *Visualizing Digital Library Search Results with Categorical and Hierarchical Axes;* Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, Texas, June 2-7, 2000.

Lonnie Harvel

Lonnie Harvel is a senior research scientist in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. He is the director of the Digital Media Lab in ECE, the associate director of the enter for Distributed Engineering Education, and a member of the Graphics, Visualization and Usability Center. His current interests in education include computer enhanced learning spaces and distributed learning. His other research interests are in context-aware computing and future computing environments.

Monson H. Hayes

Dr. Hayes is a Professor of Electrical and Computer Engineering at the Georgia Institute of Technology in Atlanta, Georgia. He received his B.A. degree in Physics from the University of California, Berkeley, and his M.S.E.E. and Sc.D. degrees in Electrical Engineering and Computer Science from M.I.T. His research interests are in digital signal processing with applications in image and video processing, nonlinear signal processing and modeling, stereographic image processing, image and video coding, image synthesis, multimedia signal processing, and DSP education. He is currently in charge of putting the Georgia Tech Masters Degree program in ECE on-line. He has contributed more than 100 articles to journals and conference proceedings, and is the author of two textbooks, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, 1996, and *Schaum's Outline Series on Digital Signal Processing*, McGraw-Hill, New York, 1999

Yu-Xi Lim

Yu-Xi Lim is pursuing a degree in Computer Engineering at the Georgia Institute of Technology. He is currently involved in other information processing projects. His interests also include multimedia, communications and networking.

Jialin Tian

Jialin Tian received the B.S. degree in electrical engineering from North Carolina State University in 1997, and the M.S.E.E. degree from Purdue University in 1999. She was a product engineer at Motorola Semiconductor Product Sector in 1998. She is currently employed as a research assistant in the Center for Signal and Image Processing at Georgia Institute of Technology while working toward the Ph. D. degree.

Sangkeun Lee

Sangkeun Lee received the B.S degree and the M.S degree in electronics engineering from the Chung-Ang University, Korea in 1996 and 1999, respectively. Since 1999, he has been a Ph.D student in the School of Electric and Computer Engineering at Georgia Institute of Technology. His current interest is mainly focused on Content-based Video Abstraction. His research interests include multimedia databases, content-based video abstraction, software-development environments, and human-computer interaction.