Universal Multimedia Access and Semantic Summarization for Presentations

by

Mohammed Ajmal

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical & Computer Engineering
University of Toronto

Mohammed Ajmal

Master of Applied Science, 2007

Graduate Department of Electrical & Computer Engineering

University of Toronto

Abstract

Multimedia content is being both disseminated and consumed on a larger scale than ever before. In particular, the growth of collaborative applications (e.g. Distance Learning (DL)) means that multimedia-rich audio-visual presentations are increasingly available on the Internet. At the same time, mobile devices capable of processing multimedia applications have gained popularity. Consequently, designers of multimedia systems face two major challenges driven by these factors: First, users require access to personalized content seamlessly over differing networks and on virtually any terminal device. Second, users require a means of navigating and digesting the vast content efficiently.

This work proposes a presentation summarization framework for audio-visual presentations in a DL environment which addresses the aforementioned challenges. The framework is designed based on emerging international standards to provide a Universal Multimedia Access (UMA) solution that is able to deliver content to heterogeneous users. A novel summarization scheme is proposed which combines information from multiple sources such as audio data and presentation slides to improve accessibility of presentation archives for users.

Acknowledgements

First, I would like to thank my supervisor, Professor K. N. Plataniotis, for his guidance and assistance throughout my graduate studies. His recommendations and demand for excellence have been invaluable toward the improvement of this thesis. Thank you.

I would also like to acknowledge the members of my defense committee, Professor D. Hatzinakos, Professor W. J. MacLean and the chair, Professor A. Moshovos, for both their time and their efforts in providing feedback on this work.

I have been lucky to meet and work with some very talented people at U of T. In particular, I wish to thank Azadeh, Karl, Abida and Jason for all their help throughout the past two years. They have been the best colleagues imaginable.

None of my achievements would be possible without the endless love and support of my parents. This thesis is as much a product of their hard work as my own. I want to thank my sister for always being willing to help, and for entertaining me to no end. And finally, to my wife: thank you for all your love and encouragement. You are truly amazing.

Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
ASR	Automatic Speech Recognition
AV	Audio-Visual
CBIR	Content-Based Image Retrieval
D	Descriptor
DDL	Description Definition Language
DI	Digital Item
DIA	Digital Item Adaptation
DIBO	Digital Item Base Operation
DID	Digital Item Declaration
DIDL	Digital Item Description Language
DIM	Digital Item Method
DIP	Digital Item Processing
DIXO	Digital Item Extension Operation
DS	Description Scheme
DL	Distance Learning
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
IR	Information Retrieval
IS	International Standard

JPEG	Joint Photographic Experts Group
MAD	Motion Activity Descriptor
MPEG	Moving Picture Experts Group
OCR	Optical Character Recognition
PDA	Personal Digital Assistant
QoS	Quality of Service
SOLA	Synchronized Overlap-Add
TF	Term Frequency
UED	Usage Environment Description
UMA	Universal Multimedia Access
UofT	University of Toronto
VSM	Vector Space Model
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transform

Chapter 1

Introduction

The amount of multimedia content available to users on the Internet has risen dramatically in recent years. Specifically, collaborative applications such as Distance Learning (DL) (sometimes referred to as ‘E-Learning’) and web meetings have thrived in light of the technological advancements in end-user devices and the ubiquity of the Internet. There has been a particular focus on DL systems due to the benefits offered by distance learning over traditional teaching methodologies [1]. Chief among these advantages are convenience and flexibility to students in terms of ability to learn independently at a self-imposed pace from any physical location [1, 2].

Within the scope of DL systems, research has been conducted into the design of ‘learning systems’ [3, 4]. This research however, has been shaped mainly from an educator’s viewpoint: the goal there is to design a system that is pedagogically robust and affords users the best conditions to learn while offering teachers the best tools to produce material for consumption. By contrast, the aim of this thesis is to design an innovative system and algorithm which addresses the limitations of existing learning systems from a user’s viewpoint.

This chapter explores the shortcomings of current DL applications which expose a need for audio-visual (AV) presentation summarization and Universal Multimedia Access (UMA) within the DL context motivating the research undertaken in this thesis. The key technical problem involved in the development of an AV presentation summarization algorithm is discussed, followed by an overview of the proposed solution. Finally, the chapter concludes with the specific contributions made in this work and an outline for the remainder of the thesis.

1.1 Motivation for Audio-Visual Presentation Summarization & Universal Multimedia Access

One of the distinguishing features of a collaborative setting is that participants in the activity may be at geographically different locations. In a typical Distance Learning scenario, for example, a webcasting setup is used to stream captured video, along with accompanying audio, any optional presentations slides and miscellaneous text (e.g. from a live user chat) in real-time over the Internet to a remote learner [5]. Upon conclusion of the live event, the multimedia material is typically archived, and made available online so that those users who were unable to join the proceedings in real-time have an opportunity to benefit from the material at their convenience. As a result, archives of multimedia-rich audio-visual presentations are commonly found for public consumption [6–8].

As described above, there are two distinct modes of operation in a DL setting: *on-line* access of a current event that is proceeding live in real-time, or *off-line* access of an archived presentation. The primary difference between these two modes of operation is in the way in which users are able to interact with the presentations. During on-line access of a real-time event, for example, a user does not have access to the entire content comprising the presentation. Consequently, a user is unable to 'skip-ahead' to particular points of interest, or otherwise skim the contents to evaluate their interest in the presentation. In this on-line setting, one enhancement particularly useful to a remote user of the system would be *pause and resume* functionality which would enable a user to step away from the event momentarily. When the user rejoins the event, the content missed by the user should be presented succinctly, retaining all the important information so there is no gap in knowledge. An AV presentation summarization scheme would achieve this objective, re-synchronizing a remote user with the live lecture. In this case, the summarization technique must be efficient and capable of near real-time operation.

This is to be contrasted with the off-line mode of operation. In this case, a user has the opportunity to navigate directly to specific portions of the presentation, and moreover, a user may be interested in an overview (or skim) of the presentation to judge its relevance to their needs. This thesis focuses on the off-line mode of operation, although where appropriate, the applicability of the proposed summarization system to the on-line mode is highlighted.

Ultimately, then, users face a daunting challenge in easily navigating and accessing presentation archives. Consider, for instance, a common task in which a user wishes to locate a particular presentation from a list of presentations generated in response to a user query. The situation is similar to a typical image search engine, in which a set of images is displayed to a user in response to a query. In the case of images, it is a relatively straightforward task to skim the images to locate the desired image. This task is complicated in presentation repositories because it is difficult to capture the gist of a presentation.

It is therefore necessary to provide tools to users which allow them to skim and find relevant content efficiently. This is an important problem because of the pervasiveness of presentation material on the Internet. In addition, as described above, there has been research in the area of learning systems, and such systems would benefit greatly through the addition of browsing technology. Current state-of-the-art systems provide only rudimentary tools for browsing large presentation repositories, failing to address the needs of users. Moreover, presentation slides are often completely ignored as a source of semantic information for both retrieval and summarization purposes [6].

System designers must ensure that such content is available for consumption to a heterogeneous set of terminal devices. A multitude of new devices capable of processing multimedia content such as Personal Digital Assistants (PDAs) and SmartPhones have risen in popularity. Wireless networks are becoming more prevalent and affordable in both industrial and academic institutions. In short, there is an ever-growing desire to gain access to information-rich multimedia content through any terminal, anywhere and at anytime. This has spawned an initiative known as Universal Multimedia Access (UMA) (see Figure 1.1) which seeks to provide the best user Quality of Service

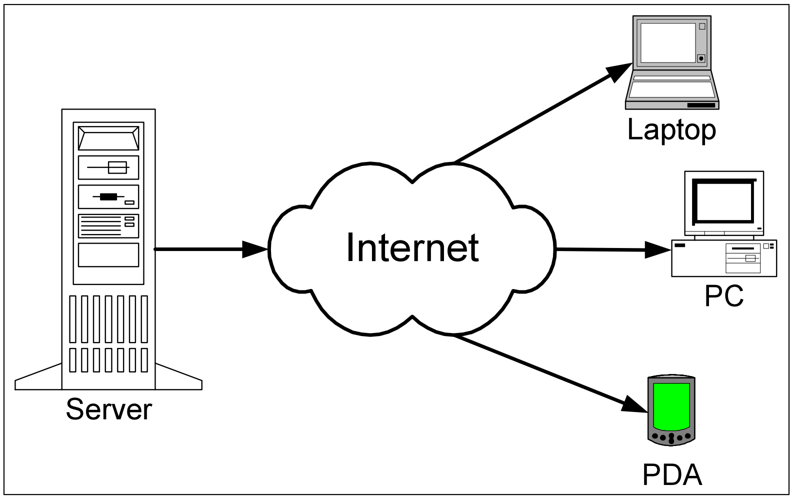


Figure 1.1: Schematic representation of a UMA system.

(QoS) while enabling access to any multimedia information by any terminal over any network [9]. Current systems [3, 6, 7] largely ignore this concern in the context of presentation repositories for DL systems.

This thesis proposes an audio-visual presentation summarization framework in a DL environment adhering to the principles of Universal Multimedia Access (UMA) to address these challenges.

1.2 Key Technical Challenges

The summarization problem in any domain (e.g. text, audio, video) is a challenging task. The major complication arises from the fact that high-level human concepts, such as ‘importance’, are difficult to ascertain from low-level features extracted from the underlying content. Accordingly, the primary goal of summarization systems, in general, is to devise novel methods to bridge this semantic gap and produce effective summaries.

In the area of video summarization, previous works have sought to overcome the semantic gap using a multi-modal approach in which the combined effect of audio and video analysis is used to guide summarization [10–13]. In a similar vein, a multi-modal approach holds the most promise for presentation summarization [12].

For the problem of presentation summarization, the main sources of information available to a summarization system are the slides and the accompanying audio. As argued in [6], the video in such presentations is usually a low-resolution face shot of the presenter. Thus, it does not provide substantial information for summarization as compared with the audio domain and slide material. In [6], the video channel is ignored in the development of a presentation summarization algorithm. Although it is possible to apply certain techniques such as gesture analysis [14] and motion activity analysis [15] to the video content, video processing is computationally expensive and marginal information is gained from presentation video. Consequently, this thesis focuses on the audio channel and presentation slides as primary sources of information for the purpose of summarization.

At a high level, the principal technical challenge to AV presentation summarization, then, is identifying *important* segments from the presentation that should be retained in a summary. This in turn requires analysis of several sources of information mentioned above, including the audio channel and presentation slides. Thus, from an engineering perspective, this thesis addresses the challenge of extracting importance information from audio data and from DL presentation slides to guide AV presentation summarization. An additional challenge discussed in this work is the fusion of importance data over multiple sources of information. Finally, the design of any AV presentation summarization system is constrained to adhere to the UMA paradigm as dictated by the needs of present day users.

1.3 Overview of Proposed Solution

This section briefly outlines the major components in the proposed framework. Figure 1.2 provides an overview of the framework. The key component of the framework is the summarization engine, which operates on an audio-visual presentation (also referred to as 'informational-talk' [6]) guided by data pertaining to the user environment and preferences to produce a summarized version of the presentation.

A general AV summarization system can be partitioned into three tasks: *segmentation*, *analysis*

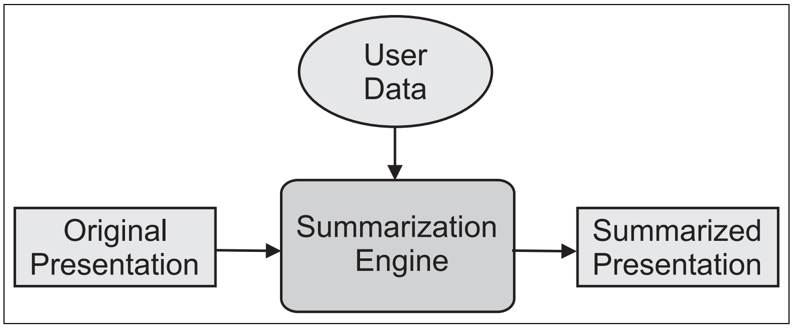


Figure 1.2: Overview of the proposed framework.

and *presentation* [14]. The second task, analysis, implies deeper semantic comprehension of the content [16], for example, determining *important* or *informative* portions of content, and is critical to the performance of summarization systems. Analysis is usually performed on a domain-specific level [14, 17]; in this thesis, the domain is restricted to AV presentations.

A novel summarization engine which extracts semantic information from multiple sources including audio data and presentation slides to guide summarization is proposed for AV presentations. In particular, this thesis demonstrates the applicability of spectral entropy information in the audio domain in identifying important segments of speech. Furthermore, an innovative approach is presented for XML-based semantic processing of presentation slides based on a fuzzy set framework, which leads to improvements in slide retrieval and enables slide transcoding - that is, the excision of data from slides to allow small-screen devices to view presentation slides. Finally, an AV presentation summary is generated through a process of opinion-fusion. Specifically, by using multi-modal analysis of audio data, presentation slides and a priori heuristics, opinions are formed as to the importance of a segment of the presentation. Then, these opinions are combined using a weighted summation operator to compute a final importance score for each segment of the presentation. A greedy strategy of retaining the highest scoring segments results in a summary for the presentation.

The UMA initiative has changed the way in which AV presentation content must be processed from content creation to content consumption. In keeping with the UMA paradigm, the entire framework is designed based on the emerging international standard MPEG-21 [18] to provide a UMA solution to the problem of presentation summarization. Figure 1.3 provides an overview of the

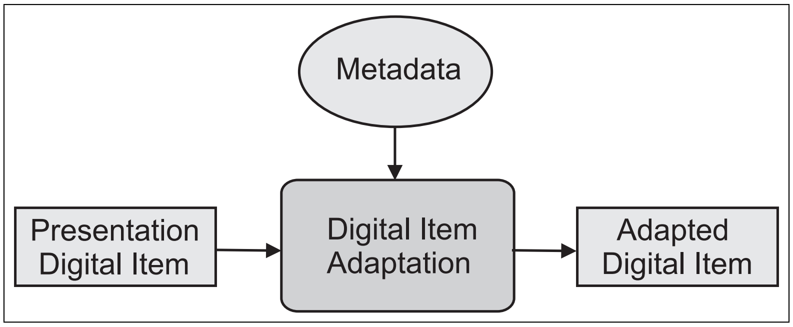


Figure 1.3: UMA system overview of proposed solution.

proposed framework using the language of MPEG-21, as opposed to the generic overview provided in Figure 1.2. In this context, AV presentations are encapsulated within a Digital Item (DI), which is then processed by a Digital Item Adaptation (DIA) engine guided by metadata describing users, the environment as well as the multimedia components of the presentation. Content analysis is performed on the audio information to extract metadata which is then packaged within the proposed Digital Item.

1.4 Contributions

This thesis focuses on the problem of audio-visual presentation summarization in a DL context, within a UMA paradigm to allow users to interact with presentation repositories efficiently. A novel presentation summarization framework is developed which adheres to the principles of UMA. The framework works by first extracting importance information from multiple sources, and then combining these to give an overall indication of the importance of various segments of the AV presentation. The specific contributions of this work are:

1. Development of an AV presentation summarization algorithm which exploits information in multiple domains.

2. Design of a UMA system based on international standards such as MPEG-7 and MPEG-21.

The proposed AV presentation summarization algorithm represents a novel contribution both in the way in which the individual sources of information are analyzed and in the way in which a multimodal approach is used for presentation summarization. Previous work in the area of presentation summarization has relied on difficult pitch estimation techniques for audio analysis, and has entirely ignored the semantic content of presentation slides [6] although these contain valuable data to aid the summarization process. This work describes a spectral entropy based approach to audio analysis and extraction of important segments. Moreover, through the development of a slide analysis framework using fuzzy logic, it is possible to perform slide transcoding enabling the delivery of personalized slide content to users.

These contributions advance the state-of-the-art in AV presentation summarization by improving the quality of the summary produced, as demonstrated through a task-based user study. In addition, the slide analysis framework represents an improvement in the state-of-the-art XML based slide retrieval systems. Finally, the proposed summarization system represents a significant improvement in DL application systems by adhering to the UMA principle. This allows future systems to leverage emerging international standards to deliver tailored AV presentation content to DL users in an interoperable, seamless manner.

1.5 Outline of Thesis

The remainder of this thesis develops the proposed AV presentation summarization system and algorithm in detail. Chapter 2 reviews approaches in presentation summarization and the design of UMA systems to gain an understanding of current state-of-the-art approaches. The design of the presentation summarization framework within the UMA paradigm, along with the proposed summarization algorithm is detailed in Chapter 3. Chapter 4 details the methodology used to evaluate the effectiveness of the proposed summarization algorithm as a whole. Furthermore, results of various experiments as well as a discussion is presented in this chapter. Finally, conclusions and directions for future work are presented in Chapter 5.

Chapter 2

Prior Work

This thesis addresses the challenge of audio-visual presentation summarization within a Universal Multimedia Access (UMA) framework. The focus of this chapter is to discuss prior work in the area of AV presentation summarization as well as existing UMA systems. To that end, this chapter presents prior work in presentation summarization, audio analysis for summarization and presentation slide analysis.

Before embarking on this discussion, an overview of the *Multimedia Content Description Interface (MPEG-7)* and *Multimedia Framework (MPEG-21)* standards is provided. It is important to be familiar with these MPEG standards as they are the key enabling technologies for UMA systems. Where relevant, an emphasis is placed on those components of the technologies that relate to the summarization problem at hand. It should be noted that additional components of the standards will be introduced as needed in the remainder of this thesis.

2.1 UMA Systems - MPEG Standards

2.1.1 MPEG-7: Multimedia Content Description Interface

The MPEG-7 standard, formally known as *Multimedia Content Description Interface* [19, 20], is an international standard which includes tools to describe audio-visual information for a wide range of applications. The MPEG-7 standard was proposed by the Moving Pictures Expert Group (MPEG) and the first version of MPEG-7 attainted International Standard (IS) status in 2001. It is important to note that MPEG-7 is fundamentally different from previous MPEG-standards (such as MPEG-1, MPEG-2, MPEG-4). The focus of the latter standards is on coding and representation of audio-visual data [19], whereas the goal of MPEG-7 is to provide tools for *describing* multimedia content in an efficient and standardized manner.

MPEG-7 data is expressed using the Extensible Markup Language (XML) [21] and is commonly referred to as *metadata*. This metadata provides insight into the semantic meaning of the underlying multimedia content, which can be processed or utilized by various devices or computers independent of the content itself [22]. In fact, the main motivation behind the development of the MPEG-7 standard was to aid in the management, indexing, search and retrieval of content-rich multimedia, a task which is dramatically aided by MPEG-7 metadata.

There are three main components of the MPEG-7 standard, namely *Descriptors (D)*, *Description Schemes (DS)* and the *Description Definition Language (DDL)* [22]. Descriptors are designed for describing features extracted from the multimedia content. For example, descriptors can describe low-level audio-visual features such as colour, texture, motion and audio energy. In addition, high-level features such as events and information about storage media may be described using descriptors.

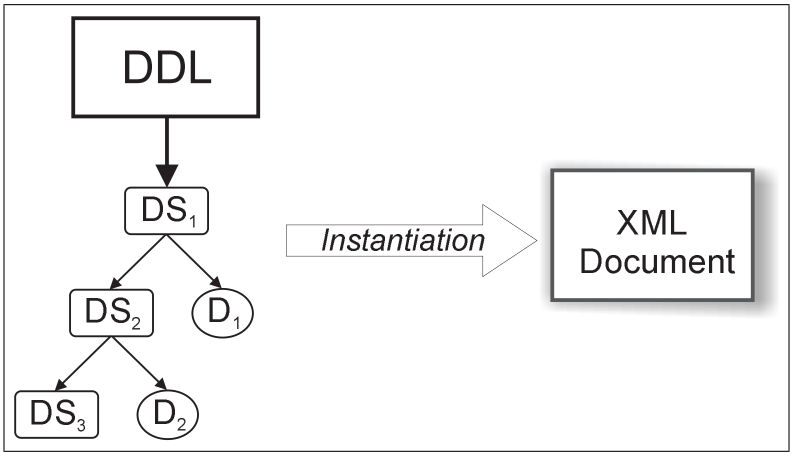


Figure 2.1: Relationship between Ds, DSs and DDL in MPEG-7.

Description schemes build on the notion of descriptors by combining individual descriptors and other description schemes into more complex structures. Description schemes also define the relationships between the various components of a particular description scheme.

Both the MPEG-7 descriptors and description schemes are defined using the MPEG-7 DDL, which itself is an extension of XML Schema Language [23] defining the syntax of valid XML descriptors and description schemes. An MPEG-7 description is generated for a particular piece of multimedia content by instantiating the MPEG-7 descriptors and descriptions schemes which conform with the syntax of the MPEG-7 DDL [22]. The power of MPEG-7 resides in the fact that the DDL may be used to create custom descriptors which are not already part of the MPEG-7 standard to describe certain content. By adhering to the rules defined by the DDL, any custom descriptors enjoy the same advantages as the existing MPEG-7 descriptors, namely greater interoperability due to a standard structure.

The function of MPEG-7 may finally be described as standardizing a core set of quantitative measures of audio-visual features (descriptors), the relationships between descriptors (description schemes) and a language for specifying custom descriptors and description schemes (Description Definition Language (DDL)) [24] (see Figure 2.1). The remainder of this section highlights some of the descriptors and description schemes relevant to the summarization problem.

The standard defines several visual descriptors, broadly grouped as *Colour Descriptors*, *Texture Descriptors*, *Shape Descriptors* and *Motion Descriptors* [22]. Likewise, several descriptors are defined for the audio domain. These are shown in Figure 2.2 (adapted from [20]).

Besides descriptors for audio-visual material, the MPEG-7 standard includes definitions for other useful tasks such as content organization and user interaction. One particular description scheme of interest is the `Summarization DS`. The purpose of this DS is to describe different compact representations of audio-visual content, by abstracting out the underlying key information. The `Summarization DS` links with the audio-visual content at the level of frames, and it can describe multiple summaries of the same audio-visual content at once, to provide potentially different levels of detail, or to highlight varying features or aspects of the content. The benefit of including links

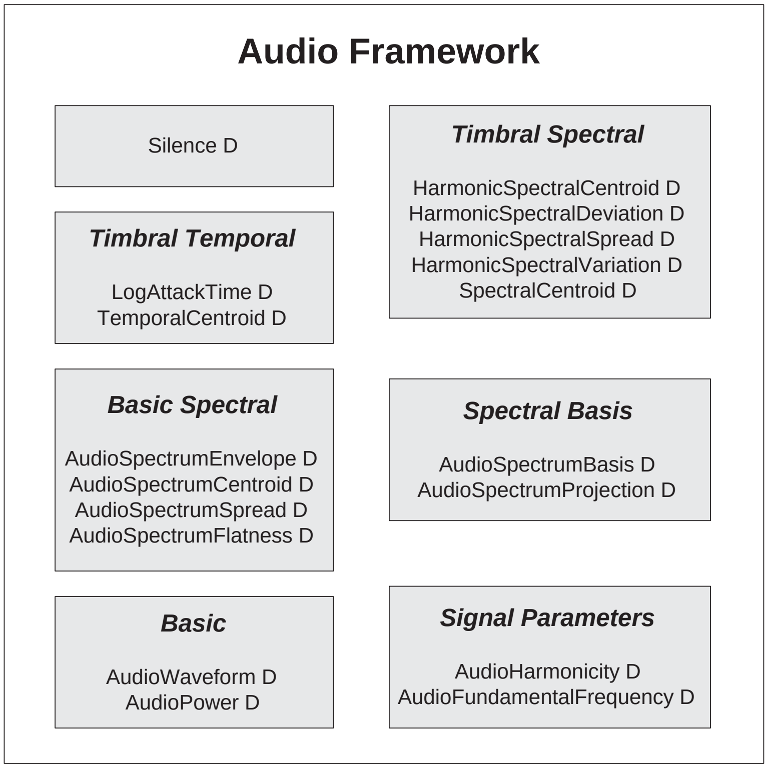


Figure 2.2: Overview of Audio Framework, including descriptors.

to the audio-visual content in the DS is that only one version of the content must be stored, and multiple summaries can be created for this same content.

There are two basic DSs within the `Summarization DS`: `HierarchicalSummary DS` and `SequentialSummary DS`. The `HierarchicalSummary DS` describes the organization of summaries into multiple levels. The building blocks of a `HierarchicalSummary DS` are temporal segments within the audio-visual content; these segments are in turn described by the `HighlightSegment DS`. Each `HighlightSegment DS` contains pointers to identify the associated key frames and key sounds. In addition, the `HighlightSegment DS` may contain textual annotation to provide semantic meaning to the overall highlight. Figure 2.3 shows a typical usage of the `HierarchicalSummary DS`. Note that the top-level DS contains two `HighlightSummary DSs`, an example of storing two different summaries of the same content within one DS. Here, the first summary contains four `HighlightSegment DSs` whereas the second summary contains only three. These two summaries may correspond to different themes in the video, or may simply be the result of summarization algorithms which emphasize different portions of the content in response to user input, for example.

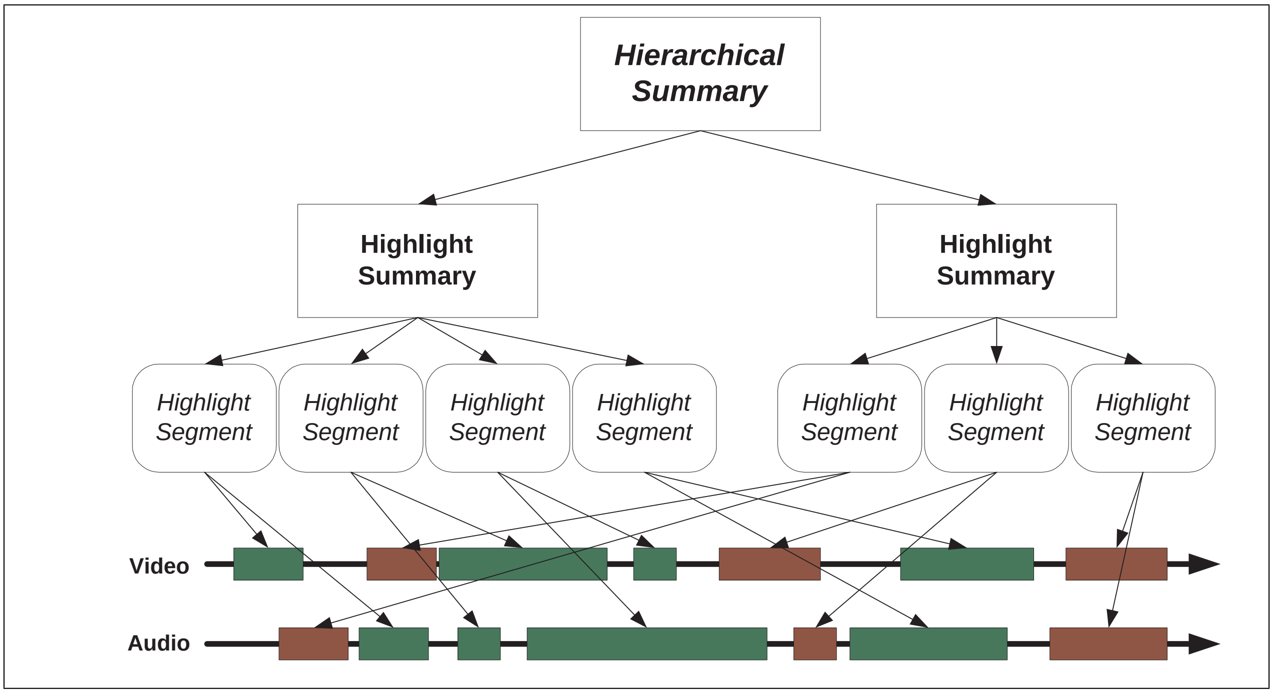


Figure 2.3: Overview of HierarchicalSummary DS, including two summaries.

A critical description scheme with regard to UMA is the `Variation DS` [9]. This DS describes variations in the multimedia content, for example, low-resolution versions, summaries, different languages or even modalities of the source content. This allows systems to make intelligent decisions with regard to the content presented to a user by matching terminal device properties and user preferences with the information contained in the `Variation DS`.

This subsection has provided a brief introduction to the MPEG-7 standard, describing its role in UMA systems. It is clear that MPEG-7 metadata can provide valuable information to a UMA system to guide application level processing of content. This section explained the `HierarchicalSummary DS` in depth, and its applicability to the summarization problem.

2.1.2 MPEG-21: Multimedia Framework

The MPEG-21 standard, formally known as *Multimedia Framework*, aims to standardize a framework to facilitate the transparent use of multimedia content across a multitude of networks and end-user devices [9, 18]. The MPEG-21 standard was proposed by MPEG in an attempt to fill in the perceived gaps in the multimedia delivery chain. Consequently, MPEG-21 standardizes a framework for packaging and processing multimedia resources [25] throughout the lifetime of the content, from creation to consumption. The overall aim of the framework is to allow universal delivery of multimedia content to a heterogeneous set of end users [26, 27].

The two fundamental concepts in MPEG-21 are the *User* (with a capitalized 'U') and the *Digital Item (DI).* A User in MPEG-21 is any entity which interacts with a Digital Item, such as individuals, institutions or governments [25,26]. This concept is larger than the idea of a traditional user; the notion of a User in MPEG-21 incorporates every participant in the delivery chain, from the point of content creation to content consumption. There is no distinction between a content provider and consumer when speaking of Users in the context of MPEG-21.

Digital Items (DIs) are the basic unit of transaction within the framework [25,26]. They can be viewed as an abstraction of multimedia content, packaging together not only the content itself, but any associated metadata (in particular, MPEG-7 metadata), identifiers, licenses and methods (see Figure 2.4) that enable interaction with the Digital Item. It is this notion of content abstraction that allows heterogeneous users to interact with the same content in a seamless fashion. The function of MPEG-21 may then be described as standardizing a framework to enable the interaction of Users where the object of interaction is a Digital Item.

The MPEG-21 standard currently consists of 18 parts. The three parts most relevant to UMA system design in the current setting are Part 2, *Digital Item Declaration (DID)* [28], Part 7, *Digital Item Adaptation (DIA)* [29] and Part 10, *Digital Item Processing (DIP)* [30] of the standard. Figure 2.5 presents a high level overview of the role of DID and DIA within a MPEG-21 compliant system. DID standardizes creation of a DI from the multimedia content, DIP enables dynamic interaction with the DI and DIA standardizes tools to process DIs to enable UMA.

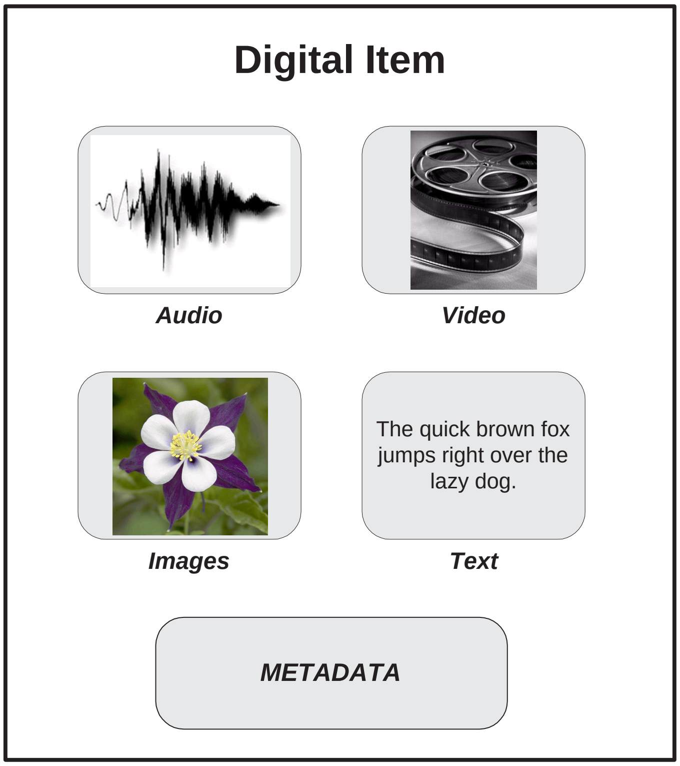


Figure 2.4: An MPEG-21 Digital Item: a collection of multimedia content and metadata.

Digital Item Declaration (DID) provides the structure to a Digital Item, by binding together resources and metadata. The representation of DIs in MPEG-21 is divided into 3 parts: *DID Model* – a set of abstract terms and concepts to form a model for defining DIs, *Representation* – the description of the syntax and semantics of each of the DID elements represented in XML, specifically using *Digital Item Description Language (DIDL)* and *Schema* – an XML Schema comprising the entire grammar of the DID representation in XML [28].

Figure 2.6 shows a simple, valid Digital Item declaration. This particular DI encapsulates the audio track from a single lecture. The details of the syntax are unimportant. The key points to note are the structured nature of the Digital Item, the ability to add descriptive content and the ability to link to multimedia content stored on a server, for example. It is also possible to embed the binary content within the DI itself.

With the above declaration, a DI corresponding to a single lecture has been created. As it

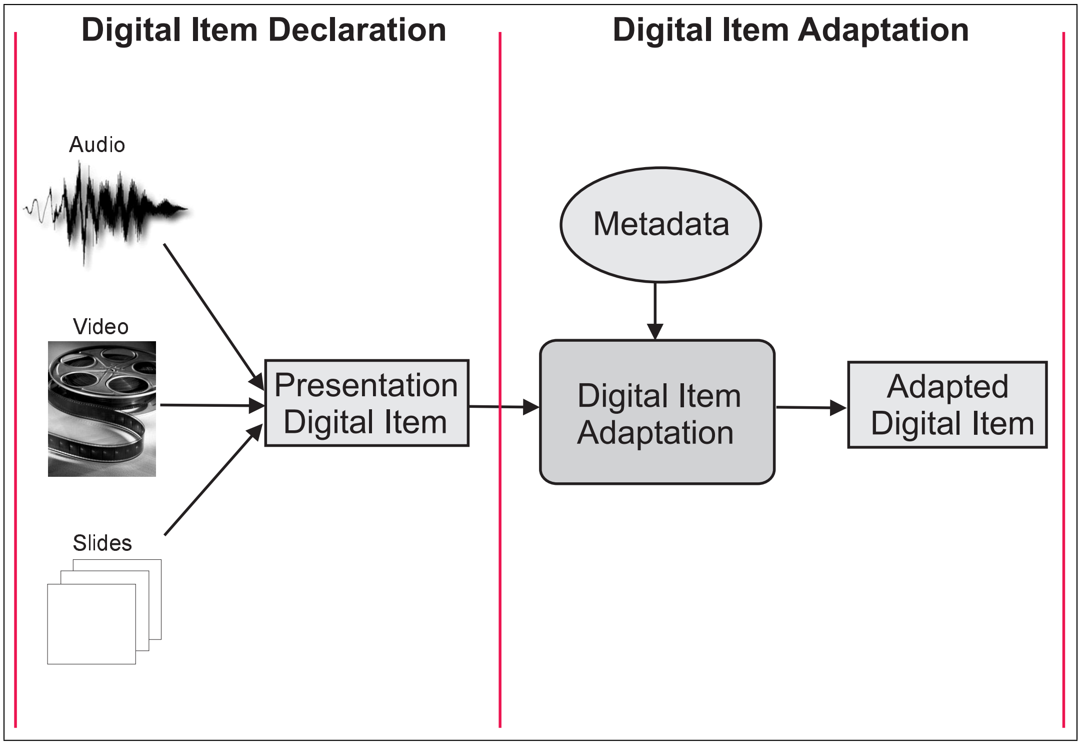
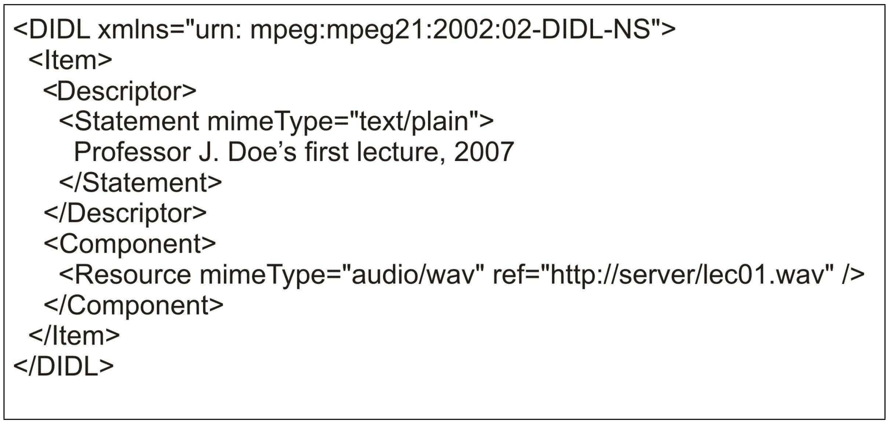


Figure 2.5: The role of DID and DIA in a MPEG-21 system.

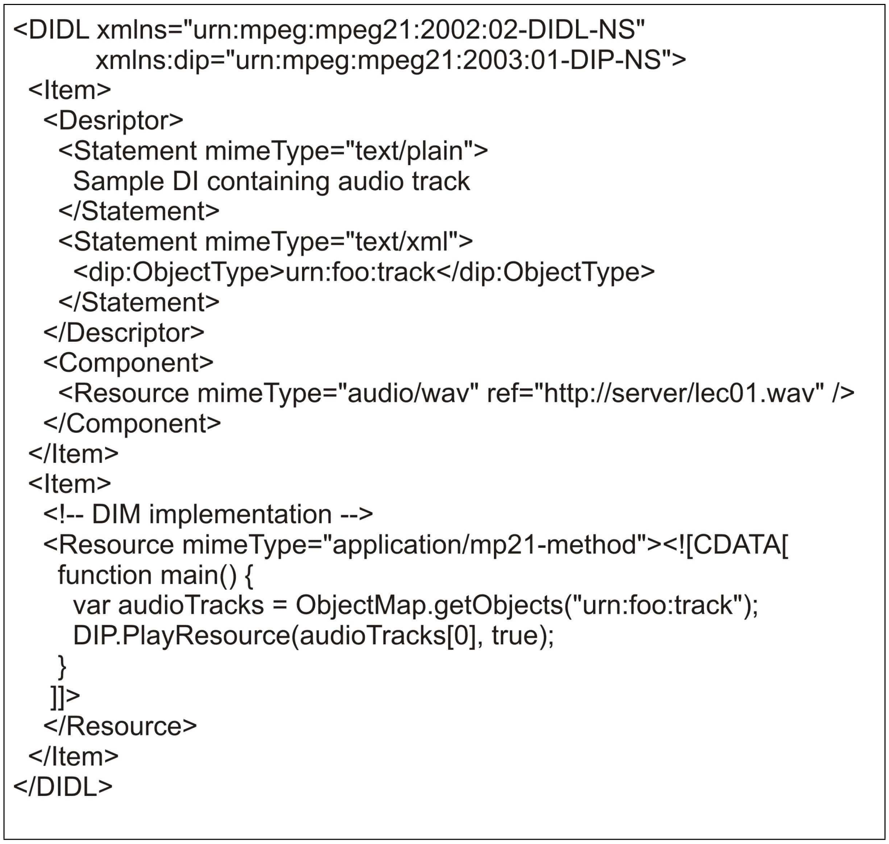


```
<DIDL xmlns="urn: mpeg:mpeg21:2002:02-DIDL-NS">
  <Item>
    <Descriptor>
      <Statement mimeType="text/plain">
        Professor J. Doe’s first lecture, 2007
      </Statement>
    </Descriptor>
    <Component>
      <Resource mimeType="audio/wav" ref="http://server/lec01.wav" />
    </Component>
  </Item>
</DIDL>
```

Figure 2.6: Simple Digital Item of an audio lecture.

stands, however, the DI is a strictly static entity. Of course, it may be the case that external programs (commonly referred to as *MPEG-21 peers* [30]) interact with the DI and process it, however the DI itself can perform no actions. In certain scenarios, it is helpful to add interactive, dynamic functionality to a Digital Item. MPEG-21 Digital Item Processing addresses exactly this need, and standardizes a manner to add such capability to a Digital Item [26, 30].

MPEG-21 DIP functionality is closely related with DID. When a DI is first declared using DID, the author has the ability to include *Digital Item Methods (DIMs)* into the declaration. These DIMs in turn contain calls to standardized application programming interfaces (APIs) in the MPEG-21 library. The collection of library functions offered to authors of DIMs are referred to as *Digital Item Base Operations (DIBOs)*. In case there does not exist a DIBO to accomplish a certain task, it is possible to declare custom functions - *Digital Item eXtension Operations (DIXOs)* - in a language of the author's choosing. Currently, MPEG-21 DIP supports only the Java programming language, however other languages will be included in future versions of the standard. In this manner, by adding DIMs to static DI, a dynamic entity is created.



```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
        xmlns:dip="urn:mpeg:mpeg21:2003:01-DIP-NS">
  <Item>
    <Desriptor>
      <Statement mimeType="text/plain">
        Sample DI containing audio track
      </Statement>
      <Statement mimeType="text/xml">
        <dip:ObjectType>urn:foo:track</dip:ObjectType>
      </Statement>
    </Descriptor>
    <Component>
      <Resource mimeType="audio/wav" ref="http://server/lec01.wav" />
    </Component>
  </Item>
  <Item>
    <!-- DIM implementation -->
    <Resource mimeType="application/mp21-method"><![CDATA[
      function main() {
        var audioTracks = ObjectMap.getObjects("urn:foo:track");
        DIP.PlayResource(audioTracks[0], true);
      }
     ]]>
    </Resource>
  </Item>
</DIDL>
```

Figure 2.7: DIM added to static DI, enabling 'play' functionality.

Figure 2.7 shows the same DI as above with added DIP functionality. In particular, note the addition of a 'main()' method within the DID. In this simple example, the method simply plays the audio resource associated with the Digital Item. Note that the call `DIP.Play()` is a DIBO call to the default MPEG-21 DIP API. This simple example demonstrates the intended use of DIP as a way of adding dynamic abilities to otherwise static DIs.

Digital Items and Classes: An Analogy

The following analogy is instructive in grasping the role of DIs and DIP in a multimedia framework. Consider any high-level object-oriented programming language. In the following, Java is used as a typical example. In Java, the main entity is a class. Java classes contain both data members as well as member functions. One cannot do much, however, with a simple class definition. An instance of the class is required and this in turn may process data, or be processed by other classes or programs. This instance of a class is referred to as an object.

MPEG-21 operates in a similar manner to Java. For example, the DIDL serves the role of a class definition. Both are static declarations which impose structure on conforming objects. In MPEG-21, DID plays the role of 'an instance of a class' in Java. That is, a DID is valid if it conforms to the requirements of the DIDL. A DI is analogous to an object. Finally, note that if a Java class contains only data members, with no member functions, then the class is simply a container of data, and cannot perform any actions. It is in fact static (the use of the word static here should not be confused with the reserved word `static` in the Java programming language). By adding member functions, the object gains dynamic functionality. Similarly, DIMs and DIP extend static DIs to dynamic entities.

The last part of the standard discussed here is DIA. From the point of view of delivering customized content to users, there are two ways in which UMA can be achieved: a) store different versions of the content on a server, and stream the appropriate version for a given client request or b) adapt the content *on-the-fly*. Digital Item Adaptation (DIA) standardizes on-the-fly adaptation of content. Such adaptation is referred to as *transcoding*, that is, the process of adapting and transforming multimedia content from one format to another usually in response to some control input such as changing network conditions or the characteristics of the user device.

To aid in DIA, MPEG-21 standardizes *Usage Environment Descriptions (UED)*. UEDs have been broadly divided into the categories listed below [29]. UEDs are an important tool in designing UMA systems since they allow content adaptation to maximize benefit to the end user.

- **Terminal Capabilities**: codec capabilities, input-output characteristics, device properties.
- **Network Characteristics**: *capabilities* – define the maximum capacity of a network, minimum bandwidth it can provide, and *conditions* – define the available bandwidth, errors, delays.
- **User Characteristics**: User Info, Usage Preferences and History, Presentation Preferences, Accessibility Characteristics.
- **Natural Environment Characteristics**: Physical conditions around a user, such as lighting, noise level, time of day.

This subsection has touched upon some of the relevant parts of the MPEG-21 standard to highlight its use in the development of UMA systems. In particular, we have introduced the idea of a Digital Item and explained how the abstraction of specific multimedia content into a DI with associated metadata enables UMA. Furthermore, we have described Digital Item Adaptation (DIA) and the associated Usage Environment Descriptions (UEDs).

2.1.3 Existing Systems Using MPEG-7 and MPEG-21

This section provides a review of current systems which use the MPEG-7 and MPEG-21 standards described above. Discussion is not limited solely to presentation summarization systems with the intent that typical uses of the standards can be learned from the literature.

In [31] and [32], a comprehensive work in video summarization based on MPEG-7 and MPEG-21 is presented. The system consists of several components: server, middleware and client. The server stores the actual multimedia content in MPEG-1 or MPEG-4 format, along with MPEG-7 metadata extracted through content analysis [33]. On the client end, the usage environment is maintained using MPEG-21 UEDs and MPEG-7 User Preferences metadata. When a client requests a video, this data is passed along to the middleware. The middleware itself is broken up into two units, the *Personalization Engine* and the *Adaptation Engine*.

The personalization engine gathers all the metadata associated with the content as well as the client request to produce a set of personalization rules which are passed on to the adaptation engine. The adaptation engine in turn summarizes the content subject to the personalization constraints while attempting to maximize the retained semantic information. To be more specific, each video shot stored in the database server is augmented to include a 'semantic score', indicating the amount of semantic content conveyed by the shot. Then, the goal of the adaptation engine is to extract the appropriate shots and combine them based on the personalization information.

The system described in [32] effectively combines the MPEG-7 and MPEG-21 standards to achieve a semantic summarization system for video content. Note that the system uses metadata to guide content transcoding at the application level. However, much of the semantic information is extracted from the video domain. For presentation summarization, the video domain is not a good source of information [6]. Thus, the presentation summarization problem is more general in that several forms of multimedia must be exploited for semantic analysis and summarization hints. The design of an adaptation engine which operates on MPEG-21 DIs is the relevant point to the AV presentation summarization task at hand.

In [34], another system is presented which performs summarization based on MPEG-7 descriptions. The overall algorithm is simply to select video segments or key frames (depending on whether a video skim or still, mosaic summary is required) based on a relevance measure. The key frame summary is represented by an MPEG-7 `SequentialSummary DS`, where each key frames location relative to the original video content is described using a `VisualSummary DS`. On the other hand, the video skim summary is represented by a `HierarchicalSummary DS`, where the location and duration of video shots relative to the original content are described using the `KeyAudioVisualClip DS` and `SummarySegment DS`.

An interesting aspect of this work is that the summarization is carried out in combination with user queries, which act as a guide to interesting events in the video content. In particular, this work focuses on skin color as captured by the MPEG-7 `DominantColour` descriptor to guide summarization. This work provides another example of using MPEG-7 metadata and the standardized descriptor schemes to aid in summarization. This work fails to address the UMA needs of users. In addition, the focus on Visual descriptors, rather than a combination of several multimedia resources to guide summarization, is a marked difference from the current problem of presentation summarization.

Another summarization system based on MPEG-7 descriptors is presented in [15]. Here, the MPEG-7 `Motion Activity` Descriptor (MAD) is extracted in the compressed MPEG video domain. Then, it is argued that the intensity of the motion activity within a segment of the video is a direct indication of its importance in summaries. This information is then augmented through the use of MPEG-7 audio descriptors and speech recognition to further extract semantic segments from the audio. Initially, the approach in [15] is shown to work well for news broadcasts and it is modified to work with sports videos as well. Once again, this work relies of heavily on visual cues. It is interesting to note that the general method had to be modified to work with sports videos. This is also an indication that presentation summarization systems may require their own specifically designed algorithms for the best summarization results. This sentiment is echoed in [12] which states that the best results may be achieved through domain-specific summarization rules.

A particularly interesting application of MPEG-21 appears in [35]. This work considers the use of Digital Items as an XML-packaging method for describing digital libraries. [35] provides a step-by-step guide to creating an appropriate DI for representing digital library items. The particular advantages of MPEG-21 mentioned by the authors over existing XML methods (e.g. [3]) include increased interoperability and the ability of MPEG-21 to accommodate any media type (in all multimedia domains, such as video, audio, images). This is another reason to believe MPEG-21 is well suited to the problem of presentation summarization since presentations may be expressed in many formats, but should be handled seamlessly. It is concluded that MPEG-21 is a particularly attractive option due in part to the well-specified data model provided by the DIDL.

In the area of Distance Learning, [4] proposes an advanced learning system. In contrast with earlier DL systems [3, 36] the technical data structure used for learning content is the MPEG-21 Digital Item. It is explained that adoption of MPEG-21 offers several advantages, including increased interoperability due to the use of an international standard and the separation of content from display information. [4] mentions adaptation of content for delivery to heterogeneous user devices as a possibility for future systems.

This section has presented some works which use the MPEG-7 and MPEG-21 standards in the area of semantic summarization. The majority of these past works have focussed on video summarization, and in particular, using visual cues (such as MPEG-7 color descriptors) or motion activity to guide summarization. Moreover, the area of presentation summarization using the existing MPEG standards has not been addressed by past work.

At a broader level, this section has shown the move toward application level processing of multimedia content by leveraging the latest MPEG standards. It has been shown that these standards provide the necessary tools to develop a Universal Multimedia Access system where content is delivered seamlessly to any user, at any time over any network. These general ideas of application level processing of content based on metadata and the UMA paradigm should shape any future work on multimedia systems.

2.2 Presentation Summarization

This section describes prior work in summarization systems focused on presentation summarization. In presentation summarization, the main sources of information available to a summarization system are the slides and the accompanying audio. The video in such presentations is usually a low-resolution face shot of the presenter [6]. Thus, it does not provide substantial information for summarization as compared with the audio domain and slide material.

In [6], the main source of information was the audio channel. Specifically, characteristics of the audio signal such as pitch, pause and intensity were extracted to guide summarization. Speech pitch information was used to identify the speaker's emphasis on certain portions of the presentation as described in [37]. Besides this information, several heuristics were employed by the summarization system. Slide transition points were used to both impose structure on the content as well as to assign relative importance to portions of the content: the time spent on a slide (i.e., the difference between transition points) was used as a heuristic measure of the importance of that section of the presentation. Thus, the longer the presenter spent on a slide, the more important that section of the presentation. A further heuristic that was used was to assume that most of the important information is spoken at the beginning of a slide.

[6] also factored in extensive data on user patterns while viewing the full presentations. This data was collected over a video server, and annotated points in the presentations at which users connected to begin viewing a presentation, the point at which they left, and the points at which they voluntarily skipped ahead in the presentation to the next slide. Through this data, [6] defined importance measures for slides based on user counts, and used this to additionally guide summarization. After creating summaries using the above methods, [6] concluded that the relatively simple summarization rule of allocating time from the beginning of a slide in proportion to the time spent on a slide was sufficient for a 'good' summary. There are two drawbacks to this algorithm. First, the semantic content of slide material is wholly ignored, and second, pitch activity analysis is difficult to perform [38].

The same group later performed a study comparing the effectiveness of different modalities of presentation summarization, such as slides only, a text transcript of the presentation or an AV summary of the presentation [39]. The research concluded that users found audio-visual summaries far better than textual transcriptions or slides only. Furthermore, rather counter-intuitively, users preferred slides with detailed information rather than slides which just highlighted the "big points".

The Video Manga project also attempted presentation summarization [40], in the context of team meetings, by extending previous work on video summarization. The approach described in [40] exploits information in the video domain to create a hierarchical cluster of video frames. Next, the clusters are associated with video segments, and an importance score is calculated for each cluster. This score is a function of the length of the segments in each cluster (a longer segment is relatively more important than a shorter one) as well as rarity as determined by the proportion of total segments contained in a cluster.

This technique is extended to meeting summarization by modifying the importance score formula to combine additional sources of information. A multiplicative factor is added to quantify the relative importance of different sources of information. Then, certain events which are deemed more important affect the overall importance measure through this multiplicative factor. In [40], closeups of individuals during the meeting are deemed important, along with events in the audio domain such as applause or laughter. The importance measure is further augmented through the extraction of captions from the meeting video. These captions are extracted from the video (where they appear as parts of slides, for example) using Optical Character Recognition (OCR) techniques and from the audio domain through speech recognition techniques. In this way, [40] argues that through the addition of context specific information, the summary may be finely tuned for the given application. Once again, the drawback of the algorithm devised here is the reliance on computationally intensive techniques such as OCR and speech recognition to extract semantic information from the content, and the lack of emphasis on information in presentation slides.

There have been other attempts at using OCR and speech recognition algorithms for presentation summarization. In [41] and [42], OCR and speech recognition are used to link material from slides with corresponding video and audio segments. In this scheme, a video is partitioned into shots based on slide transitions. Then, a commercial OCR tool recognizes any text which appears in the video, linking it to the slide used for the transition. In another approach, the team first extracts keywords from the slides and represents them in XML. Then, the audio and this XML representation are passed to a speech recognition engine. Based on the XML, the engine returns a series of keywords that were recognized in the audio. This information is then further processed to finally link portions of the audio with the slide. The drawback of these approaches is the reliance on OCR techniques, which are ineffective in assigning semantic meaning to slide information.

In another related work targeted specifically at lecture presentations, [43] explored the use of XML to both represent lectures and build structural summaries for content based retrieval. In this work, a *video shot* is defined as a segment of the lecture video that corresponds to a single lecture slide. Again, recorded slide transitions are the basis of this association between video shots and slides. It is assumed that each slide has a corresponding textual summary which is then written into XML. Using these definitions, an XML based tree-like structure is created to describe a lecture. This structure allows flexibility in terms of querying and indexing. This work does not summarize content from a semantic standpoint. It is included here to demonstrate the use of XML to represent presentation material.

2.3 Audio Analysis

As previously mentioned, typically the most important source of information in informational-talks is the audio channel, which can provide important hints for summarization [44]. There have been several previous works which perform audio analysis to guide summarization. Each such work addresses the fundamental challenge - determining important segments in a speech waveform - in a different way. In [37], a pitch-based importance detection scheme is presented. The method is based on several studies in the linguistics community which show that the introduction of a new topic or emphasis on a particular section of a sentence by a speaker is strongly correlated with an increase in pitch activity. [37] mentions that previous methods employed Hidden Markov Models (HMMs) to detect emphasized portions of speech. In this approach, a HMM was trained with signal energy and pitch as parameters until the HMM was able to recognize word intonation patterns. However, these HMM-based approaches are computationally complex and require large amounts of training data.

On the other hand, the method in [37] does simple signal processing of the pitch signal for emphasis detection. The algorithm was devised mainly by correlating various metrics with a hand-marked transcript of a speaker. Initially, a 10-15 minute segment of audio was recorded for a male speaker. Then, a linguist manually annotated the transcript, indicating emphasized portions of the speech. Finally, the author calculated several metrics (e.g. mean, range, standard deviation) using the pitch profile of the signal and observed the correlation of these metrics with the hand-marked transcript produced by the linguist. It was found that standard deviation and range of the pitch were good indicators of emphasized speech. It is noted in [37] that these metrics essentially measure the activity and variability in the fundamental frequency of the speech signal. This is an important point and we will return to it later.

The approach described above from [37] is directly applicable to the speech from informational-talks. However, there are a couple of drawbacks in this approach. First, pitch information is difficult to extract, as mentioned in several works [37, 38]. As a result, pitch activity is difficult to measure, especially under situations where the recorded speech may be noisy as is the case in spontaneous classroom lecture situations. Next, [37] mentions that because the pitch range of each individual speaker is different, the threshold values used in the algorithm must be adapted for each speaker. This is a difficulty due to the noisy nature of speech in classroom settings. Furthermore, it is not guaranteed that the same speaker will speak throughout the talk. For example, students may ask questions for clarification.

The most comprehensive work in the area of time-compression of speech to date is SpeechSkimmer [45], which uses “simple” speech-processing techniques to allow users to hear recorded speech at several levels of detail, depending on user feedback through an external device. Some of the techniques explored in [45] include automatic emphasis detection based on pitch activity (as detailed in the earlier work [37] above), loudness and pauses in the speech and speedup based on the Synchronized Overlap-Add (SOLA) method [46]. The SOLA method performs audio speedup without altering the pitch of the resulting audio. The algorithm operates by overlapping the beginning of a segment of audio with the end of the previous segment until the point of highest cross-correlation is found. In other words, two adjacent segments are overlapped to the degree where there is maximum similarity between the end of one and the beginning of the next segment. Then, the overlapping segments are averaged as a smoothing operation to produce the final output.

The SOLA method is very effective for time-compressing speech and maintaining comprehensibility, but there is a fundamental limit to the level of compression that may be achieved before the output is no longer intelligible to a user [47]. In fact, listening experiments show that speech is comprehensible up to a speedup of 1.5-2.5 times the normal rate [6,48]. Consequently, SpeechSkimmer combines various other techniques, such as pause removal based on silent-segment detection. The approach in [45] suffers the same problems as in [37], namely the reliance on robust pitch detection. Furthermore, it is unclear whether the approach can handle different speakers and a noisy environment seamlessly.

Recently, [49] have applied information theory measures to the problem of shot detection and video summarization. Although this work does not directly address the problem of speech time-compression, the techniques used are relevant to this thesis. In this scheme, important events are identified on the basis of the joint entropy and mutual information [50] between successive frames. Since independent frames will not be correlated, shot detection may be carried out based on the mutual information between frames. Once the video has been segmented, [49] performs intra-shot clustering based again on mutual information. Then, key frames are selected from significant clusters (those that have greater than some threshold number of frames) to produce a video summary. This is a novel approach based on basic measures of information.

One of the most recent works is [51]. In [51], the time-compression strategy is divided into two distinct parts, namely acoustic techniques and semantic techniques. The acoustic techniques focus on signal-processing methods of time-compression such as silence removal and constant playback-rate speedup. The semantic approach utilizes transcripts of speech generated through Automatic Speech Recognition (ASR). Using Information Retrieval (IR) methods, the transcripts are analyzed and important segments are determined in the speech. Then, the unimportant segments are excised. This work introduces a novel method of determining the importance of speech segments. However, the reliance on ASR is a major limitation due to the high error rates involved.

Another similar work is presented in [12]. The approach is very similar to the one in [51], in that automatically generated transcripts using ASR are used to guide time-compression. As mentioned in this work, however, error rates for even powerful ASR systems can be as high as 65% in unscripted, spontaneous and noisy speech environments such as a classroom setting as in our scenario.

This section has described a few works related to determination of important segments in speech signals. These works have been selected due to their particular relevance to the task of audio analysis for presentation summarization.

2.4 Presentation Slides Analysis

One of the main sources of information in audio-visual presentations are the presentation slides. Yet, there has been little prior work in the area of presentation slide analysis for the purposes of summarization. In fact, for the task of AV presentation summarization, existing systems have commonly used slides - in particular, slide transition information - to impose structure on the presentation multimedia content [6, 40–42]. Semantic information from slides has been largely ignored. This section describes prior work in the area of slide analysis.

In the past, slide analysis has often been performed in systems with the goal of slide retrieval in mind. Slide retrieval functionality, in turn, is usually provided as a feature in presentation management systems. ProjectBox [52] is an example of a system used to capture, manage and index presentations. In the ProjectBox system, slide text is extracted using OCR techniques and an index for search and retrieval is created.

One of the more recent works in slide retrieval is presented in [53]. In this system, a framegrabber captures slides as they are displayed through a projector during a presentation. OCR is again used to transcribe slide contents; the system is able to extract text from figures as well, for example, labels on a plot. Once the slide contents have been extracted, a Vector Space Model (VSM) [54] is used for retrieval. In the VSM approach, each text document and user query are represented as vectors whose entries correspond to terms in a global dictionary. The dictionary is constructed by filtering all words in the entire document collection and retaining only unique words. The set of unique words is further processed through *stemming* - the process of text normalization by removing suffixes and prefixes, elimination of stop words - that is, words that are semantically unimportant such as ‘and’, ‘or’, ‘to’, and the removal of special characters - for example, punctuation.

The VSM approach continues to define a term-by-document matrix, whose rows and columns correspond to documents and terms, respectively. The matrix entries (or weights) are calculated using the traditional information retrieval Okapi formula [53, 54]. The Okapi formula is given by:

$$w_{i,j} = \frac{tf_{i,j}.idf_i}{k.[1 - b + b.dl_j/avg\_dl] + tf_{i,j}}, \tag{2.1}$$

where $tf_{i,j}$ is the occurrence frequency for term i in document j (term frequency), $idf_i = \log(N/N_i)$ is the inverse document frequency, N and N_i are the total number of documents and the number of documents that contain term i, k and b are tuning parameters, and dl_j and avg_dl are the length of document j and average document length. The similarity between a document and a query term is obtained by summing the matrix entries of the query terms for each document. In the end, documents deemed to be the most similar to the query term are retrieved up to a pre-determined number of retrievals.

In several works, audio information from the presentation has been used for generating slide transcriptions. In [55], for example, ASR is used to obtain a set of meaningful phrases for indexing and retrieval. The word error rate reported in this work is nearly 75%. A similar approach is explained in [56]. In this work, ASR is used along with text and metadata to generate three indexes. The TF.IDF method is then used for retrieval from each index. The TF.IDF method calculates the similarity of two document vectors as a weighted inner product of the vectors, where the components of each vector are in turn a product of term frequency and inverse document frequency [54].

The approaches described thus far have processed slides without regard to their format. Specifically, the particular software used to generate slides for the presentation is irrelevant. This is the main advantage of OCR and ASR based approaches. However, these techniques have their drawbacks. Specifically, both OCR and ASR techniques are prone to high error rates and are computationally intensive techniques. Moreover, although OCR may be used to access the text on slides, it is unable to associate semantic meaning with objects on a slide. In other words, OCR techniques cannot identify a term on a slide as belonging to a table, for example, or as being the caption of an image, or an axis label.

An alternative to such format-agnostic approaches is direct processing of slides, which generally requires access to proprietary API [56, 57]. Fortunately, there has been a recent move to standardized open file formats for presentations. In particular, Microsoft Office 2007 is based entirely upon the open source format OpenXML, which provides an accessible XML representation of presentation slides.

The pervasive use of the XML format on the Internet has motivated much research in the area of XML document retrieval. For example, the Initiative for the Evaluation of XML Retrieval (INEX) [58] conducts extensive research in this area. Retrieval systems designed specifically for XML documents can exploit the structured nature of XML. Indeed, the extraction of information such as indentation level, and various other feature is simplified in an XML based format, since this data can be conveyed through arguments in XML tags. This advantage of XML-based representation of slides will be leveraged in this thesis, toward the design of an AV presentation summarization algorithm.

In [57], an XML based representation of presentations is processed to extract indentation level of a keyword in addition to time spent on a slide (slide duration). This information is used, along with the TF.IDF method for slide retrieval. [57] calculates a context impression indicator for each slide, which is the average of scores obtained for each occurrence of the query term in a slide. The scores for each term are computed as the geometric mean of the indentation level, slide duration and frequency of term occurrence. The score for each slide is augmented by considering neighbouring slides using an exponentially decaying window. This work does not provide any justification for the use of the geometric mean for feature combination as it relates to human assessment of the importance of a slide.

This section has reviewed some of the existing works in the area of slide analysis and processing. Although there has been some work in the area of slide retrieval, there are three major shortcomings in the work to date: slide information has never been used to guide presentation summarization; slide transcoding has never been performed in order to adapt slide information for varying terminal devices; several features of slides have been left unexplored, such as typeface, font size and descriptive metadata from images.

2.5 Chapter Summary

This chapter has presented an overview of the UMA enabling technologies MPEG-7 and MPEG-21. Next, current systems utilizing these standards were presented to demonstrate their applicability to the design of a UMA system. In particular, the use of MPEG-21 Digital Items in describing complex collections of multimedia content was shown. The concept of a Digital Item lends itself naturally to describing an AV presentation. This task is undertaken in the following chapter, which describes in detail the proposed `Presentation DI`.

In addition, this chapter reviewed recent approaches in the analysis of the key sources of information available in AV presentations, namely audio data and presentation slides. The major limitation of current audio analysis systems is twofold. First, many current systems rely on pitch information extracted from the audio signal. As discussed, accurate pitch estimation is both a difficult and computationally complex task. Second, several ASR based approaches to audio importance analysis have emerged. These have the drawback that ASR systems are computationally complex and suffer high error rates. Moreover, ASR systems require extensive training before use. Finally, the vocabulary of commercial ASR systems may not be well suited to academic settings, which use uncommon jargon. To address these limitations, a spectral entropy based audio analysis scheme is proposed in the next chapter.

The main limitation of current presentation slide analysis systems is the lack of slide transcoding methodologies, as well as unexplored features including typeface and font size. Further, slide analysis has not been used effectively in presentation summarization systems. In the following chapter, a novel fuzzy set based slide analysis approach is explained with applications to slide transcoding as well as presentation summarization.

Chapter 3

UMA DL System Design and Summarization Engine

Emerging multimedia systems are increasingly required to conform to the Universal Multimedia Access (UMA) paradigm to satisfy the needs of users. In present times, users demand access to personalized content seamlessly using any terminal device at any time. The delivery of presentations and presentation summaries to end users is no different, and requires a UMA-based solution. The first part of this chapter presents the proposed UMA system in detail. In the remainder of the chapter, the summarization engine is described in detail.

3.1 UMA DL System Design

Recall the form of the overall system, depicted in Figure 3.1. This section focuses on the outlined components of the total system which comprise the main parts of the UMA DL system design.

Before describing the outlined components, the operation of the complete system is described. For AV presentation summarization, the input to the summarization system is a complete presentation, including a video component, the accompanying audio track, presentation slides and *usage information*, that is, the amount of time spent on each slide during the presentation. In this work, the video component of the presentation is ignored and discarded, since it is not a rich source

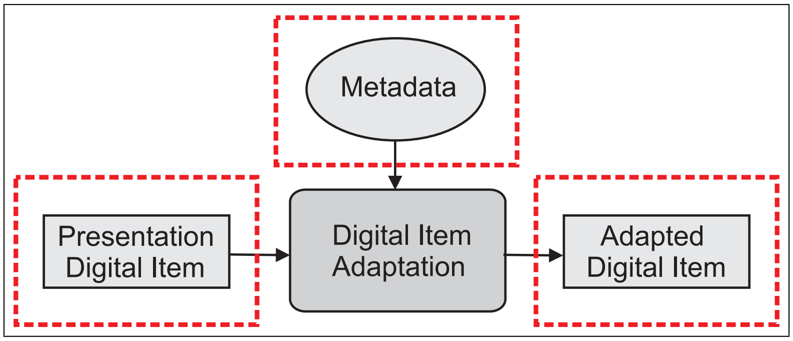


Figure 3.1: Components discussed in this chapter.

of information [6] as explained in Chapter 1. A `Presentation DI` is created to represent each presentation. The details of the `Presentation DI` are given below in subsection 3.1.1.

When a summary of a particular presentation is requested, the system records any keywords issued by the user to guide summarization. In other words, the user is able to affect the summary based on specific user needs on a particular topic. This notion of user-influenced summarization is a natural extension of the work in [59]. In [59], a real-time content filtering system that allows for summarization of TV programs is proposed. When a user begins to watch TV, the user is able to specify filtering options, such as scenes of interest. Then, the real-time content filtering functionality in the TV automatically processes all the live video streams, and when a scene matches the user-defined preferences (e.g. a goal in a soccer match), a sub-image is displayed on top of the program the user is watching to show the scene of interest. This work extends this concept to AV presentations, where user supplied keywords guide summarization.

After gathering any user input, the `Presentation DI` is passed on to a Digital Item Adaptation engine. From henceforth, the phrase 'adaptation engine' and 'summarization engine' are used interchangeably to refer to the same entity. The goal of any summarization system is to determine the important segments of the content to be summarized. By operating on the `Presentation DI`, the proposed summarization engine calculates a score for each second of the presentation, indicating the importance of the second, where higher scores indicate greater importance. Although discussion of the exact algorithm and method used to calculate these scores is delayed until the next section, it is mentioned in passing that these second scores are calculated as a function of information from all content in the Presentation DI, namely the audio track, the slides, the user supplied keywords and usage information for the particular presentation. Finally, in consideration of the desired degree of summarization, a summary is produced in the form of an adapted Presentation DI which is delivered to the user.

3.1.1 Presentation Digital Item

This subsection details the design of the proposed Presentation DI. The Presentation DI is displayed graphically in Figure 3.2. In the graphical version, the MPEG-21 names of the various parts are displayed in *italicized* font. Where applicable, it is explicitly stated that MPEG-7 descriptors are used.

The Presentation DI displayed in Figure 3.2 is the initial version of the proposed DI. As it stands, the Presentation DI contains only data relevant to the presentation, along with a portion of the data used for summarization. Elements will be added to the Presentation DI in what follows.

To describe the role of each element in the Presentation DI, and where needed the XML snippet that represents the particular element, the following will use XML terminology (e.g. child node) to refer to various elements. The top-level *Item* represents the entire presentation, with the corresponding XML given in Figure 3.3.

The first child of this node is a *Descriptor* element used to include metadata about the entire presentation. For example, details about when this presentation took place, who delivered the presentation and the methods used to capture the presentation may be included here. The next child is another *Descriptor* element. This element includes MPEG-7 metadata indicating how long the presenter spent on each slide of the presentation. The exact form of this metadata is given in Section 3.1.2.

The next child of the top-level *Item* node is a *Component* node. This element holds information related to the audio track of the presentation, as depicted in Figure 3.4. In particular, a custom

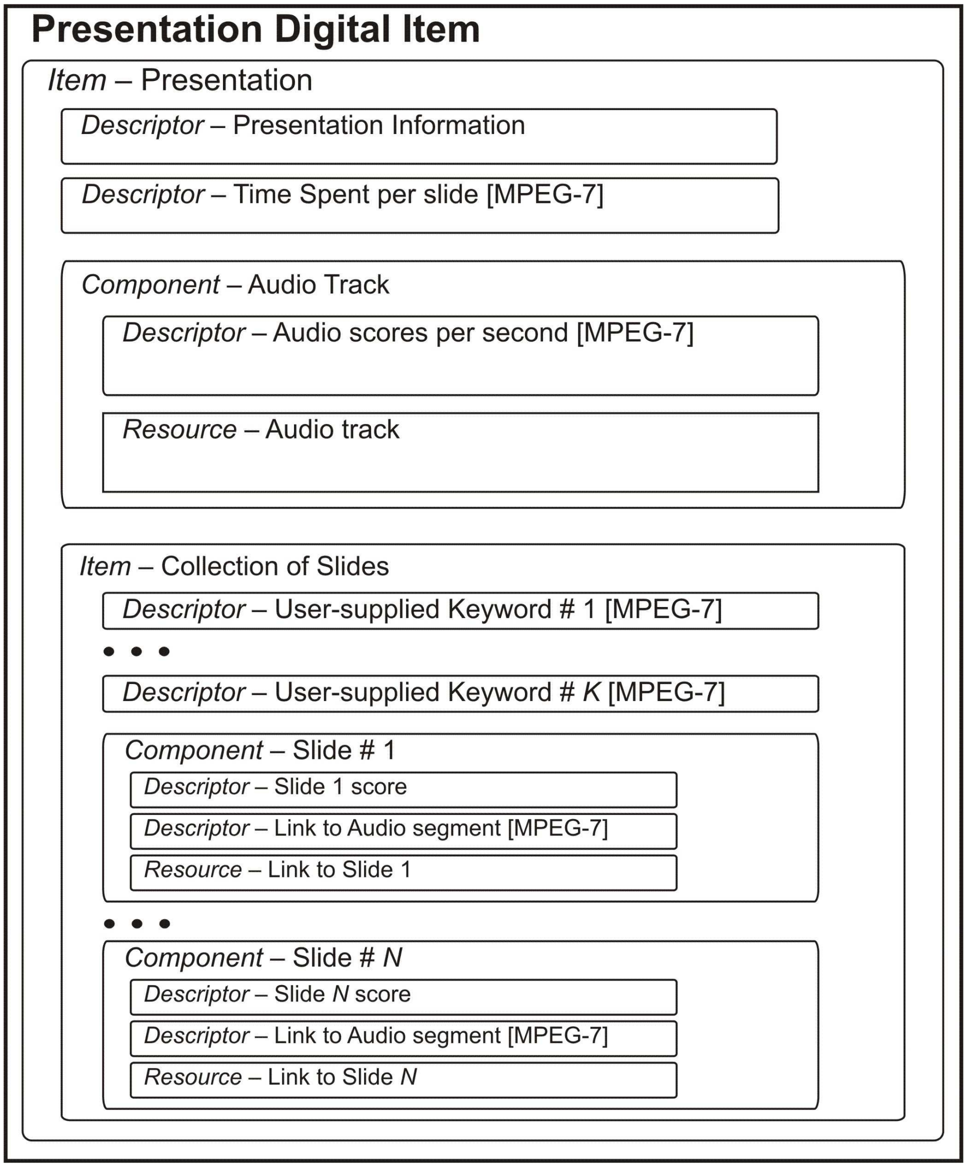
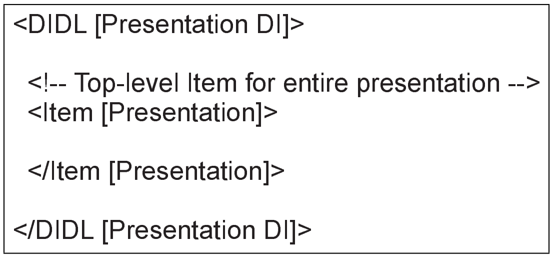

Presentation Digital Item
Item – Presentation
Descriptor – Presentation Information
Descriptor – Time Spent per slide [MPEG-7]
Component – Audio Track
Descriptor – Audio scores per second [MPEG-7]
Resource – Audio track
Item – Collection of Slides
Descriptor – User-supplied Keyword # 1 [MPEG-7]
• • •
Descriptor – User-supplied Keyword # *K* [MPEG-7]
Component – Slide # 1
Descriptor – Slide 1 score
Descriptor – Link to Audio segment [MPEG-7]
Resource – Link to Slide 1
• • •
Component – Slide # *N*
Descriptor – Slide *N* score
Descriptor – Link to Audio segment [MPEG-7]
Resource – Link to Slide *N*

Figure 3.2: Graphical representation of `Presentation DI`.

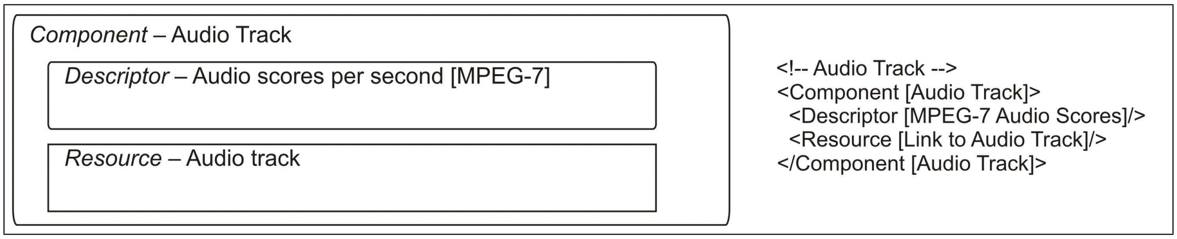
MPEG-7 Descriptor is created to hold the importance scores for each second of the audio track (see Section 3.1.2). In addition, a link is stored pointing to the audio track on the server.

Finally, the last child in the `Presentation DI` is an *Item* containing a list of slides in the presentation. Figure 3.5 represents an XML skeleton of the proposed mechanism to hold the presentation slides. The first several children of this *Item* node are MPEG-7 Descriptors which



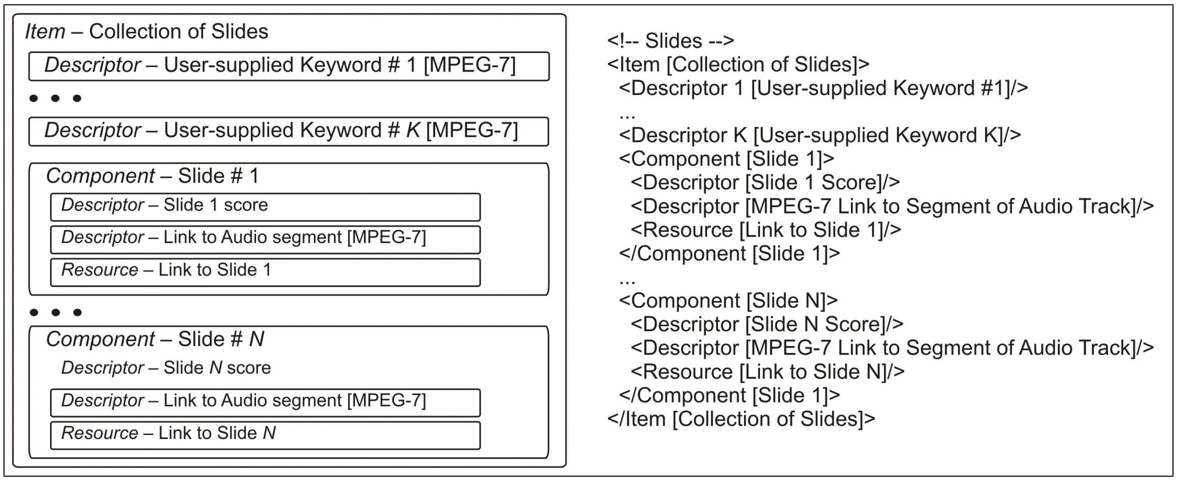
```
<DIDL [Presentation DI]>

  <!-- Top-level Item for entire presentation -->
  <Item [Presentation]>

  </Item [Presentation]>

</DIDL [Presentation DI]>
```

Figure 3.3: Top-level *Item* in proposed `Presentation DI`.


Component – Audio Track
Descriptor – Audio scores per second [MPEG-7]
Resource – Audio track

```
<!-- Audio Track -->
<Component [Audio Track]>
  <Descriptor [MPEG-7 Audio Scores]/>
  <Resource [Link to Audio Track]/>
</Component [Audio Track]>
```

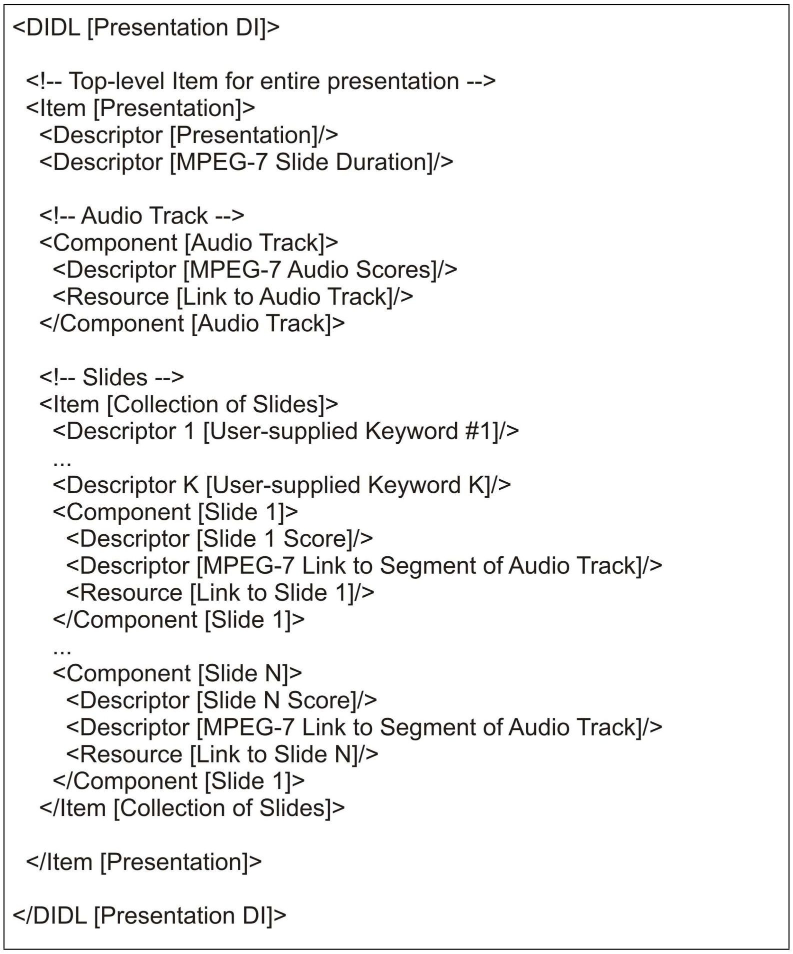
Figure 3.4: Audio track stored in the proposed `Presentation DI`.


Item – Collection of Slides
Descriptor – User-supplied Keyword # 1 [MPEG-7]
• • •
Descriptor – User-supplied Keyword # *K* [MPEG-7]
Component – Slide # 1
Descriptor – Slide 1 score
Descriptor – Link to Audio segment [MPEG-7]
Resource – Link to Slide 1
• • •
Component – Slide # *N*
Descriptor – Slide *N* score
Descriptor – Link to Audio segment [MPEG-7]
Resource – Link to Slide *N*

```
<!-- Slides -->
<Item [Collection of Slides]>
  <Descriptor 1 [User-supplied Keyword #1]/>
  ...
  <Descriptor K [User-supplied Keyword K]/>
  <Component [Slide 1]>
    <Descriptor [Slide 1 Score]/>
    <Descriptor [MPEG-7 Link to Segment of Audio Track]/>
    <Resource [Link to Slide 1]/>
  </Component [Slide 1]>
  ...
  <Component [Slide N]>
    <Descriptor [Slide N Score]/>
    <Descriptor [MPEG-7 Link to Segment of Audio Track]/>
    <Resource [Link to Slide N]/>
  </Component [Slide 1]>
</Item [Collection of Slides]>
```

Figure 3.5: A list of slides in the proposed `Presentation DI`.

hold the user-supplied keywords. Next, each slide is enclosed within a *Component* node. Each slide's *Component* node holds two *Descriptors* providing the slide's importance score as well as an MPEG-7 descriptor linking the slide to a temporal segment of the audio track. Finally, a link is provided to the slide itself. The representation of each slide is discussed in Section 3.1.4 below. A complete XML skeleton of the proposed `Presentation DI` is given in Figure 3.6.



```
<DIDL [Presentation DI]>

  <!-- Top-level Item for entire presentation -->
  <Item [Presentation]>
    <Descriptor [Presentation]/>
    <Descriptor [MPEG-7 Slide Duration]/>

    <!-- Audio Track -->
    <Component [Audio Track]>
      <Descriptor [MPEG-7 Audio Scores]/>
      <Resource [Link to Audio Track]/>
    </Component [Audio Track]>

    <!-- Slides -->
    <Item [Collection of Slides]>
      <Descriptor 1 [User-supplied Keyword #1]/>
      ...
      <Descriptor K [User-supplied Keyword K]/>
      <Component [Slide 1]>
        <Descriptor [Slide 1 Score]/>
        <Descriptor [MPEG-7 Link to Segment of Audio Track]/>
        <Resource [Link to Slide 1]/>
      </Component [Slide 1]>
      ...
      <Component [Slide N]>
        <Descriptor [Slide N Score]/>
        <Descriptor [MPEG-7 Link to Segment of Audio Track]/>
        <Resource [Link to Slide N]/>
      </Component [Slide 1]>
    </Item [Collection of Slides]>

  </Item [Presentation]>

</DIDL [Presentation DI]>
```

Figure 3.6: XML skeleton of `Presentation DI`.

Advantages of the `Presentation DI`

The use of a `Presentation DI` to represent presentations offers several advantages over existing methods of packaging together various elements of a presentation:

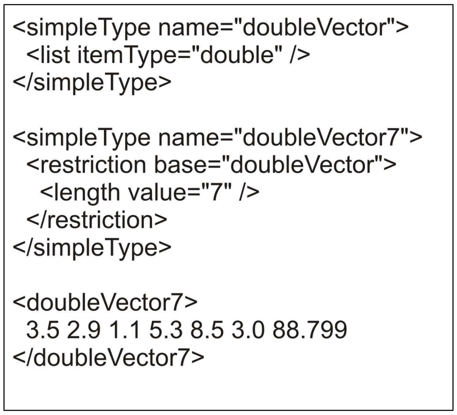
- Format independence. Note that no mention has been made of the particular coding format used for the audio track. Indeed, the audio may be in any format, and it would have no affect on the structure of the `Presentation DI`.

- Interoperability. The fundamental unit of interaction between users and systems providing presentations would be the proposed `Presentation DI`. The proposed `Presentation`

DI is implemented in a standard manner, which allows various entities to interact with the Presentation DI seamlessly. For example, the University of Toronto may decide to produce and archive several lectures from the Physics department. These lectures would be available online to the public. The University simply has to ensure that the lecture is packaged within a Presentation DI, and then it may be accessed by any person with an MPEG-21 enabled peer.

3.1.2 MPEG-7 Descriptors

This section describes the MPEG-7 descriptors used in the Presentation DI. Refer back to Figure 3.2 to determine the elements of the Presentation DI which contain MPEG-7 descriptors. The first two MPEG-7 descriptors in the proposed Presentation DI represent a vector of floating-point values. In the first case, the descriptor describes the time duration spent on each slide by the presenter during the presentation. In the second, the descriptor represents the importance score per second of the audio track. The proposed Presentation DI uses the MPEG-7 list data type to store this information. An example of this descriptor is shown in Figure 3.7.

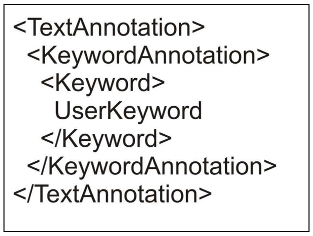


```
<simpleType name="doubleVector">
  <list itemType="double" />
</simpleType>

<simpleType name="doubleVector7">
  <restriction base="doubleVector">
    <length value="7" />
  </restriction>
</simpleType>

<doubleVector7>
  3.5 2.9 1.1 5.3 8.5 3.0 88.799
</doubleVector7>
```

Figure 3.7: MPEG-7 descriptor for vector of floating-point values.

The user-supplied keywords are also represented using MPEG-7 Descriptors. For these descriptors, the built-in KeywordAnnotation descriptor as shown in Figure 3.8.

Finally, the last set of MPEG-7 descriptors used allow linking of slide information to the audio



```
<TextAnnotation>
  <KeywordAnnotation>
    <Keyword>
      UserKeyword
    </Keyword>
  </KeywordAnnotation>
</TextAnnotation>
```

Figure 3.8: MPEG-7 descriptor for user-supplied keywords.

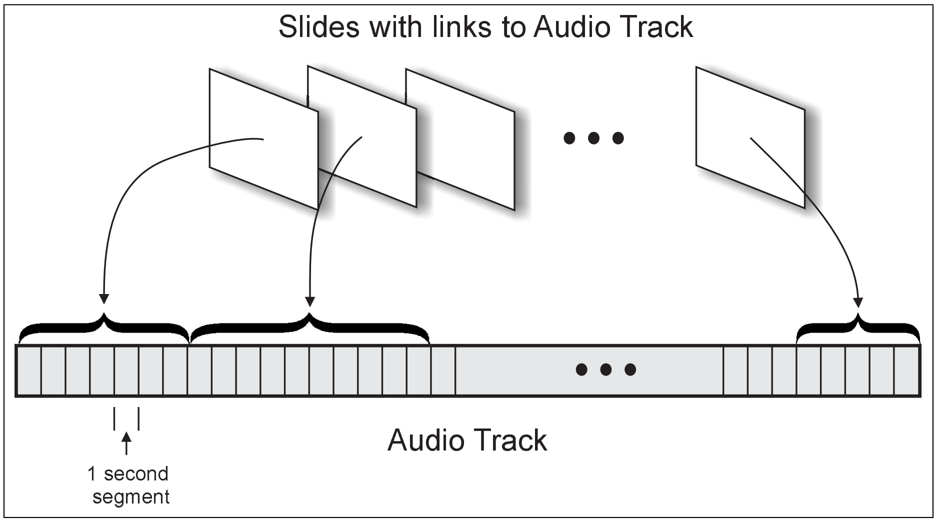
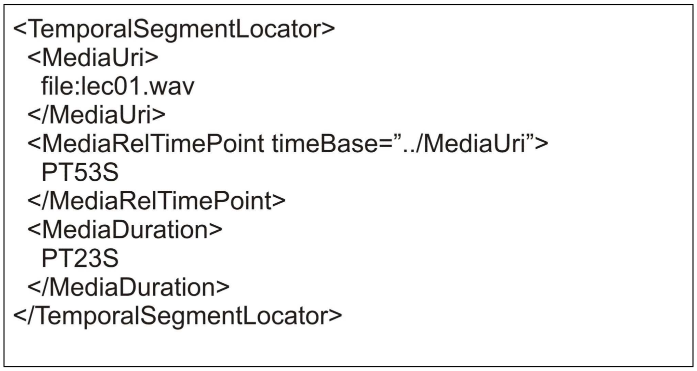


Figure 3.9: MPEG-7 descriptors used to link slides with audio track.

track. Previous works have used complicated methods to achieve this linkage, such as performing video analysis to detect slide changes in the background [41,42]. MPEG-7 provides an elegant way of achieving this link based on metadata which gives the time spent on each slide by the presenter. This is easily recorded by simple software executing on the presenter's PC during the presentation. The situation is as depicted in Figure 3.9.

The MPEG-7 descriptor which allows such linkage is `TemporalSegmentLocator`, as shown in Figure 3.10. The descriptor requires the start of a temporal segment in the particular multimedia content (audio track, in this case) as well as the length of the segment. In the example given in Figure 3.10, the temporal segment begins at an offset of 53 seconds from the start of the audio track and lasts for 23 seconds.



```
<TemporalSegmentLocator>
  <MediaUri>
    file:lec01.wav
  </MediaUri>
  <MediaRelTimePoint timeBase=”../MediaUri”>
    PT53S
  </MediaRelTimePoint>
  <MediaDuration>
    PT23S
  </MediaDuration>
</TemporalSegmentLocator>
```

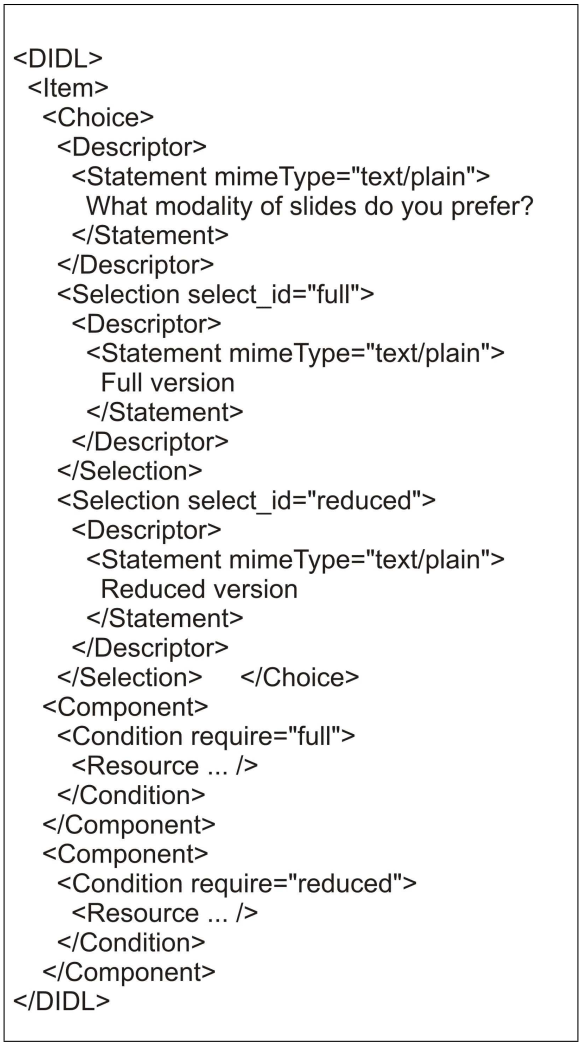
Figure 3.10: MPEG-7 descriptor for temporal linking.

3.1.3 User Environment

This section describes the way in which the user environment is described in the proposed `Presentation DI`. The problem of delivering customized content to users based on user preferences within a distance learning setting has been studied previous [60]. In fact, in the approach described in [60, 61] proposes Digital Item Adaptation based on user preferences such as desired modality and coding parameters. The limitation in the current work is an inability to perform slide transcoding for small-display devices such as PDAs and SmartPhones. In this context, slide transcoding refers to the processing of slides to reduce their size for suitable display on such devices. The proposed slide transcoding procedure is outlined in the following chapter.

To address this problem, the proposed `Presentation DI` includes a MPEG-21 Choice/Selection clause. This allows consumers of presentation content to interact with the `Presentation DI` and specify their terminal display capabilities. The semantics of the MPEG-21 Choice/Selection clause are shown in Figure 3.11. The operation of the Choice/Selection clause is straightforward. Initially, the user is presented with a choice between two options: the full version of the slides or the transcoded (‘reduced’) version. Based on the user selection, the appropriate content will be delivered to the user. Note that although it may appear as though both versions of the slides will have to be included in the `Presentation DI`, this is not the case. In fact, the appropriate version will be generated on-the-fly by the summarization engine.

As an alternative, the proposed `Presentation DI` also includes a MPEG-21 DIA Usage Envi-



```
<DIDL>
 <Item>
  <Choice>
   <Descriptor>
    <Statement mimeType="text/plain">
     What modality of slides do you prefer?
    </Statement>
   </Descriptor>
   <Selection select_id="full">
    <Descriptor>
     <Statement mimeType="text/plain">
      Full version
     </Statement>
    </Descriptor>
   </Selection>
   <Selection select_id="reduced">
    <Descriptor>
     <Statement mimeType="text/plain">
      Reduced version
     </Statement>
    </Descriptor>
   </Selection>     </Choice>
  <Component>
   <Condition require="full">
    <Resource ... />
   </Condition>
  </Component>
  <Component>
   <Condition require="reduced">
    <Resource ... />
   </Condition>
  </Component>
</DIDL>
```

Figure 3.11: MPEG-21 Choice/Select clause.

ronment Description (UED). This DIA UED may be populated ahead of time for users with known device capabilities. This alternative is not discussed here further.

3.1.4 Presentation Slide Representation

The aim of this section is to describe the representation of presentation slides in the proposed `Presentation DI`. For concreteness, this discussion assumes the presentation slides are originally in Microsoft's OpenXML format, although the techniques are applicable to any XML-based presentation format (e.g. OpenDocument).

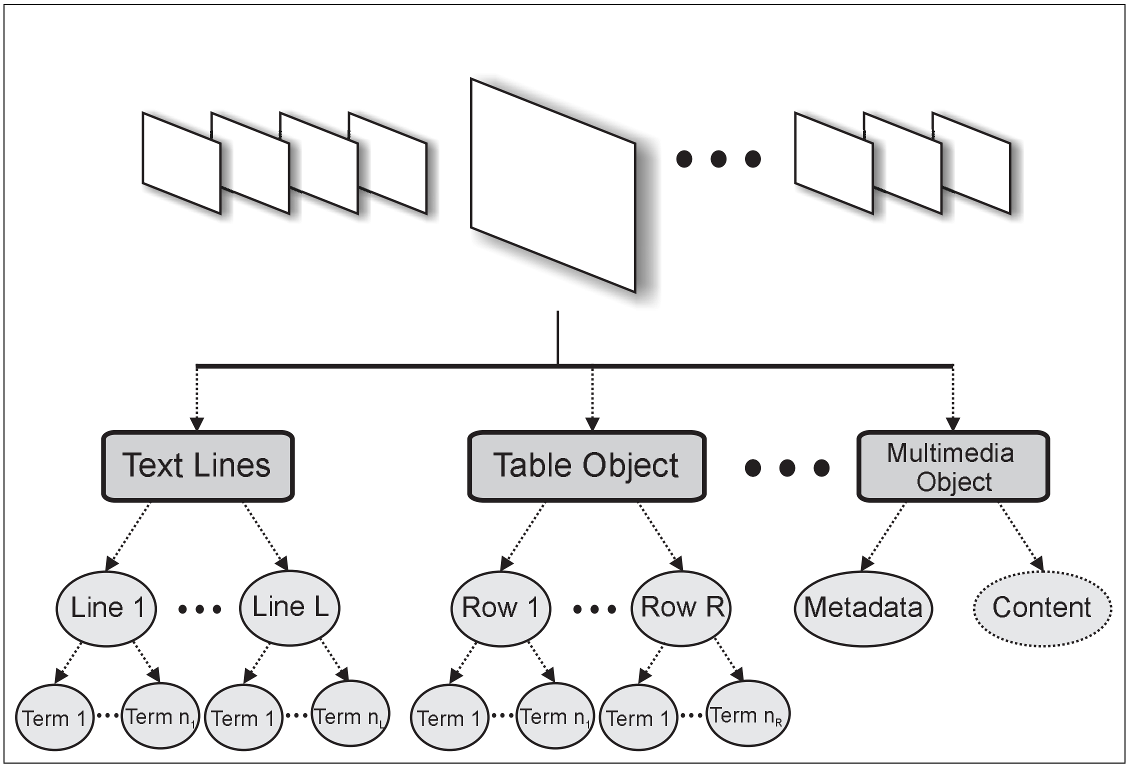
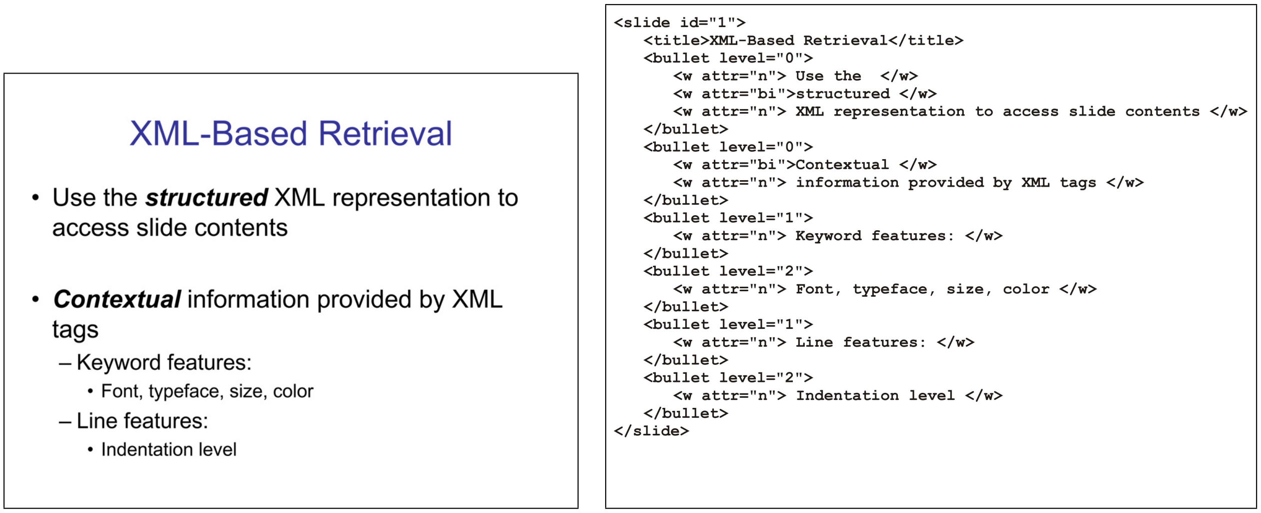


Figure 3.12: Hierarchical representation of a slide.

A slide is naturally represented in a hierarchical manner, as illustrated in Figure 3.12. As shown in the Figure, a slide can be viewed as a collection of text elements (commonly referred to as 'bullets'), tables and multimedia objects (e.g. embedded images, audio clips, etc.). It is desired to express slides in this hierarchical manner within the proposed `Presentation DI`. Although the OpenXML formats achieve this goal, there are a few drawbacks to using the OpenXML files directly. First, the OpenXML file formats include a vast amount of additional data required by PowerPoint to render the slides. This additional data bloats the size of the file unnecessarily. In addition, the addition of this data in the file results in a representation that is simply not comprehensible to humans by viewing the XML directly. Furthermore, it is a complicated task to display the slides represented by OpenXML files without writing a complete parser which accounts for all the syntactic constructs of OpenXML.

The major hinderance to using OpenXML directly is that fact that the summarization engine cannot annotate the file with additional metadata, since the resulting file would no longer be compatible with PowerPoint. Examples of additional information that could potentially be marked on the slides are protected keywords - if a protected keyword appears on a slide, all occurrences of this keyword should be hidden before the slide is displayed. For all these reasons, a simpler representation was designed to represent slides in the proposed `Presentation DI`. An example slide and its representation is shown in Figure 3.13.



XML-Based Retrieval

- Use the ***structured*** XML representation to access slide contents
- ***Contextual*** information provided by XML tags
  – Keyword features:
    - Font, typeface, size, color
  – Line features:
    - Indentation level

```
<slide id="1">
   <title>XML-Based Retrieval</title>
   <bullet level="0">
      <w attr="n"> Use the  </w>
      <w attr="bi">structured </w>
      <w attr="n"> XML representation to access slide contents </w>
   </bullet>
   <bullet level="0">
      <w attr="bi">Contextual </w>
      <w attr="n"> information provided by XML tags </w>
   </bullet>
   <bullet level="1">
      <w attr="n"> Keyword features: </w>
   </bullet>
   <bullet level="2">
      <w attr="n"> Font, typeface, size, color </w>
   </bullet>
   <bullet level="1">
      <w attr="n"> Line features: </w>
   </bullet>
   <bullet level="2">
      <w attr="n"> Indentation level </w>
   </bullet>
</slide>
```

(a) Example slide.

(b) Simplified XML representation.

Figure 3.13: Example slide and its simplified XML representation.

Note that the hierarchical structure of the slide is retained in the proposed XML representation. Moreover, additional XML tags have been introduced to annotate the slide with metadata. In Figure 3.13(b), for example, the *level* and *attr* attributes appearing within the *bullet* and *w* tags describe the indentation level and text formatting features, respectively. Also, note that this format is easily accessible to a human reader. Lastly, it is a simple task to write a XSLT sheet to display the proposed XML representation in, for example, HTML.

This section has described the UMA system design for the proposed AV presentation summarization system. The concept of a `Presentation DI` is discussed in great depth. The applicability of existing MPEG-7 and MPEG-21 standards to the present problem was demonstrated. With the definition of the proposed `Presentation DI`, we are in a position to process the DI to produce a summary.

3.2 Presentation Summarization Engine

In this section, details of the summarization engine are presented. In terms of the overall system, the focus is on the component outlined in Figure 3.14. Recall once again the fundamental problem in summarization: identifying important segments of the content that must be retained in a summary for delivery to end users. In the AV presentation summarization problem, the main sources of information to guide summarization are the audio track, presentation slides and usage information. Presently, some notation is introduced to allow further analysis of the problem.

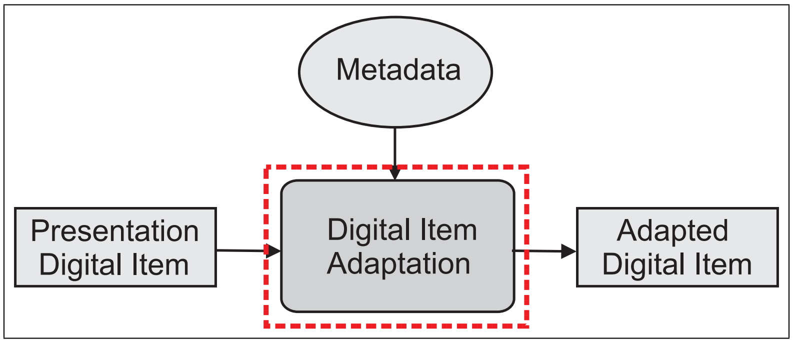


Figure 3.14: Summarization engine discussed in this section.

Let N represent the total length of the presentation, in seconds. Let

$$PS = \{s_1, s_2, ..., s_K\} \tag{3.1}$$

represent the set of presentation slides associated with this presentation. For any $k \in \{1, 2, ..., K\}$, denote by $dur(s_k)$ the duration of time spent on slide s_k during the presentation.

For any $n \in \{1, 2, ..., N\}$, the present *state* of a presentation at time n is entirely characterized by the vector

$$\vec{S}(n) = [n \;\; s_i \;\; e], \tag{3.2}$$

where n is the current position in the presentation, in seconds, s_i is the slide on display at second n (that is, the i^{th} slide is on display at this particular time instant n) and e represents the percentage of the time to be spent on the slide that has elapsed. In other words, if the current slide s_i has been on display for t seconds, then $e = \dfrac{t}{dur(s_i)}$.

Then, for any $n \in \{1, 2, ..., N\}$, let $\psi(n)$ represent the *importance score* for the n^{th} second of the presentation. The importance score $\psi(i)$ represents an opinion as to the importance of the i^{th} second of the presentation, such that if $\psi(i) > \psi(j)$ then the i^{th} second is deemed more important than the j^{th} second of the presentation. By construction, $\psi(n)$ will only assume values between $[0, 1]$, inclusive.

Similarly, for any $n \in \{1, 2, ..., N\}$, let $\alpha(n)$ represent an audio importance score, $\gamma(n)$ represent a usage importance score, and for any $k \in \{1, 2, ..., K\}$, let $\beta(s_k)$ represent a slide importance score. The exact methods to calculate these scores will be detailed in the following subsections. At this point, however, it should be noted that the audio score is a function of the audio signal alone, the usage importance score is a function of both the current slide on display as well as the time spent on the slide so far, and the slide importance score is strictly a function of the slide. Although $\gamma(n)$ is a function of two variables, there is no ambiguity in expressing it simply as $\gamma(n)$ since there is a unique slide on display at any time instant. Finally, note that $\beta()$ is a piece-wise constant function, since, from the second a slide s_i is displayed until $dur(s_i)$ seconds later, $\beta(s_i)$ will remain unchanged. By construction, each of these importance scores will take on values in the range $[0, 1]$. Once again, these functions $\alpha(n)$, $\beta(s_k)$ and $\gamma(n)$ represent opinions as to the importance of the second or slide under consideration.

The AV summarization problem then reduces to determining the presentation importance scores, $\psi(n)$ for all n. Since a presentation is completely characterized by its state $\vec{S}(n)$ at any time instant n, we have that

$$\psi(n) = f(\alpha(n), \beta(s_i), \gamma(n)) \tag{3.3}$$

where $f()$ presently represents an arbitrary aggregation function, and each of $\alpha(n)$, $\beta(s_k)$ and $\gamma(n)$ is evaluated with respect to the state at time n, $\vec{S}(n)$.

At this point, there are two main challenges: calculating the importance scores for audio, slides and usage information, and finally, devising a function $f()$ which operates on these scores

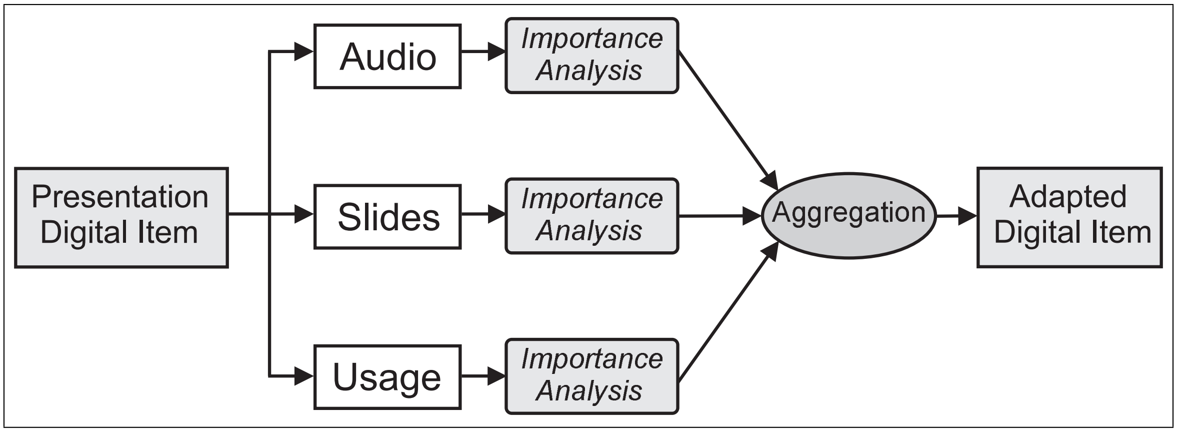


Figure 3.15: System overview of Digital Item Adaptation engine.

to compute a final score. This situation is represented schematically in Figure 3.15; individual importance scores (or opinions) are formed for each channel of information. Then, these scores are aggregated to produced the summarized presentation. The remainder of this section addresses these challenges. One of the key advantages of the proposed solution is the multi-modal approach to summarization. As mentioned in [12], using multi-channel information to guide summarization is the key to generating powerful summaries. Another critical advantage is the extensibility offered by the algorithm. It is an extremely straightforward task to include further importance scores into the aggregation function. Indeed, if other sources of information can be exploited to guide summarization, this simply enhances the quality of the final summary produced.

3.2.1 Audio Importance Scores

In this section, the algorithm used to calculate $\alpha(n)$ is described. Since $\alpha(n)$ represents an importance score derived from the audio (speech) signal of the presentation, the aim is to identify important, emphasized portions of the speech. As described earlier, the work in [37, 45] presents a pitch-based importance detection scheme. The rationale behind this approach is that linguistics research indicates that emphasized portions of speech correspond to an increase in pitch activity. The main drawback with this method is that pitch information is difficult to extract [37, 38], especially in noisy environments such as presentations.

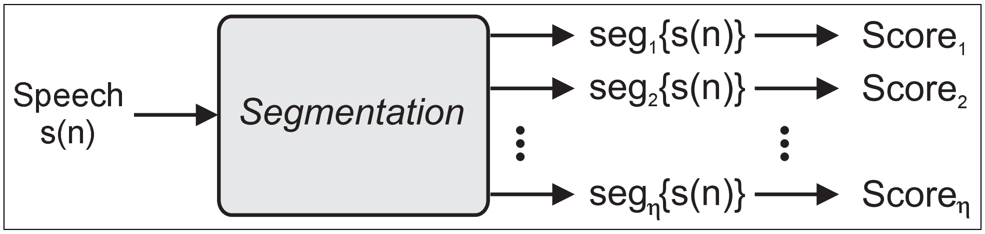


Figure 3.16: System overview for calculation of audio importance scores.

This thesis proposes to exploit the correlation between pitch activity and importance of speech segments using the information theoretic measure of entropy, hence providing an efficient and robust strategy for calculating importance scores in noisy speech [62]. An overview of the proposed algorithm appears in Figure 3.16.

Consider a discrete speech signal $s(n)$, divided into η segments of length L so that:

$$s(n) = \left[seg_1\{s(n)\} seg_2\{s(n)\} \cdots seg_\eta\{s(n)\} \right], \tag{3.4}$$

where $seg_i\{s(n)\}$ is composed of samples from $s(n)$ such that $n \in [(i-1)L+1, iL]$. As shown in Figure 3.16, the objective of the proposed method is to generate a score for each segment of the signal. In the case where each segment is one second in length, these scores correspond exactly to the $\alpha()$ sought. For now, the score for a segment i is denoted by $\alpha(seg_i\{s(n)\})$.

In [37], the importance score $\alpha(seg_i\{s(n)\})$ was calculated as the pitch activity in $seg_i\{s(n)\}$. In this thesis, these scores will be calculated as a function of the spectral entropy of each segment. This proposed approach is significantly more efficient than [37] and ASR-based methods, and capable of real-time operation. In addition, spectral entropy is easy to compute in contrast with pitch estimation, which is a difficult task.

Spectral entropy, a measure of uncertainty, has been successfully used as a speech feature in several applications including speech recognition [63, 64] and voice activity detection [65, 66]. Here, it is proposed to use spectral entropy as a feature to determine important segments in speech. This is motivated by the fact that emphasized portions of speech result in an expanded pitch range [37]. Since spectral entropy is a measure of the non-uniformity of a signal [63], a relation between the importance of a segment of speech and spectral entropy is expected. This is confirmed by empirical results reported in the next chapter. It remains to outline the exact procedure to obtain the importance scores, $\alpha(seg_i\{s(n)\})$.

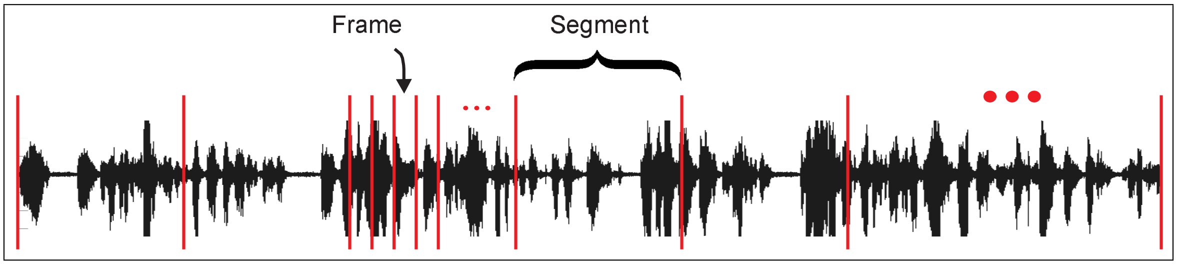


Figure 3.17: Speech waveform broken into segments consisting of audio frames.

Spectral entropy is calculated in the frequency domain [63]. Since speech is a non-stationary signal, frequency analysis is performed using the short-term Fourier transform (STFT). The STFT for the i^{th} segment in $s(n)$ is given by:

$$S_i(\omega, k) = \sum_{n=(i-1)L+1}^{iL} seg_i\{s(n)\} w(n - \lambda(\frac{1}{2} + k)) e^{-j\omega n} \tag{3.5}$$

where $w(n - \lambda(\frac{1}{2} + k))$ is a Hamming window of length λ centered at $\lambda(\frac{1}{2} + k)$, $k = 0, \ldots, \frac{L}{\lambda} - 1$. This results in L/λ non-overlapping windows, partitioning $seg_i\{s(n)\}$ into non-overlapping frames known as *audio frames* as in [37]. The relationship between audio frames, segments and the speech waveform itself is shown in Figure 3.17. A Hamming window is applied in order to minimize leakage. The value of λ is experimentally determined and divides L. The score $\alpha(seg_i\{s(n)\})$ is then computed as follows:

1. For each k, compute the power spectrum $|S_i(\omega, k)|^2$.

2. Generate a probability mass function (pmf) by normalizing each sub-band in $|S_i(\omega, k)|^2$. Thus,

$$p_i^k(\omega = \omega_m) = \frac{|S_i(\omega_m, k)|^2}{\sum_{\omega_m} |S_i(\omega_m, k)|^2} \tag{3.6}$$

3. Compute the spectral entropy as:

$$H_i^k = -\sum_{\omega_m} p_i^k(\omega_m) \log p_i^k(\omega_m) \tag{3.7}$$

A provisional score is calculated from the set of all spectral entropies as:

$$\alpha'(seg_i\{s(n)\}) = \sqrt{\frac{\lambda}{L}\sum_k (H_i^k - \mu_i)^2} \tag{3.8}$$

where $\mu_i = \frac{\lambda}{L}\sum_k H_i^k$.

Finally, a W-second sliding window is used to smooth the scores in each segment to provide a score for the approximate duration of a complete sentence, as in [37]. In the end:

$$\alpha(seg_i\{s(n)\}) = \sum_{a=max(1,i-W/2)}^{min(\eta,i+W/2)} \alpha'(seg_a\{s(n)\}) \tag{3.9}$$

To conclude, a simple threshold based on the total audio power in a one second segment of the speech signal is used to determine silent segments. The scores for these segments is set to be 0. As a final step to ensure the scores lie in the range $[0, 1]$, each score is divided by the maximum score among all scores.

3.2.2 Slide Importance Scores

This section outlines the procedure used to obtain importance scores for slides, $\beta(s_k)$. The importance of slides can only be judged with respect to a query keyword. In other words, it does not make sense to ascertain the importance of a slide under no constraints. Rather, under the constraint that a user is interested in a given topic, one can calculate the relevance (that is, importance) of a slide. This is exactly the role played by the user-supplied keywords discussed in the previous chapter. The entire set of presentation slides is first filtered with respect to each user-supplied keyword, and a score is calculated for each slide with respect to each keyword. A fuzzy set framework is proposed to calculate the importance scores for slides. In addition, the hierarchical structure of slides is exploited by the proposed algorithm.

In order to calculate a score for a slide, we begin by extracting slide features. To reflect the structured nature of XML-based slides, a *feature hierarchy* is constructed as depicted in Figure 3.18.

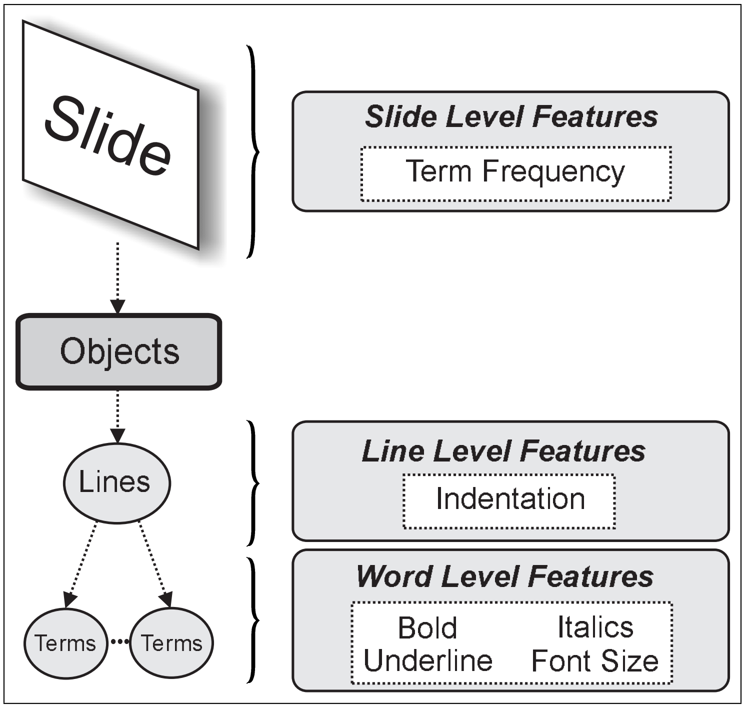


Figure 3.18: Slide feature hierarchy.

Note that a simplified, skeletal version of the slide structure tree shown in the previous chapter appears on the left side of Figure 3.18. This serves to illustrate the link between the hierarchical structure of the slide and the features composing the feature hierarchy. As shown in the figure, three sets of features are extracted from a slide. Word level features are the base of the hierarchy. These represent such features as typeface, and font size, which can vary from term to term on a slide. At one higher step in the hierarchy are the line level features, such as indentation level of a bullet. Clearly, this feature applies to all terms on a particular line or bullet, but varies between different bullets. Finally, features such as term frequency are slide level features, since they are constant over an entire slide. The following subsections detail the features at various levels of the feature hierarchy.

Before proceeding, it is mentioned that the terms extracted from a slide in this proposed method are not restricted only to textual words displayed on the slide. In fact, in the proposed scheme, terms may be extracted from metadata stored in images, for example, using Exif or JPEG-2000 metadata. Note that this is yet another aspect which sets the proposed method apart from existing methods, which focus solely on text visible on the slide itself. Metadata extracted from images may provide valuable summarization hints.

Word Level Features

The features composing the lowest level of the hierarchy describe characteristics of individual terms on a slide. These characteristics are typeface, of which three in particular are considered - bold, italics and underline, and font size. These features are extracted since they are most commonly adjusted by presenters to place emphasis on certain terms in the slide [67].

- **Typeface:** Typeface features represent formatting attributes of a term. These attributes include bold (B), italics (I), and underlined (U). These features are denoted as $B(t)$, $I(t)$, and $U(t)$ for a term t and are binary in nature; that is, a given term is either bold, or not, and so on. Mathematically, these features are defined by an indicator function as follows:

  $$B(t) = \begin{cases} 1 & \text{if } t \text{ appears in bold,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

  $I(t)$ and $U(t)$ are defined analogously.

- **Font size:** This feature indicates the font size of a term t, denoted as $size(t)$. The size of a term is related to importance since prominent terms, such as slide titles, are generally set apart by an increase in font size. The font size for a term t is given by:

  $$size(t) = z, \qquad \text{for } z \in \mathbb{N}, \tag{3.11}$$

  where $\mathbb{N}$ is the set of positive integers.

Line Level Features

At the next level in the feature hierarchy, features that are constant for a bullet are defined. From this set of features, this work focuses on the indentation level of a bullet, denoted as $indent(t)$ and mathematically calculated as:

$$indent(t) = d, \tag{3.12}$$

where $d \in (0, D)$ and 0 corresponds to the depth of the slide title and D is the maximum indentation level in the slide. Note that while indentation is considered as a line feature, $indent(t)$ is defined for an individual term t for convenience.

The role of indentation level in perceived importance is clear: generally, finer details will have deeper indentations, whereas high-level concepts will appear as main bullets.

Slide Level Features

Slide level features compose the highest level in the proposed hierarchy. This thesis computes a term frequency, defined as the number of occurrences of a term within a particular slide as a slide level feature. Mathematically, this feature is defined as:

$$TF(t) = n, \tag{3.13}$$

where $n \geq 0$ is the number of times the term t appears on the given slide.

The above discussion characterizes the complete set of features extracted from slides for the purpose of slide importance score calculation. The slide importance score cannot be calculated simply as a combination of these features, directly, however. Note that the features are on completely different mathematical scales. For example, the typeface features take on only binary values, whereas term frequency may conceivably take on any positive integer. Consequently, rather than combine the extracted features directly, this thesis proposes aggregating scores calculated based on the individual features. This idea is shown in Figure 3.19.

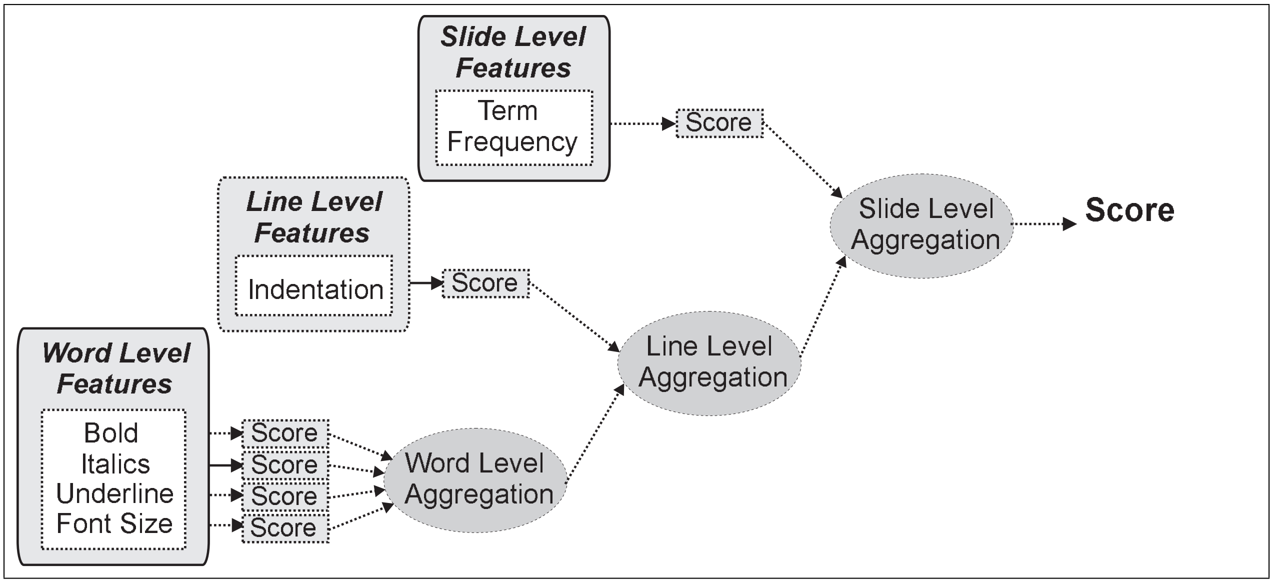


Figure 3.19: Score aggregation in a hierarchical manner to calculate slide importance score.

As mentioned at the start of this section, this thesis proposes the use of fuzzy sets [68] to calculate importance scores. This is motivated by the effectiveness demonstrated by fuzzy sets in modeling high-level human concepts [69] such as importance. In [69], the high-level concept of similarity between images is modeled using fuzzy sets and applied to the problem of content-based imaged retrieval (CBIR).

In order to calculate the scores for each feature in the feature hierarchy, a fuzzy set *importance* is defined. This is a complex high-level human concept that can be modeled based on the lower level concepts in the proposed feature hierarchy. A feature score is then the membership grade of a term to the fuzzy set *importance* based on a given feature. The grade of membership - a value between $[0, 1]$ inclusive - of an object to a fuzzy set is defined by a membership function [68], and indicates the degree to which the object is a member of the particular fuzzy set. This should be contrasted with the concept of membership to a traditional set which is binary in nature. Thus, the main challenge that remains is to devise appropriate membership functions that map an extracted slide feature value to a score in the range $[0, 1]$.

Fortunately, in [70], a generalized functional form for the membership function is derived. Thus, for $x \in [a, b]$, the generalized member function $\mu(x)$ is given by:

$$\mu(x) = \frac{(1-\nu)^{\lambda-1}(x-a)^{\lambda}}{(1-\nu)^{\lambda-1}(x-a)^{\lambda} + \nu^{\lambda-1}(b-x)^{\lambda}}. \tag{3.14}$$

Equation 3.14 defines a family of s-shaped curves, with the parameters λ and ν controlling the sharpness and inflection point of the curves. Note that for $\lambda = 1$, Equation 3.14 reduces to a linear function given by:

$$\mu(x) = \frac{x-a}{b-a} \tag{3.15}$$

The use of the linear form of the membership function given in Equation 3.15 is motivated mainly by its simplicity.

One added complication is the fact that each of these features is context-dependent. In other words, if a particular term t is bold, this does not convey any measure of importance in isolation. Indeed, the remaining terms on the slide may themselves be bold, in which case the bold typeface of term t is inconsequential. Similar arguments apply to the other features. Thus, the membership functions sought should be of the form given by Equation 3.15 while accounting for the context of the term t. In the end, the following membership functions arise and are used for calculating feature scores:

For typeface (given in terms of the bold feature $B(t)$, but it is the same for the remaining typeface features):

$$\mu(B(t)) = \begin{cases} \dfrac{C-\Sigma}{C-1} & \text{if } B(t) = 1, \\ 0 & \text{if } B(t) = 0. \end{cases} \tag{3.16}$$

where C is the total number of terms on the slide, and Σ is the total number of bold terms on the slide. Note the linear form of membership function, and note further the context dependence through the inclusion of Σ.

The membership function for $size(t)$ is defined as:

$$\mu(size(t)) = \frac{size(t) - min_{s_i}(size(\bar{t}))}{max_{s_i}(size(\bar{t})) - min_{s_i}(size(\bar{t}))} \tag{3.17}$$

where $min_{s_i}(size(\bar{t}))$ is the minimum font size and $max_{s_i}(size(\bar{t}))$ is the maximum font size of any term $\bar{t}$ on the current slide s_i. Note again the linear form and dependence on font size information in the entirety of the slide.

Finally, the membership function for $indent(t)$ is defined in a similar fashion, given by:

$$\mu(indent(t)) = \frac{indent(t) - min_{s_i}(indent(\bar{t}))}{max_{s_i}(indent(\bar{t})) - min_{s_i}(indent(\bar{t}))} \tag{3.18}$$

where $min_{s_i}(indent(\bar{t}))$ is the minimum indentation level and $max_{s_i}(indent(\bar{t}))$ is the maximum indentation level of any term $\bar{t}$ on the current slide s_i.

What remains is to combine these feature scores in an appropriate manner. For this thesis, two aggregation operators were considered, the Generalized Means [71], given by:

$$\mathcal{A}(\mu_1, \ldots, \mu_N) = \left(\sum_{i=1}^{N} w_i \mu_i^p \right)^{1/p} \tag{3.19}$$

where $p \in \mathbb{R}$, and $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$, and the particular form of operator from the family of Compensatory Operators, given by:

$$\mathcal{A}(\mu_1, \ldots, \mu_N) = \gamma \max(\mu_1, \ldots, \mu_N) + (1 - \gamma) \min(\mu_1, \ldots, \mu_N) \tag{3.20}$$

The final score calculated for each slide is in the range $[0, 1]$ as required.

In addition to the proposed method aiding in AV presentation summarization, the proposed scheme for slide transcoding is evident at this point. In this thesis, the concept of slide transcoding is restricted to excising bullets from slides to reduce their display size for small-screen devices. Due to the manner in which slide scores were calculated, each line, or bullet, already has an importance score associated with it. Thus, it is a simple matter to remove as many of the lowest scoring bullets from a particular slide as required to reduce the display size of the slide.

3.2.3 Usage Importance Scores

In this section, the details for calculating $\gamma(n)$ are provided. As stated earlier in this section, $\gamma(n)$ is a function of both the current time n, and the slide that is on display at the current time s_i.

More specifically, $\gamma(n)$ is a function of e. Recall that e is a member of the vector $\vec{S}(n)$ describing the fraction of $dur(s_i), s_i \in \vec{S}(n)$ that has elapsed since s_i was first displayed.

The usage importance score seeks to factor in two heuristics determined by the research in [6] through user pattern analysis. The first heuristic states that the amount of time spent on a slide during a presentation is proportional to the overall importance of that segment of the presentation relative to the other segments. Mathematically, for any two slides s_i and s_j, if

$$\frac{dur(s_i)}{N} > \frac{dur(s_j)}{N}$$

then the heuristic suggests the segment of the presentation coinciding with slide s_i is more important than the segment coinciding with slide s_j. Note that in the above condition, a factor of $\frac{1}{N}$ is included simply to normalize the duration values to lie in the range $[0, 1]$. Finally, owing to the first heuristic, the usage importance score for a slide is given by:

$$\gamma_{h1}(n) = \frac{dur(s_i)}{N} \tag{3.21}$$

where the subscript $h1$ indicates 'heuristic 1'. Note once more that there is no ambiguity in writing s_i, since the slide on display at any particular time instant is unique.

The second heuristic mentioned in [6] is based on empirical results of usage patterns. Usage data was collected for over $6,000$ views of over 150 presentations. This data indicated that the more important information was presented toward the beginning of a slide, rather than the end. In short, users were able to assess the relevance of presentation content quickly upon a slide transition. Thus, the portion of a presentation immediately following a slide transition are more important than other portions. Then, over the duration of an entire slide, it is possible to quantify the second heuristic with a curve of the shape shown in Figure 3.20 for a sample slide duration of 100 seconds. Thus, the value of the curve at any fraction of the total duration of the slide e gives the score for this heuristic, denoted by $\gamma_{h2}(n)$. In practice, it is found that fixing the first 25% duration of a slide at the constant value 1.0 produces better results.

Finally, the overall usage importance score is given by a linear combination of the two heuristic

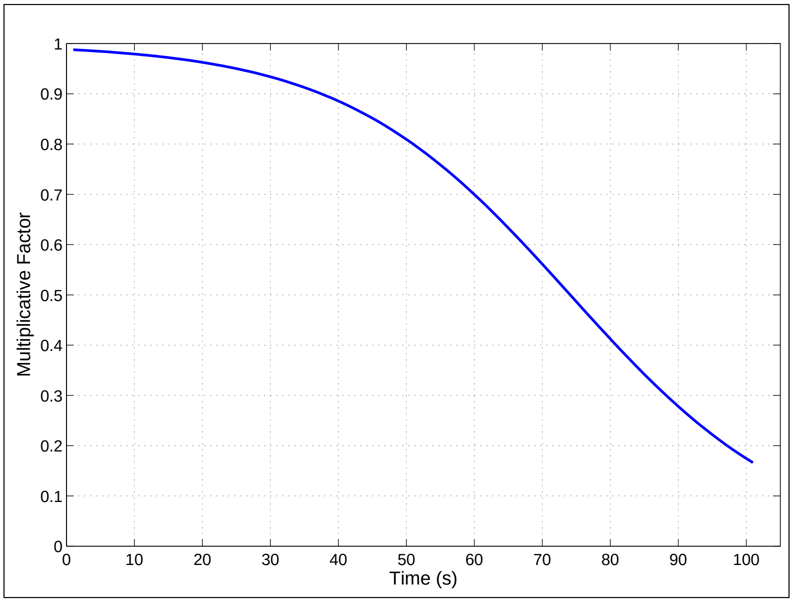


Figure 3.20: Decaying Importance over Slide Duration.

scores:

$$\gamma(n) = \xi_1 \gamma_{h1}(n) + \xi_2 \gamma_{h2}(n) \tag{3.22}$$

where ξ_1, ξ_2 are determined experimentally.

3.2.4 Aggregation of Scores

In the preceding sections of this chapter, the algorithms used to calculate the three importance scores, namely $\alpha(n)$, $\gamma(n)$ and $\beta(s_k)$ have been discussed. This section will discuss the aggregation of these scores to result, at the end, in an overall importance score for each second of the presentation. The problem of score aggregation is really one of data fusion. There has been considerable research in the area of data fusion. Before analyzing available data fusion approaches, it is important to understand what is actually being combined here.

Data fusion may be performed at several stages. As described in [72], there are three distinct stages at which fusion is possible: Premapping Fusion (Early Integration), Midst-Mapping Fusion (Intermediate Integration) and Postmapping Fusion (Late Integration). In the first possibility, data fusion occurs before the features have been processed and mapped into decision/opinion space.

There are two possibilities in this case: either the data itself is combined, or the features extracted from the data are combined. In the second stage, fusion is performed while the mapping from features to opinions is occurring. Finally, in Late Integration, data fusion, or more correctly in the present case, *opinion fusion* occurs after an opinion has been reached through utilizing several sources of information. The situation is depicted in Figure 3.21.

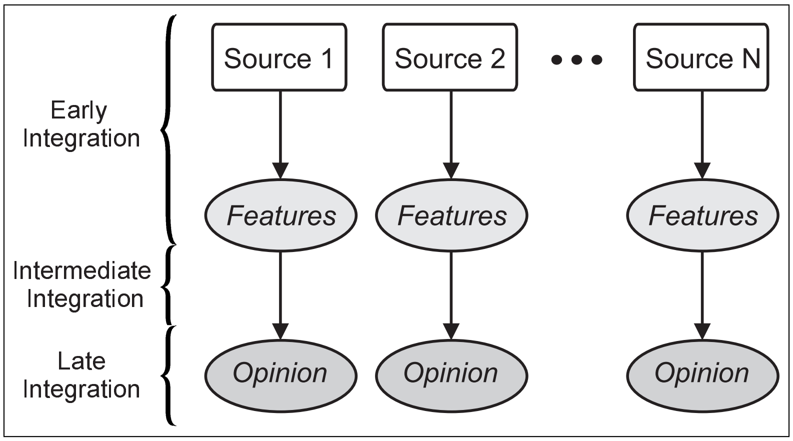


Figure 3.21: Data Fusion stages.

In the present case, a late integration opinion fusion scheme is proposed. The most common fusion operators in this case are the weighted summation and weighted product operators [72]. In [10, 11], a multi-modal approach to video summarization is presented. In this approach, an audio, visual and user attention model are created, and the overall importance of each segment of the video is obtained as a linear combination (weighted summation) of each of these models. This thesis proposes the use of a weighted summation operation to aggregate importance scores from audio, slides and usage information due to its simplicity and effectiveness as a 'middle-ground' in opinion fusion operators, between the two extremes conjunctive (AND) and disjunctive (OR) operators, which can be obtained from Equation 3.20. Furthermore, the weights on the summation may be individually varied easily to generate higher quality summaries.

So finally, the overall importance score for each second of the presentation is calculated as:

$$\begin{aligned} \psi(n) &= \rho_1\alpha(n) + \rho_2\beta s_k + \rho_3\gamma(n) && (3.23) \\ &= \rho_1\alpha(n) + \rho_2\beta(s_k) + \rho_3\gamma_{h1}(n) + \rho_4\gamma_{h2}(n) && (3.24) \end{aligned}$$

where the last term has been expanded, since $\rho_3\xi_1$ and $\rho_3\xi_2$ represent weights only, and may be suitably relabeled.

3.3 Presentation Summarization

This chapter concludes with the procedure to generate a presentation summary from all the scores above.

First, note that the scores $\alpha(n)$ and $\gamma(n)$ are calculated offline and inserted (or linked) into the `Presentation DI` at the time of creation, since these scores are constant over the lifetime of the presentation. The $\beta(s_k)$ scores, on the other hand, change dynamically in response to user-supplied keywords to guide summarization. Thus, at each summarization request, the $\beta(s_k)$ scores must be re-calculated.

The proposed `Presentation DI` is now augmented with the addition of a DIM, to enable DIP as outlined earlier in the thesis. Then, upon receiving a summarization request, the DIM queries the user for keywords, whereupon the scores $\beta(s_k)$ are computed. Next, a desired summarization ratio is obtained from the user. Finally, the highest scoring 1-second segments from the presentation are included in the output summary until the desired summarization ratio is reached. This summary is stored and delivered as an MPEG-7 `HierarchicalSummary DS`.

3.4 Chapter Summary

This chapter has described the proposed UMA AV presentation summarization system design in detail, along with a description of the proposed summarization algorithm. In terms of system design, the key concept of a `Presentation DI` was introduced. The packaging of various presentation multimedia elements was given in a step-by-step manner, along with appropriate XML code snippets to describe elements of the proposed `Presentation DI`. The proposed `Presentation DI` is a novel contribution of this thesis that advances the state-of-the-art in DL applications. The design of the `Presentation DI` addresses the UMA concerns described at the beginning of this work to allow for delivery of personalized content to a diverse set of users.

The proposed summarization algorithm was described as a combination of importance opinions from three subsystems: audio analysis, presentation slide analysis and usage analysis. The audio analysis framework was based on the information theoretic measure of spectral entropy which exploited the natural pitch range expansion that occurs in emphasized human speech. The application of spectral entropy to the detection of importance segments in audio data is another novel contribution of this work. The effectiveness of the proposed audio analysis approach in identifying important segments is tested in the following chapter.

Presentation slide analysis is performed within a proposed fuzzy set framework. The use of fuzzy sets allowed the modeling of high-level human concepts such as importance, a key technical challenge as outlined earlier in the thesis. By exploiting the hierarchical structure of presentation slides, a novel feature hierarchy was proposed. Scores were evaluated for features using a linear membership function and finally, these scores were combined hierarchically resulting in an overall slide importance score. The application of fuzzy sets to the problem of presentation slide analysis is an innovative approach and is a specific contribution of this thesis. The viability of the proposed slide analysis scheme is examined by utilizing the scheme in a slide retrieval scenario in the next chapter.

Finally, the importance scores for presentation segments were obtained as a weighted average of the importance scores of the individual subsystems. One key advantage of the proposed opinion fusion scheme is the ability to incorporate additional sources of information easily. Thus, the proposed summarization framework is easily extensible. Experimental results indicating the superior performance of the proposed algorithm when compared to a current approach is presented in the following chapter.

Chapter 4

Evaluation of Summarization Algorithm

This chapter presents an experimental evaluation of the various proposed algorithms in this thesis. Specifically, the efficacy of the proposed audio analysis subsystem, presentation slide analysis subsystem and overall presentation summarization algorithm is evaluated in this chapter.

4.1 Objectives and Evaluation Methodology

To effectively gauge the operation of the algorithms, each proposed algorithm is tested both independently as well as ensemble as described below. Before providing details of the experimental setup for each algorithm, the general evaluation methodology is discussed in this section.

Presentation summarization is a difficult task due to the complexity associated with determining the importance — a high-level human concept — of segments of a presentation. Indeed, the evaluation of the quality of a presentation summary is a complicated, ill-defined task for exactly the same reason [10–12]. In particular, for “uniform-content” multimedia, such as presentations, it is hard to assign quantitative measures of summary quality. This should be contrasted with multimedia such as sporting events. In this case, important events are more readily recognized, for example, a goal scored in hockey or a three-point shot made in basketball. In this case, one metric for summary quality is the coverage of these events in an automatically generated summary.

One possible evaluation method based upon the observation above as regards sporting events is to have a set of experts define important events from the full-length presentation. Then, much like goals and three-point shots, a metric that may be used to judge summary quality is to determine the coverage of the expert-defined important events in the summary. However, there are drawbacks to this evaluation approach [73]. Specifically, experts rarely agree on a set of important events; it is not clear how to resolve difference between expert opinions. Most importantly, however, this is a laborious time consuming task to be performed by experts. For this reason, this thesis does not utilize this evaluation method.

In the present case, the primary objective of the experiments is to determine whether important segments are correctly identified by the individual subsystems and thus retained by the overall presentation summarization algorithm. This objective can be achieved through the design of an appropriate quiz-based user study. The use of user studies to evaluate the performance of summarization algorithms and systems is commonplace in current literature [6, 10–12, 48, 51, 74–76]. In this thesis, a quiz-based user study is employed to evaluate the performance of the proposed audio analysis algorithm and the overall presentation summarization algorithm. The steps followed to design a user study in this context are adapted from [77] and are outlined in Figure 4.1.

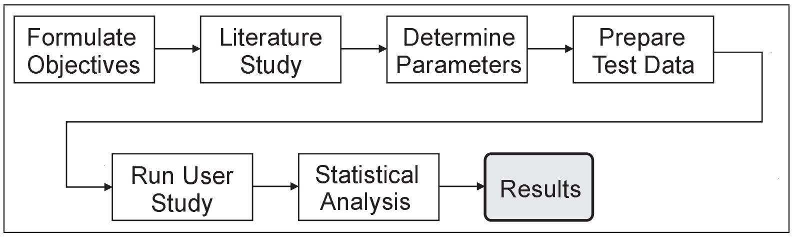


Figure 4.1: Steps involved in the design of a user study.

The specific parameters as they relate to particular experiments are given in the sections pertaining to each algorithm below. It should be noted that the scale of the user studies are influenced by two factors: first, the specific objectives of the experiment and second, by similar studies from the literature. For statistical analysis, the audio analysis and presentation summarization user studies use the two-sample t test (two-tailed). Further details are given in the appropriate sections.

Recall that the goal of the presentation slide analysis subsystem was to calculate the perceived importance of a slide relative to user-supplied keywords. Accordingly, the primary objective of the experimental evaluation of this subsystem is to verify that the most important slides are assigned the highest scores by the proposed algorithm. This objective is achieved through the design of a slide retrieval system utilizing the proposed algorithm as in [53]. Details about the slide retrieval system and experiments are presented in Section 4.3.

The remainder of this chapter is organized as follows: the next section provides results related to the audio score calculation method described in the previous chapter. Section 4.3 gives the performance results of the proposed slide importance score calculation algorithm and Section 4.4 presents results of the overall presentation summarization algorithm. Finally, the last section offers conclusions drawn from these experiments.

4.2 Audio Analysis

As mentioned, a user study was conducted to evaluate the performance of the proposed audio processing algorithm. Recall that the primary objective of this experiment was to evaluate the effectiveness of the proposed algorithm in identifying important segments. To this end, a time-compression algorithm was devised with the proposed audio processing algorithm used to determine important segments in the audio content. A secondary objective of this experiment was to evaluate the applicability of the proposed algorithm for the on-line mode of operation for a DL system.

4.2.1 Experimental Setup and Parameters

The study was conducted in the Multimedia Lab at the University of Toronto (UofT). A total of 29 students, proficient in English, from the Faculty of Engineering took part in the study. The students were randomly divided into two groups of 10 members each and one group of 9 members. Three talks were selected (designated **A**, **B**, **C**) from the ePresence archives[1] maintained by the

[1] http://epresence.tv/mediaContent/

Knowledge Media Design Institute at UofT. These clips reflected the type of presentations targeted by the proposed system. The clips were chosen to minimize user expertise in the subject matter.

Three minute segments were extracted from each of the three talks and a target compression factor 2.5 times the original length was chosen. The length of the clips were chosen to represent a typical gap in knowledge in the aforementioned on-line mode of operation of a DL system. The decision to time-compress the clips by a factor of 2.5 was based on previously discussed listening experiments [6, 48]. Further, it allowed for the testing of the assertion that speech sped up up to 2.5 times remains intelligible to listeners. Note that the clips used in the experiment represent real-world data derived from actual talks which took place on the UofT campus. They are corrupted with natural ambient noise from the surroundings. This enables testing the performance of the proposed method under naturally noisy conditions. In addition, the clips were chosen to cover a variety of characteristics, such as speaker gender, number of different speakers per clip and pace of the speaker.

The value of λ was chosen to result in audio frames of length $10ms$ (i.e. for speech sampled at a rate of $44.1kHz$, $\lambda = 441$). We set $W = 8$ throughout the experiment. Lastly, a 4^{th} order Butterworth low-pass filter was applied to the compressed speech signal, to enhance playback quality.

Three algorithms were compared: SOLA [46], the pitch-based algorithm described in [37] and finally the proposed approach (designated **SOLA**, **PITCH**, **SE** respectively). For the **SOLA** algorithm, the *respeed* program available in the *Speech Filing System*[2] speech research tool was employed. The autocorrelation method available in the Matlab speech analysis tool COLEA for pitch estimation was used in the **PITCH** method. The arrangement and order of clips is given in Table 4.2.1, where each entry indicates the clip and algorithm used to generate the summary. Such an arrangement of clips ensured that each group of students heard each of the three clips and each of the three algorithms.

Students were given a multiple choice quiz with each clip. The quiz consisted of six questions,

[2] http://www.phon.ucl.ac.uk/resource/sfs/

Group	Clip 1	Clip 2	Clip 3
Group 1	A-SOLA	B-PITCH	C-SE
Group 2	C-PITCH	A-SE	B-SOLA
Group 3	B-SE	C-SOLA	A-FREQ

Table 4.1: Order of clips presented to each group

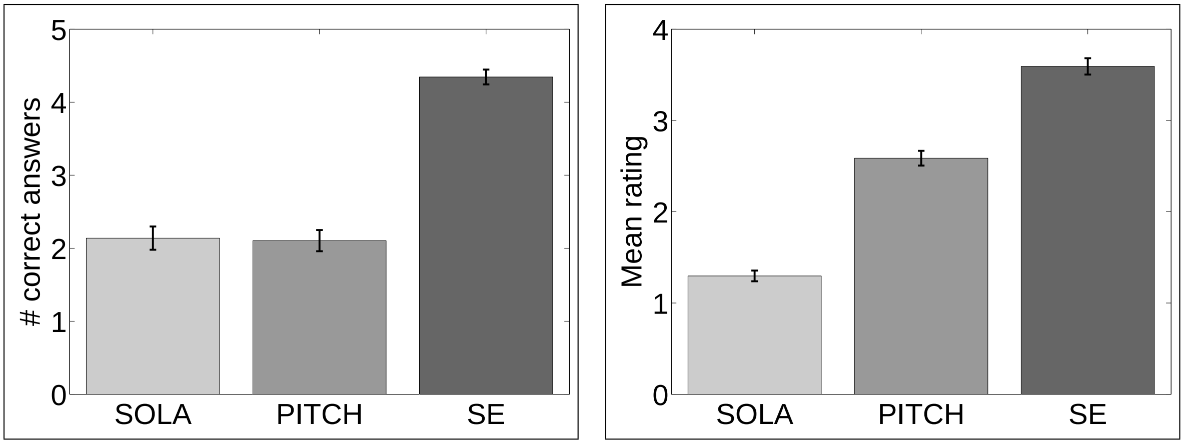
with four choices per question, exactly one of which was correct. The questions were prepared by two individuals using transcripts of the original clips. Every effort was made to ensure that the questions were simple and covered the majority of the content of the clip. Students were allowed to answer questions while listening to the clip. They were not allowed to pause or review any portion of the clip. A question was added to the end of the quiz to facilitate *intrinsic evaluation* [12] of the time-compressed clip. Students were asked to rate the ease of understanding of the clip (i.e. the audio quality of the clip) on a Likert scale [78], where a score of 1 indicated it was very difficult to understand the clip, whereas a score of 5 indicated it was very easy.

4.2.2 Results and Discussion

	SOLA	**PITCH**	**SE**
est. mean, $\hat{\mu}$	2.1379	2.1034	4.3448
est. standard deviation, $\hat{\sigma}$	1.7469	1.5889	1.1109
est. standard error, $\hat{\sigma}/\sqrt{n}$	0.31894	0.29009	0.20281

Table 4.2: Results of audio user study, grouped by method

The mean number of correct answers for the quizzes grouped by method are presented numerically in Table 4.2.2 and graphically in Figure 4.2(a). The estimated statistical values are calculated using the entire set of 29 quizzes per algorithm, without any group distinction. The error bars shown in Figure 4.2(a) represent the estimated standard error. It is evident that the proposed algorithm, **SE**, performs better than the other two algorithms under consideration. A two-sample



(a) Mean no. of correct answers per quiz, grouped by method.

(b) Intrinsic scores, grouped by method.

Figure 4.2: User study results.

t test (two-tailed) is used to assess the statistical significance of the results, with $p = 0.05$ as the cutoff for deciding significance. Then, the difference between the two algorithms **SOLA** and **SE** is statistically significant ($t = -5.7408, p = 3.9986\text{x}10^{-7} < 0.05$). The difference between **PITCH** and **SE** is also statistically significant ($t = -6.2259, p = 6.5057\text{x}10^{-8} < 0.05$).

The results in Table 4.2.2 indicate that the **PITCH** and **SOLA** algorithms perform similarly. However, when clip C is excluded from the results, the **PITCH** method outperforms the **SOLA** method (mean no. of correct answers are 1.8421 and 1.35, respectively). This is explained by the fact that the speaker in clip C spoke at a slow pace. Consequently, the time-compressed clip purely due to a speedup by a factor of 2.5 remained comprehensible.

Based on the intrinsic ease-of-understanding scores shown in Figure 4.2(b), it is clear that time-compressed speech produced by the proposed **SE** method is easier to understand than the output of either **SOLA** or **PITCH** methods. Student responses indicate that the constant speedup as in the **SOLA** algorithm results in difficult to understand time-compressed speech. This suggests that a speedup by a factor of 2.5 will not, in general, result in intelligible time-compression.

The results of this study demonstrate the viability of spectral entropy as a feature in determining important segments in speech. User study results demonstrated the effectiveness of the method applied to real-world speech archives over existing methods based on pitch estimation. Further, user feedback indicated that the time-compressed speech which results from the proposed algorithm is easier to understand than the output of current methods. Also, the results indicate that the proposed audio processing algorithm provides an effective means for summarizing relatively small portions of presentation missed by users during the on-line mode of operation.

4.2.3 Complexity Analysis

The runtime complexity of the proposed audio processing algorithm is certainly dominated by the time required to calculate the spectral entropy for the entire audio sample. In turn, the runtime to calculate spectral entropy is dominated by the time required to calculate the STFT of each segment. To calculate the STFT of a segment of length L takes $O(L \log L)$ time. For an audio clip with η segments, the running time then is $O(\eta L \log L)$.

Note again that this operation need only be performed once per presentation. Thereafter, the audio scores may be stored in the `Presentation DI` proposed in the previous chapter.

4.3 Presentation Slides Analysis

This section describes the methodology used to evaluate the performance of the proposed slide analysis technique. The objective of the current experiment is to evaluate the effectiveness of the proposed algorithm in correctly identifying important slides. This may be measured through a slide retrieval application. To this end, a complete slide retrieval system was implemented in the C# language. The system is designed in a hierarchical manner reflecting the structure of the proposed slide representation. The slide retrieval system includes the implementation of an OpenXML PowerPoint slide parser. The parser interacts with provided API to query the OpenXML tree through the Document Object Model (DOM). In addition, XPath queries are used to extract bullets, tables and images from slides. For the purpose of this study, however, tables and images are ignored in terms of further semantic processing. Text extracted from tables is still included in the entire text of the slide. Furthermore, the system is able to render OpenXML PowerPoint slides in HTML format for web-based interaction.

4.3.1 Experimental Setup and Parameters

Once again, three algorithms were compared: TF.IDF [54], the method proposed in [57] and finally the proposed approach (designated **TF-IDF**, **UPRISE**, **FUZZY** respectively). The retrieval tests were conducted on a repository of 134 presentations, consisting of a total of 2892 slides, giving an average of approximately 22 slides per presentation. The collection of presentations is diverse, including presentations from graduate Engineering courses to IEEE conference planning meetings to presentations on how to create good presentations. This selection of presentations is intentional to give a good representation of different types of presentations, and not bias the results.

The objective of slide retrieval systems is to return a ranked list of slides in response to a user query. In an ideal setting, the returned list of slides is in exactly the 'correct' order with respect to perceived importance. Note that this 'correct' order will in general not correspond with the chronological order of slides in a presentation, since the importance of a slide with respect to a keyword is not a function of its position within the presentation. In the real world, however, this is rarely the case. Thus, a common measure of performance in retrieval systems is *precision*. Precision represents the ratio of the number of relevant documents retrieved to the total number of relevant and irrelevant documents retrieved in response to a query. In the following, results are reported by measuring precision values for a fixed number of retrieved slides, over a range of retrieved slides.

One of the most difficult components of evaluating retrieval systems is generating the *ground truth* data, that is, the 'correct' ranking of relevant slides with respect to a particular keyword. In this study, ground truth data was generated manually over 2892 slides for each of the keywords used. In addition, keywords were only allowed to be used if a minimum of 10 slides in the repository contained the keyword. This was to ensure reasonable length of data for calculating precision values. Ultimately, 22 keywords were used for testing.

4.3.2 Results and Discussion

In the previous chapter, two aggregation operators for features were explored, namely Generalized Means:

$$\mathcal{A}(\mu_1, \ldots, \mu_N) = \left(\sum_{i=1}^{N} w_i \mu_i^p \right)^{1/p} \tag{4.1}$$

where $p \in \mathbb{R}$, and $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$, and the Compensatory Operator, given by:

$$\mathcal{A}(\mu_1, \ldots, \mu_N) = \gamma \max(\mu_1, \ldots, \mu_N) + (1 - \gamma) \min(\mu_1, \ldots, \mu_N) \tag{4.2}$$

For this experiment, the weights w_i in Equation 4.1 were fixed to $1/N$. Experiments were conducted to determine which of the two aggregation operators exhibits better performance. First, to determine the optimal value of the parameter p in Equation 4.1, precision curves were determined for 11 discrete values of p, given by $p \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. This range of values was chosen since Equation 4.1 represents a geometric average for $p \to 0$, whereas when $p = 1$, an arithmetic average is obtained. The resulting precision curves are shown in Figure 4.3.

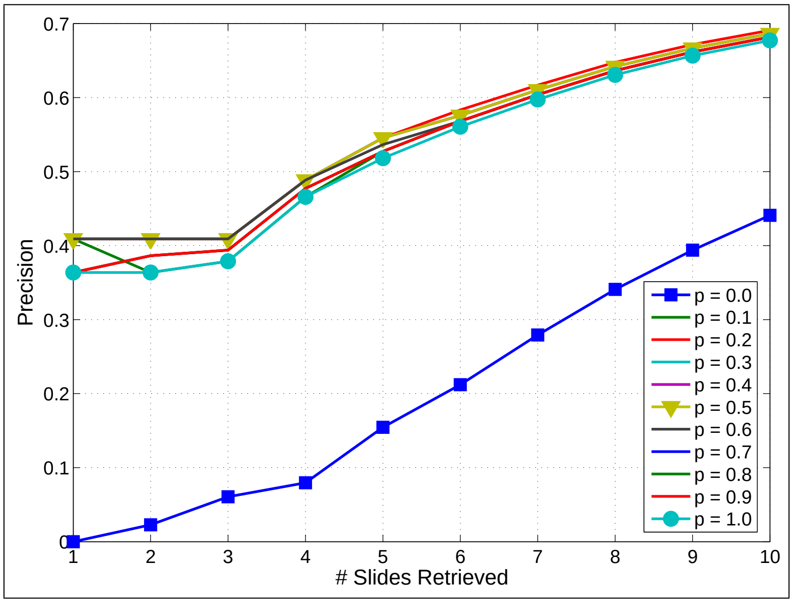
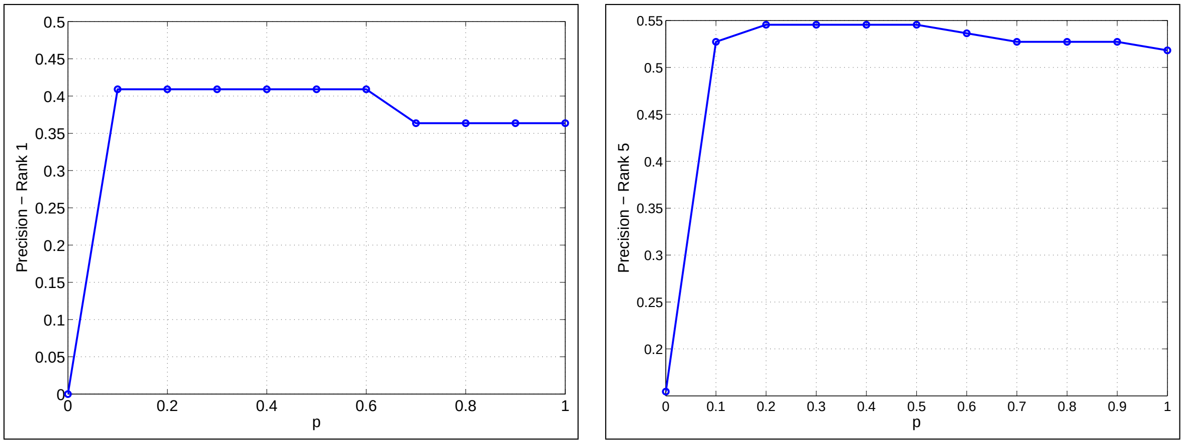


Figure 4.3: Precision curves for varying p in Generalized Means equation.

As can be seen, the performance of the aggregation operator for all values of p except $p = 0$ is very similar. For the case where $p \rightarrow 0$, note that Equation 4.1 approaches a geometric mean. Since some of the score μ_i may be 0 according to the definition of the membership functions, this degrades the performance of the operator in this circumstance. Of particular interest in the fact that for $p = 1$, the operator functions as a simple arithmetic mean yet the performance is comparable to all other values of p. To estimate the optimal value of the parameter p, plots of precision vs. p were made for slide retrieval ranks 1 and 5, shown below in Figure 4.4. As shown in the figures, the values of $p \in [0.2, 0.5]$ all result in equally good performance of the operator. Thus, a nominal estimate for the optimal value of p is 0.5.



(a) Precision vs. p, Rank 1.

(b) Precision vs. p, Rank 5.

Figure 4.4: Precision vs. p for Ranks 1 and 5.

A similar experiment is carried out with regard to Equation 4.2. Here, γ assumes values in the set $\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99, 1.0\}$. Figure 4.5 shows the performance of the operator over varying values of the parameter γ. Once again, the performance of the operator is comparable for most values of γ. However, at the extreme end when $\gamma = 1$, the performance is severely degraded. When the parameter $\gamma = 1$, Equation 4.2 behaves in a purely disjunctive fashion. This can be interpreted as overly optimistic behaviour which results in poor performance. Proceeding as before, an estimate for the optimal value of γ can be determined from Figure 4.6. In fact, this value is given by $\gamma = 0.2$.

At this point, a comparison can be made between the two operators when their parameters are

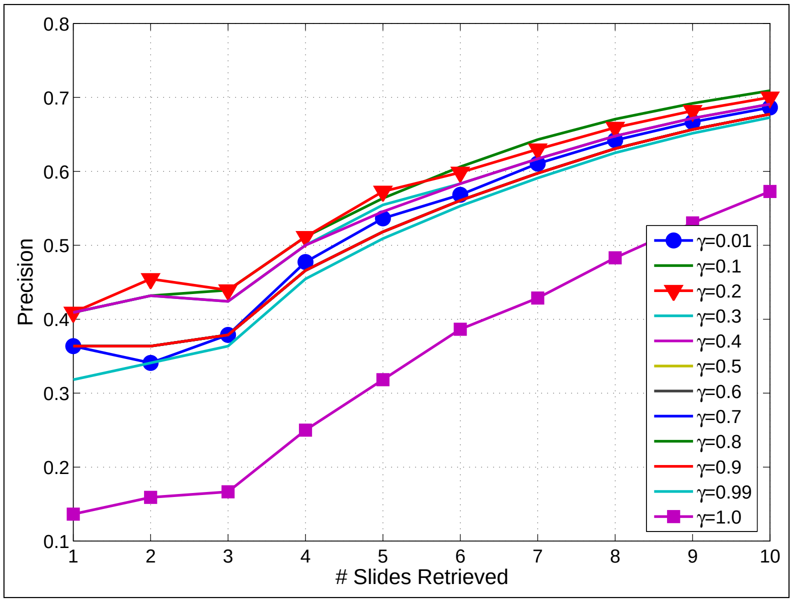
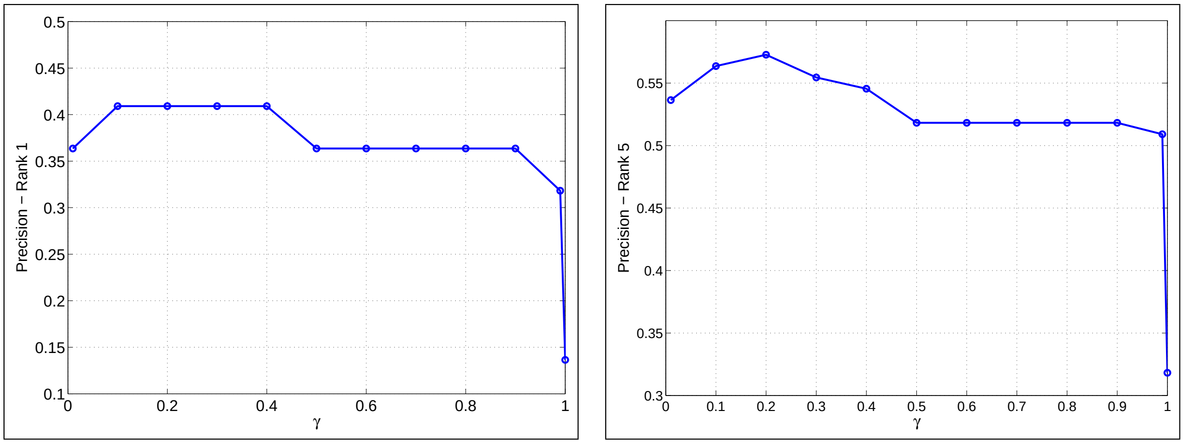


Figure 4.5: Precision curves for varying γ.

fixed at optimal values. This is shown in Figure 4.7. It is evident that the operator defined by Equation 4.2 outperforms the Generalized Mean operator. However, the difference between the two operators is minimal, with the maximum difference in precision values being approximately 4%. Consequently, for the remainder of this experiment, the Generalized Mean operator with $p = 1$ (that is, the arithmetic mean) is used. Note that as shown in Figure 4.3, the performance of the Generalized Mean operator is near optimal even in the case when $p = 1$. Besides the simplicity of the form of the operator in this case, another reason to set $p = 1$ is because the arithmetic mean exactly balances the two extreme operator behaviours (conjunctive and disjunctive).

Presently, the performance of the proposed features is evaluated. This is done by considering each proposed feature in isolation and computing the precision curves for a retrieval system based solely on the particular feature. Recall the features considered here are: TF (term frequency over a slide), ATTR (typeface attributes - bold, italics and underline), SIZE (fontsize) and INDENT (the indentation of a particular bullet). Figure 4.8 displays the results of this experiment.

From the figure, it is clear that the best performance is given by the TF feature, indicating that term frequency of a keyword over a slide is most strongly correlated with the human concept of



(a) Precision vs. γ, Rank 1.

(b) Precision vs. γ, Rank 5.

Figure 4.6: Precision vs. γ for Ranks 1 and 5.

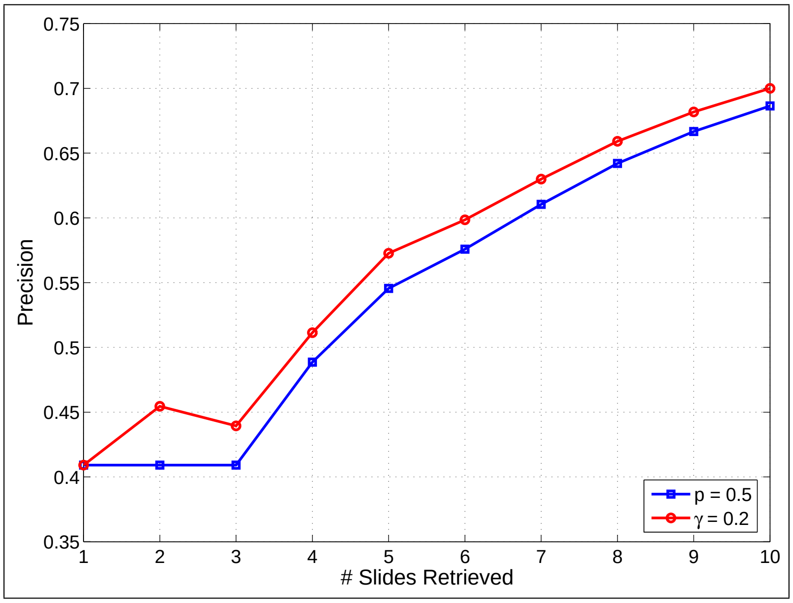


Figure 4.7: Comparison of aggregation operators.

importance. Note that the TF curve in Figure 4.8 is higher than the **TF-IDF** curve. Although the both these curves are calculated based on the frequency of occurrence of a keyword, the proposed feature in this thesis considers the frequency on a per-slide basis, whereas the TF.IDF method considers the frequency over the entire collection of documents. Consequently, the proposed TF feature alone outperforms existing systems.

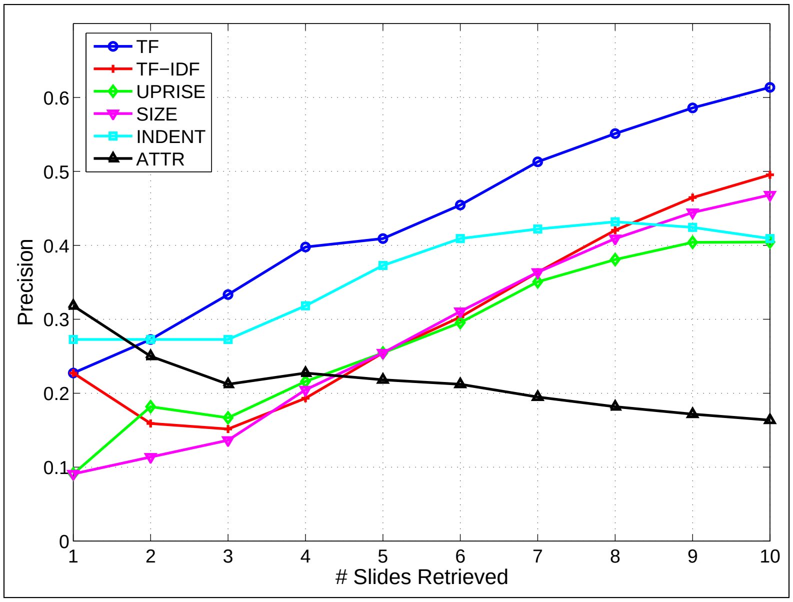


Figure 4.8: Performance of individual proposed features.

Another point to note from the figure is the poor performance of the SIZE feature. In fact, the SIZE feature never outperforms the TF feature. This suggests the SIZE feature should be dropped from consideration since it does not improve the slide importance score in any case, due to the arithmetic mean aggregation operator. Moreover, the remaining features, ATTR and INDENT only perform marginally better than TF in the early ranks. This suggests that the slide importance score calculated may be further improved by adjusting the weights w_i in the Generalized Means Equation 4.1. For the purposes of this thesis, however, the standard configuration of equal weights is retained, giving a sort of 'worst-case' performance.

Having analyzed the individual features one by one, this section concludes by presenting the overall results of the proposed system. The results in Figure 4.9 indicate the improved performance of the proposed **FUZZY** method as compared to a traditional **TF-IDF** method or an XML-based retrieval scheme such as in **UPRISE**. Specifically, the proposed method obtains a higher precision for a fixed number of retrieved slides. This result supports the claim that the features composing the proposed feature hierarchy provide additional semantic information for presentation slides.

Figure 4.10 confirms the earlier hypothesis that removing SIZE from the feature hierarchy would

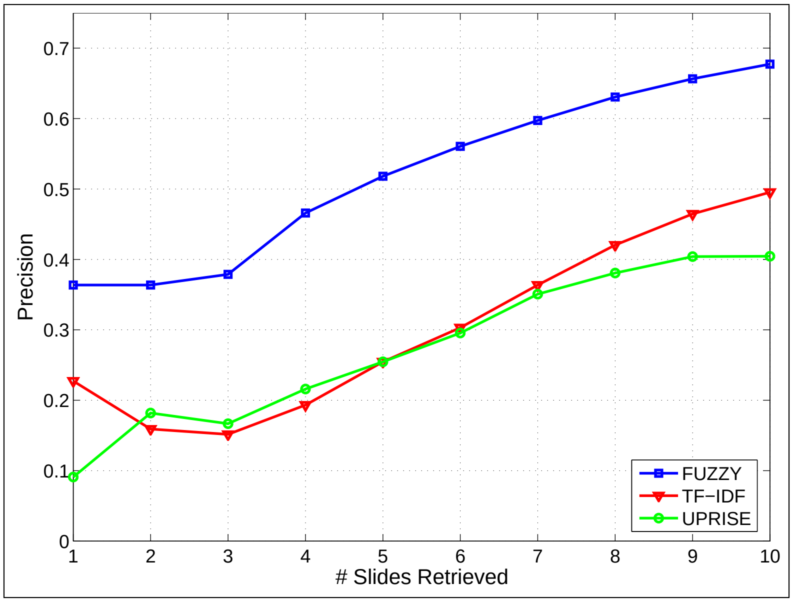


Figure 4.9: Slide retrieval results comparing three methods.

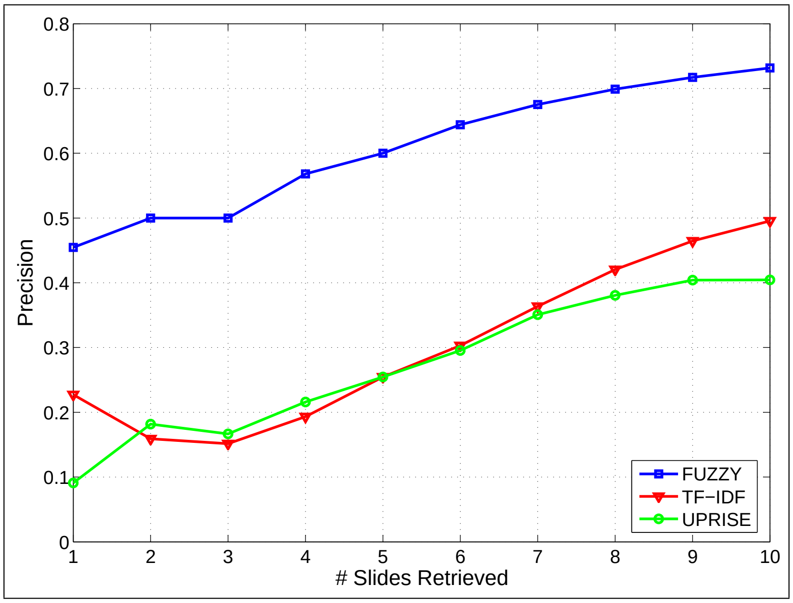


Figure 4.10: Slide retrieval results comparing three methods, ignoring SIZE attribute.

result in retrieval performance improvement. As can be seen, the precision curve is marginally higher in this figure where the curve is obtained by performing the retrieval experiment and ignoring the SIZE feature. The reason for the poor performance of this feature is the fact that font size does not change a great deal over the same presentation slide. One possible way to better utilize size information would be to consider size as a presentation level feature, rather than a word level feature, resulting in greater font size variation.

4.3.3 Complexity Analysis

The complexity analysis of the proposed slide processing method is presented here. The calculation of slide scores, along with all the other scores in the feature hierarchy for a typical presentation (e.g. 50 slides) is instantaneous. The performance bottleneck in this case is simply reading the OpenXML format off the hard-drive and parsing the XML representation of the presentation. Executing the query itself is much faster.

# of presentations	# of slides	Load time (s)	Query time (s)
1	17	0.046875	0.015625
10	170	0.4375	0.015625
100	1700	3.75	0.15625
200	3400	7.390625	0.328125
400	6800	14.796875	0.6875
800	13600	30.375	1.5

Table 4.3: Complexity of the implemented slide retrieval system.

The complexity of the slide retrieval system described above scales with the size of the slide repository. Table 4.3 gives an indication as to the performance of the system, with the results shown graphically in Figure 4.11. The 'Load time' column reports the time taken to load all presentations, which includes parsing the XML over all slides and building an index for future querying. As can be seen, with over $13,000$ slides, the system takes approximately 30 seconds to load and index the entire collection. The time to execute a query in the largest case is only 1.5 seconds, a very acceptable time. Overall, this system exhibits good performance, with the query times growing slowly with a large increase in total number of slides.

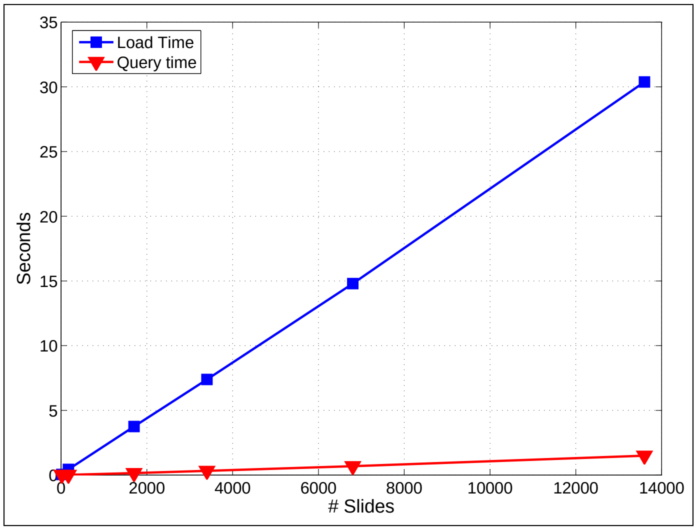


Figure 4.11: Slide retrieval load and query times.

4.4 Presentation Summarization

This section presents results of a user study conducted to test the viability of the proposed AV presentation summarization algorithm. The proposed algorithm was compared against the algorithm proposed in [6], specifically the ‘S only’ variant. In this variant of the algorithm, time is allocated to each slide in proportion to the total time spent on each slide by the presenter. For example, if the total duration of a presentation was 100 seconds, and there were 3 slides with the per slide durations being 10, 80 and 10 seconds respectively, the algorithm would assign 10% of the summary to the first slide, 80% to the second and the remaining 10% to the last slide. By following the second heuristic from the previous chapter, [6] assigns these portions of time starting at the beginning of each slide.

Recall the formula to calculate the overall importance score for a particular second of the presentation. This score is given by:

$$\psi(n) = \rho_1\alpha(n) + \rho_2\beta(s_k) + \rho_3\gamma_{h1}(n) + \rho_4\gamma_{h2}(n) \tag{4.3}$$

This is essentially an adaptive solution, since the weights ρ_i can take on any value in the range $[0, 1]$. Thus, in addition to comparing the proposed solution against an existing method, the study included audio only and slide only summaries. Note that these are easily obtained through Equation 4.3. Prior to conducting the main user study to assess the performance of the proposed summarization algorithm, a smaller study of 14 students was conducted with the purpose of determining the relative weights of slide and audio data. It is crucial here to distinguish between determining the relative *importance* of slides and audio versus determining their relative weights. It was found that in general, the audio information should be weighted greater than the slide information. This is because, even with a lower weight for slides than audio information, a slide would be displayed for a long enough period of time to allow users to skim slide contents. In essence, the infrequent rate of change of presentation slides meant that even with lower weights they would be visible in an automatically generated summary. This information was particularly useful for setting initial weights ρ_i. Note that the weights ρ_i for the proposed method were further refined experimentally. It is also possible to set these weights by using *a priori* knowledge, for example, biasing the importance score towards slide importance due to knowledge of the relative importance of the various components.

An interesting observation is that the algorithms proposed in [6] can be obtained through the general proposed method captured by Equation 4.3. For example, the ‘S only’ variant described above is obtained by setting $\rho_1 = \rho_2 = 0$ and $\rho_3 = 5\rho_4$. Thus, the method presented in [6] may be considered a special case of the proposed algorithm.

4.4.1 Experimental Setup and Parameters

The study was conducted in the Multimedia Lab at the University of Toronto (UofT). A total of 20 students, proficient in English, from the Faculty of Engineering took part in the study. The scale of the experiment was in line with similar works in the literature. The students were randomly divided into 4 groups of 5 members each. Two presentations were obtained from archives on the World Wide Web (designated **CERN** and **PODCAST**, respectively). Approximately 25 minutes of the beginning of each clip were extracted to be used as test data for the study. The choice of clip length was again guided by the literature, adhering to the steps suggested in Figure 4.1. A summarization factor of 20% was chosen, resulting in summarized versions that were 5 minutes in length. The keywords used to generate slide scores for the proposed method were chosen to represent the main topics of the presentations.

The four versions of summaries that were compared are denoted by **AUDIO**, **SLIDES**, **HE**, **PROPOSED**. Table 4.4.1 details the arrangement and order of clips presented to the four groups. The intersection of row and column in the Table indicates which group viewed the particular clip under the particular summary method. Due to the length of the clips, each group viewed only two methods. However, each method and clip was viewed by at least two independent groups.

Group	**AUDIO**	**SLIDES**	**HE**	**PROPOSED**
Group 1	CERN	PODCAST		
Group 2	PODCAST	CERN		
Group 3			CERN	PODCAST
Group 4			CERN	PODCAST

Table 4.4: Order of clips presented to each group

Students were given a multiple choice quiz with each clip. The quiz consisted of 10 questions, with four choices per question, exactly one of which was correct. The questions were prepared from the original full-length clips. Every effort was made to ensure that the questions were simple and covered the majority of the content of the clip. Students were allowed to answer questions while viewing the clip. They were not allowed to pause or review any portion of the clip.

4.4.2 Results and Discussion

In an attempt to gain a deeper understanding of the exact workings of the algorithm, this section begins with a discussion of the effect of varying the weights ρ_i and the summaries induced by these variations. Initially, consider the case where the audio information and slide information are weighted equally, with the other weights set to 0. The top half of Figure 4.12 shows the portions of the clip included in the summary as determined by the proposed algorithm; the bottom half of the Figure shows the portions of the clip included in the summary generated by **HE**.

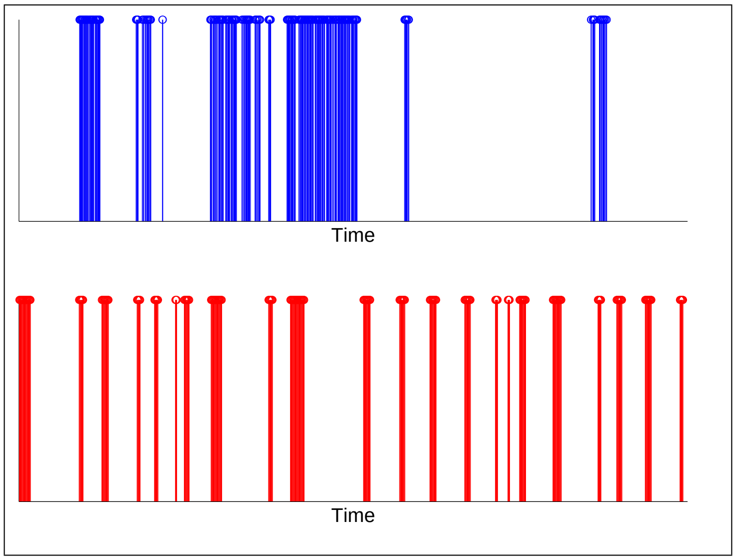


Figure 4.12: Overall summarization algorithm - audio and slides weighted equally.

In Figure 4.12, it is evident that the portions of the clip included in the summary coincide exactly with slides of non-zero score. Note how clusters have formed at portions of the clip corresponding to non-zero slide scores, giving poor coverage of the total presentation. By contrast, consider the image in Figure 4.13. Here, the audio weight is 7 times the slide weight. Note the formation of clusters of segments included in the summary generated by the proposed method are more evenly distributed throughout the length of the clip. Thus, the proposed summarization algorithm can be viewed as a clustering algorithm configured by adjusting the weights ρ_i. In this context, a clustering algorithm refers to the inclusion of a cluster of segments from the presentation within the summary. These clusters form within time intervals during which the slide on display contains the user-supplied keywords. Furthermore, Figure 4.13 confirms the results of the pilot user study which indicated a greater audio weight than slide weight. Lastly, Figure 4.14 demonstrates that for suitable values of ρ_i, the proposed algorithm generates very similar summaries to **HE**. In other words, the proposed algorithm can be seen as a generalization of current state-of-the-art presentation summarization schemes.

The results of the user study are given in Figure 4.15. The figure indicates the mean number of

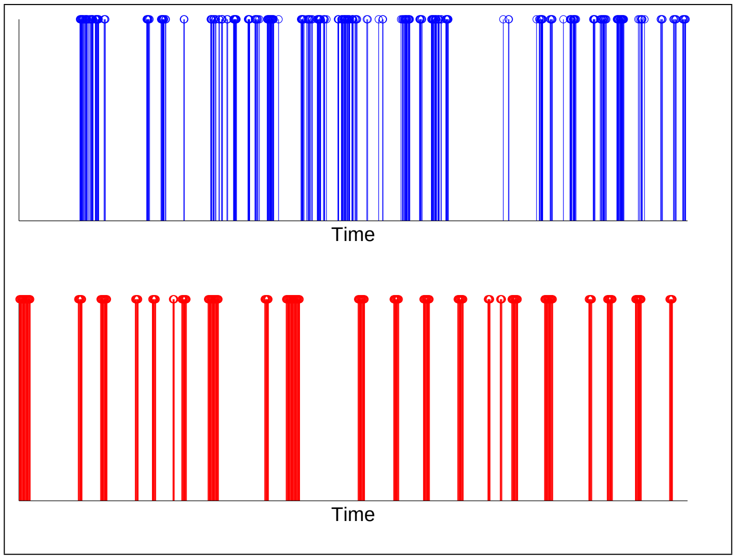


Figure 4.13: Overall summarization algorithm - $7\rho_1 = \rho_2$.

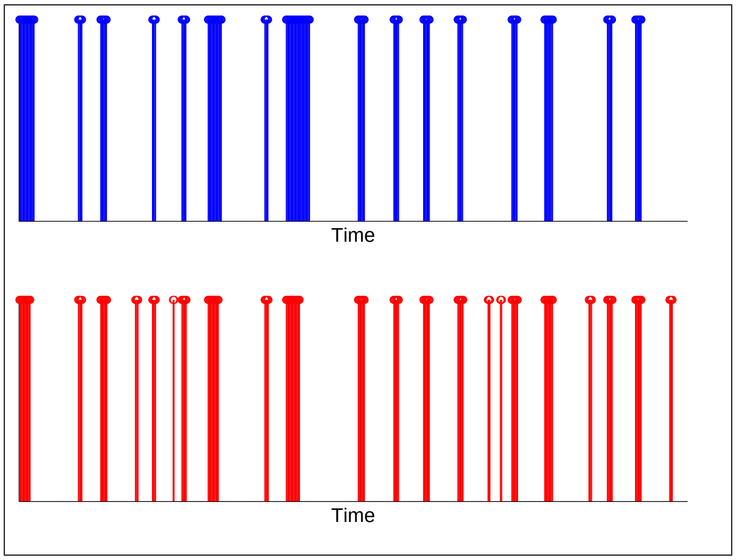


Figure 4.14: Proposed algorithm generates similar summary to existing algorithm for $\rho_3 = 5\rho_4$.

correct answers on the quiz, grouped by method. In a similar fashion to the audio results described in Section 4.2, a two-sample t test (two-tailed) is used to assess the statistical significance of the results, with $p = 0.05$ as the cutoff. The, difference between the two algorithms **HE** and **PROPOSED** is statistically significant ($t = -5.5468, p = 3.5788\text{x}10^{-4} < 0.05$). Thus, the user study demonstrates the effectiveness of the proposed method in performing presentation summarization.

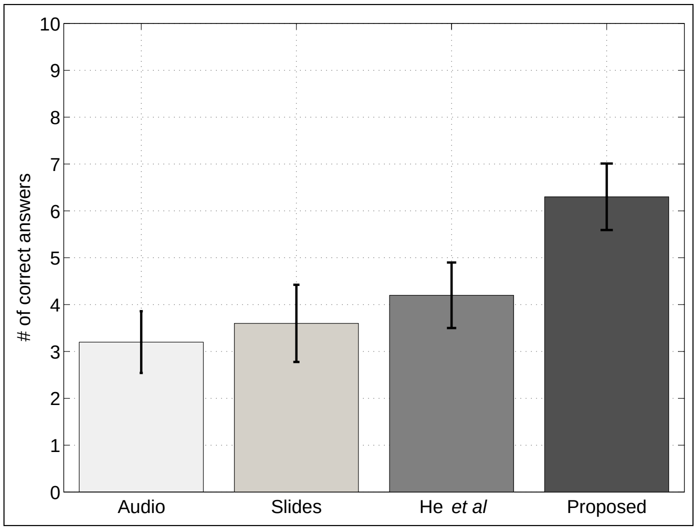


Figure 4.15: Mean number of correct answers, grouped my method

It should be noted that in general, the two methods which used a combination of both the audio domain and presentation slides performed better than the audio or slide only methods. Again, this gives credence to the fact that a multi-modal approach is a more powerful means of automatically generating presentation summaries.

4.4.3 Complexity Analysis

The proposed presentation summarization algorithm relies on multi-modal information, specifically importance analysis from the audio channel, presentation slides and usage information. Recall that the importance information extracted from the audio channel can be cached for future use in the proposed `Presentation DI`. Thus, there is a one-time fixed cost for determining audio importance. For every summary request, however, the scores $\beta(s_k)$ must be recomputed, since these are a function of the particular user-supplied query keyword. Assuming a large presentation of 100 slides, the total load and query time would still be under 1 second. The usage history information is also encapsulated in the `Presentation DI`.

Thus, the proposed summarization algorithm is very efficient, since it is required to simply compute a weighted average of these importance curves. The bottleneck in the entire system is determining the threshold to achieve a particular summarization ratio, and creating the summarized audio data through excision, since this requires examining each value in the overall presentation importance scores.

4.5 Chapter Summary

This chapter has detailed the methodology involved in evaluating the proposed summarization algorithm as well as the constituent subsystems. Based on the desired objectives, a task-based user study was chosen as the method of evaluation for the proposed audio analysis technique and overall summarization algorithm, in line with existing literature in the area. The proposed slide analysis system was evaluated within a slide retrievals application.

The user study results from the audio analysis subsystem demonstrate the applicability of spectral entropy as a feature in determining important segments from audio data. It was shown that the proposed system outperforms existing methods, and is capable of near real-time operation, ideally suited for the on-line mode of operation of existing DL webcast systems.

Next, the slide retrieval results indicated the superiority of the proposed slide processing algorithm. The fuzzy logic based design performed better at retrieval tasks than traditional methodologies including the text-based TF.IDF method. The impact of features from the proposed feature hierarchy were examined in isolation, providing a better indication of the performance of each feature. It was found that the font size feature does not provide useful information toward the determination of slide importance.

Finally, the proposed algorithm was analyzed through another task-based user study. It was shown that existing presentation summarization algorithms [6] can be seen as a special case of the proposed summarization algorithm. Moreover, an interpretation of the proposed technique as a clustering algorithm was presented, and the influence of the various subsystems was explored. Lastly, user study results indicated that the proposed solution generates better automatic summaries than existing methods.

Chapter 5

Conclusions

Multimedia-rich presentation archives are increasingly common on the World Wide Web. With the rapid adoption of distance learning pedagogy, there is a need for effective methods of accessing these presentation repositories. Moreover, the explosion in popularity of myriad mobile personal devices necessitates a conscientious effort to design systems that can deliver multimedia content to users anytime, anywhere, in any format and over any network with ease. These are the fundamental issues addressed by this thesis.

5.1 Conclusions

This thesis proposes a novel UMA system designed to perform AV presentation summarization. The system design is based on existing and emerging international standards MPEG-7 and MPEG-21. The advantages of using these standards are that they promote interoperability, and enable both user-preference based personalization and seamless processing of content. The design of the system revolved around the concept of a `Presentation DI`. The proposed `Presentation DI` encapsulates all content related to presentations, such as the audio track, presentation slides stored in a custom proposed format and assorted metadata extracted through content analysis to guide future processing.

In addition, an innovative summarization algorithm based on multi-modal opinion fusion is proposed. The summarization system operates separately on the audio (speech) track, the presentation slides and the usage data to arrive at separate opinions as to the importance of segments of the overall presentation. Then, using a weighted summation as fusion operator, these opinions were combined to form the final summarized presentation.

In the area of audio analysis, a novel method to detect emphasized portions of speech based on spectral entropy was presented. The effectiveness of this method when compared to existing pitch-based methods is demonstrated through a user study. Moreover, a novel fuzzy set framework for slide processing is proposed. The framework operated in a structured manner on a proposed feature hierarchy, calculating low-level term scores before passing these upwards in the hierarchy, computing line (bullet) level scores and finally a slide level score. The impact of the various proposed features, as well as the overall performance of the proposed slide processing algorithm is evaluated within the context of a slide retrieval system. Results demonstrate the efficacy of the proposed algorithm over existing techniques. The operation of the algorithm naturally lends itself to slide transcoding for small-screen devices.

The viability of the proposed summarization algorithm isdemonstrated again through a user study. Existing methods may be seen as a special case of the proposed summarization algorithm. One of the key features of the proposed solution is its extensibility. Indeed, additional subsystems (for example, video analysis) may be added with ease in order to ameliorate automatically generated summaries.

5.2 Future Work

One interesting aspect of this work which may be pursued further is the development of techniques to automatically determine the weighted summation weights, ρ_i. In the current approach, these weights are determined a priori heuristically, and further refined through experimentation. Future work may include machine learning techniques which attempt to learn these weights over time. In this vein, usage history may be tracked along with user feedback as to the usefulness of a particular automatically generated summary. By utilizing this additional information of user feedback, an adaptive approach may be developed which models this feedback through weight adjustments. It is particularly easy to track usage history due to the design of the `Presentation DI`. In fact, MPEG-7 `Usage History Descriptors` provide the perfect means to store this information within a `Presentation DI`.

Another avenue for future research is in the area of user-centric summarization [73]. This work has brushed on this paradigm through the addition of user-specified query keywords to guide summarization. In a full-fledged user-centered summarization system, the user would be an integral part of a dynamic back and forth process to create summaries. This sort of *interactive summarization* has the potential to deliver higher QoS to end users since the specific summaries will be perfectly tailored to their needs. In such a system, users would be able to specify summarization ratios in real-time, while a DL event is in progress. In response to this stimulus, the summarization engine would adaptively change summarization strategies.

Bibliography

[1] A. Pahwa, D. M. Gruenbacher, S. K. Starrett, and M. M. Morcos, "Distance learning for power professionals: Virtual classrooms allow students flexibility in location and time," *IEEE Power and Energy Magazine*, vol. 3, no. 1, pp. 53–58, 2005.

[2] H.-I. Liu and M.-N. Yang, "Qol guaranteed adaptation and personalization in e-learning systems," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 676–687, 2005.

[3] B. Olivier and O. Liber, "Learning content interoperability standards," *Reusing online resources: a sustainable approach to e-learning*, pp. 146–155, 2003.

[4] J. Lukasiak, S. Agostinhoa, S. Bennetta, B. Harpera, L. Lockyera, and B. Powleya, "Learning objects and learning designs: an integrated system for reusable, adaptive and shareable learning content," *ALT-J, Research in Learning Technology*, vol. 13, no. 2, pp. 151–169, 2005.

[5] E. Toms, C. Dufour, J. Lewis, and R. Baecker, "Assessing tools for use with webcasts," in *5th ACM/IEEE-CS Joint Conference on Digital Libraries*, Y.-P. P. Chen, Ed., 2005, pp. 79–88.

[6] L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *Proceedings ACM Multimedia*, 1999, pp. 489–498.

[7] "epresence interactive media," http://epresence.tv/.

[8] "Research channel," http://www.researchchannel.org/.

[9] C. Timmerer and H. Hellwagner, "Mpeg standards enabling universal multimedia access," in *1st International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, December 2005.

[10] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, 2002, pp. 533–542.

[11] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. on Multimedia*, vol. 7, no. 5, pp. 907–919, 2005.

[12] C. M. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. J. Delp, "Automated video program summarization using speech transcripts," *IEEE Trans. on Multimedia*, vol. 8, no. 4, pp. 775–791, 2006.

[13] H. D. Wactlar, "Informedia - search and summarization in the video medium," in *Imagina 2000 Conference*, February 2000.

[14] S. Ju, M. Black, S. Minneman, and D. Kimber, "Summarization of videotaped presentations: Automatic analysis of motion and gesture," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 686–696, September 1998.

[15] A. Divkaran, K. A. Peker, R. Radhakrishnan, Z. Xiong, and R. Cabasson, "Video summarization using mpeg-7 motion activity and audio descriptors," Mitsubishi Electric Research Laboratories, Tech. Rep. TR-2003-34, May 2003.

[16] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, June 1999.

[17] A. Divakaran, R. Radhakrishnan, and K. A. Peker, “Blind summarization: Content-adaptive video summarization using time-series analysis,” in *Proc. of SPIE - Multimedia Content Analysis, Management, Retrieval*, E. Y. Chang, A. Hanjalic, and N. Sebe, Eds., vol. 6073, 2006.

[18] “Mpeg-21 overview,” http://www.chiariglione.org/MPEG/standards/mpeg-21/mpeg-21.htm, October 2002.

[19] S.-F. Chang, T. Sikora, and A. Puri, “Overview of the mpeg-7 standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 688–695, 2001.

[20] “Mpeg-7 overview,” http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm, October 2004.

[21] The World Wide Web Consortium (W3C). (2006, September) Extensible Markup Language (XML) 1.0 (Fourth Edition). [Online]. Available: http://www.w3.org/TR/REC-xml

[22] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., 2002.

[23] The World Wide Web Consortium (W3C). (2004, October) XML Schema Part 0: Primer Second Edition. [Online]. Available: http://www.w3.org/TR/2004/REC-xmlschema-0-20041028

[24] F. Nack and A. T. Lindsay, “Everything you wanted to know about mpeg-7: Part 1,” *IEEE MultiMedia*, vol. 6, no. 3, pp. 65–77, 1999.

[25] F. Pereira, J. R. Smith, and A. Vetro, “Introduction to the special section on mpeg-21,” *IEEE Trans. on Multimedia*, vol. 7, no. 3, pp. 397–399, 2005.

[26] I. S. Burnett, F. Pereira, R. V. de Walle, and R. Koenen, Eds., *The MPEG-21 Book*. John Wiley & Sons, Inc., 2006.

[27] F. Pereira, “Mpeg-21 standard: Defining an open multimedia framework,” in *47th International Symposium ELMAR-2005*, 2005, pp. 5–8.

[28] I. S. Burnett, S. J. Davis, and G. M. Drury, “Mpeg-21 digital item declaration and identification - principles and compression,” *IEEE Trans. on Multimedia*, vol. 7, no. 3, pp. 400–407, 2005.

[29] A. Vetro and C. Timmerer, “Digital item adaptation: Overview of standardization and research activities,” *IEEE Trans. on Multimedia*, vol. 7, no. 3, pp. 418–426, 2005.

[30] F. D. Keukelaere, S. D. Zutter, and R. V. de Walle, “Mpeg-21 digital item processing,” *IEEE Trans. on Multimedia*, vol. 7, no. 3, pp. 427–434, 2005.

[31] “Video semantic summarization systems,” http://www.research.ibm.com/MediaStar.

[32] B. L. Tseng, C.-Y. Lin, and J. R. Smith, “Using mpeg-7 and mpeg-21 for personalizing video,” *IEEE Multimedia*, vol. 11, no. 1, pp. 42–52, 2004.

[33] Y. Wang, Z. Liu, and J.-C. Huang, “Multimedia content analysis - using both audio and visual cues,” *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.

[34] P. M. Fonseca and F. Pereira, “Automatic video summarization based on mpeg-7 descriptions,” *Signal Processing: Image Communication*, vol. 19, pp. 685–699, 2004.

[35] J. Bekaert, P. Hochstenbach, and H. V. de Sompel, “Using mpeg-21 didl to represent complex digital objects in the los alamos national laboratory digital library,” *D-Lib Magazine*, vol. 9, no. 11, 2003.

[36] I. G. L. Consortium. (2003) Ims learning design best practice and implementation guide version 1.0 final specification. [Online]. Available: http://www.imsglobal.org

[37] B. Arons, “Pitch-based emphasis detection for segmenting speech recordings,” in *Proccedings, Int. Conf. on Spoken Languages*, vol. 4, 1994, pp. 1931–1934.

[38] Y. Medan, E. Yair, and D. Chazan, “Super resolution pitch determination of speech signals,” *IEEE Trans. on Signal Processing*, vol. 39, pp. 40–48, 1991.

[39] L. He, E. Sanocki, A. Gupta, and J. Grudin, “Comparing presentation summaries: Slides vs. reading vs. listening,” in *Proc. of Conference on Human Factors in Computing Systems*, 2000, pp. 177–184.

[40] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, “Video manga: Generating semantically meaningful video summaries,” in *ACM Multimedia*, 1999, pp. 383–392.

[41] C.-W. Ngo, F. Wang, and T.-C. Pong, “Structuring lecture videos for distance learning applications,” in *Proc. of IEEE 5th Annual Symposium on Multimedia Software Engineering*, 2003, pp. 215–222.

[42] F. Wang, C.-W. Ngo, and T.-C. Pong, “Synchronization of lecture videos and electronic slides by video text analysis,” in *Proc. of the 11th ACM international conference on Multimedia*, 2003, pp. 315–318.

[43] G.-H. Cha. (2002) Cova: A system for content-based distance learning. [Online]. Available: http://www2002.org/CDROM/alternate/683/

[44] S. Venugopal, K. R. Ramakrishnan, S. H. Srinivas, and N. Balakrishnan, “Audio scene analysis and scene change detection in the mpeg compressed domain,” in *IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp. 191–196.

[45] B. Arons, “Speechskimmer: A system for interactively skimming recorded speech,” *ACM Trans. on Computer-Human Interaction*, vol. 4, no. 1, pp. 3–38, 1997.

[46] S. Roucos and A. M. Wilgus, “High quality time-scale modification for speech,” in *Proceedings, IEEE Int. Conf. on ASSP*, 1985, pp. 493–496.

[47] G. W. Heiman, R. J. Leo, G. Leighbody, and K. Bowler, “Word intelligibility decrements and the comprehension of time-compressed speech,” *Perception and Psychophysics*, vol. 40, pp. 407–411, 1986.

[48] S. Tucker and S. Whittaker, "Time is of the essence: an evaluation of temporal compression algorithms," in *Proceedings, Computer, Human Interaction*, 2006, pp. 329–338.

[49] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82–91, January 2006.

[50] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley & sons, 1991.

[51] S. Tucker and S. Whittaker, "Novel techniques for time-compressing speech: An exploratory study," in *Proceedings, IEEE Int. Conf. on ASSP*, vol. 1, 2005, pp. 477–480.

[52] D. M. Hilbert, M. Cooper, L. Denoue, J. Adcock, and D. Billsus, "Seamless presentation capture, indexing, and management," *Proceedings of SPIE*, vol. 6015, 2005.

[53] A. Vinciarelli and J.-M. Odobez, "Application of information retrieval technologies to presentation slides," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 981–995, 2006.

[54] D. A. Grossman and O. Frieder, *Information retrieval : algorithms and heuristics.* Springer, 2004.

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

[56] W. Hrst and N. Deutschmann, "Searching in recorded lectures," in *Proceedings of the World Conference on E-Learning in Corporate Government, Healthcare and Higher Education (E-Learn)*, 2006, pp. 2859–2866.

[57] H. Yokota, T. Kobayashi, T. Muraki, and S. Naoi, "UPRISE: Unified Presentation Slide Retrieval by Impression Search Engine," *IEICE Transactions on Information and Systems*, vol. E87-D, no. 2, pp. 397–406, 2004.

[58] N. Fuhr and M. Lalmas, "Introduction to the Special Issue on INEX," *Information Retrieval*, vol. 8, no. 4, pp. 515–519, 2005.

[59] S. H. Jin, J. H. Cho, Y. M. Ro, and H. K. Lee, “Archiving of meaningful scenes for personal tv terminals,” in *Proc. of SPIE - Internet Imaging VII*, S. Santini, R. Schettini, and T. Gevers, Eds., vol. 6061, 2006.

[60] M. Almaoui, “Metadata driven multimedia transcoding,” Master’s thesis, University of Toronto, 2005.

[61] M. Almaoui, A. Kushki, and K. N. Plataniotis, “Metadata-driven multimedia transcoding for distance learning,” *To appear, Springer/ACM Multimedia Systems*, 2007.

[62] M. Ajmal, A. Kushki, and K. N. Plataniotis, “Time-compression of speech in informational talks using spectral entropy,” in *To appear, Proceedings of the Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2007)*, 2007.

[63] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky, “Spectral entropy based feature for robust asr,” in *Proceedings, IEEE Int. Conf. on ASSP*, 2004, pp. 193–196.

[64] A. M. Toh, R. Togneri, and S. Nordholm, “Spectral entropy as speech features for speech recognition,” in *Proceedings of PEECS*, 2005, pp. 22–25.

[65] P. Renevey and A. Drygajlo, “Entropy based voice activity detection in very noisy conditions,” in *Proceedings of 7th European Conference on Speech Communication and Technology, EUROSPEECH*, 2001, pp. 1887–1890.

[66] Z. Tuske, P. Mihajlik, Z. Tobler, and T. Fegyo, “Robust voice activity detection based on the entropy of noise-suppressed spectrum,” in *Proceedings of 9th European Conference on Speech Communication and Technology, INTERSPEECH*, 2005, pp. 245–248.

[67] W. Hurst, “Indexing, searching, and skimming of multimedia documents containing recorded lectures and live presentations,” in *MULTIMEDIA ’03: Proceedings of the eleventh ACM international conference on Multimedia*, 2003, pp. 450–451.

[68] L. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[69] A. Kushki, P. Androutsos, K. Plataniotis, and A. Venetsanopoulos, “Retrieval of images from artistic repositories using a decision fusion framework,” *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 277–292, 2004.

[70] J. Dombi, “Membership function as an evaluation,” *Fuzzy Sets and Systems*, vol. 35, no. 1, pp. 1–21, 1990.

[71] H. Dyckhoff and W. Pedrycz, “Generalized means as model of compensative connectives,” *Fuzzy Sets and Systems*, vol. 14, no. 2, pp. 143–154, 1984.

[72] P. S. Aleksic and A. K. Katsaggelos, “Audio-Visual Biometrics,” *Proceedings of the IEEE*, vol. 94, no. 11, pp. 2025–2044, 2006.

[73] C. M. Taskiran and F. Bentley, “Automatic and user-centric approaches to video summary evaluation,” *SPIE: Multimedia Content Access: Algorithms and Systems*, vol. 6056, 2007.

[74] S. Lu, M. R. Lyu, and I. King, “Semantic video summarization using mutual reinforcement principle and shot arrangement patterns,” in *11th Annual International Conference on Multimedia Modeling*, Y.-P. P. Chen, Ed., 2005, pp. 60–67.

[75] M. G. Christel, “Evaluation and user studies with respect to video summarization and browsing,” *SPIE: Multimedia Content Analysis, Management, and Retrieval*, 2006.

[76] J. C. S. Yu, M. S. Kankanhalli, and P. Mulhen, “Semantic video summarization in compressed domain mpeg video,” in *Multimedia and Expo, 2003. ICME ’03*, vol. 3, 2003, pp. 329–332.

[77] S. Bech and N. Zacharov, *Perceptual Audio Evaluation - Theory, Method and Application.* John Wiley and Sons, Ltd, 2006.

[78] R. Likert, “A technique for the measurement of attitudes,” *Archives of Psychology*, vol. 140, 1932.