Automatic Transformation of Video with Meta-Data into Web Contents

Mahendren MUNISAMY†, Kazutoshi SUMIYA†, and Katsumi TANAKA†

† Graduate School of Informatics, Kyoto University  Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan
E-mail: †{mahen, sumiya, tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract Meta-data describes an information resource, it is data about data. A meta-data record is made up of a set of attributes necessary to describe the resource in question. The introduction of MPEG-7 (Multimedia Content Description Interface) has made it possible to apply a uniform standard in describing the content of audiovisual data. Most research about multimedia meta-data deals with presenting a portion of the data after some form of query has been performed. However, there is no research about converting meta-data into webpages. In this paper we propose a model using an example of a newcast to automatically transform video with meta-data into web pages. We also propose an architecture to automatically transform video with meta-data into web pages.

Key words Video database, metadata, web contents

1. Introduction

Rapid progress in Web database technologies and digital television technologies have made it possible to provide a vast volume of information to both internet users and television audiences. The introduction of MPEG-7 (Multimedia Content Description Interface) has made it possible to provide a representation framework for expressing the content of audiovisual documents. In other words, it is now possible to obtain a uniform standard to describe and categorize video data.

Up to now, a lot of research of video databases have concentrated in extracting the semantic contents of a video file. Most of this research is based upon segmenting video files into shots and scenes using scene change detection techniques. A popular way to represent these contents would be by presenting keyframes in either a static storyboard or dynamic slide show display [3].

The process of navigating a video is the ability to use meta-data to direct a search to a specific interval of time. In order to perform navigations through a video it is necessary to structure the data into organized units based on it’s semantic contents. The ability to navigate through a video will allow a user to view the contents from different angles and possibly perform comparisons with other forms of media (like newspapers). Meta-data can also be organized in such a way that a user will be able to obtain a quick review of the entire video without having to entirely view it.

Internet users obtain information from the WWW by navigating web hyperlinks, performing walkthroughs in 3D virtual spaces and by interacting with programs through a browser. Conventional television users (audiences) however do not perform any interactions when watching a program. Studies by Nadamoto [6] and Tanaka [7] have looked into converting the contents of web pages into television programs. However no research has been performed into converting video with meta-data into web pages. Therefore, in this paper we propose an architecture to convert video with meta-
data into web pages using MPEG-7's schema definition to specify the meta-data. The purpose of this research is to identify a standard method to view meta-data as web pages. It is hoped that by doing this, a user will be able to grasp the semantics of the video file in question.

The following section describes the contents of a video file. Section 3.1 explains the model used in conventional hypermedia design (the Dexter model) and the reason it is not a suitable approach for audiovisual data. An very simple example of transforming a news cast into a web page is presented in Section 3.2. In Section 4, we present an architecture for transforming video with meta-data into web pages. Finally, in Section 5, we present the major contributions and future works of this paper.

2. Contents of Video Data

2.1 Visual contents

Petkovic and Jonker [1] have proposed a layered video data model shown in Figure 1. We will use this model to briefly describe the visual contents of video data. The raw video data layer is at the bottom. This layer consists of a sequence of frames as well as video attributes such as compression format, frame rate, number of bits per pixel, color model, duration etc. The next layer is the feature layer that consists of domain-independent features that can be automatically extracted from the raw data. There are two types of video features:

(1) Static features characterizing a still image (frame) such as shapes, textures, color histogram, etc.

(2) Dynamic features characterizing frame sequences such as temporal motion, etc.

The features are assigned to regions. A region is defined as a contiguous set of pixels that is homogeneous in one or more features.

The highest layer is the concept layer. It consists of logical concepts that are subject of interest to the users or applications. The concept layer is divided into the object and event layer. The concept layer consists of entities (logical concepts) which can be assigned to one or more regions. A video object is defined as a collection of video regions, which have been grouped together under some criteria defined by the domain knowledge. Some examples of video objects are a specific player or the ball in a soccer game.

The event layer is the highest layer in the model. It consists of events which describe object interactions in the spatio-temporal manner. An example of an event type in the soccer domain, if the ball object type is inside the goalpost object type for a while and this is followed by a very loud shouting and long whistle, that might indicate that someone has scored a goal, which should be recognized as a goal event.

The contents of video data are not equally important. The choice and importance of the features depend on the purpose and use of the video data. There may also be additional meta information which is usually application specific and cannot be directly obtained from the video data itself. This is usually added during the annotation step of inserting video data into the database, for example, positional information of a ball in video images of a soccer game allows the user to perform searches about a ball's trajectory.

In our case study of transforming a news cast into webpages, we made use of the feature layers static feature's in order to break down a news clip into shots and scenes. We used additional meta-data in order to obtain a title for the news cast and determine the location at which each scene within a news clip was filmed at.

2.2 Audio Contents

We consider the following two forms of high-level audio data when describing sounds in a video:

- Spoken content: This includes all words that are spoken out during the course of a video. We use Automatic Speech Recognition (ASR) technologies in order to transform spoken contents into textual data. Another way to obtain the spoken content of a video file is by directly obtaining it through text captions which are used to help people with hearing disabilities.
- Non-spoken content: This includes all sounds that are not included under spoken content. We use volume distribution technologies to obtain values to express sound intensity.

Most of the spoken contents within a newscast are in form of reported speech (news commentary) and can be extracted and directly applied into a webpage. However, speech in the form of comments and conversations (discussions) have to be modified to show a dialogue is occurring.

2.3 Text Data

The following forms of text data may appear in a video file:

- Scene text: Text which appears on objects in an image of the scene. Examples of scene text includes writings on signs or billboards, text on the sides of trucks and writings on shirts.
- Graphic text: Text which is mechanically added to a video to supplement the visual and spoken information it contains. Examples of graphic text are subtitles and credits that appear at the start and end of a movie.

In this paper, we used only graphic text to obtain information about the location of each scene in a newscast. Any useful information concerning scene text was usually expressed in the commentary (spoken content) of each news clip.
3. Representation of audiovisual contents

3.1 Conventional hypermedia representation

Most hypertext and hypermedia models are designed as static networks of nodes and links based on the Dexter model [8]. In conventional HTML web pages, nodes (components) contain information in the form of text or still images. Links are created by the author and used by readers to browse this information by "jumping" from one anchor to another. Figure 2 shows the Dexter model approach to designing hypermedia. Here, components containing audiovisual data are usually created in the following way:

(1) Authors add new components to the hypermedia network.
(2) These components are filled in by selecting a temporal segment of audiovisual data from an audiovisual file.
(3) Components are indexed by typing some textual information in predefined fields.
(4) Presentation specifications controlling the execution of the application at the runtime layer is defined.

This type of approach is acceptable for authors who’re dealing with only a few seconds of audiovisual data or when all segments to be integrated in the hypermedia components are available as well defined files on a disk that only have to be imported without being watched. However, the cost of dealing with hours of audiovisual data in this way is too high.

3.2 Example of Transforming a Newsclip Into a Webpage

3.2.1 Segmentation Layer

The conventional hypermedia model introduced in the previous section is not suitable for video because it treats each component as a discrete static document. The contents of video data are continuous (streaming data) and change dynamically with a timeline. Using a newscast as an example we introduce a model to transform video with meta-data into webpages.

The first step is to segment the video data into meaningful semantic units. As illustrated in Figure 3, we first segment the newscast into newsclips. Each newsclip is further seg-
mented using scene change detection methods \cite{2} into scenes and shots.

3.2.2 Storage Layer

As shown in Figure 3 the next step is to attach each segmented layer with the appropriate meta-data for storage. The top layer is the newscast itself and this is given a title. The length of each news clip and the time it appears in relation to the entire newscast is also annotated onto this layer. The first frame of every news clip is chosen to be the key frame that will represent video clips at this level. At the scene segmentation level, the location at which each scene was filmed is annotated and the first frame of each scene is extracted to be the key frame of each scene. This will be used later when we form visual summaries for each scene. Finally, at the shot segmentation level, the spoken audio contents for each shot are extracted and stored as described in Section 2.2.

3.2.3 Runtime Layer

Finally, we apply HTML tags to present the data we have attached above. Figure 4 shows an illustration of our method on a selected newscast. From the top layer of the segmented newscast we use the title information and apply it to the \texttt{<TITLE>} tag. We also apply the title information to the \texttt{<H1>} tag, and this serves as a banner to the webpage.

The video clips of each scene are included in a separate frame in the same page. We use the extracted key frames at this layer to produce a storyboard representation of the entire newscast \cite{3}. The key frames (video clips) of each news clip is shown in a vertical manner with the news clips that appear first in the newscast located at the top of the frame. Scrolling up or down allows the user to view key frames that were not located in the initial page. By clicking on the key frames, the user can view the video for a particular news clip. This was done using both the \texttt{<A HREF>} and \texttt{<IMG SRC>} tags. For example, if the key frame for the first scene is "KeyFrame1.gif" and the video clip for that scene is "NewsClip1.mp4", then the following code would allow a user to download or play the first news clip of the newscast:

\begin{verbatim}
<A HREF = "NewsClip1.mp4"> <IMG SRC = "KeyFrame1.gif"> </A>
\end{verbatim}

The time captions at the bottom of each key frame show the user how far they have browsed through the page in terms of time.

Studies by Christel \cite{5} looked into various ways to present visual summaries for news clips. In our example, we use a storyboard consisted of key frames to represent a scene and we also use a simple map digest to act as a geographic reference for the entire scene. Here, we make use of the location information which was annotated at the scene segmentation level.

At the shot segmentation level, we add the spoken contents of each scene below the visual summaries. We use shot changes to indicate the start of a new paragraph. This is accomplished using the \texttt{<P>} tag in HTML.

This is a very simple implementation of transforming video with meta-data into web contents. We used scene change detection in order to segment each news clip because the contents that are broadcasted in a newscast are predetermined before being aired. Therefore, it is possible to grasp the se-
mantics of a newcast by segmenting each news clip in this way. News commentary is synchronised with the events that are being aired. In other words, the audio and visual contents of a newcast are harmonised during broadcast. This makes it possible to present an explanation for the visual summaries we developed using the spoken content in the shot segmentation layer.

As mentioned in Section 2.1 there may also be additional meta-data that may not be directly obtained from the video itself. This meta-data is dependent on the semantics of the video data. The types of meta-data data available may cause a change in the way a video is segmented. For example, in [4], we argued that scene change detection methods are not suitable for videos of soccer games. We also introduced a method to segment and extract key frames from such videos.

In the next Section we describe the architecture we intend to use in order to transform video with meta-data into web pages. This architecture makes use of MPEG-7’s schema definition to specify exactly how a meta-data instance document will look like.

4. System Architecture

The crucial challenge of audiovisual-based hypermedia environments is to provide a representation framework for expressing the content of audiovisual documents. This representation has to be formally structured at the document level to allow easy computer-aided access to relevant parts of the audiovisual file.

MPEG-7 (Multimedia Content Description Interface), overcomes this by providing standardized core technologies allowing description of audiovisual data content in multimedia environments [9]. The objective of MPEG-7 is to standardize the following:

- A set of description schemes and descriptors.
- A language to specify description schemes, i.e., a Description Definition Language (DDL).
- A scheme for coding the description.

The MPEG-7 Description Definition Language (DDL) is based on XML schema [12]. MPEG-7 will therefore mainly consist of a collection of schema components suitable to describe audiovisual material.

When transforming meta-data into web pages, it is important to have a specific schema definition to specify exactly which components will be put together to describe audiovisual contents. Our application’s schema definition will make use of components already defined in the MPEG-7 schema definition to define its own schema. We use an XML schema processing module to parse, validate and map the required meta-data to memory according to generic MPEG-7 schema definition.

Figure 5 shows the architecture we will use in order to transform video with meta-data into web pages. This architecture is similar to the one used by Christel et. al [10] and produces XML results in the web server. These XML documents are processed with different XSL style sheets to produce an HTML or XHTML view. The Extensible Stylesheet Language (XSL) is a language for expressing stylesheets. Given a class of arbitrarily structured XML documents or data files, designers use an XSL stylesheet to express their intentions about how that structured content should be presented [11]. An XSL stylesheet processor accepts a document or data in XML and an XSL stylesheet and produces the presentation of that XML source content that was intended by the designer of that stylesheet. As mentioned in the previous section, the way in which a video should be segmented is dependent on its semantic contents. The contents of the XSL stylesheet will change based on the way each video is segmented.

5. Conclusion and Future Works

In this paper we presented a model for transforming video-meta data into webpages. This model first segments a video file based on its semantic contents. We add meta-data to index each segmented layer with information that will be used at the runtime layer. We also segment certain layers with information useful when forming visual summaries.

In this paper we presented an architecture for transforming video-meta data into webpages. The architecture we presented makes use of components already defined in the MPEG-7 schema definition to define its own schema. The meta-data extracted is stored in a database and converted
into XML documents in a web server. XSL stylesheets and
an XSL processor converts the data into an HTML view.

The next step of this research is to define exactly which
components will be put together to describe the audiovisu-
als contents. In our case study, we presented extracted key
frames using a static storyboard structure. Sumiya et al. [14]
proposed a method to view web pages dynamically using an-
imation sequences. We propose that WebSkimming methods
be modified in order to view key frames for videos as part of
our future works.

Acknowledgments

This research was partly supported by The Special Re-
search Area’s Grant In Aid For Scientific Research (2) For
the Year 2002 under the project “Research for New Search
Service Methods Based on The Web’s Semantic Structure”
(project no: 14019048, Representative: Katsumi Tanaka)
and by The Scientific Research Fund Foundation (A)(2)
for the year 2002 under the project “Multimedia Searches And
Views For Mobile Contents And The Generation Of Broad-
casting Contents” (project no: 14208036, Representative:
Katsumi Tanaka)

文 献

18th IASTED Conference on Applied Informatics, Inns-
bruck, Austria, 2000.
Parsing and Indexing Digital Video, Journal of Visual Com-
munication and Image Representation, Vol.7, No.1 pp. 28
- 43, March 1996.
[3] Anita Komlodi and Gary Marchionini: Key frame preview
techniques for video browsing, Proceedings of the third ACM
conference on Digital Libraries, Pittsburgh, Pennsylvania,
Tanaka: A Spatial User Interface for Browsing Video Key
- 127, pp. 41 - 48, 2002 年 5 月.
raries, Proceedings of the seventh ACM international con-
ference on Multimedia (Part 1) Orlando, Florida, United
States, ACM Press, pp. 303-311[1999].
[6] Katsumi Tanaka, Akiyo Nadamoto, Machiko Kusahara,
Taeke Hattori, Hiroyuki Kondo, and Kazutoshi Sumiya:
Back to the TV: Information Visualization Interfaces Based
on TV-Program Metaphors, Proc. IEEE International
Conference on Multimedia and Expo 2000[ICME2000],
pp.1229-1232[July-Aug 2000].
[7] 瀬本 明代, 服部 多栄子, 近藤 宏行, 沢中 郁夫, 荒原 真知子,
田中 甲己 Web 情報の番組化のためのオーサリング機構, 情報処
理学会研究報告, Vol.2000, No.10 06-DBS-1 20-14 , pp.99-106,
2000 年 1 月.
[8] Gwendal Auffret, Jean Carriere, Olivier Chevet, Thomas
Dechilly, Remi Ronfard and Bruno Bachmann: Audiovisual-
based hypermedia authoring: using structured representa-
tions for efficient access to AV documents, Proceedings of the
ten ACM Conference on Hyper-text and hypermedia : re-
turning to our diverse roots: returning to our diverse roots,
Darmstadt, Germany, ACM Press, pp. 169 - 178[1999].

[9] Jose M. Martinez: Overview of the MPEG 7 Standard
(version 5.0), INTERNATIONAL ORGANISATION
FOR STANDARDISATION, CODING OF MOVING PIC-
TURES AND AUDIO.
[10] Michael G. Christel, Bryan Malher and Andrew Begun:
XSLT for tailored access to a digital video library, Proceed-
ings of the first ACM/IEEE-CS joint conference on Digital
libraries , Roanoke, Virginia, United States, ACM Press,
ble Stylesheet Language (XSL) Version 1.0[available from
http://www.w3.org/TR/xsl/]
[12] World Wide Web Consortium: XML Schema, Parts 0, 1,
2; W3C Working Draft, 7th April 2000 [available from
work (MDF) for content description of audio/video docu-
ments, Proceedings of the fourth ACM conference on Digital
libraries , Berkeley, California, United States, ACM Press,
p. 67 - 75[1999].
[14] Kazutoshi Sumiya, Minoru Takahashi and Katsumi
Tanaka: WebSkimming: An Automatic Navigation
Method along Context-Path for Web Documents, The
Eleventh International World Wide Web Conference
Hondlhu, Hawaii, USA, 7-11 May 2002 [available at