Semantic Web Techniques for Searching and Navigating Video Shots in BBC Rushes

Bradley P. Allen[1)], Valery A. Petrushin[2)], Damian Roqueiro [3)], and Gang Wei [2)]
[1)] Siderean Software, Inc.
[2)] Accenture Technology Labs
[3)] University of Illinois at Chicago

Abstract

In this paper we describe our approaches to creating content-based search and navigation systems that help a TV program or movie maker reuse rushes. We developed two prototypes: the first one uses traditional content-based search with fast browsing and the second one combines AJAX Web technology with Semantic Web technologies (RDF, SKOS) and content based multimedia information annotation and retrieval techniques (MPEG-7).

1. Introduction

In broadcasting and filmmaking industries "rushes" is a term for raw footage, which is used for productions such as TV programs and movies. Not all raw footage goes into a production. A typical "shoot-to-show" ratio for a TV program is in the range from 20 to 40. It means that up to 40 hours of raw footage is converted into one hour of TV program. Using rushes for creating TV programs manually is hard and inefficient. However managers believe that rushes could be valuable if technology could help program makers to extract some "generic" episodes with high potential for re-use. Such generic footage is called "stockshots". There are commercial stockshot libraries which provide shots for particular geographical locations, vehicles, people, events, etc. So, the technical problem is how to mine rushes for golden nuggets of stockshots.

Currently, there are several digital asset management systems (for example, *Arkemedia* by Harris Corp. [1]). These systems allow manually providing some data about the video clip, automatically split the clip into shots and select key frames for each shot. These systems are very helpful for archiving media data, but are not powerful for searching for useful shots. On the other hand, there are many experimental systems for news search, annotation and summarization (e.g., Informedia [2]). These systems are heavily based on textual information that comes from close captions or speech transcripts. In rushes textual information is rather sparse and unreliable. Rushes' soundtracks can be noisy and indecipherable for automatic speech recognition. Moreover, the soundtracks of stockshots are rarely used. They separated from the video stream and substitute by another soundtrack. These peculiarities of rushes and stockshots require developing different data mining techniques that combine sparse textual data with dominant visual data.

In this paper we present two prototypes of systems that helps TV program makers select relevant shots from a repository of shots that were created automatically from rushes. The program maker can use both textual metadata and visual "words" for search.

2. Data

The BBC Rushes data consists of two datasets – development and test sets. Each set has 48 clips which correspond to physical tapes (or rolls). The duration of each clip is from 22 to 35 minutes. The data is arranged in a way that every other clip goes into different dataset. Some descriptions of the tapes (mostly from the camera movement and type of shot view points) were provided separately. The descriptions covered 66 tapes out of 96. No common shot boundaries data was provided. The videos are devoted to French speaking countries – Senegal, Canada and Guadeloupe. Many clips have interviews in French of good quality, but no speech transcripts were provided.

3. Data preprocessing

The BBC Rushes data required a lot of manual or semi-manual preprocessing to extract useful structural and semantic information. The following preprocessing has been done.

1. *Finding shot boundaries*. An automatic shot extraction tool has been used. It produced many very short shots, which were merged manually. A keyframe for each shot was selected automatically.
2. *Deleting junk shots*. Many shots, such as "rainbow" shots of idle camera, or black screen shots in transition between shots, do not carry any useful information. We developed simple recognizers which have been applied to keyframes. They recognized and marked shots as junk shots, which were automatically deleted.
3. *Assigning keywords.* Using a proprietary tool we manually assigned keywords to each shot. The provided textual descriptions were used on this stage.
4. *Creating stories.* Shots were manually united into stories, which are time ordered sequences of logically related shots. Each story is devoted to some topic and location, for example each interview was considered as a separate story.

4. Feature extraction

Besides stories' titles and keywords that were assigned manually, we extracted low level visual features. A number of visual features can be used for describing color, texture, shape and motion characteristics of shots. We used the following MPEG-7 descriptors [6]: for color –dominant color, color structure, and color layout; for texture – homogenous texture, and edge histogram, for shape – region-based shape and contour shape, for motion – motion activity. The above mentioned color, texture and shape features have been extracted for each keyframe using the MPEG-7 XM tools. The motion features have been extracted for each shot using MPEG-1 micro block motion vectors data. We used only intensity of motion activity and average direction of motion as our motion features. Then Self-Organizing Map (SOM) clustering has been applied for each feature. After human evaluation of clustering results the following features were selected: color structure, homogenous texture and region-based shape for representing similarity by color, texture and shape correspondingly. For each map the keyframes that are closest to the SOM nodes' centroids form "visual words". Thus, we obtained three sets of "visual words" that capture the relationships among key frames by color, texture, and shape. For motion we have got motion intensity quantized as a number from 1 (no or low intensity) to 5 (high intensity) and average direction quantized into eight directions (north, north-east, east, south-east, south, south-west, west, north-west).

5. Proposed solutions

We created two prototype systems for searching relevant shots. The first prototype combines traditional keyword and visual feature based similarity search with fast browsing of results. The second prototype uses the Semantic Web techniques and AJAX technology for navigating through the shot repository. The general idea is to use Semantic Web techniques to represent relationships among both textual and visual metadata of various types. Each type of data forms a facet with its own ontology. The relationships among resources (concepts, objects) are described using the Resource Description Framework (RDF) [3] or ontologies and tools, such as the Dublin Core (DC) and the Simple Knowledge Organization System (SKOS) [4] that are based on RDF and RDFS representations.

5.1. Fast Bowser: Metadata representation

The video data in the first prototype is represented as 3-tier hierarchy. On top level all data organized in three folders by location – Senegal, Guadeloupe and Canada. Inside each folder video data is represented as sequence of stories ordered by shooting time. And each story is a time ordered sequence of shots. Each shot has both textual (emotion, subject) and visual data (color, texture, shape, motion) associated with it. The subject keywords are used for textual search and color, texture and shape data is used for visual similarity search. The motion and emotion data serve as restriction for search.

5.2. Fast Bowser: User interface

The video data is represented as tree folders accordingly to locations – Senegal, Guadeloupe and Canada. Clicking on a folder the user can see stories and shots. Each story is represented as a block of shots. Each shot is depicted as a small rectangular. The user can browse the shots by putting the cursor over the rectangular. The corresponding keyframe is displayed in the window in the top right corner. Left click puts the shot into the favorite panel. Right click calls the video player to play the shot. The user can type a keyword for textual search or put an image to search for similar search by example. The user can use both

modes at the same time and assign a weight to each component. The results go into the results' folder and the user can browse them.

5.3. Faceted Navigation: Metadata representation

We have two levels of data presentation: stories and shots. The following textual metadata were used for stories: title, location, and duration. These are encoded as Dublin Core attributes occurring in descriptions of individual story. Shots are related to the story from which they have been extracted using the *dcterms:partOf* attribute.

For shots we have both textual and visual metadata. The textual metadata includes location, emotion, and subject. Subject metadata is represented as a set of SKOS concepts, related to shots using the *dc:subject* tag. Concepts that are synonymous with terms that are values of concepts in the Library of Congress Thesaurus of Graphical Materials 1 [5] are represented using TGM-1 concepts. This allows them to be viewed and selected in a faceted navigation interface using the hierarchy defined by the broader term/narrower term relationships between thesaurus concepts.

For visual metadata the low level facets are color, texture, shape and motion. Three SOM clusterings have been produced using the color, texture, and shape features as it was described above obtaining three sets of "visual words" that capture the relationships among key frames by color, texture, and shape. Each node of the SOM is represented as a SKOS concept, with the value being the image associated with the node. Each shot is related to the concepts associated with the nodes that its keyframes are members of using the *dc:subject* attribute.

5.4. Faceted Navigation: User interface

A user interface for being useful for a program maker should have means for:

- Navigation over the shot database using a combination of facets derived from textual and visual metadata.
- Selection and manipulation of relevant shots found during the user session.
- Saving the results of a session in a form that can be useful for further processing or usage.

The BBC Rushes search user interface is built as an HTML page using AJAX [7], communicating via HTTP calls with an instance of the Seamark Navigator system [8]. Seamark Navigator is used to store the textual and visual metadata describing clips and extracted shots, and provides SOAP services that support faceted navigation over the metadata, with facets defined using textual and visual attributes of the clip and shot metadata. Seamark provides both free text querying and facet value selection for user search of the shot metadata repository.

The basic unit for navigation is a shot. Navigation is provided in two manners: through the ability to enter free text search queries against the textual facets, and through the ability to left click on textual and visual facet values. In either manner, the system updates the user interface with a set of search results and a new set on faceted navigation options. The initial state of the interface shows an overview in terms of facet values of the most frequently occurring facet values per facet across the entire collections of shots.

The results of a search are represented as a sequence of shots sorted in accordance with criteria combination of free text queries and facet value selections. Information related to every shot includes data related to the story (title, location) and data related to the shot (subject, emotion, duration).

Left clicking on any key frame or on a title hyperlink launches a video player for playing back the shot. Shots in the results that are of interest to the user can be copied to a storyboard using drag-and-drop.

The storyboard contains the total number of shots, total duration of shots, and a list of selected shots with key frames and attributes (subject, duration, etc). Left clicking on the keyframe image runs a video player for the shot. The user can reorder and delete shots on the storyboard using drag-and-drop.

Pressing on "Play All" button invokes an HTTP call to a REST Web service that takes the URLs of selected shots and generates a SMIL [9] document composing the shots into a single virtual clip that allows the user to watch the shots played sequentially in the order as they are listed., and then launches a player that plays the virtual clip.

6. Additional processing

We also did some additional data processing. However we did not obtain the final results for their performance yet. Taking into account that the collection of videos contains many interviews we decided to

use audio data to separate interviews into “question – answer” chunks. We noticed that all interviews are conducted by the same female speaker and decided to build a model for the speaker and a speaker verification classifier. The interview separation algorithm uses the speaker verification algorithm for finding the interviewer speech and marks speech between them as “question – answer” chunks. We used Gaussian Mixture models (GMM) and mel-frequency cepstrum coefficients (MFCC) as features.
We also did research on automatic image classification and tagging using two approaches. In the first approach we used SOM nodes that were created using the development set data as GMM classifiers to automatically classify the shots from the test set. In the second approach we created specialized classifiers for semantic concepts such as “sky”, “water”, “sand”, “greenery”, “person”, “building” using neural network and SVM classifiers and tried to combine them to derive such as “city”, “outdoors”, “beach” and “interview”. We shall present results for this research during the presentation and in the extended version of our paper.

References

[1] Arkemedia digital asset management system: http://www.broadcast.harris.com
[2] Informedia: http://www.informedia.cs.cmu.edu/
[3] RDF: http://www.w3.org/RDF/
[4] SKOS: http://www.w3.org/2004/02/skos/
[5] TGM: http://www.loc.gov/rr/print/tgm1/toc.html
[6] B.S. Manjunath, Ph. Salembier, Th. Sikora (Eds.) Introduction to MPEG-7. Multimedia Content Description Interface. John Wiley & Sons, Ltd., 2002, 371 p.
[7] AJAX: http://en.wikipedia.org/wiki/AJAX
[8] Seamark: http://www.siderean.com/Seamark_datasheet.pdf
[9] SMIL: http://www.w3.org/TR/REC-smil/