HTML Paper

VISUAL INFORMATION RETRIEVAL FROM ANNOTATED LARGE AUDIOVISUAL ASSETS BASED ON USER PROFILING AND COLLABORATIVE RECOMMENDATIONS

K. Ntalianis, S. Ioannou, K. Karpouzis, G. Moschovitis and S. Kollias

National Technical University of Athens

Electrical and Computer Engineering Department, Athens, 157 73, Greece, E-mail: kkarpou@softlab.ece.ntua.gr

ABSTRACT

Current multimedia databases contain a wealth of information in the form of audiovisual and text data. Even though efficient search algorithms have been developed for either media, there still exists the need for abstract data presentation and summarization. Moreover, retrieval systems should be capable of providing the user with additional information related to the specific subject of the query, as well as suggest other, possibly interesting topics. In this paper, we present a number of solutions to these issues, giving an integrated architecture as an example, along with notions that can be smoothly integrated in MPEG-7 compatible multimedia database systems. Initially, video sequences are segmented into shots and they are classified in a number of predetermined categories, which are used as a basis for user profiles, enhanced by relevance feedback. Moreover, this clustering scheme assists the notion of "lateral" links that enable the user retrieve data of similar nature or content to those already returned. In addition to this, the system is able to "predict" information that is possibly relevant to specific users and present it along with the returned results.

1. INTRODUCTION

Raw film footage has been the primary source of material for news broadcasts, documentaries and film making since the advent of the portable camera. However, content producers willing to use such material in their own broadcasts were hampered by restrictions, imposed by the media itself (older filmstrips require specific hardware for playback, which is usually incompatible with computerized editing systems), as well as the lack of any indexing or summarization of the data contained in the strips.

Current and evolving standards, such as MPEG-7 [1], provide solutions to these problems, by means of notions that enable the efficient, abstract retrieval and exploitation of specific material. This is very important in time-critical operations, such as televised news broadcasts or newspaper publishing, or applications that require advanced quality, such as entertainment. Users of this kind will benefit from the advanced summarization schemes and will be able to retrieve specific and atomic material as a result of simple and descriptive queries. In this context, queries need not be restricted to textual values but also incorporate “by-example” schemes, e.g. queries by sketch or queries for segments that contain the face of a specific person. Reversely, the results may be presented in a fashion that provides the user with an abstract understanding of the content through the use of automatic feature extraction techniques, such as shot detection and characteristic frame extraction.

Furthermore, integrated systems should be able to support diverse groups of users; for example, historians or print journalists usually prefer to concentrate on the historical and cultural background of the story. To provide users with such capabilities, semantic metadata can be attached to the video data by experts. These metadata can also be used to retrieve supplementary information, related to that actually retrieved by the query.

Several techniques and systems have been proposed in literature coping with the problem of adjusting information retrieval to particular users' needs. These approaches can be divided into two main categories: (a) content-based recommendation and (b) collaborative recommendation. A content-based recommendation system, which has its roots in the information retrieval research community, makes its recommendations by constructing a profile for each user and using this profile to judge whether discovered information will be of interest to the user or not. In the case of collaborative recommendation, discovered information is filtered according to the preferences of users with habits similar to those of the served user. Thus, items preferred by users of similar profiles are considered to be of possible interest and are presented as top suggestions to the particular user.

Several examples of personalizing information systems exist. Examples of content-based recommendation systems include the “Syskill & Webert” [3] agent, which suggests links that a user would be interested in or constructs LYCOS-compatible queries and the “InfoFinder” which scores pages based on the extraction of phrases of significant importance. On the other hand, collaborative recommendation systems include “GroupLens”, which collaboratively filters netnews and the ”WebHound” agent that locates users with similar ratings to specific pages and suggests unread pages that are preferred by them. In general, one disadvantage of the collaborative filtering approach is that when new information becomes available, other users must first rate this information before it may be recommended to others. On the contrary, the user profile approach can help to determine whether a user is likely to be interested in specific new information without relying on the opinions of other users.

2. SYSTEM OVERVIEW

Access to mere text data is far more straightforward than to multimedia data, such as video, mainly because semantic features are well defined and the relevant representation is universal. On the other hand, image and video information is far richer than text in terms of ideas and notions beyond the actual content of a documentary. For that reason, we have employed a combination of either media in our archive, which introduces a number of arguments, such as the need for abstract presentation of data and semantic mapping between visual and textual information. The MPEG-7 can help standardize the representation of a hierarchy of the supplied data and enable querying in abstract or lower levels.

On the development side, we have employed the popular three-tier architecture. In a three-tier context, the client tier is responsible for the formation and transmission of users’ input data, as well as for presentation (rendering) of the retrieved data. A typical web browser is used, since the underlying principle is restricted to calls

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

to pure JavaScript code. On the other end of the data flow, the database module handles pure SQL requests and returns database objects in the form of data types that are determined during the design phase of the project. This means that the middle tier acts as a "negotiator" between the two ends of the data flow and forms standard SQL queries from the textual or other user inputs and, reversely, create the necessary code for HTML documents that present the retrieved data in the browser window. In addition to that, this module can include any system policy issues that need to be enforced. This effectively separates the business logic from the data itself, thus making it easier to change one or the other without necessarily affecting the whole system. Other advantages of this architecture include *data security*, as the client is restrained from querying critical data, such as the database schema or security policy options, *advanced resource management* and *easy maintenance and redesign*: since all business logic is separated from the data and presentation layer, any solitary changes are not cascaded to other modules.

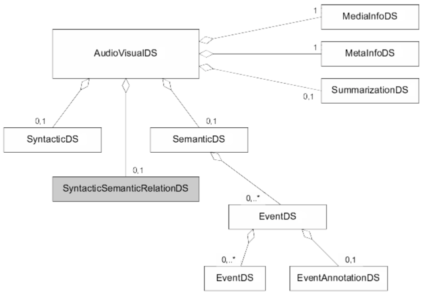

Figure 1: Representation of the AudioVisual DS hierarchy

3. MPEG-7 AND ASSET DATABASES

3.1 Organization and Description of the Material

In order to exploit the classification of the material in different categories, we employed the popular scene-shot-characteristic frame hierarchical scheme. The initial material was automatically segmented to more than sixty scenes, which in total comprise more than ten thousand shots. Each scene is described using technical features, such as the total number of frames or sound quality and annotated by an expert historian, thus providing clues on the historical and cultural environment of the subject, in addition to the textual description of the visual data. Besides that, the expert also comments on characteristic frames extracted from each shot. This assists the summarized presentation of the shot, while giving the expert the opportunity to add extended commentary to the material.

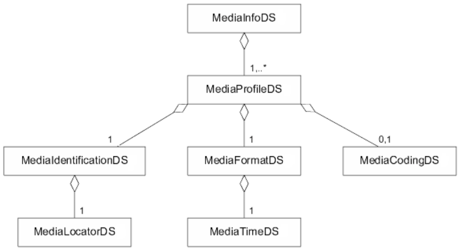

Figure 2: Technical data contained in the MediaInfo DS

A nice advantage of this description scheme is the straightforward introduction of concepts included in MPEG-7, such as Multimedia Description Schemes (MMDS) [1]. The target of these concepts is to standardize a set of tools dealing with description and management issues, as well as navigation and retrieval in multimedia entities. Since the latest generation of web browsers offer inherent support of XML, efficient separation of content, business logic and presentation of results are possible, without having to rearrange the employed schemes.

Even though the Descriptors (Ds) and Description Schemes (DSs) proposed by the MPEG Group are more than enough for the most systems, they can be extended to suit specific needs or match existing data and applications. The hierarchical structure of our system is shown in Figures 1, 2 and 3 in UML format; this format is used here instead of the usual text-based Data Definition Language (DDL) so as to illustrate the employed hierarchy and DSs in a more efficient way. In these figures, grayed objects and dotted-line connections represent notions not implemented in our system.

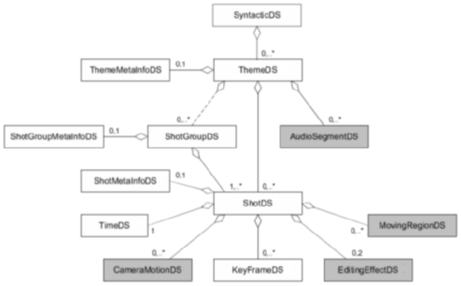

Figure 3: Structure of the video material in the Syntactic DS

In general, the AudioVisual DS is designed as a metaphor of the typical method of organizing the content in a written document, i.e. with the use of a *Table of Contents* and an *Index*. In such a context, the Table of Contents aims to define the structure of the archive, as it does in a book or document, using linear syntax regardless of the internal organization of the material and the linking which occurs with respect to its semantic content. Inversely, the goal of the Index is not to describe the structure of the content but to provide useful references to the actual material. These references are usually not complete, in the sense that the Table of Content essentially provides access to *every* piece of information in the archive, but are selected based on their semantic value to humans and may be recurring for the same item. In our implementation, syntactic information is contained in the Syntactic DS, shown in Figure 3, while the semantic content is described with the aid of the Semantic DS and Event DS hierarchies. The Syntactic DS contains information about the organization of the content in the physical level, as well as signal-based properties, such as camera movement or definition of shot groups. The inclusion of recurring Theme and Shot DSs allows the creation of hierarchical Tables of Content, where the actual material and accompanying meta-information are presented in a way that preserves the required level of abstraction. In essence, the temporal structure and overall visual properties of a high-level object, i.e. a *Scene*, are represented as a single node and may be decomposed to shorter lower-level *shots* or *shot groups*.

While this representation is critical for easier access to both high- and low-level video information, a video archive also includes references to semantic notions, which help humans interpret the

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

actual context and background information of the presented video shots. The Semantic DS – Event DS hierarchy provides references to actual visual data, through their respective syntactic description; this results to a mapping of semantic notions to time intervals in the video shots. The descriptors related to the content of each interval may be predefined in the sense of a dictionary or include text annotations. The latter case is useful when users need to read unformatted descriptions so as to handily comprehend the actual events, while dictionary entries are required on summarization and classification applications. Such information may also be included in an instance of a Summarization DS or a MetaInfo DS, but these are usually reserved for high-level audiovisual objects, even the complete archive itself.

3.2 User DS

In the same fashion as with the AudioVisual DS, MPEG-7 facilitates the description of a user's preferences, usage history and statistical data through a User DS. This information may be used to filter the actual data that is contained in the archive, with respect to a specific user's individual needs or constraints and recommend other related or updated material.

The actual data contained in an instance of the User DS range from static demographic information, such as name, address or educational background, to a dynamic record of choices and preferences of a user. This semantic information is used to determine the default view for the results, for example presentation of a keyframe or just the textual description. In practice, the User DS includes support for filtering and search preferences, as well as the browsing history for the current session of a specific user. The former may be considered as the *static* knowledge of the system, incorporating information that is used by the filtering subsystem, while the history entries are *dynamic*. The off-line duty of the system to integrate this dynamic information with the predefined user preferences is easily accomplished through the formation of a user profile and its update with respect to actual choices.

4. ASSET RETRIEVAL

4.1 Summarization of the Textual Descriptions

The first filtering step takes into consideration "noise words" such as 'a', 'the', 'in' etc. and "noise stems", which should not be included in the summarization process. In this procedure, input text words are compared against the exact noise words, and again, after stemming, against the noise stems. After considering all the previous cases, we reduce the redundancy of the remaining words, by detecting the specific stem of each word. An algorithm based on the Porter stemmer is used as a process for removing common morphological and inflectional endings from words in English.

After performing the aforementioned analysis, the keyword extraction phase is activated. In this step, information-based approaches are adopted to determine which words can be used as features. We employ the vector space information retrieval paradigm, where documents are represented as vectors [4]. To determine word weights, a TF-IDF (Term-Frequency / Inverse Document Frequency) scheme is adopted to calculate how important a word is, based on how frequently it appears. Here, the weight for a word ***w*** belonging to a document **d** is given by:

$$w_{ds}= f_{ds} \cdot \log(N_D/n_s)$$

where w_{ds} is the weight of the word, f_{ds} is the frequency of the word ***w*** in the document, N_D is the total number of documents and n_s is the number of documents containing the word ***w***.

In our approach we include the twenty highest-weighted words of a document to construct a document's vector. This is done in an attempt to reduce memory charge, decrease communications load and avoid over-fitting, as too many words lead to a performance decrease during the classification process, while our experiments for a small vocabulary have shown that recommendation results were poor. Table 1 shows some of the most informative words from a collection of documents concerning historical events.

War	island	army	leader	revolution
Europe	Running	june	people	cause
Bridge	Politician	gun	prepare	bleeding
Cold	Notice	iron	first	condition
Victory	Peace	plane	fighting	exhaustive

Table 1: Keywords used as features

Such a table is constructed for each document; the elements of a document's table are assigned weights with respect to the categories that the document belongs in. After a certain number of keywords (those with the highest weights concerning a number of documents) have been picked out, the information is supplied to a learning subsystem. Then, with each page access, the weights of their profile are updated according to new pages' analysis. A simple way to update profiles is by addition of new document information to the user profile, which is referred in the information retrieval community as relevance feedback.

4.2 User Profiling

In order to reduce the complexity of a query, it is desirable to rank the results according to the actual relevance to the query statement. For that reason, we employ a user profiling mechanism to rank the returned material, optimize the precision score [2] and recommend relevant additional shots for further study.

For each shot, the system produces a feature vector that consists of sixteen content category weights (see Table 2), followed by five user category weights. The user category weights correspond to five typical users of the system, namely *Historian*, *Journalist*, *Cinephile*, *Director* and *Casual User*. According to this scheme, a specific shot is predicted to interest a given user if the respective vectors are relatively close in this vector space.

Sports	Travel	Industry	Transportation
Celebrations	Religion	Army	Government
Services	Artistic	Politics	Education
Tourism	Celebrities	Hist. Events	Head of State

Table 2: The categories that the material is classified in

To measure the proximity of feature vectors we employ the standard dot product metric:

$$\mathrm{r}\,(\mathbf{c}, \mathbf{u}) = \mathbf{c} \bullet \mathbf{u}$$

where **u** is the user profile vector, **c** is the shot vector and r is the relevance function. This function is used to sort the returned shots, as the user is probably more interested in them. Similar to the relevance function, dynamic profile updating also corresponds to a vector operation. In this case, a simple relevance feedback algorithm is used for computing the vector increment Δ**u**:

$$\Delta\mathbf{u} = \mathrm{s} \bullet \lambda \bullet \mathbf{c}$$

where s = 1 if the user selects **c** and s = -1 if the user ignores **c** and λ is a positive parameter, typically lower than 0.001, ensuring smoothness of the updating procedure.

4.3 Video Shot Recommendation

Our system supports two types of dynamic recommendation services: content-based, where video shots similar to the ones retrieved are suggested and collaborative, where the system recommends shots viewed by users similar to the current. Here, standard clustering algorithms are used to segment the content and user spaces in 'similar' groups. Likewise, the user profile space is segmented in clusters containing users with similar profile vectors. We assume those users share common interests, so it makes sense to recommend shots viewed by "neighbors" with respect to the user profile cluster. These suggestions are called "lateral", because they might diverge from the users' path towards information retrieval while still being of interest to them.

4.4 A Hands-on Scenario

The ranking mechanism is demonstrated in an example: the user is interested on videos of the "King George of Greece". The system queries the database and returns two video shots. In the following, the vector representations of the user profile and the matched shots are presented, along with the relevance function evaluation and the final sorting.

Figure 4. Video shot #1

Figure 5. Video shot #2

0.1	0.0	0.0
0.4	1.0	0.0
0.3	0.0	0.0
0.6	0.4	0.0
0.8	0.8	1.0
0.9	0.0	0.0
0.3	0.2	0.8
0.1	0.4	0.8
0.1	0.0	0.7
0.2	0.0	0.0
1.0	0.8	0.8
0.4	0.0	0.0
0.9	0.1	0.0
0.8	0.9	0.0
0.9	0.1	1.0
0.3	0.9	0.1
0.8	0.8	0.9
0.0	0.2	0.5
0.6	0.4	0.2
0.1	0.2	0.4
0.5	0.7	0.7

Table 3. User profile vector (**u**) **Table 4.** Vector **c1** **Table 5.** Vector **c2**

Shot #1 shows the return of King Constantine, son of King George, after his trip to the States in 1967, while shot #2 is taken from a parade in downtown Athens in 1938. Although King George is actually missing from video shot #2, his absence is noted by the annotator. The relevance functions give r(**c1**) = norm(**c1**)•norm(**u**) = 0.732 and r(**c2**) = norm(**c2**)•norm(**u**) = 0.6319, where norm(**v**) denotes the normalized version of vector **v**. As a result, the system gives priority to **c1** over **c2**. Moreover, the recommendation system suggests the video shot #3, based on its close proximity to the aforementioned items. This shot, from 1921, shows King Constantine, father of King George, during a highly celebrated visit to an Orthodox church in Asia Minor.

0.0	0.1
0.9	0.2
0.0	0.2
0.3	0.3
0.7	0.9
1.0	0.6
0.3	0.1
0.1	0.1
0.0	0.1
0.0	0.2
0.2	0.9
0.0	0.2
0.1	0.9
0.9	0.9
0.5	0.9
0.9	0.1
0.9	0.8
0.4	0.0
0.2	0.3
0.9	0.1
0.1	0.2

Table 6: The 21-D vector for suggested shot #3 **Table 7:** Profile cluster vector

The user in question is classified to a profile cluster with the mean vector shown in Table 7 and the collaborative subsystem also suggests the shot in Figure 7.

Figure 6. Video shot #3

Figure 7. "Lateral" video shot

This video shot is taken from a military celebration in 1938. The King does appear in this video, but the key figure is the dictator of Greece and head of the Greek Army at the time; this explains why this video shot was not retrieved from the initial query, but it is suggested as highly relevant from the system.

5. ACKNOWLEDGEMENT

This work was partially funded by the Greek Ministry of Press and Mass Media and the program "Digitization, archiving and access to audiovisual data". The Ministry holds copyright for all presented shots and stills.

6. REFERENCES

[1] Koenen, R. and Pereira, F., MPEG-7: A standardized description of audiovisual content, Signal Processing: Image Communication, Vol. 16, Nos. 1-2, Sept. 2000

[2] Montebello M., "Optimizing Recall/Precision scores in IR over the WWW," *Procs. of the 21st ACM SIGIR Conference on Research and development in information retrieval*, ACM Press, NY, USA, 1998

[3] Pazzani, M. et al, "Syskill & Webert: Identifying interesting web sites", *Proc. of the Natl Conference on AI*, AAAI Press, 1996.

[4] Salton, G. and Buckley, C., "Term weighting approaches in automatic text retrieval", TR 87-881, Cornell University, 1987.