Multimedia Tools and Applications, 22, 215–233, 2004

Web Access to Large Audiovisual Assets Based on User Preferences

K. KARPOUZIS kkarpou@image.ece.ntua.gr
G. MOSCHOVITIS
K. NTALIANIS
S. IOANNOU
S. KOLLIAS
Image, Video and Multimedia Laboratory, Electrical and Computer Engineering Department, National Technical University of Athens, Heroon Polytechniou 9 15780, Zografou, Athens, Greece

Abstract. Current multimedia databases contain a wealth of information in the form of audiovisual as well as text data. Even though efficient search algorithms have been developed for either media, there still exists the need for abstract presentation and summarization of the results of database users' queries. Moreover, multimedia 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 topics which could be identified to attract the interest of users with a similar profile. In this paper, we present solutions to these issues, giving as an example an integrated architecture we have developed, along with notions that support efficient and secure Internet access to audiovisual/video databases. Segmentation of each video in shots is followed by shot classification in a number of predetermined categories. Generation of users' profiles according to the categories, enhanced by relevance feedback, permits an efficient presentation of retrieved video shots or characteristic frames in terms of the user interest in them. Moreover, this clustering scheme assists the notion of 'lateral' links that enable the user to continue retrieval with data of similar nature or content to those already returned. Furthermore, user groups are formed and modeled by registering actual preferences and practices. This enables the system to 'predict' information that is possibly relevant to the user's interest and present it along with the returned results. The concepts utilized in this system can be smoothly integrated in MPEG-7 compatible multimedia database systems.

Keywords: multimedia databases, web access, video summarization, dynamic search, user profiling, query expansion

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, for the greater part of the previous century, organized archives of such media had been rare thus obstructing the utilization of the material in everyday applications. In fact, producers willing to use such material in their own broadcasts were hampered by restrictions imposed by the media itself (older film strips require specific hardware for playback; such hardware is usually incompatible with computerized editing systems), as well as the lack of any indexing or summarization of the visual data that is contained in the strips.

The advent of flexible digitizing hardware, together with the augmented ability of modern computer systems to handle large audiovisual assets and with emerging multimedia database systems introduce effective solutions to these problems. In addition, current and evolving standards, such as MPEG-4 and MPEG-7 [11], support notions that aid the efficient retrieval and exploitation of specific material, without the need to manually browse through all available data. This is very important in time-critical operations, such as televised news broadcasts or newspaper publishing, or applications that require high quality, such as entertainment. Users of this kind of information will benefit from the advanced summarization schemes offered by the above standards and will be able to retrieve specific material as a result of simple and descriptive queries. In this context, queries need not be restricted to textual values but may also incorporate 'by-example' schemes, e.g., queries by sketch or queries for segments that contain the face of a specific person. 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, based on shot detection and characteristic frame extraction.

Furthermore, integrated systems should be able to support diverse groups of users; for example, historians or print journalists are usually less interested in the visual aspect of a recorded documentary and prefer to concentrate on the historical and cultural background of the story. To provide users with such capabilities, video data is annotated by experts who define the metadata for better content comprehension. Textual metadata can be also used to generate supplementary information, related to that actually retrieved by the query.

In addition to the above, the introduction of the Internet as a multimedia content transfer channel has broadened the target audience of such material, while introducing a number of additional issues, such as establishing advanced security systems and protecting existing intellectual property. Both of these matters are not necessarily associated with the content itself; however, recent work in digital video watermarking shows that in the near future one will be able to prove ownership of an image or a video clip without the need for specialized equipment.

Several techniques and systems have been proposed in the literature for 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-, or user-, 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. Profiles are mostly built up by providing material to the user, such as web pages, questionnaires and stored material, according to the application; the user rates the provided information and, thus enables the system agent to create a new profile. In the case of collaborative recommendation, discovered information is filtered by considering users with habits similar to those of the user to be served. As a result, items preferred by users with similar profiles are predicted as cases that possibly interest the specific user and are presented as suggestions to the particular user.

Several examples of personalizing information systems exist. Examples of content-based recommendation systems include the 'Syskill & Webert' [15] software agent which suggests links that a user would be interested in or constructs LYCOS-compatible queries; the 'InfoFinder' which scores pages based on the extraction of phrases of significant importance; the 'WebWatcher,' an 'information routing system' designed to suggest links to users for getting from a starting location to a goal one; the 'SIFT' system [28] which adjusts the weights of a profile by incorporating a relevance feedback approach; and the 'Amalthaea' [14], an artificial 'ecosystem' of evolving agents that cooperate and compete in a limited resources environment. In this context, agents useful to the user get positive credit, while the 'bad performers' get negative credit. Correspondingly, collaborative recommendation systems include 'GroupLens' [17], which is designed to collaboratively filter netnews; the 'Web-Hound' agent that locates users with similar ratings to specific pages and suggests unread pages that are preferred by them; the 'Ringo' [25] system, which is devoted to filter social information; and the 'Bellcore' [9], that is a video-recommender, which efficiently combines users' choices. A disadvantage of the collaborative filtering approach is that when new information becomes available, other users must first read and rate this information before it may be recommended to a specific user. 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.

Futhermore, hybrid systems have been also proposed, which recommend pages scoring highly against someone's profile (content-based recommendation) or pages rated highly by users with similar profiles (collaborative recommendation). An effective example of such a system is Fab [3]. Fab maintains two sets of profiles, that is *collection agent* profiles and *selection agent* profiles. A collection agent profile can be considered as an example of a stereotype [18]: for example, the profile of an agent that specializes in sports contains a majority of terms (from Web pages or textual descriptions) that are sports-related, as well as their corresponding weights. The functionality of a collection agent is to filter documents according to the tastes (ratings feedback) of a set of users who are interested in a specific topic; on the other hand, a selection agent acts as a filter for a single user. Over time, it is expected that these agents will learn the preferences of individual users as well as the collective population of users.

Another interesting hybrid recommendation system was presented in [4], where recommendation was reformulated as a problem in inductive learning or classification. This work focused on detecting items that would be liked or disliked by a given user, rather than predicting the exact rating of a particular item. Decisions were made using a function of both features of the user and features of the items (in the described case, movies). In the movie domain, the authors had to consider that many sources of information describing movies were available (e.g., internet resources such as the Internet Movie Database) and use these resources to extract features for their movie set (such as a movie's cast, director, producers and genre). Furthermore, a set of hybrid features that combined properties of users with properties of the movies was developed. An example of a hybrid feature could be the set of "*Comedy Movies that User X Liked*". These features were based on the user's movie ratings and on the properties associated with movies that were rated highly by the user.

Both of the aforementioned information retrieval systems contain interesting ideas on how to combine user profiling with data profiling, thus embodying content-based and collaborative recommendations into a hybrid system. However it can be argued that the most crucial factor in information retrieval systems is the quality of the multimedia material description. Lexicographic analysis of the text contained within a plain document may work well in some cases, but in the case of multimedia files such as video and images, textual

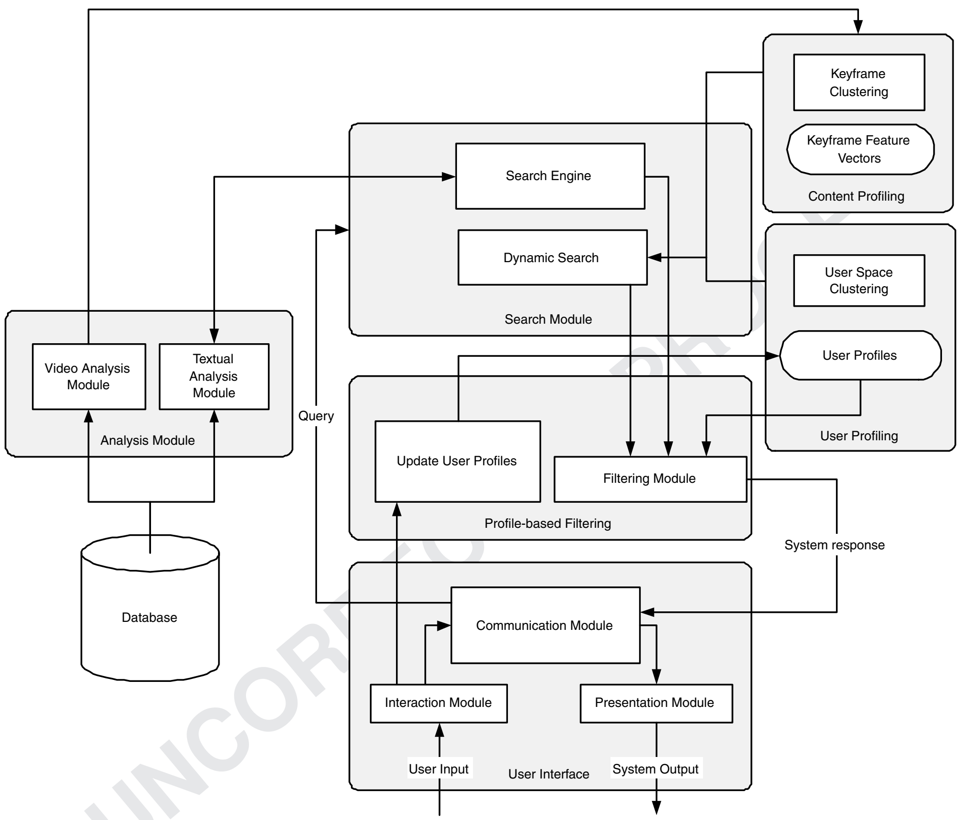

Figure 1. General system architecture.

information can only provide a poor description. This is a common drawback of most text-based recommendation systems proposed in the literature. For this reason, and in extension to the previously described IR systems, we propose an analysis and feature extraction module (left-hand side of figure 1) that is composed of two parts: (a) a lexicographic analysis sub-module and (b) a visual analysis sub-module. The first sub-module uses textual analysis techniques to extract the top referenced terms from a textual description that characterizes a multimedia file. The second sub-module uses image and video analysis methods to perform video summarization and make search easier through keyframe extraction, characterization and clustering.

2. Web-based access

We provide web based access to our system. By choosing mature open technologies like HTTP and HTML, we leverage the installed base of common web browsers to provide a

P1: ***
Multimedia Tools and Applications KL2224-02/5265452 January 14, 2004 2:28

familiar and intuitive interface to our system, that is available to all major operating systems and platforms.

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 and offers the opportunity to convey ideas and notions beyond the actual content of a documentary. As a result, we have employed a combination of either media in our archive, so as to take advantage of their respective advantages. This combination introduces a number of arguments, such as the need for abstract presentation of data and semantic mapping between visual and textual information. The introduction of MPEG-7, or the recently announced MPEG-21 standard, can help in standardizing the representation of a hierarchy of the supplied data and enable querying in abstract or lower levels.

2.1. Three-tier architecture

Instead of adopting a straightforward client-server approach, we have employed the increasingly popular three-tier architecture so as to integrate the services of each module. In fact, a two-tier system is not always feasible, especially when the database server and the web server are setup in two different computers, both behind a firewall, as part of the system requirements specifications.

In the 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 to standard JavaScript code. On the other end of the data flow, the database module handles SQL requests and returns database objects in the form of data types which were determined during the design phase of the project. In addition the three-tier architecture provides us with enhanced data security, advanced resource management (load balancing, user priorities depending on bandwith) and easy maintenance and redesign.

2.2. Secure access

User authentication follows a three-way handshaking scheme, similar to the one used in CHAP [24], in our case, this type of authentication is used only during the initial authentication phase. This procedure consists of the following steps:

- The initial login screen, containing the login and password form fields, along with a random generated number: the challenge key.
- A JavaScript implementation of the MD5 algorithm calculates the digest [19] of the user name, password and challenge key which is sent back to the server, along with the user name in plain text.
- The middle-tier computes the same digest by retrieving the additional data (random key, password) from the database. If the strings match, the user is authenticated.

3. MPEG-7 and asset databases

3.1. Organization and material description

The source material came from 6 film reels provided by the Movie Archive of the Greek Ministry of Press and Mass Media. These reels were digitized into Digital Betacam tapes, and then encoded to MPEG-1 and MPEG-2 files in our laboratory. In order to exploit the classification of the material in different categories and ensure easy upgrading to a fully MPEG-7 compatible scheme, we employed a program/shot/characteristic-frame hierarchical scheme.

3.1.1. Video analysis module—summarization. Video analysis consists of two stages:

– video shot segmentation
– characteristic (key) frame extraction from the video shots

Video analysis was employed for automatic summarization, on the one hand to facilitate the video annotators, and on the other hand to make search more efficient. At first video material was automatically segmented into shots. The algorithms used for shot segmentation are described in [1, 5]. The basic idea is that the DC coefficients of the blocks can form a sufficient representation of each frame. This spatially reduced image (DC image) is sufficient for shot detection. By examing the peak sharpness of the absolute difference of subsequent DC images, shot changes are automatically detected.

Following shot detection, a set of keyframes was extracted from each shot, providing a brief representation of the shot's content. To achieve this, each frame of the shot was segmented into homogenous regions and a feature set was created for each frame, through multidimensional fuzzy classification of the segments' properties. The feature set was in the form of a multidimensional histogram [6]. The dimension of the feature sets was n^6 corresponding to (R, G, B, x, y, size), with RGB being the 3 color components, x and y each segment's position, and size denoting each segment's size in pixels; n was the number of the histogram bins. Keyframes were optimally extracted by minimising a cross-correlation criterion in the feature set space using a genetic algorithm [2].

3.1.2. Data structures. This process resulted in sixty programs (sets of semantically related shots) which in total comprise more than ten thousand shots. Each shot's description contains technical features, such as the total number of frames and the sound quality. Each shot also contains annotation provided by an expert historian, which adds clues on the historical and cultural environment of each subject, in addition to the textual description of the visual data. Besides that, the expert also comments on the keyframes extracted from each shot (their number varying from one to seven per shot) as was described above [2]. This assists the summarized presentation of the shot, whilst giving the expert the opportunity to add extended commentary to the material.

An advantage of this scheme is the straightforward introduction of concepts included in MPEG-7, such as Multimedia Description Schemes (MMDS) [11] and XML-compatible

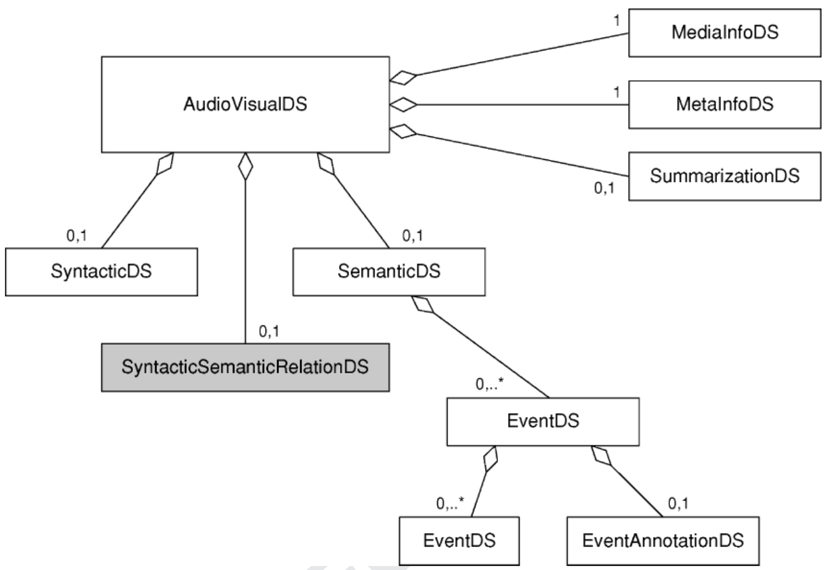

Figure 2. Representation of the AudioVisual DS description hierarchy.

content management. The target of these concepts is to standardize a set of tools dealing with description and management issues, as well as hierarchical navigation and retrieval in complex or simple multimedia entities. Since new generation web browsers offer inherent support for 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 can be extended to suit specific needs or match existing data and application schemas, they already are more than enough for the vast majority of systems. The hierarchical structure of our system is shown in figures 2, 3 and 4 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.

In general, the AudioVisual DS is designed as a metaphor for 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 4, while the semantic content is described with the aid of the Semantic DS and

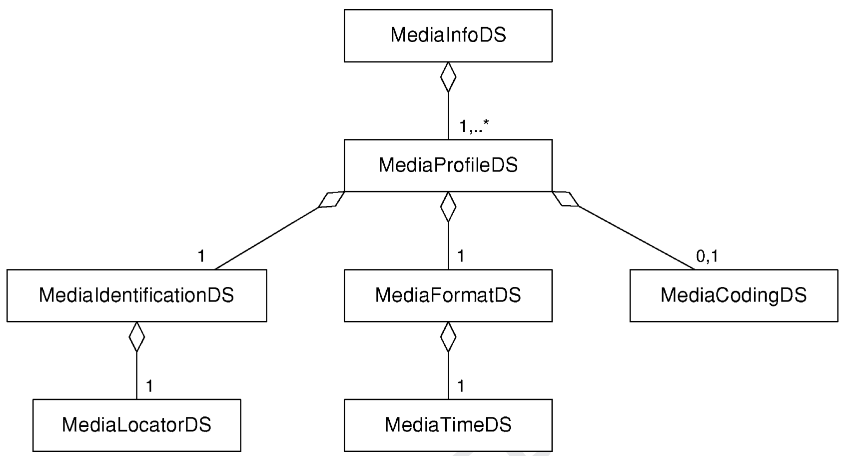

Figure 3. Technical information represented in the MediaInfo DS.

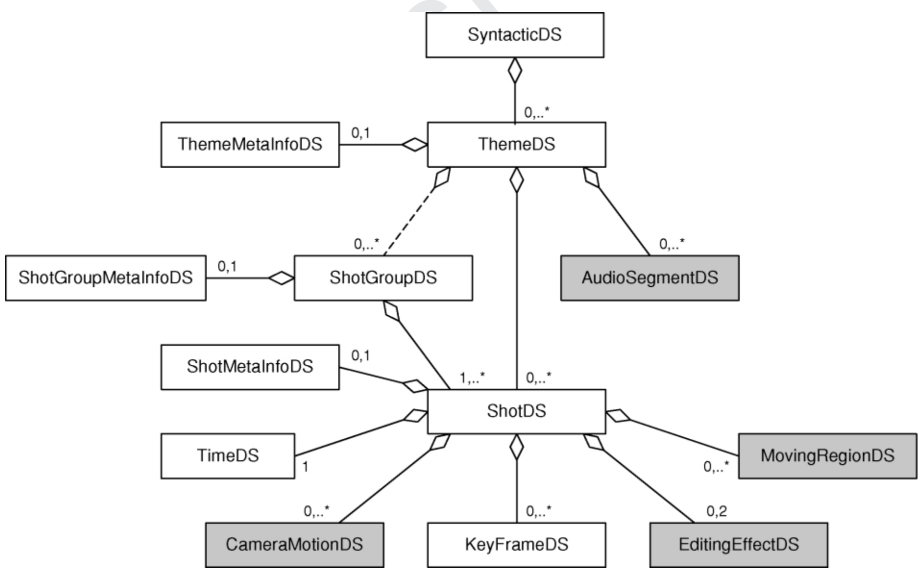

Figure 4. Structure of the video material in the Syntactic DS.

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, e.g., a *Theme*, 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 entities, which help humans interpret the 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 in a mapping of semantic entities to time intervals in the video shots. The descriptors (Ds) related to what is happening in each interval may be predefined in the sense of a dictionary or include free text annotations. The latter case is more useful when humans 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 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 description

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 technical constraints (e.g., limited bandwidth for real-time video transmission) and recommend other related or updated material. In addition to this, the archive may act as a *User Agent* or *Proxy*, locating and retrieving related data from the same archive or the Internet. The actual details 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 the actual choices and preferences of the specific 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 of a video shot. This information is utilized by a crawler to facilitate the mining for relevant content, without the explicit request of the user. In practice, the User DS includes support for either filtering and search preferences, as well as the browsing and filtering history for the current and previous session of a specific user. The former may be considered as the *static* knowledge of the system, in the sense that it incorporates the information that is used by the filtering subsystem in a format that permits immediate utilization, while the history entries are dynamic and determined at run-time; an off-line task of the archive is to integrate this dynamic information with the predefined user preferences. This is accomplished through the formation of a user profile that is updated to reflect the actual user behavior.

4. Asset retrieval

4.1. Summarization of the textual descriptions

The first step in analyzing the textual description and extracting keywords is to remove digits and punctuation, as we assume that words consist of letters only. The second filtering step takes into consideration *noise words* (or *stop words*) such as 'a', 'the', 'in' etc. and *noise stems*, for the specific topic of interest, 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; if a match occurs, the input word is ignored. Thus, common invariant words and common stems can be kept out of the index that characterizes the document. The noise stems are suggested by a specialized expert on each topic.

After considering all the previous cases, we reduce the redundancy of the remaining words again, using a stemming algorithm. For example, the words ‘characters’, ‘characterize’, ‘characteristic’ and ‘characterization’ all reduce to the root (or canonical stem) ‘character’. A well-known algorithm [7] which is based on the Porter suffix-stripping algorithm (or ‘Porter stemmer’) is used as a process for removing common morphological and inflectional endings from words in English. Descriptions in Greek are processed with the vertical stemmer described in [12]. The results of the aforementioned analysis are used in the keyword extraction phase.

In order to compensate for term ambiguity [23], we use a thesaurus to map terms with similar meaning to the same feature. This thesaurus is compiled by the expert historian and is used to provide the system with information on the semantic content of a video shot, with respect to the categories that the material is classified in. For example, the words *conquest*, *triumph*, *success* and *win* are all replaced with the term *victory*, which is a part of the thesaurus for the *Warfare*, *Sports* and *Politics* . These are included in the text analysis module, shown in figure 1. Query formation is independent of the exact phrasing that the annotator uses and does not require the user to be familiar with the specific entries of the system vocabulary.

As a general rule, every extracted word is assigned a weight corresponding to the frequency that it occurs in the ‘hotlist’ pages, and the infrequency that it occurs in the ‘coldlist’ pages [15]. This can be accomplished by finding the mutual information between the presence and absence of a word and the classification of a page. Another approach uses the vector space information retrieval paradigm where documents are represented as vectors [22]. 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. In this simple case the weight for a word **w** belonging to a document d is given by:

$$w_{ds} = f_{ds} \log \frac{N_D}{n_s} \tag{1}$$

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 in the collection and n_s is the number of documents containing the word **w**. One recent method [3] uses a more sophisticated TF-IDF scheme, which normalizes for document length, following the recommendations of [22]. According to Salton and Buckley, vector-length normalization typically does not work well for short documents. Then, the weight for a word **w** is estimated by the following formula which has been adopted in our scheme:

$$w_{ds} = \frac{\left(0.5 + 0.5\frac{f_{ds}}{f_{d\max}}\right)\left(\log\frac{N_D}{n_s}\right)}{\sqrt{\sum_{j\epsilon d}\left(0.5 + 0.5\frac{f_{ds}}{f_{d\max}}\right)^2\left(\log\frac{N_D}{n_j}\right)^2}} \tag{2}$$

Table 1. Keywords used as features for documents describing 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

where the new variable $f_{d\,\max}$ expresses the highest term frequency. 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 search complexity, decrease communications load and avoid over-fitting. Experiments in [15] have demonstrated that the number of words is crucial for constructing a robust scheme. Too many words lead to a performance decrease during the classification process of web pages even when supervised learning methods have been incorporated. Furthermore, our experiments for a small vocabulary (less than ten words) have shown that recommendation results were poor compared to cases when thirty or fifty words composed the vector of a document. Table 1 shows some of the most informative words obtained from a collection of documents concerning historical events.

As one can observe in Table 1, all words consist of letters only, and they are in lowercase form. 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. The weights correspond to the length of the document and the frequency of the specific words. Each time a user accesses a new page, the weights of their profile are updated according to new pages' analysis. The document vectors are used to update the user profiles, a process refered in the information retrieval comunity as relevance feedback [20].

4.2. User profiling

The search process in a multimedia database can produce overwhelming amounts of information, especially in the case of a user that does not look for something specific. In order to reduce transmission time and results' complexity, it is desirable to rank the results according to the user's preferences and 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 [13] and recommend relevant additional shots for further study as shown in figure 1. For each video shot, the system produces a feature vector that consists of sixteen content category weights (see Table 2), followed by five user category weights, describing in essence a fuzzy relevance to a fixed set of categories.

The actual content categories were determined by the nature of the archive; the videos were taken in a period from the beginning of the century until the early 70's. This means that themes, such as space travel or computers are not accounted for. The content is to be extended to include such videos, where the notion of text summarization, described in Section 3.1, can be employed to calculate the relevant coefficients. This can be accomplished by using the summarized keywords from each shot and taking into account their relevance

Table 2. The categories that the material is classified in.

Sports	Arrivals–Departures	Industry–Commerce	Transportation
Celebrations	Ecclesiastical themes	Military topics	Government
Public services	Artistic	Politics	Education
Tourism	Celebrities	Historical events	Head of state

to the new categories. Besides this, most new themes may be integrated into the existing ones, e.g., *Space* into *Transportation*. The user category weights correspond to five typical users of the system, namely Historian, Journalist, Cinephile, Director and Casual User. The resulting vectors are normalized for comparison purposes, thus building a 21-D unit hypercube. 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. The axis ordering is irrelevant to the process.

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

$$r(\mathbf{c}, \mathbf{u}) = \mathbf{c} \cdot \mathbf{u} \tag{3}$$

where $\mathbf{u}$ is the user profile vector, $\mathbf{c}$ is the shot vector and r is the resulting relevance function. The value of the relevance function r is used to sort the returned shots, so that the shots which are more likely to be relevant are displayed first as it is probable that the user is more interested in them. During the registration stage, new users are allowed to review their initial, neutral profile and adjust it to better match their interests and preferences. In addition, the system tracks the transactions and choices of the user so as to further refine the profile and improve the model of his persona. In contrast to other proposed architectures, our system does not require the user to rate the material retrieved from the query.

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 $\Delta\mathbf{u}$:

$$\Delta\mathbf{u} = s \cdot \lambda \cdot \mathbf{c} \tag{4}$$

where $s = 1$ if the user selects $\mathbf{c}$ and $s = -1$ if the user ignores $\mathbf{c}$ and λ is a positive parameter, typically lower than 0.001, ensuring smoothness of the updating procedure. This vector increment is calculated once per session, so as to take into account the fact that the user may look for a specific item just once, as a result of casual browsing or a specific, one-time request. If the user is not actually interested in the genre of the specific item, then the difference which results from the one-time visit should not be able to alter his/her profile significantly. On the other hand, if the specific interest does exist, the individual contributions of $\Delta\mathbf{u}$ will add up, resulting in the adaptation of the profile.

5. Video shot recommendation

Our system supports two types of dynamic recommendation services (shown in figure 1): content-based, where video shots similar to the ones the user is viewing are suggested and

P1: ***
Multimedia Tools and Applications KL2224-02/5265452 January 14, 2004 2:28

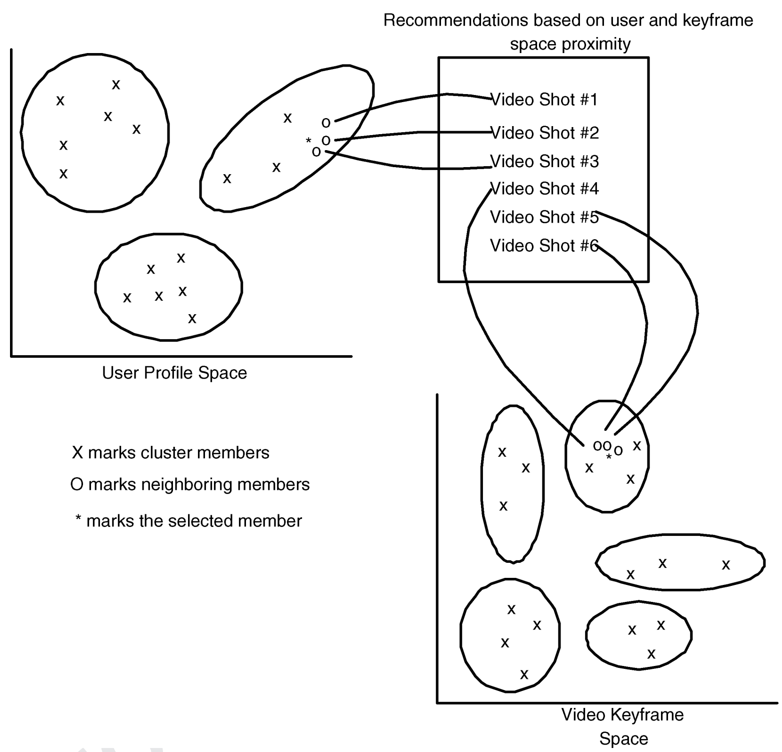

Figure 5. Recommendations based on clustered spaces.

collaborative, where the system recommends shots viewed by users that share interests with the current user (see figure 5). Both types are addressed using a similar algorithm.

More specifically, a Self-Organizing Map (SOM) algorithm is used to classify keyframes in 'similar' groups. A Kohonen Artificial Neural Network [8] is utilized to order a set of feature vectors, thus assigning keyframes of similar content to neighboring nodes. The 5 user category weights of the feature vector are ignored to prevent semantically similar content to be classified to diverse regions of the map. This process clarifies relations in the video database by revealing some inherent order.

During the training period, a set of feature vectors describing 3000 keyframes from all available programs were inserted repeatedly into a map consisting of nodes. A weight vector has been associated with each node. This vector initially consisted of random values (in essence representing a random cluster centroid). Nodes responded to the input vector according to the correlation between the input vector and each node's weight vector, using the Euclidian distance as a search criterion.

The node with the highest response to the input, as well as some nodes in the neighborhood, were allowed to learn. In our implementation we used a simple neighborhood function:

$$n(i, j) = m[k, l] \tag{5}$$

where

$$1 \leq i - w < i < i + w \leq dim, 1 \leq j - w < j < j + w \leq dim$$

where *dim* denotes the dimension of the map and *w* is the window surrounding the current node that decreases in size during the training period. Learning was achieved by adjusting the weights of the nodes by a small amount to match the input vector:

$$m[k, l](t + 1) = m[k, l](t) + a(t) * (x[i](t) - m[k, l](t)) \quad (6)$$

where a is a learning factor that decreases over time, and $x[i]$ is the input vector.

As a result of this training, a pattern of organization emerged in the map. Different nodes learned to respond to different vectors in the input set, and nodes closer to each other tended to respond to input vectors that were similar to each other. When the weights of the map nodes become stable, the training stage was considered complete. Then, the feature vectors of all keyframes were given as input to the organized map one after the other. Each input vector was associated with the node that responded the strongest (was most correlated) to that vector.

At runtime when the user is viewing a particular shot/keyframe, the system searches the shots/keyframes contained in the same content cluster and suggests the closest members according to the aforementioned dot product metric. About 100 clusters were generated and used in this procedure. This clustering provides an aggressive culling mechanism for the content database, limiting the search for similar keyframes/shots to a small subset of the database. The current implementation schedules a reclustering event once per week.

Likewise, the user profile space was segmented in clusters containing users with similar profile vectors. Due to the fact that the users' set had a considerably lower cardinality, the intra-user Euclidian distance calculation was computationally feasible. Therefore a simpler clustering scheme was utilized, with five clusters, each related with one of the five user categories, being a priori discriminated. Each user is initially associated with the cluster representing the most related user category. Due to the simple classification criterion, the clustering is updated in realtime whenever the user profile is modified.

In essence, we assume that users which belong to the same cluster share common interests, so it makes sense to recommend shots viewed by 'neighbors' with respect to the user profile cluster

For each user, we keep a record of his *Last 8 Video Selections* (*LVS* set). The colaborative subsystem recomends random shots from the difference of the user's *LVS* set and the union of the *LVS* sets of the three closest users in the cluster

$$CRS = \mathcal{R}(\cup LVS_i - LVS) \quad (7)$$

where $\mathcal{R}$ is an operator that selects random members from a set, LVS_i is the *LVS* for the neighbour user i and *CRS* is the Colaborative Recomendation Set.

We call these suggestions lateral, because they might diverge from the users' path towards information retrieval, while still being of interest to them. Our content domain (movies) is quite suitable for this kind of recommendation due to its static nature. The frequency of new additions to the database is small, enabling lots of different users to view the same items.

Au: Pls. provide Figure caption.

video shot # 1 video shot # 2 video shot # 3

Figure 6.

Furthermore, the categories are predefined, thus enabling the creation of coherent content clusters.

5.1. A hands-on scenario

We will demonstrate the ranking mechanism of our system's dynamic search with an example: the user is interested in videos referring to the 'King George of Greece' and enters that phrase in the appropriate text field of the client screen. The system queries the database and returns two video shots (shot #1 and #2 in figure 6).

The user profile vector is shown in Table 3 while the vectors of the matched keyframes are presented in Table 4. All vectors consist of the sixteen content category weights and the five user category weights; the vectors are presented in un-normalized form to show the actual weights allocated in the range [0 . . . 1].

The feature vector having the best match with the user profile (Video shot #1) is the first in Table 4. This video shot shows the return of King Constantine of Greece, son of King George, after his trip to the States in the summer of 1967. Video shot #2 (with vector elements also in Table 4) is taken from a parade in downtown Athens in 1938. Although King George is actually missing from video shot #2, his absence is strongly noted by the expert historian. The full annotation text includes '. . . *those propagandistic and nationalistic films, played in both Athens and the province from 1938 to 1940, refer to the coup of the 4th of August and I. Metaxas; King George and the rest of the royal family are absent from those films.*'

For each video shot, the calculated relevance functions are:

$$\mathrm{r}(\mathbf{c1}) = \mathrm{norm}(\mathbf{c1}) \cdot \mathrm{norm}(\mathbf{u}) = 0.732 \tag{8}$$

Table 3. User profile vector **u.**

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

Table 4. The 21-D vectors for each of the 3 shots.

			Shot #1			
0.0	1.0	0.0	0.4	0.8	0.0	0.2
0.4	0.0	0.0	0.8	0.0	0.1	0.9
0.1	0.9	0.8	0.2	0.4	0.2	0.7
			Shot #2			
0.0	0.0	0.0	0.0	1.0	0.0	0.8
0.8	0.7	0.0	0.8	0.0	0.0	0.0
1.0	0.1	0.9	0.5	0.2	0.4	0.7
			Shot #3			
0.0	0.9	0.0	0.3	0.7	1.0	0.3
0.1	0.0	0.0	0.2	0.0	0.1	0.9
0.5	0.9	0.9	0.4	0.2	0.9	0.1

and

$$r(\mathbf{c2}) = \text{norm}(\mathbf{c2}) \cdot \text{norm}(\mathbf{u}) = 0.631 \tag{9}$$

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 keyframe/video shot #3, which is also shown in figure 6, 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.

The collaborative subsystem also suggests another highly relevant keyframe/shot shown in figure 7. This video shot is taken from a military celebration in 1938. The King himself

Figure 7. ‘Lateral’ video shot.

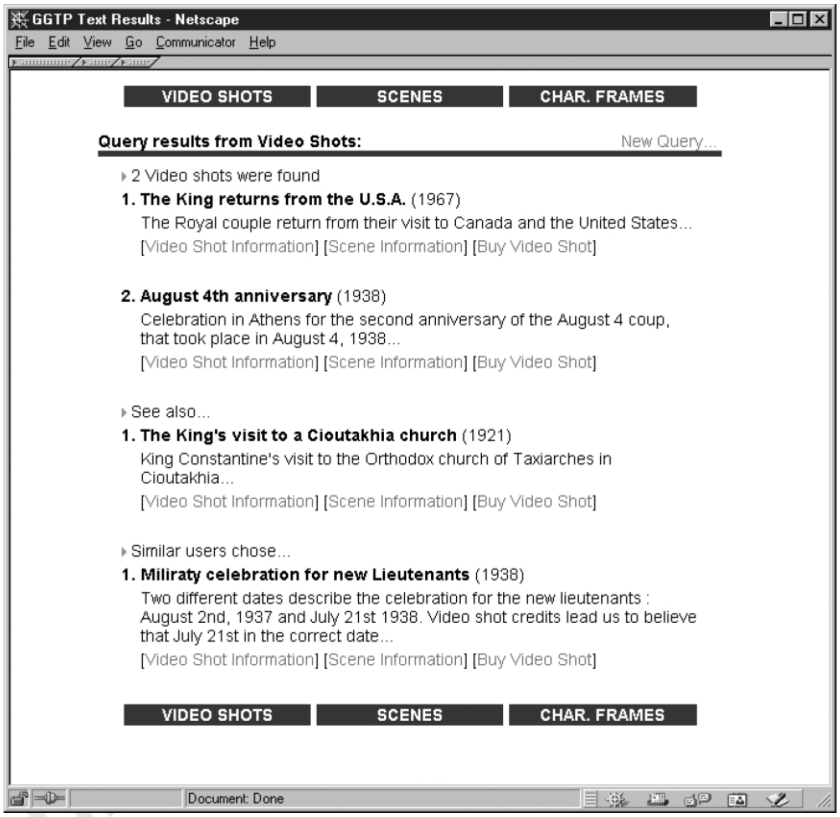

Figure 8. Retrieved and suggested shots with summarized text descriptions.

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 suggested as highly relevant from the system. The complete screen with the two retrieved shots and the suggestions made by the system, along with summarized descriptions, is shown in figure 8.

6. Conclusions

A system which provides user access to large audiovisual databases by considering their queries, alongside with their preferences, as well as the preferences of users with similar profiles, has been presented in this paper. This system has been successfully implemented in a real-life historic audiovisual asset. It is currently being extended to completely fit the MPEG-7 standard framework, especially focusing on semantic to syntactic matching issues. An extension of this system for content-based intelligent access to large heterogeneous archives is currently under development [10]. In this framework the feature sets extracted from each keyframe are used for image matching permitting sketch-based user queries. More features are introduced in this framework, such as the number of human faces appearing in the keyframes; face detection is performed using appropriate template matching [26]. Other related current work can be found in the proceedings of an International Workshop focusing on MPEG-7 and visual representation issues [16] which we recently organized.

Acknowledgments

This work was partially funded by the Greek Ministry of Press and Mass Media (MPMM) in the framework of the program ‘Digitization, archiving and access to MPMM audiovisual data’ (1999–2000). The Movie Archive of the Ministry holds the copyright for shots and stills presented in this paper. The system has been installed and used within the Ministry of Press, for both internal and external users, since December 2000.

References

1. G. Akrivas, N. Doulamis, A. Doulamis, and S. Kollias, “Scene detection methods for MPEG-encoded video signals,” in Proceeding of the 10th IEEE Mediterranean Electrotechnical Conference, Nicosia, Cyprus, July 2000, pp. 677–680.
2. Y. Avrithis, A. Doulamis, N. Doulamis, and S. Kollias, “A Stochastic framework for optimal key frame extraction from MPEG video databases,” Computer Vision and Image Understanding, Vol. 75, Nos. 1/2, pp. 3–24, 1999.
3. M. Balabanovic and Y. Shoham, “Fab: Content-based collaborative recommendation,” Communications of the ACM, Vol. 40, No. 3, pp. 66–72, 1997.
4. C. Basu, H. Hirsh, and W. Cohen, “Recommendation as classification: Using social and content-based information in recommendation,” in Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, WI, pp. 714–720, 1998.
5. Yeo Boon-Lock and Liu Bede “Rapid scene analysis on compressed video,” IEEE, CSVT, Vol. 5, No. 6, pp. 533–544, 1995.
6. A. Doulamis, Y. Avrithis, N. Doulamis, and S. Kollias, “Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback,” in Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMSC ’99), Florence, Italy, Vol. 2, pp. 954–958.
7. W.B. Frakes, “Stemming algorithms,” in Information Retrieval Data Structures and Algorithms, Prentice Hall: Upper Saddle River, NJ, USA, 1992, pp. 131–160.
8. Simon S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall: Upper Saddle River, NJ, USA, 1998.
9. Hill, Stead, “Rosenstein & Furnas recommending and evaluating choices in a virtual community of use,” in Proceedings of CHII95 Conference on Human Factors in Computing Systems, ACM Press, 1995.
10. IST Program, Unified Intelligent Access to Heterogeneous Audiovisual Content (FAETHON) 2001–2003. http://image.ntua.gr/faethon.
11. R. Koenen and F. Pereira, “MPEG-7: A standardised description of audiovisual content,” Signal Processing: Image Communication, Vol. 16, No. 1–2, pp. 5–13, Sept. 2000.
12. Y. Kotsanis, Y. Maistros, and A. Zavras, “Quicklem: A software system for Greek word-class determination,” Literary and Linguistic Computing, Oxford University Press, 1987, Vol. 2, No. 4.
13. M. Montebello, “Optimizing recall/precision scores in IR over the WWW,” in Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press: New York, NY, USA, 1998, pp. 361–362.
14. A. Moukas and P. Maes, “Amalthaea: Evolving multi-agent information filtering and discovery systems for the WWW,” Autonomous Agents and Multi-Agent Systems, 1998, Vol. 1, pp. 59–88.

15. M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting web sites," in Proceedings of the National Conference on AI, AAAI Press: Menlo Park, California, USA, 1996, pp. 54–61.
16. In Proceedings of International Workshop on Very Low Bitrate Video Coding, (VLBV01), Stefanos Kollias (Ed.), Athens, Greece, 2001. http://image.ntua.gr/vlbv01
17. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proceedings of the ACM Conference on Computer-Supported Cooperative Work, ACM Press: New York, NY, 1994, pp. 175–186.
18. E. Rich, "Users are individuals: Individualizing user models," International Journal of Man-Machine Studies, Vol. 18, pp. 199–214, 1983.
19. R. Rivest, "The MD5 message-digest algorithm," RFC 1321. http://www.ietf.org/rfc/rfc1321.txt
20. J. Rocchio Jr., "Relevance feedback in information retrieval," The SMART Retrieval System-Experiments in Automatic Document Processing, Prentice Hall: Upper Saddle River, NJ, USA, 1971, pp. 313–323.
21. P. Salembier, R. Qian, N. O'Connor, P. Correia, I. Sezan, and P. van Beek, "Description schemes for video programs, Users and Devices," Signal Processing: Image Communication, Vol. 16, Nos. 1–2, pp. 211–234, Sept. 2000.
22. G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Information Processing & Management, Vol. 24, No. 5, pp. 513–523, 1988.
23. M. Sanderson, "Retrieving with good sense," Information Retrieval, Vol. 2, No. 1, pp. 47–67, 2000.
24. W. Simpson, PPP Challenge Handshake Authentication Protocol (CHAP) RFC 1994.
25. U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating Word of Mouth," in Proceedings of the CHI '95, Denver, CO. May 1995.
26. N. Tsapatsoulis, Y. Avrithis, and S. Kollias, "Efficient face detection for multimedia applications," in Proceedings of the ICIP'00, Vancouver, BC, Canada, Sept. 2000, Vol. 2, pp. 247–250.
27. N. Tsapatsoulis, Y. Avrithis, and S. Kollias, "Facial image indexing in multimedia databases, pattern analysis and applications," Special Issue on Image Indexation, Springer-Verlag, 2001, Vol. 4, No. 2/3, pp. 93–107.
28. T.W. Yan and H. Garcia-Molina, "SIFT—A tool for wide-area information dissemination," in Proceedings of the USENIX Technical Conference, USENIX, Berkeley, CA, USA, 1995, pp. 177–186.

Au: Pls. provide Bios and photos.