Effective Access to Large Audiovisual Assets Based on User Preferences

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

Image, Video and Multimedia Laboratory

Electrical and Computer Engineering Department

National Technical University of Athens

Heroon Polytechniou 9, 157 80 Athens, Greece

email : sivann@softlab.ece.ntua.gr

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 users with a similar profile are interested in. In this paper, we present a number of solutions to these issues, giving as an example an integrated architecture we have developed, along with notions that support efficient and secure Internet access and easy addition of new material. Segmentation of the video in shots is followed by shot classification in a number of predetermined categories. Generation of users' profiles according to the same categories, enhanced by relevance feedback, permits an efficient presentation of the 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 specific users 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, text-based search, user profiling, query expansion

ACM Multimedia Workshop Marina Del Rey CA USA

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 used to be rare and occasional, 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 to that, current and evolving standards, such as MPEG-4 and MPEG-7 [5], 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 advanced quality, such as entertainment. Users of this kind will benefit from the advanced summarization schemes offered by the above standards 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 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, the video data are generally commented on by experts, generating the metadata that is necessary to better comprehend the content. Textual metadata can also be 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 the 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 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. Profiles are mostly built up by providing material to the user, such as web pages, questionnaires, stored material, etc., 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 serviced. As a result, items preferred by users of similar profiles are predicted as cases that possibly interest the specific user 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" [8] 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 [12] which adjusts the weights of a profile by incorporating a relevance feedback approach and the "Amalthaea" [7], 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" [9], which is scheduled to collaboratively filter netnews, the "WebHound" agent that locates users with similar ratings to specific pages and suggests unread pages that are preferred by them, the "Ringo" system, which is devoted to filter social information and the "Bellcore", that is a video-recommender, which efficiently combines users' choices. In general, one 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 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.

Other, hybrid, systems have also been proposed which suggest pages that score highly against someone's profile or are rated highly by users with similar profiles. An effective and robust example of such a system is the Fab [2], which is oriented towards information retrieval and relevance feedback, as well as automated filtering of incoming information.

2. WEB-BASED ACCESS

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 up in two different computers, both behind a firewall, as a part of the system requirements specifications. As far as Internet access is concerned, this setup imposes a number of restrictions, which would require resetting the existing firewall system in order to overcome them.

2.1 THREE-TIER ARCHITECTURE

The underlying principle and data flow in the three-tier system is described in Figure 1:

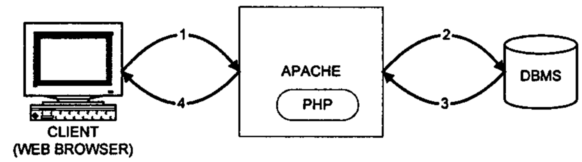

Figure 1. Three-tier architecture block diagram

In such a 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 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, any system policy issues, e.g. restrictions or logging, that need to be enforced can be included in this module. 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.

We have implemented the middle tier using PHP, a server-side, cross-platform, HTML embedded scripting language, because if offers a number of advantages over two-tier or client-server systems, such as:

- Data security: the client is restrained from querying critical data, such as the database schema or security policy options; the middleware component decides the amount and type of data permitted for transmission.
- Advanced resource management: due to security restrictions, any data revision and management is fulfilled in the middle tier. Since all traffic is controlled from here, the system is given the opportunity to perform load balancing and/or favor users with higher bandwidth or privileges.
- Easy maintenance and redesign: since all business logic is separated from the data structures and the presentation layer, any solitary changes are not cascaded to other modules. Changes in presentation and policies are handled in discrete sections of the script, while changes in the database schema are handled in isolated functions that build the queries.

2.2 SECURE ACCESS

User authentication follows a three-way handshaking scheme, similar to the one used in CHAP (RFC1994) [4] and is used only during the initial authentication phase. This procedure consists of the following steps:

- The initial login screen generated by the PHP module of the web server, contains the login and password form fields, along with a random number stored in a hidden form field. This random number is called a challenge key and is generated every time the initial login screen is requested.

- The web browser calculates the MD5 ([4] - RFC 1321) digest of a string containing the user name, password and challenge key. The constructed message digest is sent back to the server, along with the supplied user name. The complexity of the digest algorithm makes it computationally infeasible ([4] - RFC 1321) to produce two MD5 messages that map to the same digest or produce any message with a given pre-specified target message digest. The MD5 algorithm is implemented in JavaScript and, thus, is executed in the client side.

- The PHP login script computes the same digest by using the plaintext user name string received by the browser, and the random key & plaintext password retrieved from the database. If the two strings match, the user is authenticated. The whole process is depicted in Figure 2.

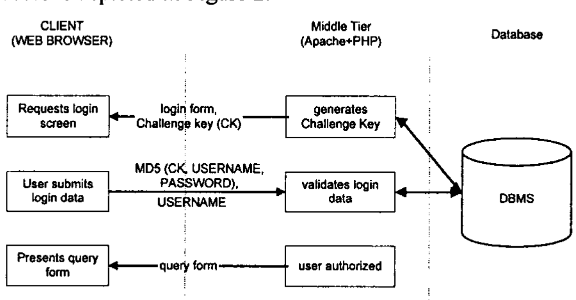

Figure 2. The authentication process

This way, even if malicious users "sniff" the network and gain access to the transferred data, they do not gain access to the database system because the actual password is never transmitted in plain text. Repeating of the encrypted string is of no use either, since the server-generated challenge key is random and changes according to authentication attempts, time of day and the client's IP address.

After users log in, they are authenticated on each subsequent query or request. This authentication is based on the combination of the user name and the client's IP number which are stored in the database, so as to prevent multiple concurrent logins. In addition to that, when a client request is made, an appropriate field containing the time of the last request is first checked and then updated in the database; thus, the system can impose an auto-logout procedure for long-inactive users.

2.3 DATABASE STRUCTURE

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 the popular scene-shot-characteristic frame hierarchical scheme. At first, the videos were digitized from the original reels and recorded to Betacam SP tapes, followed by MPEG-2 encoding. This material was then 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 [1]; this assists the summarized presentation of the shot, while giving the expert the opportunity to add extended commentary to the material.

A nice advantage of this description scheme is the straightforward introduction of concepts included in MPEG-7, such as Multimedia Description Schemes (MMDS) [5] and XML-compatible content management. 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.

3. ASSET RETRIEVAL

3.1 SUMMARIZATION OF THE TEXTUAL DESCRIPTIONS

The first step in analyzing the textual description and extract 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. For the sake of simplicity, let us assume that the noise stems are suggested by the specialized expert on each topic.

After considering all the previous cases, as a final step we reduce the redundancy of the remaining words, by detecting the specific stem of each word. For example, the words "characters", "characterize", "characteristic" and "characterization" all reduce to the root (or canonical stem) "character". A well-known algorithm [3], 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.

After performing the aforementioned analysis, the keyword extraction phase is activated. In the case of plain textual documents, information-based approaches are adopted to determine which words can be used as features. As a general rule, every extracted word can have a weight corresponding to the frequency, that it occurs in the "hotlist" pages, and the infrequency that it occurs in the "coldlist" pages [8]. 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 [11]. 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:

$$w_{ds} = f_{ds} \cdot \log \frac{N_D}{n_s} \qquad (1)$$

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 [2] uses a more sophisticated TF-IDF scheme, which normalizes for document length, following the recommendations of [11]. 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 also been adopted in our scheme:

$$w_{ds} = \frac{\left(0.5 + 0.5 \frac{f_{ds}}{f_{d\max}}\right)\left(\log \frac{N_D}{n_s}\right)}{\sqrt{\sum_{j \in d}\left[\left(0.5 + 0.5 \frac{f_{dj}}{f_{d\max}}\right)^2 \left(\log \frac{N_D}{n_j}\right)^2\right]}} \qquad (2)$$

where $f_{d\max}$ 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 memory charge, decrease communications load and avoid over-fitting. Experiments in [8] 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.

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

As one can observe, all words consist of letters only, and they are in the 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. 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, each time a user accesses a new page, 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 [10].

3.2 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 users 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 [6] and recommend relevant additional shots for further study.

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 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.

Table 2. The categories that the material is classified in

Sports	Arrivals - Departures	Industry - Commerce	Communications - Transportation
Celebrations	Ecclesiastical Themes	Military Topics	Governmental - Municipal Themes
Public Services	Artistic	Politics	Education
Tourism	Celebrities	Historical Events	Head of State

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

$$\mathbf{r}(\mathbf{c}, \mathbf{u}) = \mathbf{c} \bullet \mathbf{u} \qquad (3)$$

where **u** is the user profile vector, **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 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 $\Delta\mathbf{u}$:

$$\Delta\mathbf{u} = s \bullet \lambda \bullet \mathbf{c} \qquad (4)$$

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 0.1, ensuring smoothness of the updating procedure.

3.3 VIDEO SHOT RECOMMENDATION

Our system supports two types of dynamic recommendation services: content-based, where video shots similar to the ones the user is viewing are suggested and collaborative, where the system recommends shots viewed by users that share interests with the current user. Both types are addressed using a similar algorithm. More specifically, standard clustering algorithms are used to segment the content and user spaces in 'similar' groups.

A Kohonen Neural Network provides a topological map, where shots of similar content are assigned to neighboring nodes. The network is updated whenever new material is introduced in the database. At runtime when the user is viewing a particular shot, the system searches the shots contained in the same content cluster and suggests the closest members according to the aforementioned dot product metric. This clustering provides an aggressive culling mechanism for the content database, limiting the search for similar shots to a small subset of the database.

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. The recommendations come from a pool of most-viewed shots by members of the same cluster. We call these suggestions lateral because they might diverge form the users' path towards information retrieval while still being interest to them.

In our implementation, we store hit frequencies per cluster and video shot. Due to the dynamic nature of user profiles, the profile space clustering is updated when new users are registered or existing profiles are refined; the current implementation schedules this once per week. Our content domain (movies) is quite suitable for this kind of adaptive 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. Furthermore, the categories are predefined, thus enabling the creation of coherent content clusters.

3.4 A HANDS-ON SCENARIO

We will demonstrate the ranking mechanism with an example: the user is interested on videos referring to the "King George of Greece" and enters that phrase in the appropriate field of the client screen. 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. These vectors consist of the sixteen material category weights and the five user category weights; also, a subset of the returned shots are shown and the vectors are presented in un-normalized form to show the actual weights allocated in the range [0..1].

Figure 3. Retrieved video shot #1

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 vector for shot #1 (c1)

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

Figure 4. Retrieved video shot #2

Table 5. The 21-D vector for shot #2 (c2)

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

Video shot #1 shows the return of King Constantine of Greece, son of King George, after his trip to the States in the summer of 1967, while video 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 strongly noted by the expert historian. Given the calculated relevance functions, we have 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 following video shot based on its close proximity to the aforementioned items:

Figure 5. Suggested video shot #3

Table 6. The 21-D vector for suggested 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

This shot, from 1921, shows King Constantine, father of King George, during a highly celebrated visit to an Orthodox church in Asia Minor.

The user in question is classified to a profile cluster with the following mean vector:

0.1	0.2	0.2	0.3	0.9	0.6	0.1
0.1	0.1	0.2	0.9	0.2	0.9	0.9
0.9	0.1	0.8	0.0	0.3	0.1	0.2

and the collaborative subsystem also suggests this relevant shot:

Figure 6. "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 suggest 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 7 below.

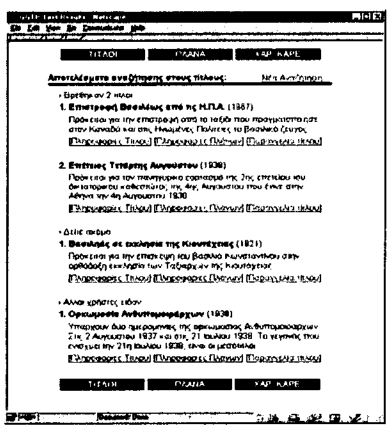

Figure 7. Retrieved and suggested shots with summarized text descriptions

4. ACKNOWLEDGEMENT

This work was partially funded by the Greek Ministry of Press and Mass Media (MPMM) and 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.

5. REFERENCES

[1] Avrithis Y., Doulamis A., Doulamis N. and Kollias S., "A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases", *Computer Vision and Image Understanding*, Vol. 75, No. 1/2, July/August 1999, pp. 3-24.

[2] Balabanovic M. and Shoham Y., "Fab: content-based collaborative recommendation", *Communications of the ACM*, 40(3), Mar. 1997, pp. 66-72.

[3] Frakes, W. B., "Stemming algorithms", *Information Retrieval Data Structures and Algorithms*, Prentice Hall, Saddle River, NJ, USA, 1992, pp. 131-160.

[4] Internet Engineering Task Force, Request for comments, http://www.ietf.org/rfc.html.

[5] ISO/IEC JTC1/SC29/WG11 N3250, "MPEG-7 Principal Concept List", March 2000.

[6] Montebello M., "Optimizing Recall/Precision scores in IR over the WWW," *Proceedings of the 21st ACM SIGIR Conference on Research and development in information retrieval*, ACM Press, New York, NY, USA, 1998, pp. 361-362.

[7] Moukas, A. and Maes P., "Amalthaea: evolving multi-agent information filtering and discovery systems for the WWW", *Autonomous agents and multi-agent systems*, Vol. 1, 1998, pp. 59-88.

[8] Pazzani, M., Muramatsu J., Billsus, D., "Syskill & Webert: Identifying interesting web sites", *Proceedings of the National Conference on AI*, AAAI Press, Menlo Park, CA, USA, 1996.

[9] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., "GroupLens: An open architecture for collaborative filtering of netnews", *Proceedings of the ACM Conf. on Computer-Supported Cooperative Work*, ACM Press, NY, 1994, pp. 175-186.

[10] Rocchio Jr. J., "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.

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

[12] Yan, T. W. and Garcia-Molina, H., "SIFT - A tool for wide-area information dissemination", *Proc. of the USENIX Technical Conference*, USENIX, Berkeley, CA, USA, 1995, 177-186.