

# Web-based Multimedia Retrieval: Balancing out between Common Knowledge and Personalized Views

<sup>1</sup>Qing Li, <sup>2</sup>Jun Yang, <sup>2</sup>Yueting Zhuang

<sup>1</sup> Department of Computer Science  
City University of Hong Kong  
Kowloon, HKSAR, China  
csqli@cs.cityu.edu.hk

<sup>2</sup> Department of Computer Science  
Zhejiang University  
Hangzhou, 310027, China  
yangjun@acm.org, yzhuang@cs.zju.edu.cn

## Abstract

*The major challenges of multimedia retrieval are the difficulty of generating semantic indexes, as well as the incapability of identifying personalized user interests. This paper attempts to address both problems by suggesting a collaborative yet personalized approach for web-based multimedia retrieval, which employs a synergy between relevance feedback technique from the Information Retrieval community, and user profiling technique from the Information Filtering community. Specifically, a “common profile” is established to represent the common knowledge on the semantics of multimedia data, which allow a user to “learn from others” in the retrieval process. On the other hand, for each user a “user profile” is constructed to characterize his/her personal views, which allow a user to “learn from own history”. Both types of profiles can be learned and updated incrementally from user feedbacks. By using an integrated retrieval algorithm based on profiles, this approach strikes the balance between exploiting the common knowledge of most users and catering for the personalized interest of a particular user. The results of some preliminary experiments have demonstrated the effectiveness of the proposed approach.*

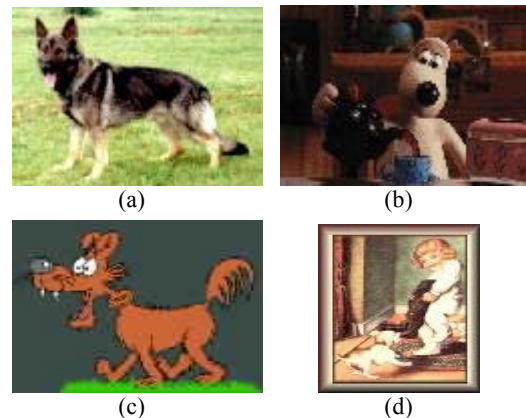
## 1. Introduction

The increasing availability of “on-line” multimedia information creates the challenge of developing effective and efficient retrieval tools that can address a variety of medias, such as text, still images, video clips. Despite the vast amount of literatures dedicated to this field, there still exist two major problems that plague the practical multimedia retrieval systems, especially when faced with the tremendous and heterogeneous Web environment accessible to a great population of users.

- **Difficulty of semantic indexing.** Human are inclined to use high-level semantics to evaluate the retrieval results. However, what current retrieval systems extract from raw data are mostly low-level features, such as color and texture feature for images, motion feature for videos. In general setting, these low-level features cannot be readily linked to semantics with today’s multimedia technology. Besides, it is prohibitively laborious and time-consuming to annotate all the data manually. Some systems derive the semantics of multimedia objects from their context in the web pages, which is nevertheless very unreliable.
- **Incapability of identifying personalized user interest.** It is quite commonplace that by the same query, different users intend to seek for different results. The four images shown in

Figure 1 are retrieved in top ranks by Ditto.com [4], a popular image search engine, in response to the query of “dog”. Although each of them shows a dog, these images differ drastically in their visual aspect, with (a) as a photo of dog, (b) as a puppet dog, (c) as a cartoon dog and (d) as a painting with a dog in it. On the user side, it is likely that a particular user is interested in only the dogs of a certain style, e.g. a child may be interested in (b) or (c), while an artist may prefer (d). However, the search engine tends to return heterogeneous results, only a small fraction of which is interesting to a particular user.

The underlying reason of the failure to identify personalized user interest is that the query keywords cannot specify the user’s need for multimedia precisely, or the user would not bother to construct a precise but complex query using a lot of keywords. Through a simple query such as “dog”, the search engine is unable to capture the preference of a particular user. Even the “by-example” search paradigm suffers from the same problem, because the object used as query sample can be by itself ambiguous or non-representative.



**Figure 1: Images retrieved by Ditto.com to query “dog”**

As an attempt to remedy the aforementioned problems, this paper proposes a collaborative yet personalized multimedia retrieval approach for the Web environment. Before proceeding with the details, we declare the following as the fundamental assumptions of our approach:

1. Web is a vast multimedia repository that is proliferated with media data. That is, new data pour into Web everyday, while the loss of old data is negligible.

2. The “correct” semantics of a multimedia object is what most people (but not necessarily all the people) agree upon.
3. Even by the same query, each user has his/her personalized interest that may not be consistent with others.

Our approach features a synergy of the typical techniques from two major research communities: relevance feedback from the Information Retrieval community and user profiling from the Information Filtering (Recommendation) community. Specifically, a common profile, which represents the common knowledge on the semantics of multimedia data, is established to enable collaboration among different users. On the other hand, a user profile is set up for each user to characterize his/her personal views. Both types of profile are learned and updated incrementally from relevance feedbacks given by users. In query processing, the common profile is used as the semantic indexes of multimedia data (thus addressing the first problem), while a specific user profile helps to construct *pseudo-feedbacks* that adjust retrieval results towards the interest of a particular user (addressing the second problem). In this way, this approach balances out between exploiting common knowledge and catering for personalized user interest.

The rest of this paper is organized as follows. Section 2 starts by reviewing the previous works related to relevance feedback and user profiling. In Section 3, we propose the representation and learning algorithm for the common profile as well as for user profiles. Section 4 describes how both types of profile can be integrated into the retrieval algorithm. In Section 5, we look at several key issues regarding the implementation of a practical system based on our approach. The results of some preliminary experiments showing the performance of our approach are presented in Section 6. Section 7 summarizes the paper and discusses our future works.

## 2. Related Work

Two strands of related research constitute the foundation of our work: the research on relevance feedback and user profiling. In this section, some representative projects from both areas are described and compared to our work.

### 2.1. Relevance Feedback

Relevance feedback is a powerful technique having its root in the traditional text information retrieval (IR) research. It is an interactive process of improving the retrieval results by automatically adjusting the original query based on the information fed-back by users about the relevance of previously retrieved documents [15]. Rocchio [13] has provided a neat interpretation of relevance feedback in the vector space, where both queries and documents are represented as points. The feedback process is essentially an improvement of the original query point  $Q$  by moving it towards the points representing relevant documents and away from the points of irrelevant ones, formulated as:

$$Q' = \alpha Q + \beta \left( \frac{1}{N_R} \sum_{i \in D_R} D_i \right) - \gamma \left( \frac{1}{N_N} \sum_{i \in D_N} D_i \right) \quad (1)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are suitable constants,  $D_R$  and  $D_N$  represent relevant and irrelevant documents respectively,  $N_R$  and  $N_N$  are

the number of documents in  $D_R$  and  $D_N$ .

The basic principle of relevance feedback is then adopted to content-based image retrieval by Rui and Huang [14]. Their approach improves the weights of low-level image features based on the user evaluations of previously retrieved images. More examples of using relevance feedback for multimedia retrieval include PicHunter [3], MindReader [5] and the framework proposed by Lee et al. [8].

A common weakness of most relevance feedback methods is that the feedback process has no “memory”. Therefore, the user feedbacks conducted in the past cannot help to process the future queries, so that the retrieval accuracy does not improve over time in a long run. The *iFind* [10] image retrieval system provides a viable solution to this problem by using a keyword propagation scheme to learn image semantics progressively from user feedbacks. In our approach, this scheme is adapted for the learning of profiles.

### 2.2. User Profiling

User profiling has been extensively used in the area of information filtering and recommendation, the objective of which is to monitor a stream of information and select those that match the users’ interests. The approaches in this area can be divided into two categories: content-based and collaborative. The content-based approaches judge the interestingness of the information by matching its content with the user interest described in a user profile. Examples of this category include the “WebMate” [2] as a personal agent which learns a user profile incrementally and use it to facilitate browsing and searching in the Web, and the “Syskill & Wbert” [11] agent system which suggests web pages based on their relevance to the user interest. On the other hand, collaborative approaches make recommendation by considering the selections made by other users with similar interest, indicated by the similarity of their user profiles. For instance, the “WebHound” [7] system requires the user to rate a list of pages in order to locate other users with similar ratings. Then, the pages preferred by the similar users are suggested to this user. “GroupLens” [12] system offers a collaborative filtering service of net news. Besides, there are also hybrid approaches that take the advantages of both content-based and collaborative approaches. The Web tour guide “WebWatcher” [6] is such a system that guides a user based on the relevance of a page to the user’s interest, as well as the way that other users have interacted with the same collection of pages in the past. More examples in this category include the “Fab” [1] system and the one proposed by Loannou [9]. Whichever category they belong to, all the works mentioned above rely on user profiling to describe the interest of a particular user, usually in the form of one or several term vector(s).

In an information filtering system, users usually have long-standing interest that will not change drastically over time. In contrast, a user of information retrieval system may change his/her query from one retrieval session to another. Therefore, instead of describing “what information does the user likes”, the profiles in our work memorize “what is the meaning of each media object in the eyes of a user (or users)”. As will be seen in Section 4, we rely on the collaborative power of many users to generate the content descriptions of media objects, which are used to match with user queries. Therefore, our approach can be

described as both content-based and collaborative.

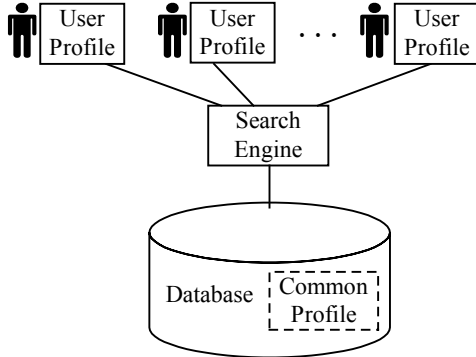
As a whole, our approach is a combination of both ideas: learning the profiles from relevance feedbacks, and then using the profiles to construct *pseudo-feedbacks* that adjust retrieval results towards the interest of a particular user.

### 3. Profile Representation and Learning Algorithm

In this section, we describe how the common profile and user profiles are internally represented, and how they are learned from user feedbacks.

#### 3.1. Profile Representation

Both the common profile and user profiles are composed of the semantic descriptions of media objects (including text, images, videos) in the database. The distinction lies in that the common profile represents the opinions that receive a majority of user consensus, while a user profile characterizes the personalized views of a particular user. As shown in Figure 2, the common profile is unique in the whole system and shared by all the users, while a user profile is created for each user and accessible only to this user.



**Figure 2: The common profile and user profiles**

For each media object  $O_i$ , the common profile maintains a list of weighted keywords as its semantic descriptor, given as:

$$D_i = \{ \langle t_{i1}, w_{i1} \rangle, \dots, \langle t_{ij}, w_{ij} \rangle, \dots, \langle t_{iN}, w_{iN} \rangle \} \quad (2)$$

where  $t_{ij}$  is the  $j$ th keyword (term) attached to  $O_i$ , with  $w_{ij}$  being its weight. The common profile is therefore a collection of such descriptors, represented as  $\{D_i\}$ ,  $i = 1, \dots, M$ , where  $M$  is the number of media objects in the database.

A user profile has the same representation with the common profile, except for the following difference: in the common profile, each keyword appears in descriptor  $D_i$  is a “positive keyword” describing what object  $O_i$  is; however, in a user profile there can be also “negative keywords” describing what the object is not. For example, a positive keyword of “dog” attached to an image indicates that the image shows a dog, whereas a negative keyword of “dog” means the image is not a dog. The reason to have this distinction will be explained in

Section 4.2. Apart from this, the user profile for user  $t$  is similarly denoted as  $\{D_t^k\}$ ,  $i = 1, \dots, M$ , where  $D_t^k$  is the descriptor of object  $O_i$  in the opinion of this specific user, given as:

$$D_t^k = \{ \langle t_{i1}^k, w_{i1}^k \rangle, \dots, \langle t_{ij}^k, w_{ij}^k \rangle, \dots, \langle t_{iN}^k, w_{iN}^k \rangle \} \quad (3)$$

Besides, for the storage concern, neither the common profile nor any user profiles actually store the empty descriptors.

#### 3.2. Learning Profile from User Feedback

The descriptors of media objects in the profile can be obtained by manual annotation, or by semantic analysis of their textual context in the web pages. However, it is more preferable to construct the profiles by learning from user feedbacks.

In a common scenario of multimedia retrieval, a user submits a set of query keywords to the system, and then marks some of the retrieved objects as positive or negative feedback examples. In this way, the user tells the system implicitly that these objects are relevant or irrelevant to the query keywords. Thus, we can confidently add the query keywords into or remove them from the descriptors of the objects marked as feedback examples. The voting algorithm proposed by Lu et al. [10] is adapted and used as a machine learning method to update the common profile. The following is the adapted algorithm applying to a feedback example  $O_i$ :

1. If there is no descriptor found in the common profile for  $O_i$ , an empty descriptor  $D_i$  is created for it.
2. If  $O_i$  is marked as a positive example, we check to see if any query keyword is not in  $D_i$ . If so, that keyword is added into  $D_i$  with its initial weight set to  $\alpha$ ; Otherwise, the weight of that keyword is increased by  $\alpha$ .
3. If  $O_i$  is marked as a negative example, we check to see if any query keyword is in  $D_i$ . If so, the weight of that keyword is decreased by  $c * \alpha$ . If the resulting weight is below zero, that keyword is removed from  $D_i$ .

The constant  $\alpha$  in the algorithm is set as smallest increment or decrement (in our experiment,  $\alpha = 1$ ) of the keyword weight. The constant  $c$  is set to a proper integer (in our experiment,  $c = 5$ ) so that the keyword with user disagreement (on its relevance to an object) will not receive a confident weight.

In the feedback operation, the above algorithm is applied on each feedback example to expand or update its descriptor with the query keywords. The common profile is updated by the feedback submitted from any user. As more and more queries and feedbacks are conducted, it gradually approximates the common knowledge of most users by improving both the coverage and the quality of the object descriptors in it.

Unlike the common profile, a user profile is updated exclusively by the feedbacks from the owner of the profile. We utilize a similar voting scheme as the learning algorithm for user profiles, along with some minor modifications made to cope with the negative keywords. Again,  $O_i$  denotes a feedback example that is designated by user  $k$  in the following algorithm:

1. If there is no descriptor found in the user profile for  $O_i$ , an empty descriptor  $D_i^k$  is created for it.
2. If  $O_i$  is a positive example, we check to see if any query keyword is not in  $D_i^k$ . If so, that keyword is added into  $D_i^k$  as a positive keyword with its initial weight of  $\alpha$ ; Otherwise, the weight of that keyword is increased by  $\alpha$  provided that it is a positive keyword, or decreased by  $c \cdot \alpha$  if it is negative.
3. If  $O_i$  is a negative example, we check to see if any query keyword is not in  $D_i^k$ . If so, that keyword is added into  $D_i^k$  as a negative keyword with its initial weight of  $\alpha$ ; Otherwise, the weight of that keyword is increased by  $\alpha$  provided that it is a negative keyword, or decreased by  $c \cdot \alpha$  if it is positive.
4. Find out all the keywords in  $D_i^k$  with weight below zero. If it is a positive keyword, label it as negative with its weight set to  $-\alpha$ ; If it is a negative keywords, label it as positive with its weight set to  $\alpha$ .

The constant  $c$  is again set to 5, which means a user's new opinion will overwhelm the old ones in the case of a contradiction between these two opinions.

## 4. Profile-Based Retrieval Approach

In this section, we describe a seamlessly integrated multimedia retrieval algorithm using both types of profiles, which allows addressing the low-level features of multimedia in addition to the semantic aspect.

### 4.1. Roles of the Profiles

Despite the similarity in their representations and learning algorithms, the common profile and user profiles play very different roles in the retrieval process. In fact, a major advantage of our approach is to allow two types of profile to benefit from each other. The descriptors in the common profile are used as the semantic indexes of the corresponding media objects. The relevance of a media object  $O_i$  to a query  $Q$  can be estimated by matching its descriptor  $D_i$  with  $Q$ , formulated as:

$$R_i = \sum_{k=1}^M w_{ik} \quad (4)$$

where  $M$  is the number of keywords that coexist in both  $D_i$  and  $Q$ , and  $w_{ik}$  is the weight attached with the  $k$ th of such common keywords in  $D_i$ . Likewise, we can calculate the relevance between two media objects by comparing their descriptors:

$$R_{ij} = \sum_{k=1}^M w_{ik} \cdot w_{jk} \quad (5)$$

where  $M$  is the number of keywords that coexist in both  $D_i$  and  $D_j$ ,  $w_{jk}$  and  $w_{ik}$  are the weights attached to the  $k$ th of such common keywords in  $D_i$  and  $D_j$  respectively.

In contrast, the descriptors in a user profile are used to

construct *pseudo-feedbacks* that can adjust retrieval results towards a particular user's interest. On acceptance of a query from a specific user, before exhaustively searching the whole database, we perform a local search by matching the query with the media objects that are profiled by the user profile of this user. (We say an object is profiled if it has a non-empty descriptor in the profile.) The matching is accomplished using Equ.4, with the weight of negative keywords reversed to negative value. The local search can be performed very efficiently because the media objects to be matched are limited to those that have been evaluated by this user in the past (so that it is profiled by the corresponding user profile). Among the results of this local search, the top matches with similarity above a positive threshold are picked out as *pseudo positive examples*, while the bottommost matches with similarity below a negative threshold are selected as *pseudo negative examples*. These pseudo examples, together with the original query, can be used to construct a relevance feedback operation, as described below.

An inherent limitation of user profile (and the pseudo-feedback approach) is that a new query will receive assistance from the user profile only if some of its keywords have been used in the former queries conducted by the same user. The new query is not necessarily identical with the former ones, but they must have some keywords in common (a query is likely to have multiple keywords) for the new query to make use of the user profile. Although empirically users are used to conduct similar queries, it undoubtedly restricts the applicability of our approach.

### 4.2. Integrated Retrieval Algorithm

Inspired by the Rocchio's feedback formula (cf. Equ.1), we propose a comprehensive similarity function that is used to evaluate the similarity of a media object to a query, by considering the evaluations made by users so far. This function differs from Equ.1 in two aspects. Firstly, it accommodates the semantics as well as the low-level features of media objects. Secondly, it is in the form of a similarity metric instead of a refinement function of the ideal query. It is formulated as:

$$S_i = \alpha R_i + \beta \left\{ \frac{1}{N_R} \sum_{k \in C_R} [(1 + R_{ik}) S_{ik}] \right\} - \gamma \left\{ \frac{1}{N_N} \sum_{k \in C_N} [(1 + R_{ik}) S_{ik}] \right\} \quad (6)$$

where  $\alpha, \beta, \gamma$  are suitable constants,  $C_R$  and  $C_N$  are positive and negative examples,  $N_R$  and  $N_N$  are the number of media objects in  $C_R$  and  $C_N$  respectively.  $R_i$  is the relevance of candidate object  $O_i$  to the original query calculated using Equ.4, based on  $O_i$ 's descriptor in common profile as its semantic index.  $R_{ik}$  is the relevance of  $O_i$  to the  $k$ th positive/negative example defined by Equ.5 (again, based on their semantic indexes in common profile), and  $S_{ik}$  is the their similarity in terms of low-level features. The set of low-level features used depends on the specific media type, e.g., color histogram for images, motion vector for videos, etc. In our experiments which focus on images, we use three types of features: 256-D color histogram in HSV space, 64-D color coherence, and 32-D Tamura directionality. The similarity metric used is Euclidean distance.

Using this similarity function, we can perform the whole process of retrieval and feedbacks via the following steps:

1. *Collect the user query keywords.*
2. *Conduct a local search in the corresponding user profile using Equ.4. Among the matches, we identify those qualified for pseudo positive examples or negative examples and add them into  $C_R$  or  $C_N$  respectively.*
3. *Compute the similarity of each candidate media object to the query using Equ.6, rank the results by similarity and present them to the user.*
4. *Collect the positive and negative examples designated by the user, add them into  $C_R$  and  $C_N$  respectively.*
5. *Update the common profile and the corresponding user profile using the method of Section 3.2 based on the user feedbacks.*
6. *Go to step 3 to start a new loop.*

The retrieval algorithm described above is able to benefit from the common knowledge while still capture the interest of a particular user. On the one hand, the common profile used as semantic indexes, allows the opinions of a majority of users to influence the retrieval process. Especially, in the case of a new user without user profile, or a new query for a specific user with no matches in his/her user profile, the common profile will suggest the results that other users evaluated as relevant to the query. Thus, the retrieval results will not be worse than average degree of user satisfactions.

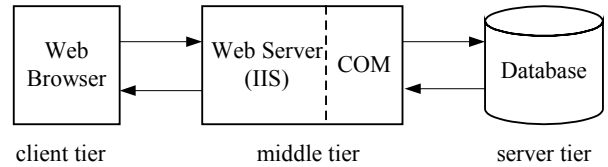
On the other hand, the user profile describes a particular user's interest by memorizing "what each object is or is not" from the view of this user. When faced with a similar query, the system can identify his/her personal interest by matching from the user profile the relevant or irrelevant samples for the query. This also justifies the introduction of "negative keywords" in user profiles. For instance, in response to the query of "dog" (see Figure 1), the user who prefers cartoon dogs will probably designate (c) as a positive example and (a) as a negative one, which cause (a) and (c) to get profiled. The next time when the user searches for "dog" again, both (c) and (a) are matched from the user profile as positive and negative example to start a pseudo feedback process. According to Equ.6, all the cartoon dogs are favored for their similarity to (c) in terms of visual content, whereas the dog photos are penalized for their similarity to (a), thereby catering for the interest of the particular user. To sum up, the common profile enables a user to "learn from others", while user profiles allow him/her to "learn from own history".

## 5. Implementation Issues

The proposed approach has been implemented in *2M2Net*, a multi-modal framework for multimedia retrieval in digital libraries or in Web. In this section, we discuss some implementation issues that are critical to the practical use of this approach. More details about *2M2Net* can be found in [16].

### 5.1. Architecture

As shown in Figure 3, a three-tier architecture is adopted in our system, which has a number of advantages over the conventional client-server architecture in terms of flexibility and extensibility.



**Figure 3: The three-tier architecture**

The client tier is an integrated user interface that can be accessed from a Web browser. It is in charge of all user-system interactions, including submission of queries and feedbacks, as well as presentation of retrieval results. The server tier is functionally a database system, which is responsible for the processing, storage, retrieval of all the available media objects. The common profile and all user profiles reside in the server tier. (The algorithm described in Section 4 is hard coded in the retrieval module of the database.) The middle tier consists of a Web Server using MS Internet Information Server (IIS), and a COM object attached to it for communication with the server tier. It acts as a proxy between the client and the server tier, by translating user queries into internal commands that can be executed by the server, as well as creating the HTML file to display the retrieved media objects in the user interface.



**Figure 4: The user interface of the client tier**

Figure 4 shows the user interface presenting the retrieved images for the query of "architecture". The top area of the pane is used for inputting queries. Below it is an image browser where retrieved images are shown as thumbnails in the descending order of their similarity to the query. Each image has a "✓" and a "x" icon attached to it that can be used to indicate positive and negative feedback examples, respectively. Other media such as text or video can be also displayed in this interface.

---

## 5.2. Safeguards for Profiles

Since the common profile is subject to the interaction of all the users, it is exposed to the attacks from malicious users. Presumably, the common profile conveys the common opinions of a majority of users. Nevertheless, according to the voting algorithm for profile learning, a user can always get a bigger “share” in the common profile by performing a great amount of identical feedbacks in a burst. By this means, the attacker can repeatedly behave opposite to most users, intending to achieve the domination of “misconceptions” in the common profile. For example, the attacker can remove a relevant keyword from the descriptor of a media object by repeatedly marking it as irrelevant to the object, or add an irrelevant keyword by doing the opposite. There is no doubt that the attacks successfully made to the common profile will severely undermine the retrieval performance of the system.

We provide two alternative solutions as the “safeguard” for the common profile. The first scheme is to restrict the number of votes each user can cast with regard to whether a specific keyword is relevant to an object. A simple implementation of this scheme is to allow each user only one vote. That means, only the first time a user says, e.g. “image A shows a dog”, causes the common profile to be updated. From then on, all the opinions from that user regarding the relevance of “dog” to image A are ignored by the common profile (but still cause the user profile updated). Note that the user can always vote on other keywords or other objects. To cope with the possibility that a user may change his/her idea, a timestamp is attached to each vote and the user is allowed to vote again regarding the same keyword and object after a certain period of time is elapsed from last vote. This mechanism can effectively prevent the attackers from “misleading” the common profile by giving frequent erroneous feedbacks.

The above scheme implies that the system must be able to automatically identify each user. We enforce a user registration procedure and require each user to login before using the system. Note that user registration is not only necessary for implementing this “safeguard” measure, but also indispensable for the personalized service provided by user profiling, which requires user identification as well. However, it may be argued that some users would not take the trouble to login every time, or would not have their private behaviors monitored by the system. A public account is therefore provided to allow users to login anonymously, but with the indication (on the interface) that the personalized service is not possible in this case. To prevent the attacks from this public account, we can restrict the number of votes from the same IP address. Admittedly, IP address is not suitable for precise user identification, but it provides a viable solution overall.

A more sophisticated safeguarding scheme is also provided, which is analogous to the concept of “versioning” in the Database research community. For each user, a local copy of the common profile is stored. All the “on-line” updates that are otherwise applied to the common profile during the retrieval process are now made to the local copy, which results in a new version of the local common profile. (For storage efficiency, we store only the changes made to the common profile in the local version). At certain intervals, the system will undergo an “off-line” process to merge all these local versions into the central common profile. Each local version will be considered

either by people or by some censoring programs to decide if the change should be made to the common profile permanently. The malicious feedbacks can be identified in the local version by some heuristics, e.g. the weight increment or decrement of a certain keyword to an object exceeds a reasonable amount. After this process, a new version of the common profile is generated and all the local versions are made consistent with it.

## 5.3. Accommodation of New Data

According to the first assumption stated in Section 1, the system is populated with new data daily. Before evaluated by any user, a new media object has no entry either in the common profile or in any user profiles. However, the retrieval algorithm of Section 4 favors the objects that have been profiled with either type of profiles. The new objects are overlooked, because their relevance to the query or any other objects is evaluated as zero. The only chance for a new object to be retrieved in top ranks, as can be seen from Equ.6, is the case when it highly resembles a positive example(s) in terms of low-level features. Hence, a so-called “vicious circle” is formed: the new objects cannot be retrieved before being profiled, and they will not be profiled before getting evaluated by the users. As a result, most new objects will stay untouched at the bottom of the database since they have been introduced.

We have devised some heuristic approaches to remedy this weakness. The first approach is to put a certain number of randomly selected new objects in each page displaying the retrieved objects. A more rational approach is to add a bonus to the similarity function of Equ.6 when a new object is encountered. Both approaches intend to “pull” some new objects into the top ranks so that they have greater opportunity to be evaluated and thus profiled, at the cost of a moderate loss in retrieval precision.

On the other hand, removal of data from the database is straightforward. When a certain object, say,  $O_i$ , is to be deleted, we deleted its corresponding descriptor  $D_i$  in the common profile, as well as its descriptor  $D_i^k$  in the user profile of each user.

## 6. Experimental Results

As our retrieval approach is Web-oriented, the experiments should be conducted in the Web environment, which may take months or even years to collect data of user behaviors. For fast evaluation of our approach, some preliminary but operational experiments are devised, which simulate the real Web with the help of Corel Image Gallery and some “virtual” users.

### 6.1. Performance of Common Profile

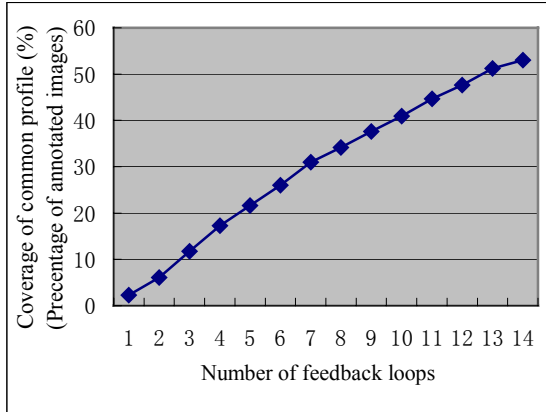
We select 5,000 images from Corel as the test data, which are pre-classified into 50 categories with exactly 100 images in each category. Each category has a title that can be used to describe all the images in this category. Therefore, if the title is used as a query, all the images within the corresponding category are regarded as the relevant results to the query.

We simulate the behaviors of real users by creating some “virtual” users, who can perform queries and feedbacks



automatically. A virtual user starts by searching for the images of a particular category in the database, using the title of the category as the query. The first 100 images with highest similarity to the query are returned by the system. The virtual user then randomly choose some images from the top 100 for evaluation, by marking it as relevant if it belongs to the intended category, or otherwise as irrelevant. Based on these user evaluations, the system updates the profiles and performs the feedback algorithm of Section 4 to improve the retrieval results. The loop of evaluation and feedback may repeat for more times. The number of images evaluated by a user in each round of feedback is set to 25 in our experiments, since a real user is unlikely to make evaluations more than that.

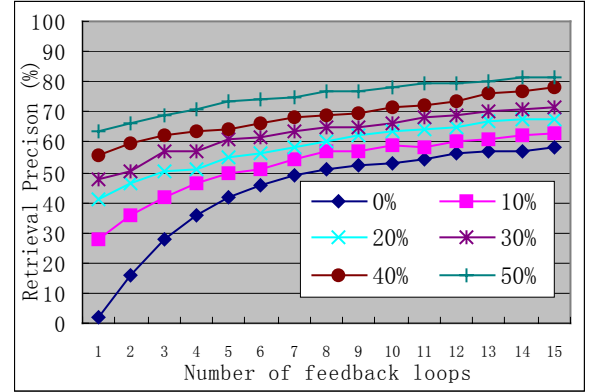
The performance of the common profile is examined on two aspects: 1) how fast it can be learned from user feedbacks, and 2) how much it helps to enhance the retrieval accuracy. In our experiment, the learning rate of the common profile is measured through its coverage in the database, interpreted as the percentage of images that are annotated with the title of the corresponding category in the common profile. To estimate this learning rate precisely, the common profile is cleared to empty and all the user profiles are disabled. Then, for each image category we activate a virtual user to execute one retrieval operation (totally 50 operations), with 14 loops of feedback in each operation. The percentage of annotated images at each loop of feedback is recorded for each category. The average profile coverage (percentage of annotated images) over 50 categories against the number of feedback loops is plot in Figure 5. As we can see, the coverage of the common profile increases steadily with the feedbacks, reaching about 50% after 12 feedback loops.



**Figure 5: Learning rate of the common profile**

The six curves shown in Figure 6 illustrate the relationship between the retrieval accuracy and the coverage of the common profile. Each curve is obtained by tracing the retrieval precision after each loop of feedback at a certain level of profile coverage. Clearly, the common profile with a large coverage greatly helps to enhance the retrieval accuracy, especially at the starting several loops of feedback. On the other aspect, higher retrieval accuracy usually encourages users to give more feedbacks, which in turn enlarges the profile coverage. Therefore, it forms a “self-reinforcing” loop between profile coverage and retrieval accuracy. The retrieval accuracy of our system is inferior to the results reported by Lu et al. [10], which evaluate the performance

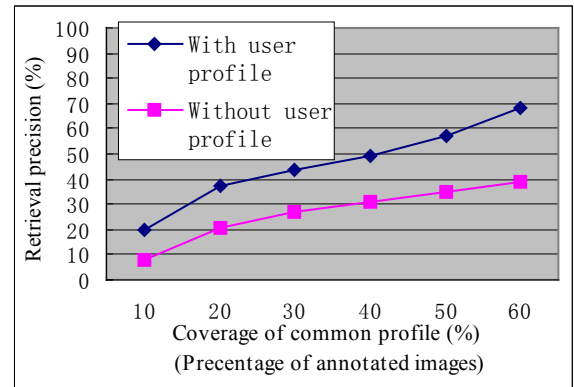
of keyword propagation in a similar experiment setting. However, we argue that the difference is due to the number of evaluations made in each round of feedback: we give only 25 evaluations, while they make 100.



**Figure 6: Retrieval precision at each level of profile coverage**

## 6.2. Performance of User Profile

The performance of user profiles is more difficult to evaluate, because the behavior of a particular user is usually unpredictable. To evaluate its performance, we use the same test data collection as in the previous experiment, but further dividing the category of “car” into four sub-categories according to the color of the car (red, black, white or yellow), with 25 images in each sub-category. We assume that a user querying by “car” targets at a specific sub-category of car, so that the retrieval precision is calculated as the percentage of the images belonging this sub-category in the top 25 images in the returned list. We compare the average precision of 8 random queries when a user profile is present to the case when it is absent, at different coverage level of the common profile. In the case of using a user profile, some images are manually “inserted” into the user profile. The result of the comparison is shown in Figure 7. The precision of using a user profile is considerably higher than that without using it, at any coverage level of the common profile.



**Figure 7: Performance of user profiles**

Unlike in Figure 6, we do not examine the change of retrieval precision during the feedback process in Figure 7. In

contrast, we focus on the precision of original queries, because a user profile is effective mainly in the original query by providing some “pseudo” feedback examples.

## 7. Conclusion

In this paper we have presented a collaborative yet personalized multimedia retrieval approach for the Web environment, which employs a synergy of incremental relevance feedback and user profiling technique. By constructing a common profile and user profiles, this approach allows to make a balance between exploiting the common knowledge of most users and catering for the interest of a particular user. We also provide viable solutions to several tricky problems regarding the practical use of this approach. The effectiveness of this approach is confirmed by some preliminary experiments.

In our preliminary experiments, the system performance under the real Web environment is yet to be studied. As our future work, some Web crawlers will be used to populate our database with multimedia objects grabbed from web pages, and the client side will be published to the access of real users. Therefore, we can examine the system performance in an environment where real users interact with real data from Web.

Another interesting line of research is to adopt linguistic approaches to improve the measurement of similarity between multimedia objects and queries. A well-studied work along this direction is using a thesaurus-based semantic metric that can address the relevancy between different keywords. Since the keyword matching is heavily used in our approach, an improvement in this direction seems to be critical and rewarding.

A more ambitious future work is to categorize users into different groups based on their behaviors in the interaction with the systems, e.g., the “children” and “artists” in the example of Figure 1. This work usually requires better knowledge about user behavior and more sophisticated user model, as well as large amount of training data about interaction, which are not currently accessible to us. A good user classification allows more intelligent and efficient retrieval algorithm with better personalization, since the users’ opinion can be deduced from his/her user group instead of learning from the former behaviors.

## Reference

1. Balabanovic, M., Shoham, Y., “Fab: content-based collaborative recommendation”, *Communication of the ACM*, 40(3), pp. 66-72, 1997.
2. Chen, L., Sycara, K., “WebMate: A Personal Agent for Browsing and Searching”, in *Proceedings of the 2<sup>nd</sup> International Conference on Autonomous Agents and Multi Agent Systems*, ACM, pp. 132 – 139, 1998.
3. Cox, I.J., Miller, M. L., Minka, T. P., Papathornas, T. V., Yianilos, P. N., “The Bayesian Image Retrieval System. PicHunter: Theory, Implementation, and Psychophysical Experiments”, *IEEE Trans. On Image Processing*, Vol 9, Issue 1, pp. 20-37, Jan, 2000.
4. Ditto.com: [www.ditto.com](http://www.ditto.com)
5. Ishikawa, Y., Subramanya, R., Faloutsos, C., “MindReader: Query Databases Through Multiple Examples”, in *Proc. of the 24<sup>th</sup> VLDB Conference*, NY, USA, 1998.
6. Joachims, T., Freitag, D., Mitchell, T., “WebWatcher: A Tour Guide for the World Wide Web”, in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 770-775, Japan, 1997.
7. Lashkari, Y., The WebHound Personalized Document Filtering System <http://rg.media.mit.edu/projects/webhound>
8. Lee, C., Ma, W. Y., Zhang, H. J., “Information Embedding Based on User’s Relevance Feedback for Image Retrieval”, Technical Report, HP Labs, 1998.
9. Ioannou, S., Moschovitis, G., Ntalinais, K., Karpouzis, K., Kollias, S., “Effective Access to Large Audiovisual Assets Based on User Preferences”, *ACM Multimedia*, 2000.
10. Lu, Y., Hu, C.H., Zhu, X.Q., Zhang, H.J., Yang, Q. ”A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems”, *ACM Multimedia*, 2000.
11. Pazzani, M., Muramatsu, J., Billsus, D., “Syskill & Websert: Identifying interesting web sites”, in *Proceeding of the National Conference on Artificial Intelligence*, Portland, USA, 1998.
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., “GroupLens: An open architecture for collaborative filtering of netenws”, in *Proceeding of the ACM Conf. on Computer-Supported Cooperative Work*, pp. 313-323, NJ, USA, 1997.
13. Rocchio, J.J. ”Relevance Feedback in Information Retrieval”. In *The SMART Retrieval System*, pp. 313-323. Prentice Hall, 1971.
14. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S., “Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval”, *IEEE Trans on Circuits and Systems for Video Technology*, Special Issue on Segmentation, Description, and Retrieval of Video Content, Vol 8, pp 644-655, 1998.
15. Salton, G., Buckley, C. “Introduction to Modern Information Retrieval”, McGraw-Hill Book Company, New York, 1982.
16. Yang, J., Zhuang, Y. T., Li, Q., “Search for Multi-Modality Data in Digital Libraries”, to be presented in the *The 2<sup>nd</sup> IEEE Pacific-Rim Conference on Multimedia*, Beijing, China, 2001.