Multimed Tools Appl (2010) 50:563–585
DOI 10.1007/s11042-010-0480-8

A multimedia recommender integrating object features and user behavior

Massimiliano Albanese · Angelo Chianese · Antonio d'Acierno · Vincenzo Moscato · Antonio Picariello

Published online: 28 January 2010

Abstract Despite the great amount of work done in the last decade, retrieving information of interest from a large multimedia repository still remains an open issue. In this paper, we propose an intelligent browsing system based on a novel recommendation paradigm. Our approach combines usage patters with low-level features and semantic descriptors in order to predict users' behavior and provide effective recommendations. The proposed paradigm is very general and can be applied to any type of multimedia data. In order to make the recommender system even more flexible, we introduce the concept of multichannel browser, i.e. a browser that allows concurrent browsing of multiple media channels. We implemented a prototype of the proposed system and tested the effectiveness of our approach in a virtual museum scenario. Experimental results have proved that the system greatly enhances users' experience, thus encouraging further research in this direction.

Keywords Recommender systems · Browsing · Information retrieval · Multimedia databases

M. Albanese (✉)
UMIACS, University of Maryland, College Park, MD 20742, USA
e-mail: albanese@umiacs.umd.edu

A. Chianese · V. Moscato · A. Picariello
DIS, University of Naples "Federico II", Via Claudio, 21, 80125 Naples, Italy

A. Chianese
e-mail: angelo.chianese@unina.it

V. Moscato
e-mail: vmoscato@unina.it

A. Picariello
e-mail: antonio.picariello@unina.it

A. d'Acierno
ISA, CNR, Via Roma 64, Avellino 83100, Italy
e-mail: dacierno.a@isa.cnr.it

1 Introduction

Due to the enormous progress in consumer electronics and the widespread availability of internet access, today's society is able to produce and share digital data—text, audio, images, and video—at an unprecedented rate. In order to facilitate browsing of large multimedia repositories, a number of algorithms and tools are being proposed. Such tools, usually referred to as "Recommender Systems", collect and analyze usage data in order to determine users' interests and preferences, and thus provide them with useful recommendations. It is the author's opinion that the problem can be effectively addressed by considering both the users' browsing behavior and the specific features of the multimedia objects that users explicitly access during a browsing session. Experimental results reported in this paper have confirmed this intuition.

Although a huge amount of work has been done in the field of content based multimedia retrieval, no significant effort has been devoted to the problem of intelligent browsing of multimedia collections. In [13], the authors provide a comprehensive survey of state of the art in Multimedia Information Retrieval and identify the major research challenges for the future, namely: (1) semantic search with emphasis on the detection of concepts in media with complex backgrounds; (2) multimodal analysis and retrieval algorithms especially to exploit the synergy between the various media, including text and context information; (3) experiential multimedia exploration systems to allow users to gain insight and explore media collections; (4) interactive search, emergent semantics, or relevance feedback systems; and (5) evaluation with emphasis on representative test sets and usage patterns.

Our work is a first important step towards addressing these challenges. In this paper we present an intelligent browsing system based on a recommendation paradigm that takes into account both a user's browsing behavior and low-level and semantic multimedia descriptors. A combined analysis of all these aspects enables the system to generate recommendations based on the expected preferences of each user.

Browsing systems have received significant attention especially in the video realm, and methods have been developed for presenting video content by hierarchical video shot clustering [25], and video storyboards [24]. Traditional browsing systems allow a user to rapidly browse through a multimedia sequence, navigate from one segment to another, and then either get a quick overview of multimedia content or zoom to different levels of detail to locate segments of interest. However, these techniques fail either in detecting semantically related units for browsing or in integrating efficient multimedia retrieval. Other approaches [26] have tried to overcome these limitations, but they failed to integrate multimodal analysis and retrieval, and focused on a single media type.

Users of a multimedia browsing and retrieval system should be able to navigate a repository of multimedia objects in a semantics-driven fashion, rather than by media type. For instance, a user watching a documentary on Shakespear might also be interested in reading one of his poems, therefore an effective multimedia browsing system should be designed to provide recommendations across media types.

In this paper, we address this issue by introducing the concept of multichannel browser, i.e., a browser that allows a user to browse multiple media channels concurrently. The recommender can then offer suggestions for each channel based on past users' behavior and features of objects displayed on all channels. We leverage the

work proposed in [2] and generalize the approach to handle multiple media types in a uniform way. Our work significantly differs from previous works, as it combines usage patterns, low-level object features and semantic descriptors in a novel approach to recommendation. In addiction, our system does not rely on explicit login—which typically discourages the users from accessing a web site—so a returning user starting a new session is considered as a new user. Such requirement makes traditional collaborative recommendation techniques inapplicable.

The paper is organized as follows. Section 2 discusses related work, whereas Section 3 introduces a motivating example that we will use throughout the remainder of the paper. Section 4 discusses our approach to evaluating object similarity and Section 5 presents the core of our work, i.e. the recommendation algorithm. Details about the implementation and tuning of the system are provided in Section 6. Finally, experimental results are reported in Section 7, and concluding remarks are given in Section 8.

2 Related work

Recommender systems for multimedia objects broadly fall into two classes.[1] Of course, many hybrid solutions also exist, as illustrated at the end of this section.

In *content based filtering* [18], the utility of an item s for a user is estimated based on the utility assigned by the *same* user to other items that are similar to s [1]. This approach is heavily based on information retrieval and information filtering. The improvement over traditional approaches comes mainly from the use of user profiles, which contain information about users' preferences. Profiling can be realized explicitly (through, for example, questionnaires) or implicitly (i.e., learned from the users' behavior over time). In [15], a method based on ontologies is used for ranking relevancy in the electronic papers domain while, in [9], content based filtering has been applied to music data using decision trees. A multi- criteria based system is proposed in [17]. The drawback of these techniques in our context is that they do not benefit from the great amount of information that could be derived by analyzing the behavior of other users.

Collaborative filtering is a good alternative to content based strategies. The main idea is to associate the current user to a set of other users having, in some way, *similar* profiles. In this way, data items are recommended based on the similarity between users, rather than on the similarity between data items themselves. Wang et al. [23] presents a probabilistic user-to-item relevance framework which introduces the concept of *relevance* and derives three different models: the *user based* model, the *item based* model and the *unified relevance* model. Collaborative filtering has been also used to build a prototype movie search and browsing engine called MAD6 [16]. In [22], several collaborative algorithms have been fused in a system that also takes into account metadata as additional knowledge . One of the main drawbacks of this techniques is the delay in considering a newly introduced data item as a candidate for recommendation: a new data item will in fact become available for recommendation only when enough users have seen and rated it. Besides, if a new user is not similar

[1] Here we do not consider link-based systems mainly used in WEB search engines.

enough to any of the previous and known users, it will not be possible to make reliable recommendations.

Content based filtering and collaborative filtering could be profitably combined to improve the effectiveness of the recommendation process [20]. In [4], the authors present a unified approach for learning a prediction function that systematically integrates all available training information, such as past user-item ratings, data item attributes and users' attributes. In [3], an recommendation approach for integrating user rating vectors with an ontology is described. Finally, [10] presents a system based on collaborative filtering that uses content based information to address the cold-start problem (giving recommendations to novel users who have no preference on any items, or recommending items that no user of the community has seen yet). More recently, a hybrid approach based on content-based and collaborative filtering, implemented in MoRe, a movie recommendation system, has been described in [11].

3 Motivating example

In this section we present a typical scenario where an effective multimedia recommender system would be desirable. We will refer to this example throughout the rest of the paper, and we will also describe a prototypal implementation of our system applied to such scenario.

We consider the case of a *virtual museum*, i.e. a museum that offers a web-based access to a multimedia collection of digital reproductions of paintings, educational videos and text documents. In order to make the user's experience in the museum more interesting and stimulating, the access to information should be customized based on the specific profile of a visitor, which includes learning needs, level of expertise and personal preferences. Fayzullin et al. [7] presents a system that assists visitors to an archeological site by delivering them highly customized stories about the subjects of paintings and statues across the site. The authors emphasize the importance of tailoring information to the specific needs of a user in this class of systems.

Let us consider users visiting a virtual museum and suppose that they request, at the beginning of their tour, some paintings depicting *imaginary* landscapes. While observing such paintings, they are attracted, for example, by a Peter Paul Rubens' painting entitled *Landscapes with the ruins on the Palatine Hill in Rome* (Fig. 1a). It would be helpful if the system could learn the preferences of the users, based on these first interactions, and predict their future needs, by suggesting other paintings (or any

a b c

Fig. 1 Paintings depicting landscapes

other multimedia objects) representing the same or related subjects, depicted by the same or other related authors, or items that have been requested by users with similar preferences. As an example, a user who is currently observing the Rubens' painting in Fig. 1a might be recommended to see a Nicolas Poussin's painting entitled *Landscape in the Roman Campagna* (Fig. 1b), that is quite similar to the current picture in terms of color and texture, and *Italian landscape—Early seventeenth century* by William Van Nieulandt (Fig. 1c), that is not similar in terms of low level features but is similar in terms of semantic content.

From the user perspective there is the advantage of having a guide suggesting artifacts which the users might be interested in, whereas, *from the system perspective*, there is the undoubted advantage of using the suggestions for pre-fetching and caching the objects that are more likely to be requested.

4 Object similarity

A key-element in the design of an effective multimedia recommender system is the definition of similarity metrics to compare multimedia objects, exploiting both low and high level features. The object comparison strategy we adopt in this work is based on combining results from low-level multimedia processing and semantic annotations of objects. Without loss of generality, we will describe such a strategy with respect to images.

In the literature, content based similarity of images has been well investigated. Images have been usually characterized through three fundamental low-level features, namely color, texture and shape [21]. Image processing algorithms can automatically extract these features and compute the distance between two images as the distance between their features in the feature space. In the prototypal implementation of our recommender system, we adopted distance function $\delta_{\mathcal{F}}$ included in the *Oracle Intermedia* extension of the Oracle DBMS, which exploits color, texture, shape and spatial information of images.

As for the high-level features, different solutions have been proposed to automatically map low-level features to semantic concepts and to compare different sets of annotations using some form of background knowledge, represented for example through an ontology. Without loss of generality, we will assume that semantic annotations of objects have been manually generated by human experts based on taxonomies. A taxonomy $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ is a hierarchical concept network, where a node $n \in \mathcal{N}$ in the hierarchy represents a concept and an edge $e \in \mathcal{E}$ represents a parent/child relationship between two concepts. This assumption is perfectly reasonable in our virtual museum scenario, where we expect that each object in the collection has been manually classified and tagged by human experts.

We can now formalize the concept of semantic annotation of an object and define a metric to compare objects based on their annotations.

Definition 1 (Annotation Schema) Given a taxonomy $\mathcal{T} = (\mathcal{N}, \mathcal{E})$, an *Annotation Schema* is a tuple

$$\Lambda_{\mathcal{T}} = (A_1, \ldots, A_n, B_1, \ldots, B_m) \tag{1}$$

where $A_1, \ldots, A_n$ are attributes s.t. $\forall i \in [1, n]$ $dom(A_i) \subseteq \mathcal{N}$ (i.e. A_i assumes values corresponding to nodes of $\mathcal{T}$), and $B_1, \ldots, B_m$ are attributes s.t. $\forall j \in [1, m]$ $dom(B_j) \nsubseteq \mathcal{N}$ (i.e. B_j does not assume values corresponding to nodes of $\mathcal{T}$)

In other words attributes $A_1, \ldots, A_n$ (taxonomic attributes) correspond to concepts that are relevant for the specific domain being modeled. Under particular circumstances a conceptual data model can be mapped into a taxonomy whose nodes are the instances of the concepts in the data model [13].

Definition 2 (Semantic Annotation) Given a taxonomy $\mathcal{T}$, an annotation schema $\Lambda_{\mathcal{T}}$ and an object O, a *Semantic Annotation* of O is a tuple

$$\Lambda_{\mathcal{T}}(O) = \left(v_1^A, \ldots, v_n^A, v_1^B, \ldots, v_m^B\right) \tag{2}$$

where $\forall i \in [1, n]$ $v_i^A \in dom(A_i)$ and $\forall j \in [1, m]$ $v_j^B \in dom(B_j)$.

Now we want to define a metric that evaluates the distance between two objects in terms of their semantic annotation. We start from the assumption that, given a taxonomic attribute A_k, the similarity between objects O_i and O_j, as discussed in [14], is inversely proportional to the length of the path between the respective values of A_k and directly proportional to the depth into the hierarchy of their subsumer. We can thus define the taxonomic distance as follows.

Definition 3 (Taxonomic Distance) Given a taxonomy $\mathcal{T}$ and an annotation schema $\Lambda_{\mathcal{T}} = (A_1, \ldots, A_n, B_1, \ldots, B_m)$, the *Taxonomic Distance* between two objects O_i and O_j is defined as

$$\delta_{\mathcal{T}}(O_i, O_j) = 1 - \frac{1}{n} \cdot \sum_{k=1}^{n} e^{-\alpha \cdot l\left(A_k^i, A_k^j\right)} \cdot \left(1 - e^{-\beta \cdot d\left(A_k^i, A_k^j\right)}\right) \tag{3}$$

where A_k^i and A_k^j are the values of attribute A_k for O_i and O_j respectively, $l\left(A_k^i, A_k^j\right)$ is the path length between A_k^i and A_k^j and $d\left(A_k^i, A_k^j\right)$ is the depth in the hierarchy of the subsumer of A_k^i and A_k^j; α and β are parameters scaling the contribution of shortest path length and depth respectively.

We remark that Eq. 3 does not take into account the attributes $B_1, \ldots, B_m$ for evaluating the similarity between objects. The values of these attributes are not represented into the taxonomy, thus it is not possible to establish any relation between them. In the case of the virtual museum scenario, we assume the availability of a taxonomy that represents the concepts of painters, pictorial genres and depicted subjects. Thus, we can assume $n = 3$, $m = 2$, and $\Lambda_{\mathcal{T}} = (A_1, A_2, A_3, B_1, B_2) = (Author, Genre, Subject, Title, Date)$.[2] Based on the above discussion we can conclude that, the closer authors, genres and subjects are, the more similar the paintings are.

[2] Ad-hoc metrics could be defined to evaluate the similarity of non-taxonomic attributes. Without loss of generality, we omit further discussions on this topic.

The distance metric we adopt in our system is a combination of feature-based and taxonomic distances, as defined below.

Definition 4 (Distance) The *Distance* between two objects O_i and O_j is defined as:

$$\delta(O_i, O_j) = \alpha_{\mathcal{F}} \cdot \delta_{\mathcal{F}}(O_i, O_j) + \alpha_{\mathcal{T}} \cdot \delta_{\mathcal{T}}(O_i, O_j) \tag{4}$$

where $\alpha_{\mathcal{F}}$ and $\alpha_{\mathcal{T}}$ are two weighting factors.

Note that, in order to ensure the scalability of the system w.r.t. high volumes of data, different indexing strategies could be adopted; in the current implementation, we have chosen to index multimedia objects using *M-Trees* [5] and the distance metric defined above.

5 Recommendation algorithm

This section presents the core of the proposed multimedia recommender system, which expands the work presented in [2]. In the following we provide some preliminary definitions, including the definition of Multichannel browser. We then introduce the concept of usage pattern and illustrate how usage patterns can be used to generate recommendations.

Definition 5 (Multichannel Browser) Given a set $\mathcal{M}$ of media types (image, video, audio, text), a *h-channel Browser* $\mathcal{B}_h$ is a h-ple $(ch_1, \ldots, ch_h)$, where $\forall j \in [1, h]\ ch_j \in \mathcal{M}$.

In other words, a multichannel browser is a browser which allows concurrent browsing of h channels, each assigned to a specific media type.

Definition 6 (Multichannel Object) Given a Multichannel Browser $\mathcal{B}_h$, a *h-channel object* O^h is a h-ple $(O_1, \ldots, O_h)$, such that $\forall j \in [1, h]\ O_j$ is an object of media type ch_j. Let $O^h[j]$ denote $O_j, \forall j \in [1, h]$ and let $\mathcal{O}^h$ denote the set of all h-channel objects.

Intuitively, each h-channel object is a snapshot of what is being displayed on a h-channel browser at a given time. For the sake of brevity, in the following we will refer to Multichannel objects as m-objects, whereas we will use the term objects to denote the component objects of a m-object. We will often abuse notation when $h = 1$ (single channel), and use O to refer both to an m-object and to its only component object.

The techniques described in Section 4 would enable a browsing system to provide users with recommendations based solely on the objects that they are currently watching on the several channels of the multimedia browser. For example, the system may suggest a user to watch the pictures that are most similar to the picture currently displayed on the image channel.

In this section we describe how to augment a recommender system by taking into account past behavior of other users, in accordance with the idea that *personalization* is the *process of customizing the content and the structure of an application* in order

to provide users with the information they are interested in, without asking for it explicitly [6].

The intuition behind our approach is that, if we can predict what objects a user is likely to request next and use such predictions as recommendations, it is very likely that the user will accept one of the recommendations, rather than jumping to entirely unrelated objects or starting a new browsing session altogether. Experimental results have confirmed this intuition.

In the following we propose an algorithm for predicting user behavior based on the concept of *usage patterns*, which is defined below.

Definition 7 (Usage Pattern) Given a Multichannel Multimedia Browser $\mathcal{B}_h$, a *usage pattern* P_i^h of length k is an ordered sequence of k m-objects visualized by a user in the same browsing session i:

$$P_i^h = \left(O_{i_1}^h, O_{i_2}^h, ..., O_{i_k}^h\right), \; with \; O_{i_j}^h \in \; \mathcal{O}^h \; \forall \; j \; \in [1, k] \tag{5}$$

Let $\mathcal{P}^h$ be the set of all the usage patterns of past users of a h-channel browser.

Note that $P_i^h[j]$ denotes the j-th m-object $O_{i_j}^h$ in P_i^h and $P_i^h[j][k]$ denotes the k-th component object of $P_i^h[j]$.

We are interested in dynamically classifying the behavior of a new user (e.g. a user visiting an online virtual museum). We remind the reader that our system does not require explicit login, so a returning user starting a new session is considered as a new user.

Our approach to recommendation consists in finding the patterns in $\mathcal{P}^h$ that best match the current usage pattern and making suggestions based on what users corresponding to those patterns have done in the past. Therefore, we are interested in the notion of similarity between usage patterns. Several algorithms have been proposed to compare sequences of symbols from a given alphabet Σ and evaluate their similarity or their distance.

A well-known algorithm in this field is the Levenshtein algorithm [12], that was designed to evaluate the distance between two words as the total cost of the basic operations (insertions, deletions and substitutions) needed to transform a string into the other. The Levenshtein distance gives a measure of how much two sequences of symbols differ in terms of alignment, without taking into account the nature of the symbols themselves: the cost of substituting a symbol a with a symbol $b \neq a$ is fixed and does not depend on the specific nature of a and b. Intuitively, one would expect that replacing a consonant with a vowel should have a higher cost than replacing a consonant with another consonant. Similarly the cost of deleting or inserting a symbol a is fixed and does not depend on the specific nature of a.

Example 1 (Similarity of Usage Patterns) W.r.t. the example in Fig. 2, let us assume that $h = 1$ (single channel browser) and consider the usage patterns $P_1 = (O_1, O_2, O_4, O_5)$ and $P_2 = (O_1, O_7, O_4, O_6)$. The Levenshtein distance between P_1 and P_2 is equal to 2. If we consider a generic pattern $P_x = (O_1, O_x, O_4, O_5)$, the Levenshtein distance between P_1 and P_x is equal to 1, independently of the specific features of object O_x, whereas we might expect that such distance depends on the distance between O_2 and O_x.

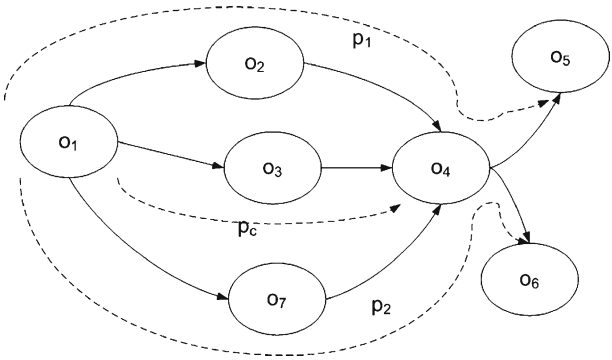


Fig. 2 An example of usage patterns

The idea behind our approach is to evaluate the similarity between patterns based on the similarity between the objects in the patterns. To this aim, we use the similarity metrics defined in Section 4 and we adopt an indexing strategy to guarantee fast access to objects and patterns of interest. We remind the reader that our approach does not rely on any a-priori knowledge of the users, therefore we need to learn their preferences in real time, as they browse the multimedia collection. The length of a usage pattern starts at zero and then increases by one unit every time the user requests a new item from the collection. For this reason, comparing the current usage pattern with full patterns in the usage log might not be effective. Instead, a measure of *local similarity* between patterns can provide better results. In other words, we are interested in finding those patterns containing subsequences that match the current pattern in an optimal way and then make suggestions based on them.

Starting from the Levenshtein theory, we have designed an algorithm that evaluates the local similarity between usage patterns, taking into account the features of the objects in them. Given two usage patterns P_1 and P_2, the algorithm computes a matrix Ω whose (i, j) element represents the maximum local similarity between two patterns, respectively containing the first i elements of P_1 and the first j elements of P_2. The highest value in Ω is the overall local similarity between P_1 and P_2 and corresponds to the best local alignment between those patterns.

Example 2 (Local Similarity of Usage Patterns) W.r.t. the example in Fig. 2, let us assume that $h = 1$ (single channel browser) and suppose that the partial usage pattern of a user who is currently browsing the collection is $P_c = (O_1, O_3, O_4)$. Also, assume that $P_1 = (O_1, O_2, O_4, O_5)$ and $P_2 = (O_1, O_7, O_4, O_6)$ are the patterns in the log containing the subsequences that optimally match P_c, i.e. local similarity between P_c and any of P_1, P_2 is high and above a given threshold. Based on P_1 and P_2, it is likely that the current user may be interested in either O_5 or O_6, as these two objects were requested right after O_4 by users with similar local behavior. Therefore, the system can recommend objects O_5 and O_6, ranking them on the basis of how much O_2 and O_7 are similar to O_3.

Definition 8 introduces the functions used to compute the cost of an alignment in terms of substitutions, insertions and deletions.

Definition 8 (Cost Functions) Let $P_1^h = (O_{k_1}^h, ..., O_{k_m}^h)$ and $P_2^h = (O_{l_1}^h, ..., O_{l_n}^h)$ be two patterns of length m and n respectively. We define the substitution, insertion and deletion cost functions as follows:

$$Sub\left(P_1^h[i], P_2^h[j]\right) = \sum_{c\in[1,h]} \frac{\tau - \chi_c\left(O_{k_i}^h[c], O_{l_j}^h[c]\right)}{1-\tau} \tag{6}$$

$$Ins\left(P_2^h[j], P_1^h[i]\right) = \sum_{c\in[1,h]} \frac{\tau - \min\left\{\chi_c\left(O_{k_i}^h, O_{l_j}^h\right), \chi_c\left(O_{k_{i+1}}^h, O_{l_j}^h\right)\right\}}{1-\tau} \tag{7}$$

$$Del\left(P_1^h[i], P_2^h[j]\right) = Ins\left(P_1^h[i], P_2^h[j]\right) \tag{8}$$

where $\chi_c = 1 - \delta_c$ is a similarity metric defined on the media type of channel c and $\tau \in [0, 1]$ is a threshold.

$Sub\left(P_1^h[i], P_2^h[j]\right)$ is the cost of replacing the *i-th* element of P_1^h with the *j-th* element of P_2^h, and it is computed as the sum of the costs of replacing each component of a m-object. Note that when the similarity between two corresponding objects in a given channel is equal to the threshold, the contribution of the channel to the overall cost is 0; when the similarity is above the threshold, the contribution of the channel is negative, meaning that it reduces the overall cost, actually rewarding the substitution. Similarly, $Ins\left(P_2^h[j], P_1^h[i]\right)$ is the cost of inserting the *j-th* element of P_2^h after the *i-th* element of P_1^h and $Del\left(P_1^h[i], P_2^h[j]\right)$ is the cost of deleting the *i-th* element of P_1^h, j being the position of the element in P_2^h aligned with $P_1^h[i-1]$. The threshold τ has been defined as a function of the size of the collection, by posing $\tau = (\lg|\mathcal{O}^h| - 0.4)/\lg|\mathcal{O}^h|$. For example, $\tau = 0.8$ when $|\mathcal{O}^h| = 100$ and $\tau = 0.9$ when $|\mathcal{O}^h| = 10{,}000$.

Figure 3 lists the algorithm used for the evaluation of local user similarity between patterns. Given an alignment, the algorithm assigns a positive score (negative cost) to each substitution of an element $O_{k_i}^h$ of P_1^h with an element $O_{l_j}^h$ of P_2^h that is similar

```
function local-similarity(P_1^h, P_2^h)
    P_1^h and P_2^h are two patterns of length m and n respectively
    Ω is a two-dimensional array with m + 1 rows and n + 1 columns
    for j ← 0 to n do
        Ω[0, j] ← 0
    end for
    for i ← 0 to m − 1 do
        Ω[i + 1, 0] ← 0
        for j ← 0 to n − 1 do
            Ω[i + 1, j + 1] ← max{0,
                                  Ω[i, j] − Sub(P_1^h[i], P_2^h[j]),
                                  Ω[i, j + 1] − Del(P_1^h[i], P_2^h[j]),
                                  Ω[i + 1, j] − Ins(P_2^h[j], P_1^h[i])}
        end for
    end for
return max_{i,j}{Ω[i, j]}/ min{m, n}
```

Fig. 3 Algorithm for evaluating the local user similarity

to $O^h_{k_i}$, with similarity above the threshold τ. Vice versa a negative score is assigned to each substitution where the similarity is below the threshold. In both cases the absolute value of the score is proportional to the similarity measure between the two objects. In a similar way the insertion of an element $O^h_{l_j}$ of P^h_2 between elements $O^h_{k_i}$ and $O^h_{k_{i+1}}$ of P^h_1 is penalized by an amount that is greater when the new element is dissimilar from both $O^h_{k_i}$ and $O^h_{k_{i+1}}$. In the following we define a measure of the similarity between m-objects that is latent in the usage patterns, in the sense that usage patterns capture choices made by users based on the perceived relationship (visual or semantic) between different objects. To this aim we need to define the following sets:

$$\mathcal{P}^h_\gamma = \left\{ P^h \in \mathcal{P}^h \mid \text{local-similarity}\left(P^h, P^h_c\right) \geq \gamma \right\} \tag{9}$$

$$\mathcal{O}^h_\gamma = \left\{ O^h \in \mathcal{O}^h \mid \exists P^h \in \mathcal{P}^h_\gamma, \text{next}_{P^h}\left(P^h_c\right) = O^h \right\} \tag{10}$$

$\mathcal{P}^h_\gamma$ is the set[3] of all the patterns in the log that are similar to the current pattern P^h_c within a threshold γ, while $\mathcal{O}^h_\gamma$ is the set of those objects that users corresponding to the patterns in $\mathcal{P}^h_\gamma$ have seen after the subsequence aligned with P^h_c. Let us now define the following sets:

$$\mathcal{O}^h_c = \mathcal{O}^h_\gamma \cup \mathsf{NN}(O^h_c, k) \tag{11}$$

$$\mathcal{P}^h_i = \left\{ P^h \in \mathcal{P}^h_\gamma \mid \text{next}_{\mathcal{P}}(\mathcal{P}_c) = \mathcal{O}_i \right\}, \forall \mathcal{O}_i \in \mathcal{O}^h_c \tag{12}$$

where $\mathsf{NN}(O^h_c)$ selects the k nearest neighbors of the current m-object O^h_c being visualized by the user. $\mathcal{O}^h_c$ is the set of candidate objects for inclusion in the recommendation list, while $\mathcal{P}^h_i$ is the subset of $\mathcal{P}^h_\gamma$ containing those patterns having O^h_i as the first element following the subsequence aligned to P^h_c.

The threshold γ is needed because we want to base recommendations on patterns that are highly similar to the current pattern. Moreover, considering only a subset of $\mathcal{P}^h$ reduces the complexity of the algorithm. The threshold γ should be close enough to 1 in order to get high precision results and it should be higher when the size of the log increases. We have chosen $\gamma = (|\mathcal{P}^h| - 0.2)/|\mathcal{P}^h|$.

Definition 9 (Implicit Similarity) The *Implicit Similarity* χ_P between a m-object O^h_i and a current usage pattern P^h_c is defined as

$$\chi_P\left(O^h_i, P^h_c\right) = \frac{\sum_{P^h \in \mathcal{P}^h_i} \text{local-similarity}\left(P^h, P^h_c\right)}{\max_i \left\{ \sum_{P^h \in \mathcal{P}^h_i} \text{local-similarity}\left(P^h, P^h_c\right) \right\}} \tag{13}$$

$\max_i \left\{ \sum_{P^h \in \mathcal{P}^h_i} \text{local-similarity}\left(P^h, P^h_c\right) \right\}$ being a normalization factor.

We can finally define how to build a ranked list of recommendations. The idea is to weight both the similarity w.r.t. the last requested object and the similarity in terms of usage patterns. In fact, when a user starts browsing the collection, her current pattern is too short to make useful recommendations based on usage patterns only. In this case, it would be useful to take into account the features of the last requested

[3]We will discuss in Section 6.2 how to build this set.

object and recommend the objects most similar to it. Let us introduce the following definition.

Definition 10 (Recommendation grade) Given the current pattern P_c^h and the last element O_c^h in P_c^h, the recommendation grade ρ of an object O_i^h is defined as:

$$\rho\left(O_i^h\right) = \alpha_c \cdot \frac{1}{h} \sum_{c\in[1,h]} \chi_c\left(O_i^h[c], O_c^h[c]\right) + \alpha_P \cdot \chi_{\mathcal{P}}\left(O_i^h, P_c^h\right) \tag{14}$$

α_c and α_P being two weighting factors.

In conclusion, the system will recommend the k m-objects in $\mathcal{O}_c^h$ exhibiting the higher values of ρ.

6 Implementation

In this section we address some fundamental implementation issues. In particular, we discuss in more details the architecture of our system, how to tune the system by setting the several parameters we have introduced, and how to make our solution scalable.

6.1 System architecture

Figure 4 shows at a glance the overall architecture of the system. The front end of the system—the multichannel browser—is implemented as a web application, therefore users can access the system through a common web browser. As a user explore the multimedia collection, the *Usage Log* records which items she requests and in which order. At the same time, the *Pattern Discovery Subsystem*, based on the behavior of past users and the metrics discussed in the previous sections, tries to classify the user and predict her future behavior.

As anticipated in the introduction, we do not use explicit login since it typically discourages the users from accessing a web site, even if the site is regarded as interesting.[4] Therefore, the precision of user classification, being exclusively based on her dynamic behavior during a single browsing session, is quite poor when the user first starts using the system and then it improves as she continues to explore the collection.

The *Recommendation Subsystem*, based on the current knowledge of the user and on the item that she is currently observing, returns a ranked list of suggested items.

Due to the large amount of data involved, we chose to implement a prototype of our browsing system using *ORACLE technologies* (Oracle Application Server, Oracle 10g DBMS, Oracle Intermedia, Oracle Text, PL/SQL Stored Procedures, PSP Server Pages).

[4]We use cookies to track sessions, and don't set an expiration date, so they will be deleted when the browser session ends. In this way, different users browsing the collection from a shared computer (e.g., in a public library) will not misinterpreted as the same user.

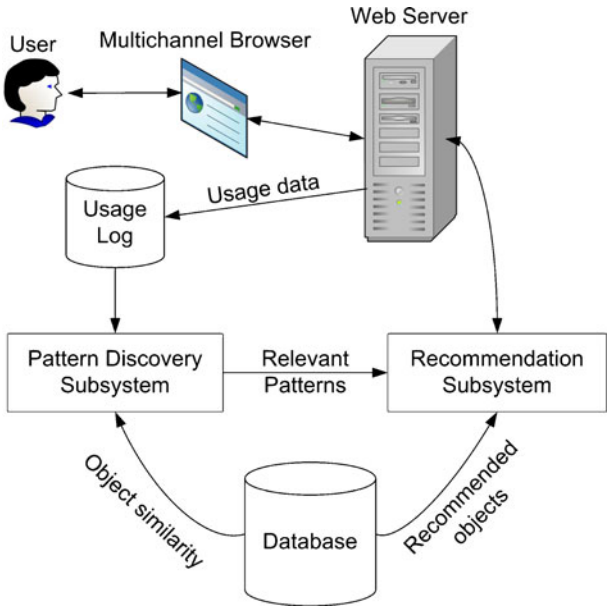


Fig. 4 System architecture

With respect to the issue of computing distance metrics, we adopted *Oracle Intermedia* to compute feature-based distance between images. Ad-hoc PL/SQL procedures were created to implement the taxonomic distance, the distance metric, the local-similarity algorithm and the M-tree indexing strategy.

6.2 System tuning

Several parameters have been introduced along the paper for weighting the contribution of different factors. In this section we discuss the strategy used to select good values for these parameters.

A signature based distance is usually an attempt to reproduce human behavior when assessing the similarity or dissimilarity of two visual stimuli. During this process each perceived feature of the stimulus is implicitly assigned a different weight. We tried to estimate such weights by means of the following experiment.

We selected about 100 pictorial images and asked a group of about 40 people[5] to judge the similarity—only in terms of visual appearance – between these images on a 1 to 10 scale. We then determined the values of the factors α_{color}, $\alpha_{texture}$, α_{shape}, and $\alpha_{location}$—used to weight the different features analyzed by Oracle Intermedia—that maximized the correlation between the average values of human judged similarity and the values of $\chi_{\mathcal{F}} = 1 - \delta_{\mathcal{F}}$. In conclusion, we obtained $\alpha_{color} = 0.3$, $\alpha_{texture} = 0.2$, $\alpha_{shape} = 0.3$ and $\alpha_{location} = 0.3$.

In the definition of Taxonomic Distance (Eq. 3), two parameters γ and β are used to scale the contribution of shortest path length and depth respectively, by tuning

[5]The people involved in the experiments were mainly students from the University of Naples, Italy.

the slope of the two exponential curves. Li et al. [14], who defined an approach for measuring semantic similarity between words, proposed to evaluate such parameters by maximizing the correlation with human similarity judgements, as in the very first experiments by Rubenstein-Goodenough and Miller-Charles. They tested several similarity metrics on a standard set of word pairs from WordNet. We repeated their experiments on a set of concept pairs from our taxonomy, obtaining $\gamma = 0.27$ and $\beta = 0.59$ (γ and β are not required to sum up to 1).

Equation 4 defines the Distance metric as a weighted sum of $\delta_{\mathcal{F}}$ and $\delta_{\mathcal{T}}$. In order to select good values for the weighting parameters $\alpha_{\mathcal{F}}$ and $\alpha_{\mathcal{T}}$, we conducted an experiment similar to the one used for selecting the values of α_{color}, $\alpha_{texture}$, α_{shape}, and $\alpha_{location}$. We asked a different group of about 40 people to judge the similarity between the pairs of pictorial images used in the previous experiment, also taking into account the semantic description of the paintings (author, genre and subject). We obtained $\alpha_{\mathcal{F}} = 0.52$ and $\alpha_{\mathcal{T}} = 0.48$.

In the definition of Recommendation Grade (Eq. 14) two parameters, α_c and α_P, are used to weight the contribution of features and pattern based similarity in evaluating the recommendation grade. This weighting scheme has been designed to assist a user even in the very first steps of her browsing session, when her current pattern is too short to predict her behavior. For this reason, we set α_c and α_P such that α_P increases and α_c decreases as the length n_c of the current pattern P_c increases ($\alpha_c = 1/n_c, \alpha_P = (n_c - 1)/n_c$). When $n_c = 1$, i.e. when the user requests the first item, $\alpha_c = 1$ and $\alpha_P = 0$, so the recommended items are the k objects having the shortest distance from the last requested object O_c^h. When $n_c = 10$, i.e. when the current pattern of the user is quite long, $\alpha_c = 0.1$ and $\alpha_P = 0.9$, so the recommendations are mainly determined by the analysis of previous patterns.

Two scale issues arise in the proposed system: how to deal with the size of the multimedia collection and how to deal with the size of usage pattern log.

We have already mentioned that an M-tree index has been adopted in order to index the objects in the collection, while in Section 5 we have used a k nearest neighbors query in defining the set of candidate objects. In [5], Ciaccia et al. demonstrated that the M-tree scales well with respect to the size of the indexed data set, and that the dynamic management algorithms do not deteriorate the quality of the search. Moreover the updates to the collection are quite rare once the system has bee set up. In fact, we have experimentally observed that the first scale issue is well addressed.

However, the most challenging scale issue and one of the most critical aspects of the whole system is the construction of the set $\mathcal{P}_\gamma^h$ defined by Eq. 9.

As discussed in Section 5, the threshold γ is defined as a function of $|\mathcal{P}^h|$. This guarantees that the size of $\mathcal{P}_\gamma^h$ does not increase with $|\mathcal{P}^h|$, since the threshold becomes more restrictive. To make our solution scalable with respect to the size of $\mathcal{P}^h$ we need to define an efficient strategy to build the set $\mathcal{P}_\gamma^h$. There is no doubt that it is not feasible to compare each element in $\mathcal{P}^h$ to P_c^h in order to assess its inclusion in $\mathcal{P}_\gamma^h$. The above consideration led us to define an indexing scheme for the pattern collection too. Since the M-tree is suitable to index a generic metric space, and a similarity measure has been defined in the pattern space, we have adopted an M-tree indexing strategy, using $\delta_P = 1 - \chi_P$ for computing the distance between patterns and partitioning the metric space. The set $\mathcal{P}_\gamma^h$ can be thus determined using a range query $\mathsf{range}(P_c^h, 1 - \gamma)$, that selects all the patterns within a distance of

$1-\gamma$ from P_c^h. We can finally conclude that the second scale issue is well addressed too. It is worth pointing out that, while updates to the object collection are quite rare, updates to the usage pattern log are very frequent and their number is directly proportional to the number of users. Although the dynamic management algorithms do not significantly deteriorate the performance of the system, the large number of updates to the usage log could be a problem. For this reason the system maintains log data about current users in a temporary data structure in memory and permanently stores such data in the log only when the system is idle.

The above discussion fully addresses all the scale issues. However, more computations can be saved by better analyzing the algorithm in Fig. 3, used in Eq. 13 for computing the local similarity between each pattern $P^h \in \mathcal{P}_\gamma^h$ and the current pattern P_c^h. The algorithm computes a $(m+1) \times (n+1)$ matrix, where m and n are the lengths of P^h and $\mathcal{P}_c^h$ respectively. When a user requests a new item, the length of the current pattern increases by one unit and a new matrix should be computed for each $P^h \in \mathcal{P}_\gamma^h$. Since the values in a column only depend on the values in the previous column, it is not necessary to recompute the whole matrix, while only the last column needs to be computed.

7 Case study and experimental evaluation

In this section we show how our prototypal system works and report the experiments we have conducted to evaluate the impact of the proposed system on enhancing users' experience in a virtual museum setting.

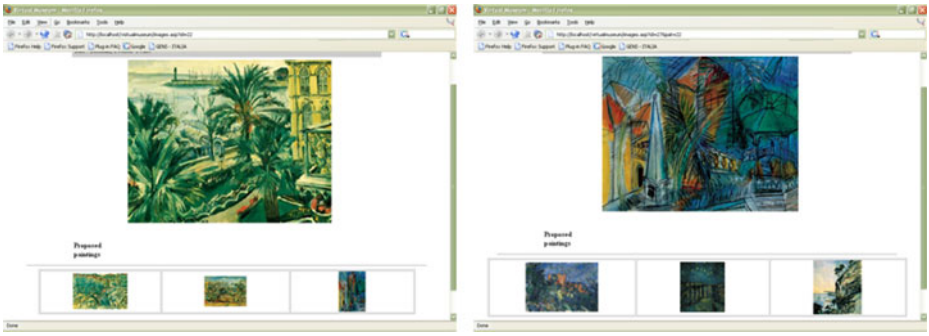
The collection used in the experiments includes 5,000 paintings encompassing 25 genres (e.g., Cubism, Baroque, Early Renaissance), about 200 authors (e.g., Caravaggio, Rubens), and about 80 subjects (e.g., Landscapes, Portraits).

7.1 Virtual gallery

A user who is just starting her tour of the virtual museum can select any of the objects in the exhibition by means of standard search methods: *search by genre*, *search by author and subject*. As she makes the first request for a painting, the system begins to assist her visit.

Figure 5a shows an example in which the first item to be selected is a painting depicting the *French Coast*. At this time, the suggestions from the system are exclusively based on the retrieval of the most similar images. If the current picture is not the first of the browsing session (see Fig. 5b), the system tries to propose both paintings that are similar to the current image and paintings requested by users with similar behavior. As a consequence, the recommendation list includes a painting apparently not related to the only one viewed so far, which was proposed because it was requested by one or more users with a (locally) similar behavior.

We remark that the user is not required to browse one of the recommended items, but she can select, at any time, any of the images in the collection. This avoids that user patterns are exclusively based on the similarity between images.



a b

Fig. 5 The web interface of the recommender system

7.2 Experimental results

7.2.1 Browsing effectiveness

This first set of experiments aims at comparing the ranking provided by our system using the proposed recommendation degree with the ranking provided by a human observer. To this end, we have slightly modified a test proposed by Santini [19], in order to evaluate the difference between the two rankings ("treatments") in terms of hypothesis verification on the entire dataset.

Consider a weighted displacement measure defined as follows. Let Q be a query on a database of $\mathtt{N}$ images that produces $\mathtt{n}$ results. There is one ordering (usually given by one or more human subjects) which is considered as the ground truth, represented as $\mathtt{R_h} = \{O_1, \ldots, O_\mathtt{n}\}$. Every image in the ordering has also associated a measure of relevance $0 \leq \mathtt{S}(O, Q) \leq 1$ such that (for the ground truth), $\mathtt{S}(O_i, Q) \geq \mathtt{S}(O_{i+1}, Q), \forall i$.

This is compared with an (experimental) ordering $\mathtt{R_s} = \{O_{\pi_1}, \ldots, O_{\pi_n}\}$, where $\{\pi_1, \ldots, \pi_n\}$ is a permutation of $1, \ldots, n$. The displacement of O_i is defined as $\mathtt{d}_Q(O_i) = |i - \pi_i|$. The relative weighted displacement of $\mathtt{R_s}$ is defined as $\mathtt{W}_Q = \frac{\sum_i \mathtt{S}(O_i, Q)\mathtt{d}_Q(O_i)}{\Omega}$, where $\Omega = \lfloor \frac{\mathtt{n}^2}{2} \rfloor$ is a normalization factor.

Relevance $\mathtt{S}$ is obtained from the subjects asking them to divide the results in three groups: *very similar* ($\mathtt{S}(O_i, Q) = 1$), *quite similar* ($\mathtt{S}(O_i, Q) = 0.5$) and *dissimilar* ($\mathtt{S}(O_i, Q) = 0.05$).

In our experiments, on the basis of the ground truth provided by human subjects, treatments provided either by humans or by our system are compared. The goal is to determine whether the observed differences can indeed be ascribed to the different treatments or are caused by random variations.

In terms of hypothesis verification, if μ_i is the average score obtained with the *i-th* treatment, a test is performed in order to accept or reject the null hypothesis $\mathtt{H}_0$ that all the averages μ_i are the same (i.e., the differences are due only to random variations); clearly the alternate hypothesis $\mathtt{H}_1$ is that the means are not equal, that is the experiment actually revealed a difference among treatments. The acceptance of the $\mathtt{H}_0$ hypothesis can be checked with the $\mathtt{F}$ ratio.

Let us assume that there are $\mathtt{m}$ treatments and $\mathtt{n}$ measurements (experiments) for each treatment. Let $\mathtt{w}_{ij}$ be the result of the *jth* experiment performed with the i^{th} treatment in place. Let us define $\mu_i = \frac{1}{\mathtt{n}} \sum_{j=1}^{\mathtt{n}} \mathtt{w}_{ij}$ the average for treatment i, $\mu = \frac{1}{\mathtt{m}} \sum_{i=1}^{\mathtt{m}} \mu_i = \frac{1}{\mathtt{nm}} \sum_{i=1}^{\mathtt{m}} \sum_{j=1}^{\mathtt{n}} \mathtt{w}_{ij}$ the total average, $\sigma_{\mathtt{A}}^2 = \frac{\mathtt{n}}{\mathtt{m}-1} \sum_{i=1} \mathtt{m}(\mu_i - \mu)^2$ the between treatments variance, $\sigma_{\mathtt{W}}^2 = \frac{1}{\mathtt{m}(\mathtt{n}-1)} \sum_{i=1} \mathtt{m} \sum_{j=1} \mathtt{n}(\mathtt{w}_{ij} - \mu_{\mathtt{i}})^2$ the within treatments variance. Then, the $\mathtt{F}$ ratio is :

$$\mathtt{F} = \frac{\sigma_{\mathtt{A}}^2}{\sigma_{\mathtt{W}}^2} \tag{15}$$

A high value of $\mathtt{F}$ means that the between treatments variance is preponderant with respect to the within treatment variance, that is, that the differences in the averages are likely to be due to the treatments.

In our experiments we employed 12 subjects selected among undergraduate students. Ten students, randomly chosen among the 12, were employed to determine the ground truth ranking and the other two served to provide the treatments to be compared with our system. Six query images were selected, and for each of them we run a query returning a result set of ten objects, for a total of 60 objects. Result sets were randomly ordered to prevent bias and the two students were then asked to rank images in each set in terms their level of recommendation with respect to the query object.

Each subject was also asked to divide the ranked objects in three groups: the first group consisted of images judged *very relevant* to the query, the second group consisted of images judged *quite relevant* to the query, and the third of *non relevant* images. The mean and variance of the weighted displacement of the two subjects and of our system with respect to the ground truth are reported in Table 1.

Then, the $\mathtt{F}$ ratio for each pair of distances was computed in order to establish which differences were significant. As can be noted from Table 2, the $\mathtt{F}$ ratio is always less than 1 and since the critical value $\mathtt{F}_0$, regardless of the confidence degree (the probability of rejecting the right hypothesis), is greater then 1, the null hypothesis can be statistically accepted.

7.2.2 User satisfaction

In order to evaluate the impact of the system on the users we have conducted the following experiments.

First, we have asked a first group of about 25 people to use the system for some days, in order to collect a significant amount of usage patterns (several hundreds).

Then we asked a different group of about 50 people to browse a collection of images and complete several browsing tasks (20 tasks per user) of different complexity (five tasks for each complexity level), using the well-known image database system *Picasa* (taxonomies are implemented as *albums*, *folders* and *descriptions*). After this

Table 1 Mean (μ_i) and variance (σ_i^2) of the weighted displacement for the three treatments (two human subjects and system)

	Human 1	Human 2	Recomm. grade $\rho(Q)$
μ_i	0.0451	0.0373	0.0279
σ_i^2	$8.145e^{-4}$	$8.928e^{-4}$	$8.970e^{-4}$

Table 2 The F ratio measured for pairs of distances (human vs. human and human vs. system)

F	Human 1	Human 2	$\rho(Q)$
$\rho(Q)$	0.472	0.799	0
Human 2	0.0896	0	
Human 1	0		

test, we asked them to browse once again the same collection with the assistance of our recommender system and complete other 20 tasks of the same complexity. We have subdivided browsing tasks in the following four broad categories:

1. *Low Complexity* tasks (Q_1)—e.g. "explore at least 10 paintings of *Baroque* style authored by *Caravaggio* and depicting a *religious* subject";
2. *Medium Complexity* tasks (Q_2)—e.g. "explore at least 20 paintings of *Baroque* authors that have *nature* as their subject";
3. *High Complexity* tasks (Q_3)—e.g. "explore at least 30 paintings of *Baroque* authors with subject *nature* and with a predominance of *red color*";
4. *Very High Complexity* tasks (Q_4)—e.g. "explore at least 50 paintings of *Baroque* authors with a predominance of *red color*".

Note that the complexity of a task depends on several factors: the number of objects to explore, the type of desired features (either low or high-level), and the number of constraints (genre, author, subject). Two strategies were used to evaluate the results of this experiment: empirical measurements of access complexity in terms of *mouse clicks* and *time*, and *TLX* (*NASA Task Load Index* factors) [8].

With respect to the first strategy, we measured the following parameters:

– *Access Time (t_a)*. The average time spent by the users to request and access all the objects for a given class of tasks;
– *Number of Clicks (n_c)*. The average number of clicks necessary to collect all the requested objects for a given class of tasks.

Table 3 reports the average values of t_a and n_c, for both Picasa and our system, for each of the four task complexity levels defined earlier.

In the second experiment, we asked the users to express their opinion about the capability of Picasa and our system respectively to provide an effective user experience in completing the assigned browsing tasks. To this end, we used the TLX evaluation form, which allows to assess the workload on operators of various human–machine systems. Specifically, TLX is a multi-dimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six

Table 3 Comparison between our system and Picasa in terms of t_a and n_c

Task class	Search engine	t_a (sec.)	n_c
Q_1	Picasa	60.2	15
Q_1	Our System	53.8	13.2
Q_2	Picasa	104.3	26.8
Q_2	Our System	62.5	21.3
Q_3	Picasa	219.8	57.1
Q_3	Our System	155.1	39.2
Q_4	Picasa	402.6	104.2
Q_4	Our Systems	240.3	60.7

Table 4 Comparison between our system and Picasa in terms of TLX factors

TLX Factor	Our system	Picasa
Effort	43.9	51.5
Mental demand	45.7	48.2
Physical demand	40.2	44.7
Temporal demand	49.4	62.3
Frustration	52.6	69.1
Own performance	31.2	39.8

sub-scales: mental demand, physical demand, temporal demand, own performance, effort and frustration (lower TLX scores are better). In other words, this experiment was aimed at measuring how difficult is for a user to use either our system or Picasa to complete a browsing task. We obtained the average result scores reported in Table 4, which show that our system outperforms Picasa in every sub-scale.

It is evident that the two aspects where our system beats Picasa by the largest margin are temporal demand and frustration. This result implies that our system allows to complete browsing tasks faster and provides a better (less frustrating) user experience. In addition, the fact that browsing tasks can be completed faster using our system is an indication that recommendations are effective, as they allow a user to explore interesting and related objects one after another, without the interference of undesired items that would necessarily slow down the process.

8 Conclusions and future directions

In this paper we presented a novel approach to the design of recommender systems in the context of multimedia browsing. Our approach is based on combining the information that is latent in usage logs with the features—both low-level and semantic descriptors—of the objects in a multimedia repository. We leveraged the work presented in [2]—which was primarily focused on image databases—and augmented its theoretic foundations in order to deal with more complex scenarios. We introduced the concepts of *Multichannel Browser* and *Multichannel Object* to model a user concurrently browsing multiple types of objects (e.g., an image and a text document displayed side by side to form a single "Multichannel Object"). In such a scenario, we want to enable a recommender to provide a "complex" recommendation (e.g., the image and the text document to be displayed next).

We conducted extensive experiments on a prototypal implementation of the proposed system, and the results are extremely promising and encourage further research in this direction. In particular, we compared our system with Picasa, and showed that it outperforms Picasa in terms of effectiveness and usability by a significant margin.

In conclusion, although the results of our work are extremely satisfying, there is still huge room for improvement. First, the assumption of not relying on explicit login—which makes the system more general—could be relaxed in order to allow profiling of both authenticated and anonymous users. This would have the undoubted benefit of improving the quality of recommendations for users who decide to use the system to its fullest potential. Second, the way usage patterns are collected and analyzed only allows to discover positive links between objects, i.e. the fact that users selected certain objects—possibly among those suggested—as the successors of

other objects. However, more precise information about users' preferences could be acquired by tracking and analyzing all the recommendations that were made to a user and then ignored by that user: the fact that a recommendation is ignored indicates that the user does not consider the suggested object related to the current object. We plan to address these and other issues in the near future.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6): 734–749
2. Albanese M, Chianese A, d'Acierno A, Moscato V, Picariello A (2009) A recommendation system for browsing digital libraries. In: SAC '09: proceedings of the 24th symposium on applied computing. ACM, Honolulu pp 1771–1778
3. Anand SS, Kearney P, Shapcott M (2007) Generating semantically enriched user profiles for web personalization. ACM Trans Internet Technol 7(4):22
4. Basilico J, Hofmann T (2004) Unifying collaborative and content-based filtering. In: ICML '04: proceedings of the 21st international conference on machine learning, ACM, Banff
5. Ciaccia P, Patella M, Zezula P (1997) M-tree: an efficient access method for similarity search in metric spaces. In: Jarke M, et al. (eds) VLDB 97: proceedings of 23rd conference on very large database. Morgan Kaufmann, Athens, pp 426–435
6. Eirinaki M, Vazirgiannis M (2003) Web mining for web personalization. ACM Trans Internet Technol 3(1):1–27
7. Fayzullin M, Subrahmanian V, Albanese M, Cesarano C, Picariello A (2007) Story creation from heterogeneous data sources. Multimed Tools Appl 33(3):351–377
8. Hart SG, Stavenland LE (1988) Development of NASA-TLX (task load index): results of empirical and theoretical research. In: Hancock PA, Meshkati N (eds) Human mental workload, chapter 7. Elsevier, Amsterdam, pp 139–183
9. Hijikata Y, Iwahama K, Nishida S (2006) Content-based music filtering system with editable user profile. In: SAC '06: proceedings of the 21st ACM symposium on applied computing. ACM, Dijon, pp 1050–1057
10. Lam XN, Vu T, Le TD, Duong AD (2008) Addressing cold-start problem in recommendation systems. In: ICUIMC '08: proceedings of the 2nd international conference on ubiquitous information management and communication. ACM, Suwon, pp 208–211
11. Lekakos G, Caravelas P (2008) A hybrid approach for movie recommendation. Multimed Tools Appl 36(1–2):55–70
12. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10(8):707–710
13. Lew MS, Sebe N, Djeraba C, Jain R (2006) Content-based multimedia information retrieval: state of the art and challenges. ACM Trans Multimedia Comput Commun Appl 2(1):1–19
14. Li Y, Bandar ZA, Mclean D (2003) An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans Knowl Data Eng 15(4):871–882
15. Maidel V, Shoval P, Shapira B, Taieb-Maimon M (2008) Evaluation of an ontology-content based filtering method for a personalized newspaper. In: RecSys '08: proceedings of the 2nd ACM international conference on recommender systems. ACM, Lausanne, pp 91–98
16. Park S-T, Pennock DM (2007) Applying collaborative filtering techniques to movie search for better ranking and browsing. In: KDD '07: proceedings of the 13th ACM international conference on knowledge discovery and data mining. ACM, San Jose, pp 550–559
17. Pasi G, Bordogna G, Villa R (2007) A multi-criteria content-based filtering system. In: SIGIR '07: proceedings of the 30th annual international ACM SIGIR conference. ACM, Amsterdam, pp 775–776
18. Pazzani M J, Billsus D (2007) Content-based recommendation systems. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The adaptive web: methods and strategies of web personalization. Lecture Notes in Computer Science, vol 4321, chapter 10. Springer, Berlin, pp 325–341
19. Santini S (2000) Evaluation vademecum for visual information systems. In: Proceedings of SPIE storage and retrieval for image and video databases, vol 3972. San Jose, USA, pp 132–143

20. Si L, Jin R (2004) Unified filtering by combining collaborative filtering and content-based filtering via mixture model and exponential model. In: CIKM '04: proceedings of the thirteenth ACM international conference on information and knowledge management. ACM, Washington, D.C., pp 156–157
21. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
22. Tso-Sutter KHL, Marinho LB, Schmidt-Thieme L (2008) Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: SAC '08: proceedings of the 23rd ACM symposium on applied computing. ACM, Fortaleza, pp 1995–1999
23. Wang J, de Vries AP, Reinders MJT (2008) Unified relevance models for rating prediction in collaborative filtering. ACM Trans Inf Sys 26(3):1–42
24. Yeung MM, Yeo B-L (1997) Video visualization for compact presentation and fast browsing of pictorial content. IEEE Trans Circuits Syst Video Technol 7(5):771–785
25. Zhang HJ, Low CY, Smoliar SW, Zhong D (1995) Video parsing, retrieval and browsing: an integrated and content-based solution. In: Proceedings of the ACM international conference on multimedia. San Francisco, California, USA, November, pp 15–24
26. Zhu X, Elmagarmid AK, Xue X, Wu L, Catlin AC (2005) Insightvideo: toward hierarchical video content organization for efficient browsing, summarization and retrieval. IEEE Trans Multimedia 7(4):648–666

Massimiliano Albanese received a Laurea degree in Computer Science and Engineering from the University of Naples "Federico II" in 2002. In 2005, he received his Ph.D. degree in Computer Science and Engineering from the same University, where he then served until 2006 as a Research and Teaching Assistant with the Multimedia Information Systems Group. In 2006, he joined the University of Maryland Institute for Advanced Computer Studies, College Park, as a Post Doctoral Researcher. His primary areas of interest are in Multimedia Databases, Information Extraction, Activity Detection, Knowledge Representation and Management.

Angelo Chianese received the Laurea degree in Electronics Engineering from the University of Naples, Federico II in 1980. In 1984, he joined the Dipartimento di Informatica e Sistemistica of the University of Naples "Federico II" as an Assistant Professor. Currently he is a full professor at the University of Naples Federico II. He has been active in the field of pattern recognition, optical character recognition, medical image processing, and object-oriented models for image processing. His current research interests lie in multimedia data base and multimedia content management for elearning. Angelo Chianese is a Member of the International Association for Pattern Recognition (IAPR).

Antonio d'Acierno received the Laurea degree com Laude in Electronics Engineering from the University of Naples Federico II . Since 1988 to 1999, he was actively integrated into the research group of IRSIP (Institute for Research on Parallel Informatic Systems) of the National Research Council of Italy (CNR). In 1999 he joined the Institute of Food Science (ISA) of CNR. His current research interests lie in the field of mobile transactions, information retrieval, semantic web, multimedia ontologies and applications, and bioinformatics.

Vincenzo Moscato received the Laurea degree (cum laude) in Computer Science and Engineering from the University of Naples "Federico II", Italy, in 2002. In 2005, he received the Ph.D. degree in Computer Science and Engineering at the same University. In 2009 he joined the Dipartimento di Informatica e Sistemistica of University of Napoli "Federico II", where he is currently an Assistant Professor of Data Base and Computer Engineering. He has been active in the field of computer vision, video and image indexing and multimedia data sources integration. His current research interests lie in the area of multimedia databases, video-surveillance applications and knowledge representation and management. He is an IEEE member.

Antonio Picariello received a Ph.D. degree in Computer Science and Engineering in 1998 from the University of Naples Federico II. He is currently an Associate Professor of Data Base and Computer Engineering at the same University. Prof. Picariello has been active in the field of computer vision, medical image processing and pattern recognition, objectoriented models for image processing. His recent research interests lies on video surveillance, multimedia data base, multimedia ontologies and summarization. He is an IEEE member.