A Multimedia Data Base Browsing System[*]

Massimiliano Albanese
malbanes@unina.it

Carmine Cesarano
cacesara@unina.it

Antonio Picariello
picus@unina.it

Dipartimento di Informatica e Sistemistica
University of Naples “Federico II”
Naples, Italy

ABSTRACT

Browsing large multimedia databases is becoming a challenging problem, due to the availability of great amounts of data and the complexity of retrieval. In this paper we propose a system that assists a user in browsing a digital collection making useful recommendations. The system combines computer vision techniques and taxonomic classifications to measure the similarity between objects and adopts an innovative strategy to take into account user behavior.

1. INTRODUCTION

Browsing and retrieval from large multimedia databases is becoming a challenging problem, due to the availability of great amounts of data and to the different retrieval strategies to be used in the multimedia domain. In order to address this issue, researchers from the computer vision community continuously propose new content-based descriptors and techniques for extracting them from multimedia objects [7, 9, 18, 21]. This approach allows similarity queries based on perceptive considerations [1]. The drawback of that approach lies in the fact that the *similarity* between images is exclusively evaluated in terms of the visual content of the images themselves. It is indeed clear that two images might be regarded to be similar if they share some common *high-level* semantics, whatever their visual content is. On the other hand, two images exhibiting similar low-level features might have different high-level semantics. It’s the authors’ opinion that, in order to improve the quality of the retrieval process, high-level semantics should be taken into account. To achieve this aim, we propose some low and high level descriptors and define a strategy to combine them in the similarity matching process. In this paper we present a system that assists a user in browsing a digital collection, providing useful recommendations, based on an innovative strategy for taking into account user behavior and usage patterns. The model behind our system is general enough to be applied whenever one wants to allow the browsing of a digital collection and, even if it is defined w.r.t. image databases, it could be easily extended to any kind of multimedia object. So, sometimes in the paper, we will be using the term *object* instead of *image*.

The remainder of the paper is organized as follows: related works are reported in section 2, while section 3 introduces a motivating example that will be the running example of this paper; section 4 describes the architecture of the system; section 5 describes the metric that combines image features and taxonomies, while section 6 explains how user preferences are taken into account to provide recommendations; system tuning and scale issues are discussed in section 7; eventually, experiments and conclusions are reported in sections 8 and 9 respectively.

2. RELATED WORKS

Several systems have been proposed in the literature in order to simplify browsing and retrieval in large multimedia databases. In [20] van Beek et al. present multimedia descriptions that can be used to facilitate rapid navigation and efficient access to different views of audiovisual programs according to personal preferences. Niblack et al. [15] describe methods for crawling, summarizing and displaying visual multimedia data, based on queries that combine text and image similarity. Drummond et al. [4] propose a technique to assist the users in their search through a multimedia database, based on the intelligent agents paradigm.

Regarding the recommendation systems realm, two main approaches have been explored in the literature: the content based filtering and the collaborative filtering. Anyway, several systems combines both of them in different ways.

A *content based filtering* approach tries to recommend the data items accessed in the past by the user. The success of this kind of approach relies on the ability to represent the data items in terms of appropriate sets of content features: the relevance of an item is proportional to the similarity of user’s profile. The drawback of this technique relies on the recommendations computed on a very limited diversity. As an example, *Infofinder* [8] is based on this kind of approach.

Collaborative filtering is a good alternative to the content based strategies. The main idea of the collaborative filtering is to associate the current user as to a set formed by all the users having a “common” profile. In this way, the data items are recommended on the basis of the similarity

[*]This work has been carried out partially under the financial support of the Ministero dell’Istruzione, dell’Università e della Ricerca (MIUR) in the framework of the FIRB Project “Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)”

CVDB'04, June 13, 2004, Paris, France.

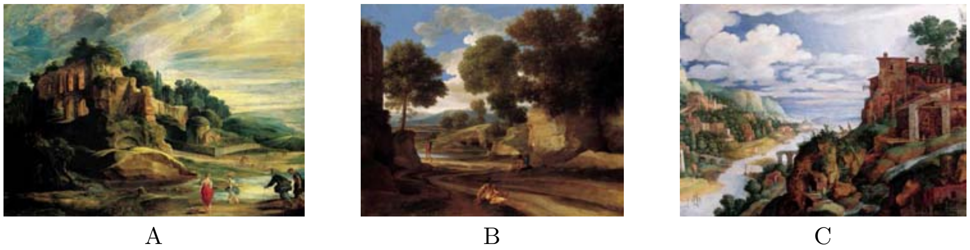

Figure 1: Paintings depicting some landscapes

between users, rather than on the similarity between data items themselves. The drawback of this technique relies on the delay in considering a newly introduced data item like a possible recommendation: the new data will become available for recommendation after that many users have seen and rated it. Besides, if there are not adequate overlaps between the current user's profile and the stored ones, it will not be possible to make a reliable recommendations using this kind of technique. Some systems like Siteseer [17] use this approach.

Some systems use both content-based and collaborative filtering techniques, such as *Personalized Television service* [19]. In these systems there is the opportunity to discover the user's interest from the accessed data items, and to rate the suggestions. At the same time, all the users that specify similar satisfactory degrees are grouped together in order to use collaborative techniques. Another example is the *PCFinder* system [22]. It combines an Order-Based Similarity Measure [2] with collaborative filtering techniques and a cluster analysis for grouping the users according their long term profiles. In this system, the users need to provide a variety of information such as: main interests, occupation, age, and so on; the system tries to provide suggestions about long-term constraints, according to the profile information, while, using the current behavior of the user, tries to make suggestions combining long-term constraints with the short-term ones.

The work presented in this paper differs from the other recommendations systems in a considerable way. This paper, in fact, represents a first step towards a much broader goal: it describes a system that may be used to produce recommendations using both human-created annotations of the images, and image analysis and processing features and, in addition, user preferences without any preliminary knowledge of the user behavior. As such, our system is general and applies to most kinds of image collections applications.

3. MOTIVATING EXAMPLE

In this work we focus our attention on the case of a *virtual museum*, i.e. a museum that offers a web based access to a collection of digital reproductions of paintings. In order to make user experience in the museum more interesting and stimulating, the access to information should be differentiated according to the visitors specific profile, that includes learning preferences, level of expertise and visiting styles.

Let us consider a user visiting the virtual museum and suppose that she requests, in the first steps of her *virtual tour*, some paintings depicting imaginary landscapes. Then she focuses her attention on a Peter Paul Rubens' painting entitled *Landscapes with the ruins on the Palatine Hill in Rome* (figure 1.A). It would be nice if the system could learn the user preferences, based on these first interactions, and accommodate her needs, by suggesting other paintings representing the same or related subjects, depicted by the same or related authors, or somehow related to the overall user behavior. From the user perspective there is the advantage of having a guide suggesting artifacts which the user might probably be interested in, while, from the system perspective, there is the undoubted advantage of using the suggestions for pre-fetching and caching the images that are more likely to be requested. Thus the user who is currently observing the painting in figure 1.A might be recommended to see a Nicolas Poussin's painting entitled *Landscape in the Roman Campagna* (figure 1.B), that is quite similar to the current picture in terms of color and texture, and *Italian landscape - Early seventeenth century* (figure 1.C) by William Van Nieulandt, that is not similar in terms of low level features but is similar in terms of semantic content.

4. SYSTEM ARCHITECTURE

Figure 2 shows at a glance the overall architecture of the system. A user connects to the web server through a common internet browser and starts exploring the images collection. As the user keeps on browsing, the system records in the *Usage Log* which items she requests and in which order. In the meantime the *Pattern Discovery Subsystem*, based on the behavior of past users and a certain metric that we will define later in the paper, tries to classify the user and predict her future behavior.

No mechanisms such as cookies or explicit user login have been implemented to simplify the task of user identification and classification, since the first ones can be deleted or disabled by the user herself, while the explicit login typically discourages the users from accessing a web site, even if it is regarded as interesting. So the precision of user classification, being exclusively based on her dynamic behavior, is quite poor when the user accesses the collection and then it gets better as she keeps on exploring the collection itself. 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 interesting items to see next.

5. A METRIC FOR IMAGE COMPARISON

The main aspect of our similarity matching strategy lies in combining image visual features and descriptive taxonomies. Thus we define computer vision mechanisms for image features extraction and comparison, and mechanisms for semantic comparison.

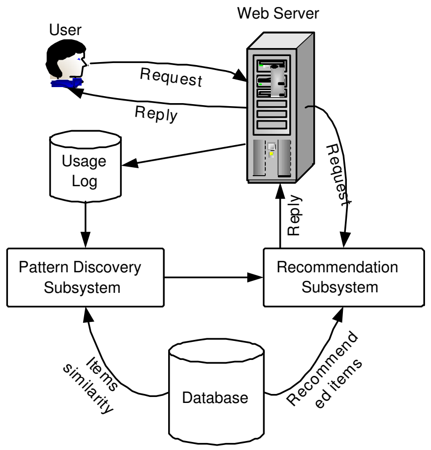

Figure 2: System Architecture

5.1 Features Based Image Similarity

In an *image data base system*, the basic question to address is to define when two images may be considered similar. In the literature, similarity of images has been characterized through three important features: color, texture and shape [18]. While shape is a more complex information to cope with, color and texture are low-level features that are very simple to process. For this reason, they are probably the two most prominent and commonly exploited low-level image features, and many methods have been proposed in the past for extracting them. In this paper we are essentially dealing with paintings and art images, so we are not interested in having high precision in retrieving similar images. If two images are similar, then their color histograms are similar; however, the opposite, might not be true: actually, two images having the same color histogram, can have very different appearance. Thus, we propose to combine a histogram technique - that takes into account a coarse and quick similarity concept - and the texture features.

In order to design a fast retrieval system, we have adopted the wavelet transform [21] as a mechanism useful for both reducing the amount of data to be analyzed and providing a suitable color and texture representation. Let us denote the wavelet coefficients as $w_l^k(x,y)$, where $(x,y) \in D_p \subseteq \mathbb{R}^2$, l is the decomposition level and k the sub-bands. A wavelet decomposition gives rise to 4 subregions of dimension $|D_p|/4$. We have used the low pass components in order to take into account the color information by means of *color histograms*.

Color features. Given a set of representative colors $Q = \{q_1, ..., q_B\}$, a color histogram $h(I) = \{h_b^I\}$ of an image I is defined on bins $b \in [1, B]$, such that, for any pixel in D_p, h_b^I is the probability that the color of the pixel is $q_b \in Q$. We remark that the low pass component is a smoothed copy of the original picture, thus allowing to avoid lightening and noise problems.

Definition 1 (Color Distance)
Given two images I_1 and I_2 and their respective color histograms $h(I_1) = \{h_b^{I_1}\}$ and $h(I_2) = \{h_b^{I_2}\}$, defined on the same number B of bins, the Color Distance *can be defined as*

$$d_{col}(I_1, I_2) = 1 - \sum_{b=1}^{B} \min\left(h_b^{I_1}, h_b^{I_2}\right) / \sum_{b=1}^{B} h_b^{I_1}, \tag{1}$$

where $\sum_{b=1}^{B} h_b^{I_1}$ is a normalization factor.

Texture features. Only the detail components of the wavelet transform are taken into account, in order to characterize texture. For $k = 1, 2, 3$, the detail sub-bands contain horizontal, vertical and diagonal directional information, respectively, and are represented by coefficient planes $\left[\{w_l^k(x,y)\}\right]_{k=1,2,3}$. Next, the Wavelet Covariance Signature is computed, i.e. the feature vector of coefficient covariances $\Sigma_C^2 = \{\sigma_{X,Y}^2\}$, where:

$$\sigma_{X,Y}^2 = \sum_{x,y} \left\{ \frac{1}{|D_p|/4} \sum_{k=1}^{3} X_k(x,y) Y_k(x,y) \right\}. \tag{2}$$

The pair (X_k, Y_k) is in the set of coefficient plane pairs $\{(w_i^k, w_j^k)\}$, i and j being used to index the three channels, and (x,y) span over the sub-band lattice of dimension $|D_p|/4$.

Definition 2 (Texture Distance)
Let C_1 and C_2 be the wavelet signatures of two images I_1 and I_2 respectively. We define the Texture Distance *between two images I_1 and I_2 as*

$$d_{tex}(I_1, I_2) = \frac{1}{R} \sum_{i=1}^{|\Sigma^2|} \frac{\left|\Sigma_{C_1}^2[i] - \Sigma_{C_2}^2[i]\right|}{\min\left(\left|\Sigma_{C_1}^2[i]\right|, \left|\Sigma_{C_2}^2[i]\right|\right)}. \tag{3}$$

where R is a normalization factor to bound the sum in $[0,1]$, and $|\Sigma^2|$ the number of features in the feature vector Σ^2 computed through equation 2.

We can now define the distance that combines color and texture distances and the related similarity measure.

Definition 3 (Features Based Distance)
The Features Based Distance *between two images I_1 and I_2 is defined as*

$$d_{features}(I_1, I_2) = \alpha_{col} \cdot d_{col}(I_1, I_2) + \alpha_{tex} \cdot d_{tex}(I_1, I_2) \tag{4}$$

α_{col} and α_{tex} being two weighting factors[1].

Definition 4 (Features Based Similarity)
The Features Based Similarity *between two images I_1 and I_2 is defined as*

$$S_F(I_1, I_2) = 1 - d_{features}(I_1, I_2) \tag{5}$$

5.2 Taxonomy Based Similarity

A taxonomy is usually a hierarchical concept network, where a node in the hierarchy represents a concept/class and an edge represents a direct association between two parent/child concept nodes. We can reasonably assume that each object in a collection has an associate semantic description, typically consisting of a set of attributes. Some of these attributes correspond to concepts that are relevant for the specific domain, being the entities in the conceptual data model. 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 [10]. Let us formalize the concept of semantic description.

[1] Section 7.1 describes the strategy used for selecting good values for α_{col} and α_{tex} as well for other parameters we will introduce in the paper.

Definition 5 (Object Semantic Description)
Given an application specific taxonomy $\mathcal{T}$, an Object Semantic Description *OSD is an ordered pair defined as*

$$OSD = (TA, NTA) \tag{6}$$

where $TA = (A_1, ..., A_\tau)$ is an ordered tuple of attributes that assume values corresponding to nodes of $\mathcal{T}$ and $NTA = (A_1^*, ..., A_{\tau^*}^*)$ is an ordered tuple of attributes whose values do not correspond to nodes of $\mathcal{T}$.

An object o in the collection can be defined as a triple $(OID, OSD, PhyObj)$, where OID is a unique object identifier, OSD is a semantic descriptor and $PhyObj$ the raw multimedia object. Let us denote by $\mathcal{O}$ the collection of all the objects o_i in the database that are exhibited through the web interface. Now we want to define a metric that evaluates the similarity between objects in terms of semantic description. We start from the assumption that, given a taxonomic attribute A_k, the similarity of objects o_i and o_j, as discussed in [12], 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. Thus we can give our definition of taxonomic similarity and, dually, of taxonomic distance.

Definition 6 (Taxonomy Based Similarity)
Let $\mathcal{T}$ be a taxonomy and $TA = (A_1, ..., A_\tau)$ the ordered tuple of taxonomic attributes. The Taxonomy Based Similarity *between two objects o_i and o_j is defined as*

$$S_T(o_i, o_j) = \frac{1}{\tau} \cdot \sum_{k=1}^{\tau} e^{-\alpha \cdot l(a_k^i, a_k^j)} \cdot \left(1 - e^{-\beta \cdot d(a_k^i, a_k^j)}\right) \tag{7}$$

where $a_k^i = t_i(A_k)$ and $a_k^j = t_j(A_k)$ are the values of attribute A_k for o_i and o_j respectively, $l(a_k^i, a_k^j)$ is the path length between a_k^i and a_k^j and $d(a_k^i, a_k^j)$ 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 equation 7 does not take into account the attributes in NTA 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.

Definition 7 (Taxonomy Based Distance)
The Taxonomy Based Distance *between two objects o_i and o_j is defined as*

$$d_{taxonomy}(o_i, o_j) = 1 - S_T(o_i, o_j) \tag{8}$$

Example 1. *W.r.t. the virtual museum example, we assume the availability of a taxonomy that manages the concepts of painters, pictorial genres and depicted subjects. Then we adopt an OSD such that $TA = (Author, Genre, Subject)$ and $NTA = (Title, Date)$. Based on the above discussion we can conclude that, the closer are the authors, the genres and the subjects, the more similar two paintings are.*

5.3 Features and Taxonomy Metric

To improve the retrieval performance, we adopt an image indexing strategy based on *M-Tree* [3]. M-tree is a well known access method in the image database community, which use a generic *metric space* in order to organize multimedia data. The M-tree strategy is independent from the adopted metric and is easily integrable with other access methods in a DBMS. In the recommendation process we take advantage of the *k nearest neighbors query* described in [3], which permits to retrieve the *k* indexed images having the shortest distance from a given image. The adopted distance is a combination of the taxonomic and feature based distances, as in the following.

Definition 8 (Image Index Metrics)
The Index Distance Metric *between two images I_i and I_j is defined as*

$$d_M(I_i, I_j) = \alpha_F \cdot d_{features}(I_i, I_j) + \alpha_T \cdot d_{taxonomy}(I_i, I_j) \tag{9}$$

α_F and α_T being two weighting factors. The Index Similarity Metric *between I_i and I_j is defined as*

$$S_M(I_i, I_j) = 1 - d_M(I_i, I_j) \tag{10}$$

The M-tree is built by iteratively partitioning the metric space into regions containing similar objects. The initial set of objects $\mathcal{O}$ is divided into clusters according to a minimum distance principle, using the following procedure.

1. $\mathcal{O}$ is divided into two subsets A and $B \mid \mathcal{O} = A \cup B$.

2. Centers and radii are computed for each cluster pair (A, B), selecting the gravity center point as center, and the distance between the center and the most distant object in the cluster as radius.

3. Step 1 and 2 are repeated for A and B until each region contains a number of elements lower than a fixed threshold, or when the maximum tree depth is reached.

6. USER PREFERENCES

The techniques described so far allow to make suggestions to a user based on the picture that she is currently watching. It would be useful if the system could personalize the recommendations taking into account the behavior of current and past users. The *personalization* is usually described as *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]. In the following we propose an algorithm for the prediction of user preferences and behavior.

Definition 9 (Usage Pattern)
A Usage pattern *p of length k is the ordered sequence of k objects requested by a user in the same browsing session.*

$$p = (o_{i_1}, o_{i_2}, ..., o_{i_k}), \text{ with } o_{i_j} \in \mathcal{O}\ \forall j \in [1, k] \tag{11}$$

Let $\mathcal{P}$ be the set of all the usage patterns of past visitors. We are interested in dynamically classify the behavior of the users visiting the virtual museum. The basic idea of our approach consists of finding the patterns in $\mathcal{P}$ that best match the current usage pattern and making suggestions based on what the users corresponding to those pattern have done in the past. So, we are interested in the notion of similarity between usage patterns. Several algorithms have been proposed [5, 11] to compare sequences of symbols from a given alphabet Σ and evaluate their similarity or their distance.

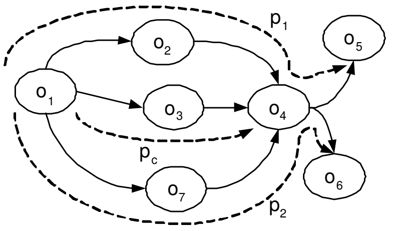

Figure 3: An example of usage patterns

A well-known algorithm in this field is the Levenshtein algorithm [11], that was designed to evaluate the distance between two words as the sum of the costs of the basic operations (insertions, deletions and substitutions) needed to transform a string into the other. Such 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 for substituting a symbol a with a symbol $b \neq a$ and the cost for deleting or inserting a symbol a are defined to be 1, whatever a and b are.

Example 2. *W.r.t. the example in fig. 3, let us 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, whatever the features of o_x are, while we might state that such distance should depend on the distance between o_2 and o_x.*

The idea of our approach is to evaluate the similarity between patterns based on the similarity defined in section 5 and taking advantage from the related indexing strategy. It's worth pointing out that the main issue here is that of dynamically identify a user as she browses the collection. The length of a usage pattern starts from zero and then increases by one unit every time the user requests a new item from the collection. To these aims it is not useful to rawly compare the current usage pattern with the full patterns in the log, while a measure of local similarity between patterns would be better. In other words, we are thus interested in finding those patterns containing the subsequences that match the current pattern in an optimal way and then make suggestions based on their structure.

Example 3. *W.r.t. the example in fig. 3, let us suppose that the partial pattern of a user that is currently browsing the collection is $p_c = (o_1, o_3, o_4)$ and 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. Thus the system can suggest the current user to see objects o_5 and o_6, ranking them on the basis of how much o_2 and o_7 are similar to o_3.*

Starting from the Levenshtein theory, we have designed an algorithm in order to evaluate the local similarity between usage patterns, taking into account the features of the objects in the patterns. The algorithm computes an array D 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 D is the overall local similarity between p_1 and p_2 and it corresponds to the best alignment between those patterns. Definition 10 introduces the functions used to reward or penalize an alignment.

Definition 10 (Sub,Ins,Del)
Let $p_1 = (o_{k_1}, ..., o_{k_m})$ and $p_2 = (o_{l_1}, ..., o_{l_n})$ be two patterns of length m and n respectively. We define the substitution, insertion and deletion functions as follows

$$Sub(p_1[i], p_2[j]) = \frac{S_M(o_{k_i}, o_{l_j}) - \delta}{1 - \delta} \quad (12)$$

$$Ins(p_2[j], p_1[i]) = \frac{\min\{S_M(o_{k_i}, o_{l_j}), S_M(o_{k_{i+1}}, o_{l_j})\} - 1}{(1 - \delta)/\delta} \quad (13)$$

$$Del(p_1[i], p_2[j]) = Ins(p_1[i], p_2[j]) \quad (14)$$

S_M being the similarity metric defined by equation 10 and δ a threshold.

$Sub(p_1[i], p_2[j])$ is the reward/penalization for the substitution of the *i-th* element of p_1 with the *j-th* element of p_2, $Ins(p_2[j], p_1[i])$ is the penalization for the insertion of the *j-th* element of p_2 after the *i-th* element of p_1 and $Del(p_1[i], p_2[j])$ is the penalization for the deletion of the *i-th* element of p_1, j being the position of the element in p_2 aligned with $p_1[i-1]$. The threshold δ has been defined as a function of the size of the collection, by posing $\delta = (\lg |\mathcal{O}| - 0.4)/\lg |\mathcal{O}|$. For example, $\delta = 0.8$ when $|\mathcal{O}| = 100$ and $\delta = 0.9$ when $|\mathcal{O}| = 10000$.

Figure 4 lists the algorithm used for the evaluation of local similarity between patterns. Given an alignment, the algorithm assigns it a positive score for each substitution of an element o_{k_i} of p_1 with an element o_{l_j} of p_2 that is similar to o_{k_i} within the threshold δ. Vice versa a negative score is assigned to each substitution of an element of p_1 with an element of p_2 not similar within 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_{l_j} of p_2 between elements o_{k_i} and $o_{k_{i+1}}$ of p_1 is penalized by an amount that is greater when it is dissimilar from both o_{k_i} and $o_{k_{i+1}}$. In the following we define a measure of the similarity between objects implicity expressed through the usage patterns. To this aim we need to introduce the following sets:

$$\begin{aligned} \mathcal{P}_\gamma &= \{p \in \mathcal{P} \mid \text{local-similarity}(p, p_c) \geq \gamma\} && (15) \\ \mathcal{O}_\gamma &= \{o \in \mathcal{O} \mid \exists p \in \mathcal{P}_\gamma, \text{next}_p(p_c) = o\} && (16) \end{aligned}$$

$\mathcal{P}_\gamma$ is the set[2] of all the patterns in the log that are similar to the current pattern p_c within a threshold γ, while $\mathcal{O}_\gamma$ is the set of those object that users corresponding to the patterns in $\mathcal{P}_\gamma$ have seen after the subsequence similar to p_c. Let us now define the following sets:

$$\begin{aligned} \mathcal{O}_c &= \mathcal{O}_\gamma \cup \mathsf{NN}(o_c, k) && (17) \\ \mathcal{P}_i &= \{p \in \mathcal{P}_\gamma \mid \text{next}_p(p_c) = o_i\}, \forall o_i \in \mathcal{O}_c && (18) \end{aligned}$$

where $\mathsf{NN}(o_c)$ selects the k nearest neighbors of last requested object o_c. $\mathcal{O}_c$ is the set of candidate objects for inclusion in the recommendation list, while $\mathcal{P}_i$ is the subset of $\mathcal{P}_\gamma$ containing those patterns having o_i as the first element following the subsequence similar to p_c. The threshold γ is needed because it makes no sense to base the recommendations on patterns that are not similar enough to the current pattern. Moreover, considering only a subset of $\mathcal{P}$ reduces the complexity of the process. The threshold γ should be

[2] We will discuss in section 7.2 how to build this set.

```
function local-similarity($p_1$,$p_2$)
    $p_1$ and $p_2$ are two patterns of length $m$ and $n$ respectively
    $D$ is a two-dimensional array with $m+1$ rows and $n+1$ columns
begin
    for $j \leftarrow 0$ to $n$ do
        $D[0,j] \leftarrow 0$
    end for
    for $i \leftarrow 0$ to $m-1$ do
        $D[i+1,0] \leftarrow 0$
        for $j \leftarrow 0$ to $n-1$ do
            $D[i+1,j+1] \leftarrow \max\{0, D[i,j] + Sub(p_1[i],p_2[j]), D[i,j+1] + Del(p_1[i],p_2[j]), D[i+1,j] + Ins(p_2[j],p_1[i])\}$
        end for
    end for
    return $\max_{i,j}\{D[i,j]\}/\min\{m,n\}$
end
```

Figure 4: Algorithm for evaluating the local similarity

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 $\gamma = (|\mathcal{P}| - 0.2)/|\mathcal{P}|$.

Definition 11 (Pattern-Based Similarity)
The Pattern-Based Similarity S_P *of an object* o_i *w.r.t. the last element of the current pattern* p_c *is defined as*

$$S_P(o_i) = \frac{\sum_{p \in \mathcal{P}_i} \textit{local-similarity}(p, p_c)}{\max_i \left\{ \sum_{p \in \mathcal{P}_i} \textit{local-similarity}(p, p_c) \right\}} \quad (19)$$

$\max_i \left\{ \sum_{p \in \mathcal{P}_i} \textit{local-similarity}(p, p_c) \right\}$ *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 object and recommend the objects most similar to it. Let us introduce the following definition.

Definition 12 (Recommendation grade)
The recommendation grade ρ *of an object* o_i*, given the current pattern* p_c *and the last element* o_c *in* p_c*, is defined as*

$$\rho(o_i) = \alpha_M \cdot S_M(o_i, o_c) + \alpha_P \cdot S_P(o_i) \quad (20)$$

α_M *and* α_P *being two weighting factors.*

The k objects in $\mathcal{O}_c$ exhibiting the higher values of ρ are the items that the system recommends to request next.

7. IMPLEMENTATION

In the previous sections we have focused our attention on presenting the main ideas behind our work, describing how computer vision techniques may be combined with the use of high level descriptors and log data in order to design a multimedia database browsing system. In this section we address some fundamental implementation issues. In particular we discuss how to tune the system, by setting the several parameters we have introduced, and how to make our solution scalable.

7.1 System tuning

Several parameters have been introduced along the paper for weighting the contribution of different terms. Let us discuss the strategy we used to select good values for these parameters.

In equation 4 the distance $d_{features}$ is defined as a weighted sum of color and texture distances. A features based distance or similarity metric is usually an attempt to reproduce the 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[3] to judge the similarity – in terms of visual appearance only – between these images as a grade between 0 and 10. We then determined the values of α_{col} and α_{tex} that maximized the correlation between the average values of human judged similarity and the values of $S_F = 1 - d_{features}$. In conclusion we obtained $\alpha_{col} = 0.67$ and $\alpha_{tex} = 0.33$.

In the definition of Taxonomy Based Similarity (equation 7) two parameters, α and β, are used to scale the contribution of shortest path length and depth respectively, by tuning the slope of the two exponential curves. Li et al. [12], 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 [16] and Miller-Charles [13]. They tested several similarity metrics on a standard set of word pairs from WordNet [14]. We repeated their experiments on a set of term pairs from our taxonomy, obtaining $\alpha = 0.27$ and $\beta = 0.59$ (α and β are not requested to sum up to 1).

Equation 9 defines the Index Distance Metric as a weighted sum of $d_{features}$ and $d_{taxonomy}$. In order to select good values for the weighting parameters α_F and α_T we carried out an experiment similar to the one used for selecting the values of α_{col} and α_{tex} in equation 4. We asked to a different group of about 40 people to judge the similarity between the pairs of pictorial images used in the previous experiment, being aware of the semantic description of the paintings (author, genre and subject). We obtained $\alpha_F = 0.62$ and $\alpha_T = 0.38$.

In the definition of Recommendation Grade (equation 20) two parameters, α_M and α_P, are used to weight the contri-

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

bution 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 these reason we have set α_M and α_P such that α_P increases and α_M decreases as the length n_c of the current pattern p_c increases.

$$\alpha_M = 1/n_c \qquad \alpha_P = (n_c - 1)/n_c \tag{21}$$

When $n_c = 1$, i.e. when the user requests the first item, $\alpha_M = 1$ and $\alpha_P = 0$, so the recommended items are the k objects having the shortest distance from the requested object o_c, according to the distance function d_M. When $n_c = 10$, i.e. when the current pattern of the user is quite long, $\alpha_M = 0.1$ and $\alpha_P = 0.9$, so the recommendations are mainly determined by the analysis of previous patterns.

7.2 Scale Issues

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

In section 5.3 we have already mentioned that the M-tree has been adopted in order to index the images in the collection, while in section 6 we have used a k nearest neighbors query in defining the set of candidate objects. In [3], 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 image collection are quite rare once the system is set up. We can thus conclude that the first scale issue is well addressed.

On the other hand, the most challenging scale issue and one of most critical aspects of the whole system is the construction of the set $\mathcal{P}_\gamma$ defined by equation 15.

As discussed in section 6, the threshold γ is defined as a function of $|\mathcal{P}|$. This guarantees that the size of $\mathcal{P}_\gamma$ does not increase with $|\mathcal{P}|$, since the threshold becomes more restrictive. To make our solution scalable with respect to the size of $\mathcal{P}$ we need to define an efficient strategy to build the set $\mathcal{P}_\gamma$. There is no doubt that it is not feasible to compare each element in $\mathcal{P}$ to p_c in order to assess its inclusion in $\mathcal{P}_\gamma$. 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 $d = (1-$local-similarity$)$ for computing the distance between patterns and partitioning the metric space. The set $\mathcal{P}_\gamma$ can be thus determined by issuing a range query $\mathsf{range}(p_c, 1-\gamma)$, that selects all the patterns within a distance of $1-\gamma$ from p_c. We can finally conclude that the second scale issue is well addressed too.

It's worth pointing out that, while updates to the image collection are quite rare, updates to the pattern collection are very frequent and their number is directly proportional to the number of users. Although the dynamic management algorithms do not deteriorate performances, the great number of updates to the pattern collection could be a problem. For this reason log data about current users are maintained in a temporary data structure and permanently stored in the log only when the system is idle. In other words, the behavior of other users currently connected to the system is not taken into account in the recommendation process.

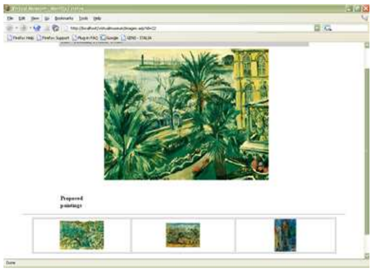

Figure 5: User interface

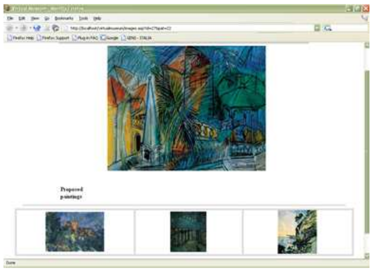

Figure 6: User interface

The above discussion fully addresses all the scale issues. However, more computations can be saved by better analyzing the algorithm in figure 4, used in equation 19 for computing the local similarity between each pattern $p \in \mathcal{P}_\gamma$ and the current pattern p_c. The algorithm computes an $(m+1) \times (n+1)$ matrix, where m and n are the lengths of p and p_c respectively. When a user requests a new item, the length of her pattern increases by one unit and a new matrix should be computed for each $p \in \mathcal{P}_\gamma$. 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.

8. EXPERIMENTS

In this section we show an example of how our system woks and report the very first experiments we have carried out for evaluating the impact of the proposed system on enhancing user's experience in a virtual museum.

A user that starts her tour in the virtual museum from the scratch can select any of the paintings in the exhibition by means of a standard search (by authors, by genre, and so on). As she makes the first request for a painting, the system begins to assist her visit. Figure 5 shows an example in which the first item to be selected has been 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 w.r.t the metric S_M. The user keeps on exploring the collection selecting, for example, one of the suggested pictures (see figure 6). At this point the system tries to propose both paintings similar to the current one and paintings inspected by similar users. Thus, among the recommended pictures in figure 6, there are two paintings that are similar to the current one and a painting, apparently not related to the ones inspected so far by the user, that has been proposed since it was requested by one or more users with a similar behavior and a similar usage pattern. We remark that the user is not required 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.

In order to evaluate the impact of the system on the users we have carried out the following experiment. We have asked two group of about 60 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 20 people to browse the collection using the standard search capabilities. After this trial we asked them to browse once again the collection, with the assistance of the recommender system, and express their opinion about the capability of the system to improve user experience. 73% of the people involved in the experiments found the system helpful, while the remaining 27% of the people said they were not able to appreciate significant differences with traditional browsing systems.

9. DISCUSSION AND CONCLUSIONS

In this paper we have presented a novel approach for managing collections of images in a museum scenario, considering both semantic concepts and low-level visual features in order to personalize the retrieval and presentation of multimedia data. The recommendation is obtained through the design of a pattern comparison algorithm which gives the users recommendations and assistance based on the behavior of previous visitors.

A prototypal system has been implemented using an appropriate indexing strategy, in order to address scale issues. We have shown that the proposed system provides the following interesting insights: (i) the recommendation algorithm does not use any preliminary knowledge about the users' behavior; (ii) the recommendation is produced using both visual and semantic description; (iii) the impact on the users at this early stage of the experimentations is promising.

Several issues remain open, most notably in extending our analysis and experiments to more general scenarios and different kind of multimedia data, such as video. In addition, more sophisticated visual features and novel matching algorithms might be adopted for improving the similarity search. Eventually, how to create an adequate semantic taxonomy for different realms still remains challenging.

10. REFERENCES

[1] M. Albanese, G. Boccignone, G. Moscato, and A. Picariello. Image similarity based on animate vision: the information path algorithm. In *Proc. of 8th Int Workshop on Multimedia Inf. Systems*, 2002.

[2] D. Bridge, R.Weber, and C. G. von Wangenheim. Product recommendation systems: A new direction. *ICCBR*, 2001.

[3] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of 23rd International Conference on VLDB*, pages 426–437, 1997.

[4] C. Drummond, D. Ionescu, and R.Holte. Intelligent browsing for multimedia application. In *Proc. of MULTIMEDIA'96*, pages 386–389, 1996.

[5] L. Dudoignon, E. Glemet, H. Heus, and M. Raffinot. High similarity sequence comparison in clustering large sequence databases. In *Proc. of Bioinformatics Conf.*, pages 228–236, 2002.

[6] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Trans. on Internet Technology*, 3(1):1–27, 2003.

[7] C. Grigorescu and N. Petkov. Distance sets for shape filters and shape recognition. *IEEE Trans. on Image Processing*, 12(10):1274 – 1286, 2003.

[8] B. Krulwich and C. Burkey. Learning user information interests throught extraction of semantically significant phrases. In *Proc. of the AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[9] A. Kushki, P. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos. Retrieval of images from artistic repositories using a decision fusion framework. *IEEE Trans. on Image Proc.*, 13(3):1057–7149, 2004.

[10] R. Leow and K. Taylor. Efficient web access to distributed biological collections using a taxonomy browser. In *Proc. of 12th Int. Conf. on Scientific and Statistical Database Management*, 2000.

[11] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, 6:707–710, 1966.

[12] Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. on Knowledge and Data Eng.*, 15(4):871–882, 2003.

[13] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[14] G. A. Miller. Wordnet: An on-line lexical database. *Int. Journal of Lexicography*, 3(4):235–312, 1990.

[15] W. Niblack, S. Yue, R. Kraft, A. Amir, and N. Sundaresan. Web-based searching and browsing of multimedia data. In *Proc. of IEEE Int. Conf. on Multimedia and Expo*, 2000.

[16] H. Rubenstein and J. Goodenough. Contextual correlates of synonymy. *Comm. ACM*, 8, 1965.

[17] J. Rucker and M. J. Polano. Siteseer: Personalized navigation for the web. *Communications of the ACM*, 40(3):73–75, 1997.

[18] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[19] B. Smyth and P. Cotter. A personalized television listings service. *Comm. ACM*, 2000.

[20] P. van Beek, I. Sezan, D. Ponceleon, and A. Amir. Content description for efficient video navigation, browsing and personalization. In *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 40–44, 2000.

[21] G. Van de Wouwer, P. Scheunders, S. Livens, and D. Van Dyck. Wavelet correlation signatures for color texture characterization. *Pattern Recognition*, 32(3):443–451, 1999.

[22] B. Xiao, E. Aimeur, and J. M. Fernandez. Pcfinder: An intelligent product recommendation agent for e-commerce. *Proc. of the IEEE Int. Conf. on E-Commerce*, 2003.