Recommending multimedia visiting paths in cultural heritage applications

Published version:

DOI:10.1007/s11042-014-2062-7

I. Bartolini; V. Moscato; R.G. Pensa; A. Penta; A. Picariello; C. Sansone; M.L. Sapino. Recommending multimedia visiting paths in cultural heritage applications. MULTIMEDIA TOOLS AND APPLICATIONS. 75 (7) pp: 3813-3842.
DOI: 10.1007/s11042-014-2062-7

Noname manuscript No.
(will be inserted by the editor)

Recommending Multimedia Visiting Paths in Cultural Heritage Applications

Ilaria Bartolini[1] · Vincenzo Moscato[2] · Ruggero G. Pensa[3] · Antonio Penta[3] · Antonio Picariello[2] · Carlo Sansone[2] · Maria Luisa Sapino[3]

Received: date / Accepted: date

Abstract The valorization and promotion of worldwide Cultural Heritage by the adoption of Information and Communication Technologies represent nowadays some of the most important research issues with a large variety of potential applications. This challenge is particularly perceived in the Italian scenario, where the artistic patrimony is one of the most diverse and rich of the world, able to attract millions of visitors every year to monuments, archaeological sites and museums. In this paper, we present a general recommendation framework able to uniformly manage heterogeneous multimedia data coming from several web repositories and to provide context-aware recommendation techniques supporting intelligent multimedia services for the users - i.e. dynamic *visiting paths* for a given environment. Specific applications of our system within the cultural heritage domain are proposed by means of real case studies in the mobile environment related both to an outdoor and indoor scenario, together with some results on user's satisfaction and system accuracy.

1 Introduction

It is widely agreed that the purpose of Cultural Heritage exhibitions is rapidly moving from an old vision, that provides a tourist with static information consisting of a large amount of cultural signs, to novel personalized services, matching the visitors' personal goals and behaviors by considering their cultural characteristics and preferences and *context* information.

· (1) University of Bologna, DISI, Viale Risorgimento 2, 40136, Bologna, Italy
E-mail: i.bartolini@unibo.it,
· (2) University of Naples Federico II, DIETI, via Clusio 21, 80125, Napoli, Italu
E-mail: {vmoscato,antonio.picariello,carlo.sansone}@unina.it,
· (3) University of Torino, DI, Corso Svizzera 185, I-10149, Torino, Italy
E-mail: {pensa,penta,mlsapino}@di.unito.it.

As reported by Bowe et al. [20], this "personalization" may be considered the shift towards a "user-centered information dialog" between a cultural space and its visitors.

The interactive dialog is surely interesting for a "virtual" cultural site: following the most advanced trends in Computer Science and Engineering such as web services, semantic web and recommender systems, a modern exhibition web site provides the users with personalized and interactive services. This enhances the experience of a virtual visitor, who spends time and usually money to discover the "deep" secrets of the exhibition and gather information about its points of interests.

However, also "physical" sites may take advantages of these modern techniques, for example connecting the visitors to the virtual worlds by means of sophisticated sensor networks. In addition, the user experience could be surely enhanced if, instead of using classic "touristic" guiding devices, she/he could be embedded in a cultural environment with a number of functionalities for representing the relevant information derived from the available digital sources about cultural heritage, such as text descriptions, pictures, and videos. In this way, a tourist would be given the opportunity of enjoying multimedia stories in real time, thus enriching his/her cultural experience.

Offering virtual navigation environments turns out to be particularly important for the valorization and promotion of worldwide Cultural Heritage. This need is particularly perceived in a country like Italy, where the artistic patrimony represents a worldwide resource of inestimable value, attracting millions of visitors every year to monuments, archaeological sites and museums.

Several points need to be addressed to create effective virtual navigation environment that can be easily customizable for a variety of applications: (*i*) information about visitors and their personal interests need to be dynamically acquired; (*ii*) the personalized functionalities that can be provided in a real space need to be identified and designed, and (*iii*) solutions to connect the "virtual" and the "physical" user experience need to be selected.

In this paper, our goal is to meet the discussed requirements "extending" classical recommendation techniques (*content-based*, *collaborative filtering* and *hybrid* strategies), usually exploited for facilitating the browsing of web large data repositories, to support useful *context-aware* services (e.g. a multimedia touristic guide) within a single framework. Such services must assist users when visiting cultural environments (indoor museums, archaeological sites, old town centers) containing several *cultural Points Of Interest* - POIs - (e.g. paintings of museum rooms, buildings in ancient ruins or in an old town center, etc.) correlated with a large amount of multimedia data available in multiple web repositories.

In particular, we present a general multimedia recommender system - that is an extension of our previous work [13] - able to uniformly manage heterogeneous multimedia data and to provide context-aware recommendation techniques supporting intelligent services - i.e. dynamic *visiting paths* - useful for the users during the exploration of different kinds of cultural sites.

In addition, we describe real case studies in the mobile environment, related both to an outdoor and to an indoor scenario, together with some results on user's satisfaction and system accuracy.

The paper is organized as follows. Section 2 illustrates the main related work concerning multimedia recommender systems and their application for Cultural Heritage. Section 3 presents at a glance a functional overview of our recommender system. Section 4 describes the techniques used for multimedia data management, while Section 5 details the proposed recommendation strategy. Section 6 outlines the chosen case studies with the related implementation details and Section 7 reports some experiments. Finally, Section 8 discusses some conclusions and future work.

2 Related Work

In its most common formulation, the *recommendation problem* is the problem of estimating *ratings*, or *utilities*, which quantify the degree of interest for a *user* for the set of *items* that have not yet been seen by him.

In *Content-Based Filtering* [43], the utility for a user of a given item is estimated using the utilities assigned by the same user to other similar items. For example, in a movie recommendation application, in order to recommend movies to a user, content-based filtering tries to recognize the commonalities among the movies the user has rated highly in the past (specific actors, directors, genres, subject matter, etc). Then, only the movies that have a high degree of similarity to the user's preferred ones are recommended. These techniques do not benefit from the great amount of information that could be derived by also analyzing the behavior of other users. Moreover, the effectiveness of the methods strongly depends on the performance of the available feature extraction algorithms, and on the ability of recognizing as similar but distinct objects with the same extracted features. Another intrinsic potential problem is *overspecialization*: the system can only recommend items that are similar to those already rated by the user.

Collaborative Filtering [2] is the process of filtering or evaluating items using the opinions of other people. Thus, unlike content-based recommendation methods, collaborative systems focus on the similarity among users: to predict the utility of items for a given user they rely on the rankings assigned to the same items by users similar to the considered one. Collaborative filtering takes its root from something human beings have been doing for centuries: sharing opinions with each others [35,46]. A major challenge faced by collaborative filtering is the need to associate each user to a set of other users having similar profiles. Thus, in order to make any recommendations, the system has to collect data either asking for explicit ratings from users, or through non intrusive profiling algorithms implicitly logging actions performed by users. Once the data has been gathered, there are two basic ways of filtering through it, to make predictions. The most basic method is *passive filtering*, which simply uses data aggregates to make predictions (such as the average rating for an item). Each user will be given the same predictions for a particular item (e.g.

digg.com). *Active filtering* instead uses patterns in user history to make predictions, thus obtaining user-specific and context-aware useful recommendations (e.g. *Amazon*). An important limitation of collaborative filtering systems is the *cold start problem*, that describes situations in which a recommender is unable to make meaningful recommendations due to an initial lack of ratings, thus degrading the filtering performance. Cold start filtering needs to be addressed in three prosily frequent scenarios: a *new user* joins the system, a *new item* is available to be recommended, but - being new - has never been rated, and a *new community* is detected, and there are "no community" data available.

Content-based filtering and collaborative filtering may be manually combined by the end-user specifying particular features, essentially constraining recommendations to have certain content features. More often they are automatically combined in the so called *hybrid approach* [16,9,15,44] that helps to overcome some limitations of each method. Different ways to combine collaborative and content-based methods into a hybrid recommender system can be classified as follows: (*i*) implementing collaborative and content-based methods separately and combining their predictions; (*ii*) incorporating some content-based characteristics into a collaborative approach; (*iii*) incorporating some collaborative characteristics into a content-based approach; (*iv*) constructing a general unifying model that incorporates both content-based and collaborative characteristics.

A recommendation strategy eventually should be able to provide users with the more relevant information depending on the *context* [23,32] (i.e. user preferences, user location, observed objects, weather and environmental conditions, etc. as in *Context Aware Recommendation Systems* [34]). In the *Contextual Pre-filtering* techniques context information are used to initially select the set of relevant items, while a classic recommender is used to predict ratings. In *Contextual Post-filtering* approaches context is used in the last step of the recommending process to contextualize, for each user, the output of a traditional recommender.

More recently, all the above discussed strategies have been extended to multimedia realm (e.g. multimedia repositories, digital libraries, multimedia sharing system, etc.) with the aim of considering in the more effective way the multimedia content of recommended objects, both in terms of low-level and high-level characteristics (i.e. multimedia features and semantics [1]), in the recommendation process together with user's social behavior and preferences.

As for content-based techniques, [40] proposes a method that exploits some ontologies for ranking items' relevance in the electronic paper domain, while in [28] a content based filtering has been applied to music data using decision trees. In the framework of multimedia sharing system, [42] introduces a recommender system that uses two ontologies (one for multimedia objects and one for users) in the context of a photo sharing system. To generate suggestions a new concept of *multirelational* social network was introduced, covering both direct as well as multimedia object-based relationships that reflect social

[1] For multimedia feature extraction and mining, good surveys are [53],[24],[19].

and semantic links between users. The authors in [41] propose a content-based recommender architecture which explores information that is available at the time users enhance content in order to capture a certain level of semantic information from the multimedia content and from user preferences, that is at the base of their video recommender system.

Among collaborative-filtering proposals, Kim et al. [36] propose a collaborative filtering-based multimedia contents recommender system in P2P architectures that rates multimedia objects of nearest peers with similar preference through peer-based local information only. Tseng et al. [49] propose a system, which combines discovered relations between user preferences and conceptualized multimedia contents by annotation and association mining techniques, to assist users in making a decision among a massive amount of multimedia items (images, videos and music).

Among the hybrid solutions, the uMender system [47] exploits context information, musical content and relevant user ratings to perform music recommendations on mobile devices. A framework for recommendation of multimedia objects based on processing of individual ontologies is proposed in [31]: the recommendation process takes into account similarities calculated both between objects (metadata) and users ontologies, which reflect the social and semantic features existing in the system. Finally, low and high level features have been used to define the similarity among multimedia items in [4,6,7]: this measure is then used to compare patterns of past users in order to identify users with similar browsing behavior.

In the area of *Cultural Heritage*, there are several multimedia systems designed and developed to help the user's exploration of available multimedia content [51,50,37,22]. Even if these systems have absorbed previous results coming from different multimedia research projects, they also pose new challenges in the recommendation process such as how different multimedia modules can be efficiently integrated, how conflicts coming from the management of heterogeneous data can be resolved or how the user with his/her preferences, habits and social relationships can be considered. In [10], the authors describe the latest approaches related to how we can model and represent the users in the context of cultural heritage applications and how we can use those models to reason with regard to the available information. All these approaches are useful to perform a *personalization* of the services [33,37,52,1,18].

In a nutshell, the majority of approaches to recommendation in the multimedia realm generally exploit *high level* metadata - extracted in automatic or semi-automatic way from *low level* features - that are in different manners correlated and compared with user preferences.

These approaches suffer from several drawbacks: (*i*) it is not always possible to extract in automatic and effective way useful high level information from multimedia features (automatic annotation algorithms have not always high performances); (*ii*) for some kinds of multimedia data there does not exist a precise correlation between high and low level information (e.g. in images the concept of "moon" is related to a region with a circular shape and white color with a given uncertainty); (*iii*) there is not always available explicit and useful information (*knowledge*) about user preferences and feedbacks (e.g. usually a user to retrieve information from a multimedia system needs a registration); (*iv*) in the recommendation process sometimes it is useful to take into account features of the objects the user is currently observing as content information (e.g. the main colors of a painting are often an indication of the related artistic movement or school).

Here, we present a general multimedia recommender system able to uniformly manage heterogeneous multimedia data and to provide context-aware recommendation techniques supporting intelligent multimedia services useful for the users. It addresses several drawbacks of state-of-the-art approaches:

- analyzing in a separate way low and high level information, since both contribute to determine the utility of an object in the recommendation process;
- exploiting system logs to implicitly determine information about users and the related community, considering their browsing sessions as a sort of "ratings";
- considering as relevant content for the recommendation the features of the object that a user is interested in (e.g. the item user is watching) ;
- exploiting user preferences and other context information (e.g. user location) to perform a pre-fitering of the candidate objects for recommendation;
- arranging the obtained recommendations in dynamic visiting paths that take into account possibile changes in user needs and in the surrounded environment.

3 System Overview

Our system has to support the described recommendation framework, providing the following functionalities:

- *fetching* of multimedia contents (i.e. raw data and the related annotation) from several web repositories;
- *indexing* of multimedia data exploiting both low and high level descriptors in order to realize a content-based retrieval;
- *recommending* the multimedia items to users using information about their preferences together with other context information (e.g. the item user is watching, user location, etc.);
- *arranging* recommended objects in visiting paths that can dynamically change with the context.
- *delivery* and *presentation* of generated visiting paths to user devices.

Figure 1 describes at a glance a functional overview of the proposed system in terms of its main components, that we are detailing in the following.

The *Multimedia Data Management Engine* (MDME) is responsible for: (*i*) accessing by the *Indexing and Access Manager* module to the media contents present in several data sources (*Multimedia Data Repositories*), (*ii*) extracting from multimedia data, by the *Feature Extraction* module, high and low

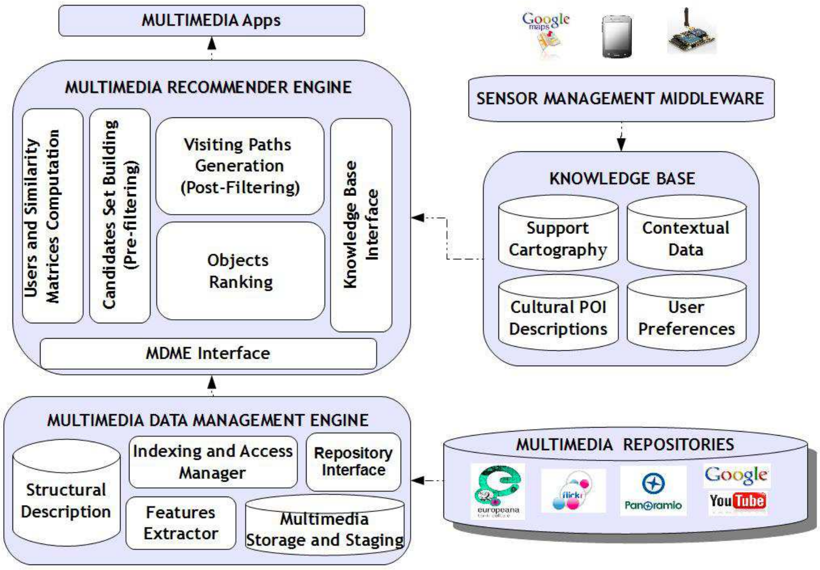


Fig. 1 System Overview.

level features useful both for indexing aims and to obtain a structured representation of the data (*Structural Description*). In particular, the *Repository Interface* provides a set of Restful API to communicate with the different multimedia repositories (e.g., *Wikipedia*, *Flickr*, *Europeana*, *Panoramio*, *Google Images*, *YouTube*, etc.). The multimedia data gathered from these sources are then stored in a *Multimedia Storage and Staging* area.

The *Sensor Management Middleware* is responsible for deriving, on the base of information accessible via physical sensors (e.g. GPS, WSN), Web-services/API or wrapping techniques, the "knowledge" related to the context in which the user is located. In particular, the *Knowledge Base* of our system consists of the *Contextual Data* (e.g. weather and environmental conditions registered for the considered place), *User Preferences* (explicitly and implicitly captured), *Cultural POI Descriptions* (in terms of general information and "pointers" to the multimedia data related to a cultural point of interest) and a *Support Cartography* useful to geo-localize users and visualize their positions with respect to POIs.

The *MultiMedia Recommender Engine* provides a set of recommendation facilities for multi-dimensional and interactive browsing of multimedia data related to cultural POIs. In particular, exploiting context information about user location and preferences, the *Candidate Set Building* module selects a set of *candidate* objects for recommendation; successively, the *Objects Ranking* module performs a ranking of such candidates exploiting a proper strategy (that uses the *Users and Similarity Matrices Computation* module). Finally, the *Visiting Paths Generation* module dynamically selects a subset of candidates, on the base of the object that a user is currently watching and context information (e.g., environmental conditions), and eventually arrange them in *visiting paths* as in a touristic guide. All information about the context and multimedia data necessary for the recommendation aims are collected from the system Knowledge Base and Multimedia Data Management Engine using the primitives provided by *Knolwedge Base Interface* and *MDME Interface*, respectively.

Each user device is then equipped with a *Multimedia Guide App* that allows the fruition of multimedia contents and visualization of visiting paths.

4 Management of Multimedia Data

Our data and retrieval models are inspired by the WINDSURF ones [12] as follows.

4.1 Data model and feature extraction

We have a database $\mathcal{O}$ of M multimedia objects, $\mathcal{O} = \{O^1, \ldots, O^M\}$, such as images, videos, and documents, where each object O is composed of m_O *elements*, $O = \{o_1, \ldots, o_{m_O}\}$ representing regions of an image, shots of a video, and parts of a document, respectively. Each element o is described by way of *low level features* F^l extracted by the *Features Extraction* module that represent, in an appropriate way, the content of o (e.g., the color distribution of image's regions or of a video keyframe). Although we consider for an image/keyframe its regions and for each region its visual features, representing an image/keyframe as a set of local features, like SIFT [39] and SURF [17], is also easily achievable within the WINDSURF framework.

In particular, images are segmented into regions, where pixels included in a single region share the same visual content (i.e., color/texture) [12]. Videos are first segmented into shots [14]. Then, each shot is represented by a single representative keyframe (e.g., the first frame of the shot). Each keyframe is first segmented into visually coherent regions, then color/texture features are extracted for each keyframe region [12]. Documents are modeled as a set of pages. The content of each page is represented by means of a set of relevant keywords extracted using $tf \times idf$ values after stopping & stemming stages [45].

In order to enrich data representation, objects are also annotated by the *Features Extractor* module with high level (semantic) descriptors F^h (e.g., annotations concerning the history of a paint, experts' descriptions of an ancient manuscript, visitors' descriptions and reviews, keywords describing what a video shot (or an image) is related to, etc.).

Semantic descriptors can be of two types: (i) meta-data, manually provided by users and/or visitors or automatically acquired by external multimedia repositories (such as Wikipedia, Google, Flickr, Europeana, YouTube, etc.) through the *Repository Interface* APIs and maintained in the *Multimedia Storage and Staging* area; (*ii*) (semi-)automatically provided annotations in the form of simple keywords (or *tags*) or semantic tags, i.e., concepts taken from tree-structured taxonomies. Semantic tags can be regarded as descriptions for objects that is more precise and powerful than tags (with no inherent semantics), yet not so complex to derive as concepts of RDF-like ontologies (whose semantics might not be so easy to grasp by end-users). Meta-data are processed as pages of documents and modeled by means of a set of relevant keywords.

We define the universe of semantic descriptors $\mathcal{F}$ as the union of all annotations (both meta-data and (semi-)automatically provided labels) associated to objects in $\mathcal{O}$. The association between an object o and its descriptors is modeled by way of a membership relation $R \subseteq \mathcal{O} \times \mathcal{F}$ that indicates that object O has assigned an annotation in $\mathcal{F}$.

4.1.1 (Semi-)automatic annotation stage

Tags and semantic tags are semi-automatically assigned to objects by means of a multimedia object annotator that, starting from a training set of pre-annotated objects, predicts sets of good keywords which effectively characterize the content of new untagged objects.

Here we provide only some basic intuition on how tag suggestion works, a detailed description being given in [11].

The annotation process is essentially based on the idea of suggesting those (semantic) tags that are assigned to objects similar to the target object. To this end, a nearest-neighbors search is first performed using low-level features, which determines a set of objects similar to the target one. For all (semantic) tags associated to at least one returned object, a frequency score is then computed as the number of objects annotated with such (semantic) tag. Then, in order to remove unrelated (semantic) tags, thus to improve the prediction accuracy, a correlation analysis is performed on each pair of (semantic) tags. The so-resulting correlation scores are then used to determine whether or not the two (semantic) tags are connected in a new graph whose nodes are the candidate tags, and where the node of a (semantic) tag is given weight equal to the frequency score. Finally, a maximum-weight clique of such a graph is determined, with nodes in the clique determining which are the tags to be suggested.

Note that, while for objects of type image tags are directly associated to images, when annotating videos, we are able to predict tags not only for shots but even for videos, by suitably propagating most representative tags at the shot level to the video level [14].

4.2 Retrieval model

With respect to the retrieval model, given a query object $Q = \{Q_1, \ldots, Q_m\}$ composed of m elements, and an element distance function δ, that measures the dissimilarity of a given pair of elements (using their features), we want to determine the top-k objects in $\mathcal{O}$ that are the most similar to Q.

Low-level similarity between objects is numerically assessed by way of an object distance function d_{F^l} that combines together the single element distances into an overall value. Consequently, object O^a is considered better than O^b for the query Q iff $d_{F^l}(Q, O^a) < d_{F^l}(Q, O^b)$ holds [30]. The computation of the object distance d_F is obtained by combining three basic ingredients: (i) the element distance δ, (ii) the set of constraints that specify how the component elements of the query Q have to be matched to the component elements of another (database) object O, and (iii) the aggregation function that combines distance values between matched elements into an overall object distance value (e.g., a simple average of distance values between matched elements).

Often, the overall object distance is computed by aggregating scores of the best possible matching, i.e., the one that minimizes the overall object distance; in this case, the computation of d_{F^l} also includes the resolution of an optimization problem in the space of possible matchings between elements of Q and elements of O. The efficient resolution of queries over low level features is ensured by the *Data Indexing and Access Manager* module which supports indices built on top of elements (e.g., image regions, and video shots) based on the *M-tree* metric index [21].

In details, image regions are compared according to their visual features using *Bhattacharyya distance* metric δ; region scores are opportunely matched by solving a one-to-one matching problem, where each element of a document can be only matched to at most one element of the other document, and vice versa. Then a "biased" average d_{F^l} is used to aggregate distance values of matched elements. This defines an assignment problem, which can be solved using the Hungarian Algorithm in $O(n^3)$ time [38]. With respect to videos, being each shot modeled by a single representative keyframe, shots comparison can be assessed by means of the above image distance function d_{F^l}. Whole videos are compared by aggregating the distances between shots (i.e., their representative keyframes). Comparison between document pages is performed by applying the vector space model [45] on pages' features. Whole documents are compared by aggregating distances between their pages.

With respect to high level features, following the well known *keyword-based paradigm*, given a user-provided set of keywords as query semantic concepts, objects are selected by the *Indexing and Access Manager* module by applying a $co-occurrence$-based distance function d_{F^h} on $\mathcal{F}$. The search provides the set of objects (i.e., images, videos/shots, documents) that share at least one keyword with the input. This can be carried out efficiently by exploiting the existence of indices, e.g., inverted files.

Finally, both low level features and high level semantic descriptors concur to determine the *multimedia relatedness* $d(O^i, O^j)$ among two objects. In details, if O^i and O^j are of the same type (e.g., we are comparing two images), we define their global distance as the average between the contribution given by low level features and the one provided by semantics, that is: $d(O^i, O^j) = (d_{F^l}(O^i, O^j) + d_{F^h}(O^i, O^j))/2$; on the other hand, if we are comparing objects of different type (e.g., a document with a video), their multimedia relatedness equals to their semantic distance only, i.e., $d(O^i, O^j) = d_{F^h}(O^i, O^j)$.

5 Context-Aware Multimedia Recommendation Services

The basic idea behind our proposal is that when a user is near to a cultural POI, the recommender system has to be able to:

1. determine a set of useful *candidate* objects for the recommendation, on the base of user location, needs and preferences (*pre-filtering stage*);
2. opportunely rank these objects exploiting their intrinsic features and users' past behaviors (*ranking stage*);
3. dynamically, when a user "selects" one or more of the candidate objects, determine the list of most suitable objects (*post-filtering stage*) and eventually arrange such items in apposite *visiting paths* considering other context information such as weather or environmental conditions.

In the following, we are detailing each one of the described stages.

5.1 Pre-filtering stage

Each object subject to recommendation may be represented in different and heterogeneous feature spaces. For instance, the picture of a monument may be described by annotations concerning history of the monument, the materials it has been built with, low-level image features, experts' descriptions, visitors' descriptions and reviews, and so on. Each of these sets of features contributes to the characterization of the objects to different extents. Hence, it is important to consider congruently each type of descriptor during the recommendation process.

The first step of the pre-filtering stage consists in clustering together "similar" objects, where the similarity should consider all (or subsets of) the different spaces of features. To this purpose, we employ high-order star-structured co-clustering techniques [29] to address the problem of heterogeneous data clustering. In this context, the same set of objects is represented in different feature spaces. Such data represent objects of a certain type, connected to other types of data, the features, so that the overall data schema forms a star structure of inter-relationships.

The co-clustering task consists in simultaneously clustering the set of objects and the set of values in the different feature spaces. In this way we obtain a partition of the objects influenced by each of the feature spaces and at the same time a partition of each feature space.

The pre-filtering stage leverages the clustering results to select a set of candidate objects by using the user's profile, which is modeled as sets of descriptors in the same spaces as the objects' descriptors.

We now provide the formalization of our problem. Let $\mathcal{O} = O^1, \ldots, O^M$ be a set of M multimedia objects and $\mathcal{F} = \{F^1, \ldots, F^N\}$ be a set of N feature spaces. A dataset can be viewed under the different views given by the different feature spaces F^k. Therefore, the view k is associated with each feature space F^k. Let $\mathcal{SD} = \{SD^1, \ldots, SD^N\}$ be a star-structured dataset over $\mathcal{O}$ and $\mathcal{F}$. Each value $sd^k_{st} \in SD^k$ corresponds to the counting/frequency/presence of feature $f^k_t \in F^k$ in object $O^s \in \mathcal{O}$. Without loss of generality, we assume that $sd^k_{st} \in \mathbb{N}$. An example of two-views star-structured data is given in Figure 2(a).

In our recommendation problem, a user is represented as a set of vectors $U = \{\boldsymbol{u}^1, \ldots, \boldsymbol{u}^N\}$ in the same N feature spaces describing the objects. Each vector $\boldsymbol{u}^k$ is updated each time the user visits (or re-visit) an object, by considering the object features in each space at the instant of the visit. Let $O^U_v \in \mathcal{O}$ be the set of objects visited by the user represented by U. Hence, the component of vector $\boldsymbol{u}^k \in U$ related to feature f^k_t is computed as:

$$u^k_t = \sum_{O^s \in O^U_v} d^k_{st}$$

Clearly, the action of updating the vectors in U can be performed incrementally, as the user visit new objects. Notice that, thanks to this approach, users are not described by sets of objects, but by sets of features that characterize the objects they visit, like or browse.

The first step consists in identifying clusters of similar objects in $\mathcal{O}$ by leveraging all feature spaces by means of a star-structured data co-clustering approach. Its goal is to find a set of partitions $\mathcal{Y} = \{Y^1, \ldots, Y^N\}$ over the feature set $\mathcal{F} = \{F^1, \ldots, F^N\}$, and a partition $\mathcal{X}$ of the object set $\mathcal{O}$ by optimizing a certain objective function. To solve the high-order star-structured coclustering problem, several algorithms have been proposed based on different approaches. In this work, we adopt a parameter-less iterative algorithm that maximizes the Goodman-Kruskal τ, a statistical measure of association that automatically identifies a congruent number of high-quality co-clusters [29]. Goodman and Kruskal τ measure [26] is one of them that estimates the association between two categorical variables X and Y by the proportional reduction of the error in predicting X knowing or not the variable Y:

$$\tau_{X|Y} = \frac{e_X - E[e_{X|Y}]}{e_X}$$

Evaluating the quality of the partition of objects, given the partitions of features, is formalized as follows. The partition of objects is considered as the dependent variable X, and the N partitions of the feature spaces are considered as many independent variables $\mathcal{Y} = \{Y^1, \ldots, Y^N\}$. Each variable $Y^k \in \mathcal{Y}$ has n_k categories $Y^k_1, \cdots, Y^k_{n_k}$, corresponding to n_k feature clusters, with probabilities $q^k_1, \ldots, q^k_{n_k}$ and X has m categories $X_1, \cdots, X_m$ corresponding to m

	$\boldsymbol{f}_1^1$	$\boldsymbol{f}_2^1$	$\boldsymbol{f}_3^1$	$\boldsymbol{f}_4^1$
$\boldsymbol{O}^1$	sd_{11}^1	sd_{12}^1	sd_{13}^1	sd_{14}^1
$\boldsymbol{O}^2$	sd_{21}^1	sd_{22}^1	sd_{23}^1	sd_{24}^1
$\boldsymbol{O}^3$	sd_{31}^1	sd_{32}^1	sd_{33}^1	sd_{34}^1
$\boldsymbol{O}^4$	sd_{41}^1	sd_{42}^1	sd_{43}^1	sd_{44}^1
$\boldsymbol{O}^5$	sd_{51}^1	sd_{52}^1	sd_{53}^1	sd_{54}^1

	$\boldsymbol{f}_1^2$	$\boldsymbol{f}_2^2$	$\boldsymbol{f}_3^2$
$\boldsymbol{O}^1$	sd_{11}^2	sd_{12}^2	sd_{13}^2
$\boldsymbol{O}^2$	sd_{21}^2	sd_{22}^2	sd_{23}^2
$\boldsymbol{O}^3$	sd_{31}^2	sd_{32}^2	sd_{33}^2
$\boldsymbol{O}^4$	sd_{41}^2	sd_{42}^2	sd_{43}^2
$\boldsymbol{O}^5$	sd_{51}^2	sd_{52}^2	sd_{53}^2

(a)

	$\boldsymbol{Y}_1^1$	$\boldsymbol{Y}_2^1$	
$\boldsymbol{X}_1$	r_{11}^1	r_{12}^1	p_1^1
$\boldsymbol{X}_2$	r_{21}^1	r_{22}^1	p_2^1
	q_1^1	q_2^1	

	$\boldsymbol{Y}_1^2$	$\boldsymbol{Y}_2^2$	
$\boldsymbol{X}_1$	r_{11}^2	r_{12}^2	p_1^2
$\boldsymbol{X}_2$	r_{21}^2	r_{22}^2	p_2^2
	q_1^2	q_2^2	

(b)

Fig. 2 An example of a star-structured dataset consisting of two feature spaces F^1 and F^2 (a) and the contingency tables associated with a related star-structured co-clustering $(\mathcal{X}, Y^1)$ and $(\mathcal{X}, Y^2)$ (b).

object clusters. However, for each variable Y^k, the m categories of X have different probabilities $p_1^k, \cdots, p_m^k$, $k = 1 \cdots N$. Probabilities p_i^k and q_j^k are computed as follows:

$$p_i^k = \frac{\sum_{O^s \in X_i} \sum_t sd_{st}^k}{\sum_s \sum_t sd_{st}^k}, \qquad q_j^k = \frac{\sum_{f_t^k \in Y_j^k} \sum_s sd_{st}^k}{\sum_s \sum_t sd_{st}^k}$$

The joint probabilities between X and any $Y^k \in \mathcal{Y}$ are denoted by r_{ij}^k, for $i = 1 \cdots m$ and $j = 1 \cdots n_k$ and are computed as follows:

$$r_{ij}^k = \frac{\sum_{O^s \in X_i} \sum_{f_t^k \in Y_s^k} sd_{st}^k}{\sum_s \sum_t sd_{st}^k}$$

Figure 2(b) provides an example of co-clustering computed on the two-space star-structured data depicted in Figure 2(a).

The error in predicting X is the sum of the errors over the independent variables of $\mathcal{Y}$: $e_X = \sum_{k=1}^N \sum_{i=1}^m p_i^k (1 - p_i^k) = N - \sum_{k=1}^N \sum_{i=1}^m (p_i^k)^2$. $E[e_{X|\mathcal{Y}}]$ is the expectation of the conditional error taken with respect to the distributions of all $Y^k \in \mathcal{Y}$:

$$E[e_{X|\mathcal{Y}}] = \sum_k^N \sum_j^{n_k} q_j^k e_{X|Y_j^k} = \sum_k^N \sum_j^{n_k} q_j^k \sum_i^m \frac{r_{ij}^k}{q_j^k}(1 - \frac{r_{ij}^k}{q_j^k}) = N - \sum_k^N \sum_i^m \sum_j^{n^k} \frac{({r^k}_{ij})^2}{q_j^k}$$

The generalized Goodman-Kruskal's $\tau_{X|\mathcal{Y}}$ association measure is then equal to:

$$\tau_{X|\mathcal{Y}} = \frac{e_X - E[e_{X|\mathcal{Y}}]}{e_X} = \frac{\sum_k \sum_i \sum_j \frac{({r^k}_{ij})^2}{q_j^k} - \sum_k \sum_i (p_i^k)^2}{N - \sum_k \sum_i (p_i^k)^2} \qquad (1)$$

If we consider Y^k as a dependent variable, and X as an independent variable, the corresponding $\tau_{Y^k|X}$ is computed as follows:

$$\tau_{Y^k|X} = \frac{e_{Y^k} - E[e_{Y^k|X}]}{e_{Y^k}} = \frac{\sum_i \sum_j \frac{(r^k_{ij})^2}{p^k_i} - \sum_j (q^k_j)^2}{1 - \sum_j (q^k_j)^2} \tag{2}$$

The adopted co-clustering approach for star-structured data is formulated as a multi-objective combinatorial optimization problem which aims at optimizing $N+1$ objective functions based on Goodman-Kruskal's τ measure. The main procedure of the algorithm is sketched in Figure 5.1. The reader may refer to [29] for further algorithmic details.

```
Input: a star-structured dataset SD and an integer N_iter
Output: a coclustering (X, Y)
  Initialize Y^1, ..., Y^N, X with discrete partitions
  i ← 0
  T ← ∅
  for k = 1 to N do
    T^k ← ContingencyTable(X, Y^k, SD^k)
    T ← T ⋃ T^k
  end for
  while (i ≤ N_iter) do
    [X, T] ← OptimizeMultiObjectCluster(X, Y, T)
    for k = 1 to N do
      [Y^k, T^k] ← OptimizeFeatureCluster(X, Y^k, T^k)
    end for
    i ← i + 1
  end while
  return Y^1, ..., Y^N, X
```

Fig. 3 Pseudo-code of the adopted star-structured co-clustering algorithm [29].

To provide a first candidate list of objects to be recommended, we measure the *cosine similarity* of each user vectors associated to the k-th space, with the centroids of each object clusters in the k-th space. Let $\boldsymbol{x}^k_i$ be the centroid of cluster X_i in the feature space F^k. The t-th component of $\boldsymbol{x}^k_i$ is computed as:

$$x^k_i = \frac{\sum_{O^s \in X_i} d^k_{st}}{|X_i|}$$

and the cosine similarity between $\boldsymbol{u}^k$ and $\boldsymbol{x}^k_i$ is evaluated as

$$sim(\boldsymbol{u}^k, \boldsymbol{x}^k_i) = \frac{\boldsymbol{u}^k \cdot \boldsymbol{x}^k_i}{\|\boldsymbol{u}^k\| \|\boldsymbol{x}^k_i\|}.$$

For each space, the most similar object cluster is chosen leading to a set of N clusters $\mathcal{X}^c = \{X^c_1, \ldots, X^c_N\}$ of candidate objects. Then, two different strategies can be adopted to provide the pre-filtered list of candidate objects $\mathcal{O}^c$:

– **relaxed strategy**: the objects belonging to the union of all clusters are retained, i.e.,

$$\mathcal{O}^c = \bigcup_k X_k^c$$

– **strict strategy**: the most represented cluster in $\mathcal{X}^c$ is retained, i.e.,

$$\mathcal{O}^c = \underset{X_k^c \in \mathcal{X}^c}{\operatorname{argmax}} |X_l^c \in \mathcal{X}^c \ \ s.t. \ \ X_k^c \equiv X_l^c| \, .$$

The first strategy is suitable when user's vectors are associated to very small clusters (e.g., because the user likes very uncommon objects). In any other situation, the second strategy is the most appropriate. As an additional step, objects already visited/liked/browsed by the user can be filtered out. We do not filter-out these objects at the beginning of the pre-filtering stage because they are relevant for the co-clustering step. In fact they are likely to be involved in important cross-associations between sets of features and sets of objects.

Finally, provided that each object in $\mathcal{O}$ is georeferenced, the set of candidate objects $\mathcal{O}^c$ issued by the above-described process can be further refined by an ordering step. To this purpose, we employ the route distance between the user's current position and the position of each object in $\mathcal{O}^c$. Closer objects are on top of the items' list, while more distant ones are on its bottom. In conclusion, at the end of the pre-filtering stage, we provide an ordered list of candidate objects $\hat{\mathcal{O}}^c$ grouped by the related cultural POI (in this manner a user can easily choose items coming from more different cultural POIs).

5.2 The ranking and post-filtering stages

The main goal of these stages is to automatically and dynamically recommend to a user a subset of $\mathcal{O}^c$ on the base of one or more *target objects* opportunely selected from $\hat{\mathcal{O}}^c$, exploiting objects' intrinsic multimedia *features* and users past browsing *behaviors*.

In particular, we use a novel technique that some of the authors have proposed in previous works, combining low and high level features of multimedia objects, possible past behavior of individual users and overall behavior of the whole "community" [5,6,7].

Our basic idea is to assume that when an object O_i is chosen after an object O_j in the same *browsing session*, this event means that O_i "is voting" for O_j. Similarly, the fact that an object O_i is very similar in terms of multimedia features to O_j can also be interpreted as O_j "recommending" O_i (and viceversa). Thus, we model a browsing system for the set of candidate objects $\mathcal{O}^c$ as a labeled graph (G, l), where:

– $G = (\mathcal{O}^c, E)$ is a *directed graph*;

– $l : E \rightarrow \{pattern, sim\} \times R^+$ is a *labeling function* that associates each edge in $E \subseteq \mathcal{O}^c \times \mathcal{O}^c$ with a pair (t, w), where t is the type of the edge which can assume two enumerative values (*pattern* and *similarity*) and w is the weight of the edge.

  We list two different cases:

1. a *pattern label* for an edge (O_j, O_i) denotes the fact that an object O_i was accessed immediately after an object O_j and, in this case, the weight w_j^i is the number of times O_i was accessed immediately after O_j ;
2. a *similarity label* for an edge (O_j, O_i) denotes the fact that an object O_i is similar to O_j and, in this case, the weight w_j^i is the similarity between the two objects. Thus, a link from O_j to O_i indicates that part of the importance of O_j is transferred to O_i .

Given an object $O_i \in \mathcal{O}^c$, its *recommendation grade* $\rho(O_i)$ is defined as follows:

$$\rho(O_i) = \sum_{O_j \in P_G(O_i)} \hat{w}_{ij} \cdot \rho(O_j) \tag{3}$$

where $P_G(O_i) = \{O_j \in \mathcal{O}^c | (O_j, O_i) \in E\}$ is the set of predecessors of O_i in G, and $\hat{w}_{ij}$ is the normalized weight of the edge from o_j to o_i. For each $o_j \in O$ $\sum_{O_i \in S_G(O_j)} \hat{w}_{ij} = 1$ must hold, where $S_G(O_j) = \{O_i \in \mathcal{O}^c | (O_j, O_i) \in E\}$ is the set of successors of O_j in G.

In [5, 7], it has been shown that the ranking vector $R = [\rho(O_1) \ldots \rho(O_n)]^T$ of all the objects can be computed as the solution to the equation $R = C \cdot R$, where $C = \{\hat{w_{ij}}\}$ is an ad-hoc matrix that defines how the importance of each object is transferred to other objects.

Such a matrix can be seen as a linear combination of:

– a *local browsing matrix* $A_l = \{a_{ij}^l\}$ for each user u_l, where its generic element a_{ij}^l is defined as the ratio of the number of times object O_i has been accessed by user u_l immediately after O_j to the number of times any object in $\mathcal{O}^c$ has been accessed by u_l immediately after O_j;
– a *global browsing matrix* $A = \{a_{ij}\}$, where its generic element a_{ij} is defined as the ratio of the number of times object O_i has been accessed by any user immediately after O_j to the number of times any object in $\mathcal{O}^c$ has been accessed immediately after O_j;
– a *multimedia similarity matrix* $B = \{b_{ij}\}$ such that $b_{ij} = \frac{1 - d(O^i, O^j)_{ij}}{\Gamma}$ if $1 - d(O^i, O^j)_{ij} \geq \tau \ \forall i \neq j$, 0 otherwise ($\tau$ is a threshold and Γ is a normalization factor which guarantees that $\sum_i b_{ij} = 1$, see [5] for more details).

The successive step is to compute *customized* rankings for each individual user. In this case, we can rewrite previous equation considering the ranking for each user as $R_l = C \cdot R_l$, where R_l is the vector of preference grades, customized for a user u_l.

We note that solving the discussed equation corresponds to finding the stationary vector of C, i.e., the eigenvector with eigenvalue equal to 1.

In [5, 7], it has been demonstrated that C, under certain assumptions and transformations, is a real square matrix having positive elements, with a unique largest real eigenvalue and the corresponding eigenvector has strictly positive components. In such conditions, the equation can be solved using the *Power Method* algorithm.

Finally, we have introduced a *post-filtering* method for generating the final set of "real" candidates for recommendation.

Assume that a user u_l is currently interested in a target object O_j. We can define the set of candidate recommendations as follows:

$$\mathcal{O}^c_{l,j} = \bigcup_{k=1}^{M} \{O_i \in \mathcal{O}^c \mid a^k_{ij} > 0\} \cup \{O_i \in NNQ(O_j, \mathcal{O}^c)\} \tag{4}$$

The set of candidates includes the objects that have been accessed by at least one user within k steps from O_j, with k between 1 and M, and the objects that are most similar to O_j according to the results of a *Nearest Neighbor Query* ($NNQ(O_j, \mathcal{O}^c)$) functionality provided by the Multimedia Data Management Engine. Note that a positive element a^k_{ij} of A^k indicates that O_i was accessed exactly k steps after O_j at least once.

The ranked list of recommendations is then generated by ranking the objects in $\mathcal{O}^c_{l,j}$, for each object O_j selected as interesting by user, using the ranking vector R_l. The ranked list can change on the base of weather and environmental situations. For example, the recommendation grades of objects, which come from certain cultural POIs with a certain number of persons or with particular values of temperature or humidity, could be in some way "penalized" and such objects could be excluded from recommendation.

Finally, the list of K most important suggested items can be organized, according to the available POIs, into apposite *visiting paths* (considering distances from user location as in $\hat{\mathcal{O}}^c$). The visiting paths will be automatically updated when the set of target objects O_j is modified.

6 Case Studies

In this section, we are considering as real case studies for our framework two different "cultural environments" presenting different problems and solutions: an outdoor archeological site and an indoor museum.

6.1 System Customization for an outdoor environment

We consider as first real case study the archeological site of *Paestum*, one of the major Graeco-Roman cities in the South of Italy. Here, the main cultural attractions for a tourist are represented by a set of ancient buildings: three main temples of Doric style (i.e. the *First Temple of Hera*, also called *Basilica*, the *Second Temple of Hera*, also known as *Temple of Neptune*, and the *Temple of Athena*), the *Roman Forum* with several ruins, and the *Amphitheater*.

All the buildings are surrounded by the remains of the city's walls. In addition, there is a museum near the ancient city containing many evidences of the Graeco-Roman life (e.g. amphorae, paintings and other objects). Thus, the cited buildings will constitute in such a context the set of cultural Points Of Interest (POIs) for our case study.

Users visiting ruins could be happy of having a useful multimedia guide able to describe the main cultural attractions and to suggest automatically *visiting paths* containing multimedia objects of interest.

For instance, when a user is approaching a particular cultural POI (e.g. Temple of Neptune), the related multimedia description and the set of candidate objects (i.e. multimedia data of several kinds as audio, images, video and texts related to the different POIs) are delivered on the user's mobile device (pre-filtering stage). The list of proposed objects depends on the user's preferences (e.g. the majority of items will be images or texts if a user prefers to see such kinds of data and will reveal effective user needs), is initially ordered according to effective user location (i.e. the closest items will appear at the top of list) and contains data grouped by the related cultural POI.

Successively, after the user has selected one or more objects as "of interest" (he/she has to select each time at least one target object, for example the item he is currently watching), the recommendation services first perform a final ranking (ranking stage) of all the candidate objects (e.g. images of Temple of Neptune, of other Temples and of Roman Forum) according to their *recommendation grades* and then filters the recommendation list considering only the most similsr items to target objects (post-filtering stage). The *Top-K* objects from the obtained recommendations are finally arranged in visiting paths, shown on a proper map together with user's location with respect to POIs.

When a user is near to a different POI, he/she can decide to modify the list of target objects (e.g. removing those related to the previous visited POI or adding new objects) and consequently the visiting path will be automatically updated, thus including new items.

The paths take into account the current context in terms of actual position (obtained in this case by GPS), the selected multimedia data and the weather and environmental conditions, thus enhancing the visiting experience. Once acquired such kind of information, the path can dynamically change also in the case of unfit to use areas (e.g. too high temperature/humidity or a closed area). Eventually, the visiting paths could be enriched with other touristic POIs (e.g. restaurants, hotels, etc.). A graphic user interface gives the detailed view of the suggested path on an proper cartography, reporting a preview of cultural POIs and allowing a *rating* of observed objects.

Figures 4 and 5 show a running example for our system concerning the building of a visiting path for the Paestum ruins. User can select target objects from the candidates set by means of a proper GUI. A user can filter objects belonging to a given POI using different criteria: type of multimedia data, language, size, etc. The candidates are then ranked, filtered and arranged in a visiting path, reporting for each POI the list of recommended objects.

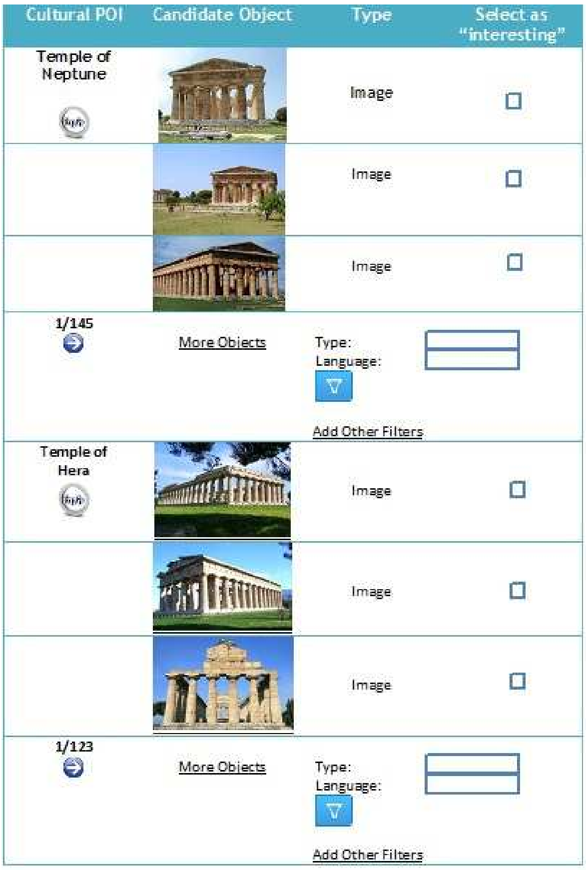


Fig. 4 GUI for the selection of target objects from the candidates set.

In the following, we report some implementation details concerning the customization of developed prototype for Paestum ruins.

Our multimedia collection consists of a database of about 10,000 images and texts coming from several Multimedia Repositories (i.e. Flickr, Panoramio, Facebook, Wikipedia) and related to all the main attractions of Paestum.

We used for raw data management a Multimedia Storage and Staging area based on a distributed *Multimedia File System*. In turn, Structural Description of multimedia objects in terms of features (low and high level) and spatial information has been managed by the *PostegreSQL* ORDBMS and its spatial extension *PostGIS*. The Indexing and Access Manager and Features Extractor modules have been implemented using the *Windsurf* library [2], while Repository Interface exploits the set of available API to gather data from the cited multimedia repositories. The Sensor Management Middleware collects and manages sensors' messages from users' mobile devices. By means of the

[2] http://www-db.deis.unibo.it/Windsurf/

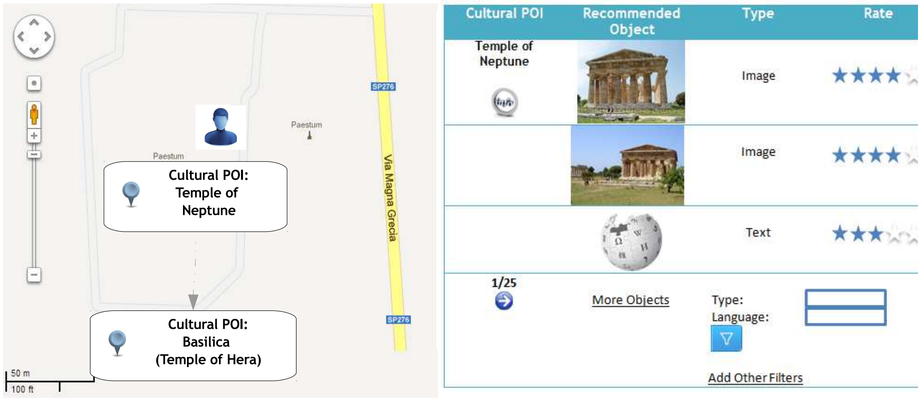


Fig. 5 GUIs for visualization of a visiting path.

GPS facilities and *Google Weather* API, it is able to capture user location and some environmental parameters for a given area (number of a persons and the related weather conditions).

The Knowledge Base, realized using different technologies, allows to manage the overall knowledge related to a given cultural environment. Contextual Data instances (messages containing information about users' position and environmental parameters) are managed by the *Cassandra* DBMS, while Cultural POI Descriptions are stored in a linked open data format based on the *RDF* model and managed by the *Sesame* Repository and *JENA* libraries.

Semantics of data can be specified by linking values of high-level attributes to some available ontological schema.

User Preferences (managed by *MongoDB* and *Neo4j*) are captured in an explicit manner by means of proper questionnaires or using information from Social Network (i.e. *Facebook*) and in an implicit way considering user's session logs. For the support cartography, we use *Google Maps*.

On the other hand, the Multimedia Recommender Engine exploits proper *JAVA* libraries (developed for the systems presented in [6, 7] and integrated with co-clustering libraries) to accomplish its tasks.

Finally, a user can interact with our system using – at the moment – an *Android* Multimedia Guide App. The presentation logic is based on apposite widgets. The client requests are elaborated by *JAVA Servlets* and the results are sent to the client in form of XML data.

6.2 System Customization for an indoor environment

We consider as second real case study the *National Museum of Capodimonte* in Naples, Italy. The museum heritage consists of many paintings from the 13th to the 18th centuries including works by famous artists such as *Caravaggio* and *Raffaello*, and of the magnificent *Farnese* collection of classical, mostly Roman, monumental sculptures.

Here, the cultural POIs consist of each single museum room and we can consider as motivating example the case of a tourist visiting an art exhibition within the museum. The cultural environment offers, through a Wi-Fi connection, a web-based access to a multimedia collection containing: digital reproductions of about 5,000 data among paintings and sculptures, educational videos, audio guides, textual and hypermedia documents with description of authors, paintings and sculptures.

In order to make the user's experience more interesting and stimulating, the access to information should be automatically delivered and customized based on the specific profile of a visitor, which includes learning needs, level of expertise and personal preferences, on user effective location in the museum, on the objects "similarity" between items user is currently watching and the other ones, and on information about the context in terms of number of persons for each room, room fitness, network performance, etc.

For instance, when a user is entering into a particular museum room (POI), the list of candidate objects are delivered on the user's mobile device (pre-filtering stage) order by the related distance from user and grouped by the related belonging room.

As in the previous example, after the user has selected one or more objects as of interest, the recommendation services first perform a final ranking (ranking stage) of all the candidate objects and then filters the recommendation list considering only target objects (post-filtering stage). The *Top-K* objects are finally arranged in visiting paths, shown on a proper museum map. When a user is approaching a different museum room, he/she can decide to modify the list of target objects (e.g. removing those related to the previous visited POI or adding new objects) and consequently the visiting path will be automatically updated, thus including different items.

The paths take into account the current context in terms of actual position (obtained in this case by a Wi-Fi positioning system or a WSN), the selected multimedia data and the environmental conditions: once acquired such kind of information, the path can dynamically change also in the case of crowded or closed room.

In this case, Multimedia Data Management Engine, Multimedia Recommender Engine, Multimedia Guide App and Knowledge Base (with the unique exception of the Support Cartography that consists of the museum maps) are realized with the same technologies of the outdoor case study. In turn, Sensor Management Middleware can collect and manage sensors' messages from users' mobile devices by means of a *Wi-Fi Positioning System* and/or *Wirless Sensor Network* (based on the Bluetooth technology) [8] facilities and *TinyDB* API [3]. In this way, it is able to capture user relative location and some environmental parameters for a given area (number of a persons and the related environmental conditions).

7 Experimental Results

Recommender Systems are very complex applications that are based on a combination of several models, algorithms and heuristics. This complexity makes evaluation efforts very difficult and thus results are hardly generalizable, as reported in the literature [3, 25]. Moreover, characterizing and evaluating the quality of a user's experience and subjective attitude toward the acceptance of recommender technology is an important issue which we will consider in the following.

The majority of research efforts on recommender system evaluation have mainly focused on prediction *accuracy* and *stability* (e.g., [3]). More recently, researchers began examining issues related to users' subjective opinions and developing additional criteria to evaluate recommender systems. In particular, they suggest that user satisfaction does not always (or, at least, not only) correlate with the overall recommender's accuracy.

Starting from these considerations and based on current trends in the literature, we decided to perform both a *user-centric* evaluation and a more traditional evaluation based on well-established accuracy metrics. In particular, the proposed evaluation strategy aims at measuring: (*i*) *user satisfaction* with respect to assigned browsing tasks in an outdoor environment, and (*ii*) effectiveness of the system in terms of *accuracy* for an indoor cultural space.

In particular, we evaluated, from one hand, how a visiting path can effectively support browsing tasks of different complexity when multimedia items of interest can come from different cultural POIs placed in not close areas (e.g. buildings in an archeological site), and from the other hand, how our ranking strategy is accurate within a single POI (e.g. a museum room) with respect to other recommendation strategies [7].

7.1 User Satisfaction

We designed and carried out several experiments to investigate how helpful the recommendations offered by our system - in terms of visiting paths - are to accomplish assigned browsing activities, demonstrating that the introduction of such techniques can improve the tourists' experience with respect to traditional and static touristic guides.

[3] http://telegraph.cs.berkeley.edu/tinydb/software.html

7.1.1 Paestum ruins

For the training of our system, we decided to implement a web-based application that allows users to browse the entire multimedia collection (about 10,000 items characterized by a set of schema-free tags) related to Paestum ruins.

In this way, we were able to capture the browsing sessions of about 50 users among graduate students (that used the system for several weeks) and to build a consistent matrix A for the described collection.

We then asked a different group of 10 profiled people (this group consisted of 5 not-expert users on graeco-roman art, 3 medium expert users and 2 expert users) to complete by the same application several browsing tasks of different complexity within the Paestum ruins collection (15 per user - 5 for each degree of complexity) and without any recommendation facility (web application provides classical search/retrieval mechanisms). After this test, we asked them to browse once again the same collection with the assistance of our recommender system (by facilities provided by visiting paths generated obligating users to choose at least one target object for each suggested POI) and complete other tasks of the same complexity. In a similar manner, in a second session we asked another group of 10 people to browse the same collection first with the assistance of our recommender system completing other different tasks and then without any help.

In particular, we have subdivided browsing tasks in the following three broad categories:

- ***Low Complexity*** tasks (T_1) - explore at least 30 multimedia objects related to 3 different POIs depicting ancient buildings;
- ***Medium Complexity*** tasks (T_2) - explore at least 50 multimedia objects related to 5 different POIs depicting graeco-roman *temples*, *amphitheaters* an *Roman forum buildings* (10 objects for each subject);
- ***High Complexity*** tasks (T_3) - explore at least 160 multimedia objects related to 8 different POIs depicting graeco-roman *temples*, *Roman forum buildings*, *amphitheaters* and *city walls' gates* (20 objects for each subject).

Note that the complexity of a task depends on several factors: the number of objects to explore, the number of POIs to explore and the type of desired subjects. Users know each browsing task's goal before selecting target objects. However, if a visiting path initially does not contain sufficiently many objects required by a browsing task, user can modify the path itself changing the list of target objects.

The strategy we used to evaluate the results of this experiment is based on NASA TLX (*Task Load Index factor*) [4].

To this aim, we then asked the users to express their opinion about the advantage of our system to provide an effective user experience in completing

[4] TLX [27] is a multi-dimensional rating procedure that provides an overall score based on a weighted average of ratings provided by users by means of proper questionnaires on six sub-scales: mental demand, physical demand, temporal demand, own performance, effort and frustration. The lower TLX scores (ranging in the 0-100 interval), the better they are.

Table 1 Comparison between our system and no facilities

	Experts		**Medium Exp.**		**Not Experts**	
TLX factor	**With rec.**	**Without**	**With rec.**	**Without**	**With rec.**	**Without**
Mental	29.2	30.1	34.5	36.2	38	45
Physical	29	35	32	39	34.1	48
Temporal	31	35.2	31	39	33	38
Effort	29.4	36	38	45	40	55
Perfomances	75	72	76	75.3	78.5	78.7
Frustation	28	38	29.9	35.2	30	35

the assigned visiting tasks. Thus, we obtained the average results scores for each of three categories of users reported in Table 1 (the lower the TLX score — in the range $[0-100]$ — the better the user satisfaction).

Note that not-expert users find our system more effective than the other users' category in every sub-scale, because they consider very helpful the provided suggestions. Instead, in expert and medium expert users' opinion, our system outperforms a classical touristic guide in every sub-scale except for *mental demand and performances*: this happens because an expert user considers sometimes not useful the automatic suggestions just because they know what they are looking for.

7.2 Accuracy

In this second series of experiments, our goal was to measure the *accuracy* of our ranking strategy with respect to other recommendation techniques, in order to have a precise idea of the real effectiveness of the proposed recommendation approach.

Generally, accuracy allows to measure the *prediction error*, i.e., how the system recommendations differ from the choice a user would probably make, and recommendation strategies are usually compared based on standard datasets of products, movies, songs, etc. (e.g., OZSTORE, Jester, BookCrossing, MovieLens, Netflix data, Last.fm and so on) that simply contain the description of user profiles and, for each item, the set of users' ratings.

Unfortunately, such datasets do not exactly fit with our strategy for different reasons: (*i*) we do not need specific ratings of dataset items for computing recommendations; (*ii*) each user rating is not absolute but depends on the related context (i.e., the items previously accessed); (*iii*) we use multimedia features and high-level semantic descriptors of items that require the availability of raw data.

For these reasons, we decided to use as dataset for the experiments our multimedia collection related to the Capodimonte Museum. We retrieved from the Web about 5,000 multimedia objects (the majority are images and texts) and extracted low and high level information using MDME facilities. For the images of paintings/sculptures we used as semantic tags *author*, *genre* and *subject* information, in turn for texts (describing paintings authors or subjects) we chose as tags *title* and *keywords*.

7.2.1 Capodimonte Museum

Also for this experimentation, we implemented a web-based application that allows users to browse the related multimedia collection.

In particular, we asked a group of 50 users to use the system for some weeks, in order to collect a significant amount of browsing sessions to populate browsing matrices. During their session, we also asked the users to rate the paintings they consider more interesting on a scale from 1 to 5.

Then, we collected – as a ground truth – the ratings of other 40 users for a subset of 2000 multimedia data (belonging to different POIs, in particular 100 for each room) with respect to several target objects of several kinds [5].

We used the *Mean Absolute Error* (*MAE*) and the *Root Mean Square Error* (*RMSE*) as metrics in our experiments. In our case, *MAE* and *RMSE* are defined as:

$$MAE = \frac{1}{N}\sum_{u,i,j}|r^j_{ui} - \hat{r}^j_{ui}| \quad ; \quad RMSE = \sqrt{\frac{1}{N}\sum_{u,i,j}(r^j_{ui} - \hat{r}^j_{ui})^2}$$

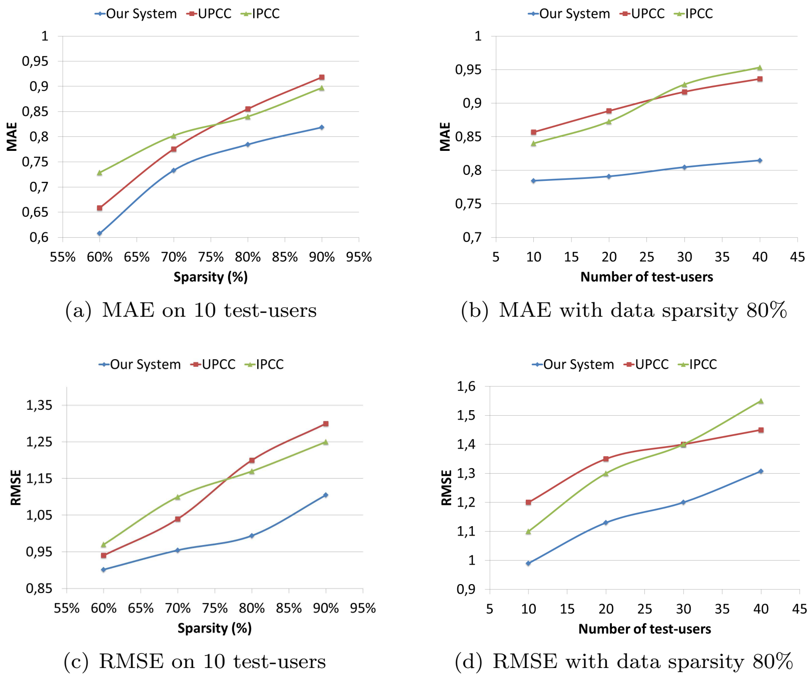
where r^j_{ui} is the actual rating that the user u has given to item i w.r.t. the target object j, $\hat{r}^j_{ui}$ is the system predicted rating (the recommendation grades were also normalized on a scale from 1 to 5), and N is the total number of test ratings. Both *MAE* and *RMSE* thus attempt to measure the prediction error (accuracy of the recommendation): *RMSE* is considered as a stronger measure than *MAE* as larger prediction errors are penalized more. For both metrics, smaller values indicate better performances.

We compared the performance of our algorithm with the two most diffused approaches: *User based Pearson Correlation* (*UPCC*) and *Item based Pearson Correlation* (*IPCC*) [48]. These techniques were implemented leveraging machine learning libraries provided by the *Apache Mahout* framework.

In our case, the *rating data sparsity* is the average percentage of database items that have not been previously rated by users of the first group. For example, a sparsity of 60% means that a user rated at least one time only 40% of images.

Fig. 6 compares the performance of our algorithm in terms of MAE and RMSE w.r.t. the other approaches varying the number of test users (with a fixed sparsity) and the sparsity (with a fixed number of test users). Note that our system achieves very good performance and outperforms the other techniques, especially for higher values of sparsity. This is due to the fact the UPCC and IPCC suffer from the cold start and overspecialization problems for

[5] We have chosen two groups of users among students and graduate students: the first one used the system for 3 weeks without recommendation facilities to capture a significant number of browsing sessions/ratings and then we asked the second one to indicate, for each target object (randomly selected), the most relevant ones among 100 multimedia items (belonging to the same POI of the target one) rating each one in a scale ranging from 1 to 5.



(a) MAE on 10 test-users

(b) MAE with data sparsity 80%

(c) RMSE on 10 test-users

(d) RMSE with data sparsity 80%

Fig. 6 Comparison between our approach and other techniques in terms of MAE and RMSE varying number test-users and rating sparsity

high values of sparsity. Moreover, in our system the prediction error increases in the most slow way w.r.t. to the number of test users, both for MAE and RMSE, demonstrating a quite good stability.

8 Conclusions

In this paper we proposed a novel multimedia and context-aware recommender platform in the Cultural Heritage domain. Basically, when a user is close to a cultural POI, our proposed recommender system is able to: (i) determine a set of useful *candidate* objects for the recommendation, considering users' location, needs and preferences (*pre-filtering stage*) and using co-clustering techniques; (ii) opportunely rank these objects exploiting their intrinsic features and users' past behaviors (*ranking stage*) by means of a proper hybrid strategy; (iii) dynamically, when a user "selects" one or more candidate objects, select the list of most suitable objects (*post-filtering stage*) and eventually arrange such items in apposite *visiting paths*, also considering other context information such as weather or environmental conditions.

We implemented our system in both outdoor and indoor environments, the *Paestum Ruins* and *Capodimonte Museum*. In both cases, we were able to provide tourists with personalized and dynamic visiting paths useful to make their visiting experience more stimulating and interesting. Then, we investigated the effectiveness of the proposed approach in the considered scenarios, based on the users' satisfaction with respect to several browsing tasks and system's accuracy in terms of prediction error. Experimental results showed that our approach is quite promising and encourages further research.

We are planning to enrich our work in several directions. Future work will be devoted to: (*i*) extend the experimental campaign on a larger multimedia data set, (*ii*) provide the synchronization and presentation of the different multimedia items related to a given POI in the shape of a *multimedia story* to be delivered to final users.

Regarding the last aspect, we are also interested in emergent research topics such as Interactive Storytelling with the aim of developing interactive media presenting Cultural Heritage stories where the presentation of a narrative, and its evolution, can be influenced in real time by the users and the context.

9 Acknowledgments

The realization of the proposed prototype was supported by *DATABENC* [6], a high technology district for Cultural Heritage management recently funded by Regione Campania - Italy.

References

1. C. Aart, B. Wielinga, and W.R. Hage. Mobile cultural heritage guide: Location-aware semantic search. In *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 257–271. Springer Berlin Heidelberg, 2010.
2. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. In *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, IEEE Computer Society, 2005.
3. G. Adomavicius and J. Zhang. On the stability of recommendation algorithms. In *ACM Conference on Recommender Systems*, pages 47–54, ACM, 2010.
4. M. Albanese, A. Chianese, A. d'Acierno, V. Moscato, and A. Picariello. A multimedia recommender integrating object features and user behavior. In *Multimedia Tools and Applications*, 50(3):563–585, Springer, 2010.
5. M. Albanese, A. d'Acierno, V. Moscato, F. Persia, and A. Picariello. Modeling recommendation as a social choice problem. In *ACM Conference on Recommender Systems*, pages 329–332. ACM, 2010.
6. M. Albanese, A. d'Acierno, V. Moscato, F. Persia, and A Picariello. A multimedia semantic recommender system for cultural heritage applications. In *IEEE International Conference on Semantic Computing*, pages 403–410. IEEE Computer Society, 2011.
7. M. Albanese, A. d'Acierno, V. Moscato, F. Persia, and A. Picariello. A multmimedia recommender system. In *ACM Transactions on Internet Technology*, 13(1), ACM, 2013.
8. F. Amato, A. Chianese, A. Mazzeo, V. Moscato, A. Picariello, and F. Piccialli. The talking museum project. In *International Conference on Emerging Ubiquitous Systems and Pervasive Networks*, EUSPN-2013, 2013.
9. S.S. Anand, P. Kearney and M. Shapcott. Generating semantically enriched user profiles for web personalization. In *ACM Transactions on Internet Technology*, 7(4), ACM, 2007.

[6] www.databenc.it

10. L. Ardissono, T. Kuflik and D. Petrelli. Personalization in cultural heritage: the road travelled and the one ahead. In *User Modeling and User-Adapted Interaction*, 22(1-2):73–99, Springer, 2012.
11. I. Bartolini and P. Ciaccia. Imagination: Exploiting link analysis for accurate image annotation. In *Adaptive Multimedia Retrieval: Retrieval, User, and Semantics*, volume 4918/2008 of *Lecture Notes in Computer Science*, pages 32–44, Springer, 2008.
12. I. Bartolini, P. Ciaccia, and M. Patella. Query processing issues in region-based image databases. In *Knowledge Information System*, 25(2):389–420, Springer, 2010.
13. I. Bartolini, V. Moscato, R.G. Pensa, A. Penta, A. Picariello, C. Sansone, and M.L. Sapino. Recommending multimedia objects in cultural heritage applications. In *International Conference on Image Analysis and Processing, Workshops*, pages 257–267, 2013.
14. I. Bartolini, M. Patella, and C. Romani. Shiatsu: tagging and retrieving videos without worries. In*Multimedia Tools and Applications*, 63(2):357–385, Springer, 2013.
15. I. Bartolini, Z. Zhang, and D. Papadias. Collaborative filtering with personalized skylines. *IEEE Trans. on Knowl. and Data Eng.*, 23(2):190–203, February 2011.
16. J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *International Conference on Machine Learning*, pages 65–72, ACM, 2004.
17. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). In *Computer Vision Image Understanding*, 110(3):346–359, Elsevier, 2008.
18. F. Bellotti, R. Berta, A. De Gloria, A. D'ursi, and V. Fiore. A serious game model for cultural heritage. *Journal on Computing and Cultural Heritage*, 5(4):pages 17:1–17:27, ACM, 2013.
19. C.A. Bhatt and M.S. Kankanhalli. Multimedia data mining: State of the art and challenges. In *Multimedia Tools Applications*, 51(1):35–76, Springer, 2011.
20. J.P. Bowe and S.F. Fantonio. Personalization and the web from a museum perspective. In *International Conference on Museums and the Web*, 2004.
21. P. Ciaccia, M. Patella and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *International Conference on Very Large Data Bases*, pages 426–435, Morgan Kaufmann Publishers Inc, 1997.
22. S. Costantini, L. Mostarda, A. Tocchio, and P. Tsintza. Dalica: Agent-based ambient intelligence for cultural-heritage scenarios. In *Intelligent Systems*, 23(2):34–41, IEEE, 2008.
23. P. Dourish. What we talk about when we talk about context. In *Personal Ubiquitous Computer*, 8(1):19–30, Springer, 2004.
24. C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. In *Computer Vision and Image Understanding*, 114(6):712–722, Elsevier, 2010. Special Issue on Multi-Camera and Multi-Modal Sensor Fusion.
25. A.G. Schulz and M. Hahsler. Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory. In *International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles*, volume 2703 of *Lecture Notes in Artificial Intelligence*, pages 100–114, Springer, 2002.
26. L.A. Goodman and W.H. Kruskal. Measures of association for cross classifications, iv: Simplification of asymptotic variances. *Journal of the American Statistical Association*, 67(338):pp.415–421, 1972.
27. S. Hart and L.E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human mental workload* pages 139–183, 1988.
28. Y. Hijikata, K. Iwahama and S. Nishida. Content-based music filtering system with editable user profile. In *ACM Symposium on Applied Computing*, pages 1050–1057, ACM, 2006.
29. D. Ienco, C. Robardet, R.G. Pensa, and R. Meo. Parameter-less co-clustering for star-structured heterogeneous data. In *Data Mining Knowledge Discovering*, 26(2):217–254, Springer, 2013.
30. I.F. Ilyas, G. Beskales and M.A. Soliman. A survey of top-k query processing techniques in relational database systems. In *ACM computing surveys*, 40(4):11:1–11:58, ACM, 2008.

31. K. Juszczyszyn, P. Kazienko, and K. Musia. Personalized ontology-based recommender systems for multimedia objects. In *Agent and Multi-agent Technology for Internet and Enterprise Systems*, Studies in Computational Intelligence, pages 275–292, Springer, 2010.
32. K. Kabassi. Personalisation systems for cultural tourism. In *Multimedia Services in Intelligent Environments*, volume 25 of *Smart Innovation, Systems and Technologies*, pages 101–111, Springer, 2013.
33. S. Karaman, A. Bagdanov, G. D'Amico, L. Landucci, A. Ferracani, D. Pezzatini and A. Bimbo. Passive profiling and natural interaction metaphors for personalized multimedia museum experiences. In *New Trends in Image Analysis and Processing*, volume 8158 of *Lecture Notes in Computer Science*, pages 247–256. Springer, 2013.
34. A. Karatzoglou, X. Amatriain, L. Baltrunas and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *ACM Conference on Recommender Systems* pages 79–86, ACM, 2010.
35. H.K. Kim, J.K. Kim and Y.U. Ryu. Personalized recommendation over a customer network for ubiquitous shopping. *IEEE Transaction on Service. Computing.*, 2(2):140–151, 2009.
36. J.K. Kim, H.K. Kim, and Y.H. Cho. A user-oriented contents recommendation system in peer-to-peer architecture. *Expert Systems with Applications*, 34(1):300–312, Elsevier, 2008.
37. T. Kuflik, O. Stock, M. Zancanaro, A. Gorfinkel, S. Jbara, S. Kats, J. Sheidin and N. Kashtan. A visitor's guide in an active museum: Presentations, communications and reflection. In *Journal Computing and Cultural Heritage*, 3(3):11:1–11:25, ACM, 2011.
38. H. W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, 2:83–97, 1955.
39. D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference Computer Vision*, vol. 2, pages 1150–1157, 1999.
40. V. Maidel, P. Shoval, B. Shapira and M. Taieb-Maimon. Evaluation of an ontology-content based filtering method for a personalized newspaper. In *ACM Conference on Recommender Systems*, pages 91–98. ACM, 2008.
41. M.G. Manzato and R. Goularte. Supporting multimedia recommender systems with peer-level annotations. In *XV Brazilian Symposium on Multimedia and the Web*, pages 26:1–26:8. ACM, 2009.
42. K. Musial, P. Kazienko, and T. Kajdanowicz. Social recommendations within the multimedia sharing systems. In *1st World Summit on The Knowledge Society: Emerging Technologies and Information Systems for the Knowledge Society*, pages 364–372. Springer, 2008.
43. M.J. Pazzani and D. Billsus. The adaptive web. In *Content-based Recommendation Systems*, pages 325–341, Springer, 2007.
44. F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook.* Springer, 2011.
45. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
46. J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen. The adaptive web. In *Collaborative Filtering Recommender Systems*, pages 291–324, Springer, 2007.
47. J.H. Su, H.H. Yeh, P.S. Yu and V.S. Tseng. Music recommendation using content and context information mining. In *IEEE Intelligent Systems*, 25(1):16–26, IEEE, 2010.
48. X. Su and T.M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4:2–4:2, Hindawi, 2009.
49. V.S. Tseng, J.H. Su, B.W. Wang, C.Y. Hsiao, J. Huang and H.H. Yeh. Intelligent multimedia recommender by integrating annotation and association mining. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 492–499, IEEE, 2008.
50. V. Vlahakis, T. Pliakas, A. Demiris, and N. Ioannidis. Design and application of an augmented reality system for continuous, context-sensitive guided tours of indoor and outdoor cultural sites and museums. In *International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pages 155–164. Eurographics Association, 2003.

51. V. Vlahakis, J. Karigiannis, M. Tsotros, M. Gounaris, L. Almeida, D. Stricker, T. Gleue, I.T. Christou, R. Carlucci and N. Ioannidis. Archeoguide: First results of an augmented reality, mobile computing system in cultural heritage sites. In *Conference on Virtual Reality Archeology, and Cultural Heritage*, pages 131–140. ACM, 2001.
52. Y. Wang, N. Stash, R. Sambeek, Y. Schuurmans, L. Aroyo, G. Schreiber and P. Gorgels. Cultivating personalized museum tours online and on-site. In *Interdisciplinary Science Reviews*, 34(2-3):139–153, 2009.
53. D. Weinland, R. Ronfard and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. In *Computer Vision and Image Understanding*, 115(2):224 – 241, Elsevier, 2011.