

Social and Content Hybrid Image Recommender System for Mobile Social Networks

Faustino Sanchez · Marta Barrilero · Silvia Uribe ·
Federico Alvarez · Agustin Tena ·
Jose Manuel Menendez

Abstract One of the advantages of social networks is the possibility to socialize and personalize the content created or shared by the users. In mobile social networks, where the devices have limited capabilities in terms of screen size and computing power, Multimedia Recommender Systems help to present the most relevant content to the users, depending on their tastes, relationships and profile. Previous recommender systems are not able to cope with the uncertainty of automated tagging and are knowledge domain dependant. In addition, the instantiation of a recommender in this domain should cope with problems arising from the collaborative

filtering inherent nature (cold start, banana problem, large number of users to run, etc.). The solution presented in this paper addresses the abovementioned problems by proposing a hybrid image recommender system, which combines collaborative filtering (social techniques) with content-based techniques, leaving the user the liberty to give these processes a personal weight. It takes into account aesthetics and the formal characteristics of the images to overcome the problems of current techniques, improving the performance of existing systems to create a mobile social networks recommender with a high degree of adaptation to any kind of user.

F. Sanchez (✉) · M. Barrilero · S. Uribe · F. Alvarez · A. Tena ·
J. M. Menendez
Grupo de Aplicación de Telecomunicaciones Visuales (G@TV),
Madrid, Spain
e-mail: fsg@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

M. Barrilero
e-mail: mbg@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

S. Uribe
e-mail: sum@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

F. Alvarez
e-mail: fag@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

A. Tena
e-mail: atg@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

J. M. Menendez
e-mail: jmm@gatv.ssr.upm.es
URL: <http://www.gatv.ssr.upm.es/>

F. Sanchez · M. Barrilero · S. Uribe · F. Alvarez · A. Tena ·
J. M. Menendez
ETSI Telecomunicación. Universidad Politécnica de Madrid
(UPM),
Madrid, Spain

Keywords aesthetics · social recommendation ·
content-based recommendation · hybrid recommender ·
image classification · user modeling

1 Introduction

Mobile social networks and social networks accessed by mobile terminals need specific tools to take into account on the one hand the limited capabilities and screen size, and on the other hand take advantage of the mobility and personal orientation of the terminals, to tailor the information presented to the user needs and interests.

This is the reason why multimedia recommender systems help to offer the user the relevant information based on his/her personal taste from a high amount of available information when accessing with mobile terminals collections of images uploaded to a social network.

In the technical domain, one of the main challenges of multimedia recommender systems in mobile social networks is to create recommenders which are independent from the knowledge domain or the manual description tagging, able to be adapted to the mobile environment specific capacities.

The most popular approach to solve this problem is the use of social techniques such as collaborative filtering. The advantage of this technique is the independency from the content of the multimedia resource: it can be applied to any image collection. Moreover, this technique is the most appropriate in social networks, because it allows using the social information from each user. However, collaborative filtering has several well-known problems [25] such as cold start or banana problem. A powerful way to solve these problems is the use of a hybrid system, combining collaborative filtering with content-based techniques. In this paper we focus the description on our content-based filtering proposal, which is used to solve the mentioned problems of collaborative filtering systems and typical content-based techniques. The novelty of the system lies in the modeling of the user taste regarding aesthetics and formal characteristics of the images, which is developed by the Content-based module.

Another critical problem is the need of applications that are easy to use and offer a light workload in the user terminal, since mobile devices present considerable restrictions, related to size screen, computational power, battery life or network connectivity. This implies optimizations in the architecture and implementation of recommenders, as it is shown in [1]. The solution proposed in this paper is a scalable hybrid image recommender system which takes into account aesthetics and formal characteristics of the images to solve the problems of current techniques, improving the performance and creating a recommender with a high degree of adaptation to any kind of user and terminal. Studies within the field of psychology have already proven the influence of aesthetics and formal characteristics of the images on the perception and taste of users [2–4], and some systems have started to take them into account in other applications [5].

This feature of the recommender generates a new necessity: the inference of the characteristics of the visual perception from each user. Starting from raw data of item ratings (obtained either explicitly or implicitly), the inference of a user's visual perception can be obtained by modeling the user's tastes. Therefore, we propose a multidomain, scalable and hybrid Multimedia Recommender System, able to solve multiple well-known problems of Image Recommenders, such as cold start and banana problem (solved by means of hybrid techniques), domain dependence (solved using low level descriptors), overspecialization of content-based algorithms (solved by means of clustering techniques). Moreover, the proposed system improves the recommendation prediction efficiency, as it is shown in Section 7. Besides, we use a modular design that eases the generation and integration of extensions or further improvements (as explained in the future work section). The paintings domain has been chosen for testing the system with real users, although the use of other domains is also possible.

A methodology has been followed for the development of the research, design and implementation: Firstly, after studying the related work, a definition of low level descriptors with relevance in the aesthetic perception of an image by a user was made. These descriptors were tested and compared with the explicit results obtained from real users (Section 7): the results was that some descriptors were validated and other descriptors rejected. Next, we designed a recommendation algorithm which takes into account the descriptors with positive and negative influence in the users' taste. Finally, a clustering technique was included in the algorithm to group different styles that the user prefers or that the user specially dislikes. Final tests were performed over the complete algorithm to check its efficiency.

The remainder of this paper is organized as follows. Section 2 describes the relevant related work. Section 3 explains our modular design and architecture. Sections 4 and 5 describe the content-based recommender system classification and recommendation processes, respectively. Section 6 describes the implementation of the algorithms in a real system and Section 7 the tests and experiments validating our system's results. Finally Section 8 explains future work and Section 9 presents conclusions.

2 Related work

Recommendation systems have become an important research area since the first researches about collaborative filtering appeared in the early 90's [6]. Image recommendation systems [7] are a specific branch in the field of visual content recommendation [8]. In fact, this type of recommendation takes great advantage of Content-Based Image Retrieval (CBIR) techniques [9], as they use the same methods, such as users' tagging [10] or semantic content analysis [11].

2.1 Image recommendation types

According to the current classification of recommendation systems, three basic image recommendation methods are available [12]:

Content-based recommendation method This method [13] is based on the similarity between items' properties and user profile information, in order to select similar content according to the user's preferences. With regards to image recommendation, image properties can be classified as either high or low level properties. High level refers to semantic and cognitive properties, whereas low level corresponds to intrinsic features of the image. In [14], an image recommendation system based on the influence of affective metadata (metadata that describe the user's emotions) is presented. Low level

properties have also been used for recommendation, taking advantage of CBIR methods combined with relevance feedback techniques (see [15]).

Collaborative recommendation method In this method, the recommendation is based on item ratings which have been previously made by a set of users. The recommendation is generated according to heuristic or probabilistic methods [16], which are applied to the known ratings in order to predict an unknown rating of a user. Regarding image recommendation, there are useful methodologies and tools which can bring considerable benefits, since they allow users to keep a record of their ratings. For instance, *Flickr* application is an important tool in the task of testing this type of systems [17], since it provides useful resources, such as users' tagging or comments, which allow the development of a semantic context helping the collaborative recommendation. In fact, there is a considerable quantity of social tagging systems which can be used to extend the capabilities of recommender systems [18].

Hybrid method It combines the advantages of content-based and collaborative recommendation methods. Burke [19] introduced the classification of the hybridization techniques into seven classes. Two of them are implemented in Section 7. Moreover, recent works on hybrid recommendation methods introduce new algorithms to avoid certain problems related to scalability and sparsity [20].

2.2 Social networks recommendation

Recommenders systems for social networks must cope with the heterogeneity of the information and the sources of information. The first challenge deals with the modeling of the information to quantify and combine the said heterogeneous data into a common recommendation engine.

One example has been proposed by Kazienko et al. [21] with a methodology to deal with heterogeneous data social networks focused on video and images, such as Flickr or Youtube. Authors create a multidimensional system with 11 layers to model the relationship between 2 users. Then, they create correlation metrics to calculate the similarity between them, to apply classic collaborative filtering tools or social filters. The explained model is based on the Arazy et al. [22] model, where 4 different sources are considered and modeled to produce 4 different metrics.

It should be highlighted that many cases are based on the calculation of the *tie strength* attribute, which was formalized by Granovetter [23] in the 70's and by Krackhardt [24] in the 90's. The purpose was to offer an estimation of the connection strength which links the members of a community. With the advent of social networks in the last decade,

the research was applied to this domain to engineer the sociologic behavior. Regarding to mobile social networks recommenders, our approach considers the scalability restrictions defined in [1].

The image recommendation system presented in this paper is a hybrid recommender which performs a new content-based algorithm, but it also takes advantage of collaborative recommendation, to get a more accurate result in social network scenarios.

3 Overview of our approach and system architecture

The objective of our system is to generate recommendations of images from both social information and aesthetics and formal characteristics. Recommendations must be adapted to the user's tastes. Therefore, modeling of user tastes is necessary and is carried out within the recommendation modules.

The system (Fig. 1) is made up of two databases and two independent processes. The first database, *Image annotation*, stores general characteristics of the images, which will be used by the recommendation processes to personalize the recommendations. The second database, *User Ratings*, stores information about the images that the user likes and dislikes, and information about the general tastes of the user (which will be automatically stored from the Content-Based filtering of the Recommendation process). The two processes involved are:

1. *Classification process*. It is an offline process, independent of the users of the system. When someone wants to insert a new image in the global corpus of the system, this process calls the algorithms of *Perceptive classification*, which extract image characteristics and store them in the *Image annotation* database.

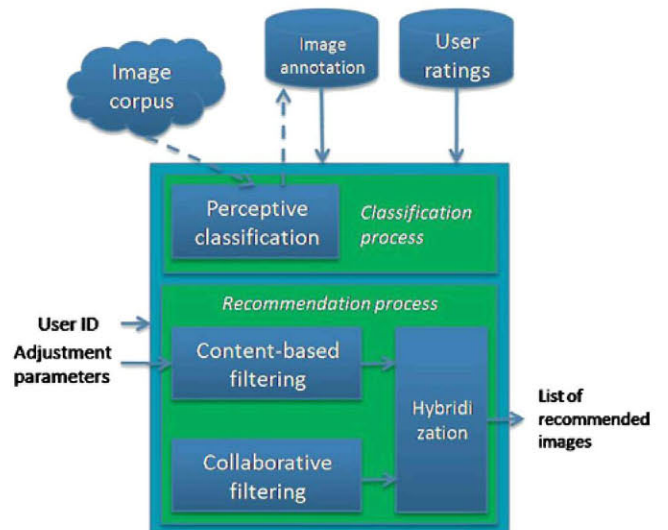


Fig. 1 General system architecture

2. *Recommendation process.* It is an online process, which generates the list of recommendations for each user. Users call the algorithms of this process when they send a request to the system. The modules involved are *Content-based filtering*, *Collaborative filtering*, and *Hybridization*.

The system operates directly with both databases in a user-transparent way. It requires the identification of the user and a collection of adjustment parameters as an input to generate the recommendations. These parameters have a default value and change according to the user's behavior. However, they can be modified manually by the user at any time. The parameters are the following:

Number of images: number of image recommendations that the user wants to receive. It is used in the hybridization stage.

Image Threshold: adjustment parameter related to the subjectivity of the ratings of a user. It sets the threshold to decide when the user likes or dislikes an image.

Attribute Threshold: parameter which sets the demand of the user related to the attributes. It determines when an attribute is meaningful to like or dislike an image.

m coefficient: parameter which adjusts the linearity of the metrics used for the final decision algorithm. It depends on the homogeneity of the different tastes of the user.

S coefficient: parameter which determines if the image characteristics which the user likes are more important than those which the user dislikes. It also quantifies this difference of importance.

The influences of S and m parameters in the final decision are explained in Section 5, in the discussion of Eq. (9). The parameter *Number of images* is used in the hybridization stage, and the other four parameters are used only in the Content-based filtering module. The Collaborative filtering module only needs the identification of the user, because the other required data are directly obtained from the databases.

The output of the system is the list of recommended images, although each one contains associated characteristics. As some studies have concluded [25, 26], these parameters are very important for the performance and the reliability of any recommender system: *Identifier*, *Utility*, *Reliability* and *Explanation*, which differentiates the weight of the usefulness prediction which comes from the content-based module and from the collaborative filtering module and estimates the reason of the recommendation.

The novelty of the system lies in the modeling of the user taste when it comes to aesthetics and formal characteristics of the images, which is developed by the Content-based module. Therefore, Sections 4 and 5 are focused on the explanation of the two processes involved in the Content-based recommendation system.

4 Content-based recommender system: classification process

Unlike common recommendation systems based only on semantic annotation, this new recommender uses low-level metadata to complete the content classification.

In order to obtain this information, a set of parameters from the MPEG-7 standard [27] is selected and applied, following the previous works performed by the authors in [25]. Although there are five types (color, texture, shape, motion and others), only color and texture descriptors have been considered, avoiding the ones dealing with motion (we are not recommending picture in motion) and shape (we are only using low level descriptors). They are obtained with the MPEG-7 Low Level Feature Extraction command line tool [28]. A bigger set of descriptors was initially defined, but some of them were finally discarded in the validation tests, explained in Section 7, Classification testing.

In order to extract the knowledge given by these parameters and to facilitate and optimize the classification process, we have produced new descriptors derived from combinations of the existing ones. These descriptors have been selected from a wide set of parameters according to how suitable they were to characterize images, based on the results presented in the classification testing of Section 7. Depending on the characteristic analyzed, the descriptors applied can be divided into three different groups [29], two of them already existing in the MPEG-7 standard (color and texture descriptors) and the third added ad-hoc to collect new descriptors (general luminance descriptors):

4.1 Texture descriptors

In this case, the designed parameters come from the MPEG-7 Edge Histogram Descriptor (EHD) [27], which finds the edge distribution along the image by obtaining local edges with different orientations in subimages.

Line energy This parameter (*El*) measures the total line density in the image according to its line energy level. It distinguishes images with no transitions from full edge images:

$$E_{ITot}^2 = E_{45^\circ}^2 + E_{90^\circ}^2 + E_{135^\circ}^2 + E_{180^\circ}^2 + E_{Other}^2 \quad (1)$$

where E_i shows the image edge distribution along each direction.

Line homogeneity This descriptor (*HI*) establishes the image line continuity, by dividing the image into 16 subimages and then calculating the distribution line variance among 4×4 neighboring subimages as follows:

$$H_1 = V_1 + V_2 + V_3 + V_4 + V_5 \quad (2)$$

where each V_i represents the variance among the four neighboring subimages except V_5 , which represents the variance among the other V_i

Entropy variety This parameter represents the variance of the entropy along the whole image. It is obtained by dividing the image into a set of 5×5 subimages and then calculating the entropy of each one of them. The final parameter is the variance of the 25 subimages

4.2 Luminance descriptors

These parameters determine the content saturation, according to different aspects of the image.

Direct luminance This parameter (DL) derives directly from the MPEG-7 Color Layout Descriptor (CLD) [27], which represents the color spatial distribution of the image obtained from the application of the Discrete Cosine Transform (DCT) to the image colors in the YCrCb color space.

DL obtains the mean saturation level of each image, in order to distinguish between clear and dark content, and is composed of two values (Lc, Lv): Lc measures the mean of the luminance and Lv its variance, which represents luminance dispersion.

Bit-plane distribution entropy This descriptor establishes the bit-plane distribution entropy according to [30]. Each element of a grayscale image is an integer lying between $[0, 255]$, which sets the intensity value of each pixel. It can be represented by an 8 bit binary vector, one for each plane ($b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$). In order to prevent small variation of intensity from affecting all bit planes, each pixel is expressed by its Gray code (which provides more significant information), computed as follows [30]:

$$g_i = \begin{cases} b_i \oplus b_{i+1}, & 0 \leq i \leq m-2 \\ b_i, & i = m-1 \end{cases} \quad (3)$$

where \oplus denotes XOR operation, b_i is the i -th bit-plane and g_i is the i -th bit plane expressed using Gray code. Then, we consider the entropy of the four highest bit-planes, which contain most of the structural information in the image.

Therefore, this descriptor is composed of four components, one for each bit-plane entropy ($E1, E2, E3, E4$).

4.3 Color descriptors

Chromatic variety This parameter (CV) comes from the MPEG-7 Dominant Color Descriptor (DCD), and analyzes the color variety along the image, according to the detected dominant colors.

CV is composed of three values ($N, VC_{intra}, VC_{inter}$): N indicates the number of dominant colors (extracted directly

from DCD), VC_{intra} expresses the variance among different values of the same color, and finally VC_{inter} measures the variance among each color.

Spatial coherence This descriptor comes also directly from MPEG-7 DCD descriptor and it determines the continuity of the color in the image.

HSV color space In order to establish the impact of the color values and their saturation in the user selection, we perform a color space change. This parameter is composed of two values: Sm , or the mean saturation value and Sv or the saturation variance.

Color planarity This descriptor (CP) represents the homogeneity on each dominant color in the image in terms of color hue, taking into account that an image with the same colors will have much more color planarity than another one with many different color hues. CP is obtained as follows: the image is divided into pixel blocks so that each block is a square with a pixel-side equal to the minimum side of the image divided by a selected factor (in the tests the factor is 8). The most frequent color in each block is then selected and the percentage of pixels with a similar color (with a maximum difference of 20 in the RGB system) exclusively in that block is calculated. The general color planarity is then the mean of the obtained percentages.

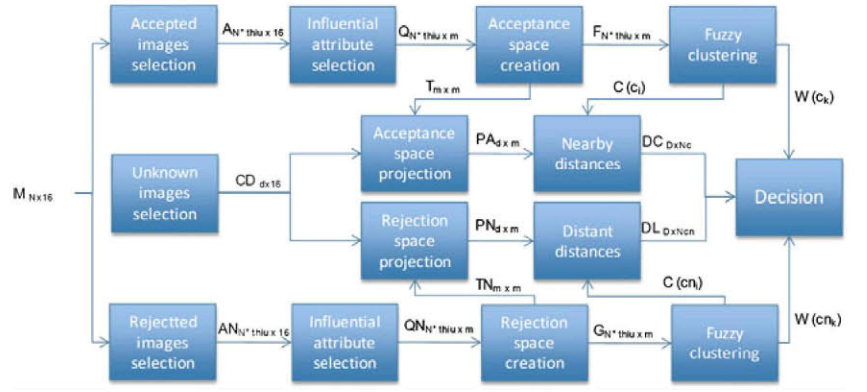
5 Content-based recommender system: recommendation process

Two algorithms have been designed for the personalized recommendation process. The main one is the *Profile-Based Recommendation Algorithm* (PBRA). It generates the content-based recommendations depending on the tastes inferred from the user preferences. The algorithm starts from a collection of ratings of images which can be obtained either explicitly (rating the images in a numeric scale, for example, from 1 to 5) or implicitly (through the user consumption). The other algorithm is the *Content-Based Initialization Algorithm* (CBIA), and it is used to avoid *cold start* problem. CBIA is applied for new users to generate recommendations until the system has collected enough information about the user. From that moment on, the PBRA is used to generate personalized and not general recommendations.

5.1 Profile-based recommendation algorithm: personalized subspaces and clustering

The whole PBRA process is shown in Fig. 2. It is composed of two branches: the first one infers and takes into account the image characteristics that make

Fig. 2 Content-based recommendation process



an image attractive for the user; the second one considers those attributes that make an image unattractive to the user. The objective of both branches is to create two personalized vector spaces: the *Acceptance space* and the *Rejection space*. The former allows the representation of images using a coordinate system adapted to the user tastes, and the latter takes the attributes which have a negative influence on the user as its coordinate system. These spaces allows the application of metrics and distances taking into account the characteristics of the images which have actual influence (be it positive or negative) on the visual impact of the user.

Next, the process followed in the *Acceptance* branch is explained. Simultaneously, an analogous process runs in the *Rejection* branch, and the final *Decision* module combines both information.

1. Accepted images selection

As the input data of the system, let us assume a user U , for whom we initially know N ratings over a collection of I images. The first filtering is the selection of the accepted and rejected images as the earliest step to build the personalized spaces. The threshold thi_u specifies the percentage of images to be taken into account for accepted images (the $N \cdot thi_u$ images with highest ratings) and for rejected images (the $N \cdot thi_u$ images with lowest ratings). thi_u value determines the compromise between reliability and sparsity of the recommendation results.

Consequently, the output matrix A has the dimension of the number of accepted images ($N \cdot thi_u$) times the number of basic general attributes which describe each image (16).

2. Influential attribute selection

The next step is to decide which attributes are really affecting the user's perception. This is done comparing the variance of each attribute at_i in the total image corpus (C_t) to its variance in the accepted images (C_a). If the variance of the attribute inside the accepted images is significantly lower than the variance in the total corpus (this is regulated by the adjustment parameter tha_u), this attribute is considered relevant. Therefore, the m attributes which satisfy condition (4) are selected for each user.

$$\frac{\text{Var}(at_i)|_{C_t}}{\text{Var}(at_i)|_{C_a}} > tha_u \quad (4)$$

3. Acceptance space creation

The relationship between the influential descriptors and its relevance degree are used to generate each personalized space vector. The coefficients of the transformation matrix T are obtained as the ratio of the covariance among the influential attributes in the total corpus and to that among the accepted images corpus.

$$T_{ij} = \frac{\text{Cov}(at_i, at_j)|_{C_t}}{\text{Cov}(at_i, at_j)|_{C_a}} \quad (5)$$

Matrix T will be used as the transformation matrix to represent every image defined by the 16 general attributes in the personalized vector space. This method is based on classical techniques used in *Content-Based Image Retrieval* (CBIR) applications, such as *Principal Component Analysis* (PCA) [31] or *Multidimensional Scaling* (MDS) [32].

4. Fuzzy clustering

After creating the new personalized subspace, it is necessary to know the reference points to measure the distance from these hotspots to the unknown images. The hotspots are obtained projecting the group of accepted images to the personalized vector space and finding the main centers in a clustering process. Each of the centers of the personalized space represents an aesthetic image style. The clustering process allows the recommendation of different style images. If only the main center is considered, the recommendation will consist of images with similar aesthetics characteristics. Consequently, the clustering adds diversity to the final results of the recommendation. A Fuzzy Clustering technique has been selected to generate the classification, because every image can be associated to every cluster (every image style) through a numeric weight. The sum of the weights associated to a cluster will give an idea of how important this cluster is for a user. This will define

the affinity with the user, and it will be used in the final recommendation stage.

The clustering algorithm was selected because of its simplicity and the trade-off between efficiency and experimental results in the Fuzzy C-Means [33]. The most important problem of the adjustment of the clustering parameters was the selection of the final number of clusters. Multiple validity indices were studied [34] and the best results were obtained with the Xie-Beni Index, which was selected [35]. It aims to quantify the ratio of the total variation within clusters to the separation among them:

$$XB(c) = \frac{\sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^m \|x_j - v_i\|^2}{N \cdot \min_{i,j} \|x_j - v_i\|^2} \quad (6)$$

Where c is the number of clusters, N is the number of input data (number of images), μ_{ij} the value of the hard partition, x_j the value of the input data and v_i the value of the centers.

5.2 Profile-based recommendation algorithm: distances and decision

The pair of personalized vector spaces (Acceptance Space and Rejection Space) and the results of the clustering for each user are stored in the user profile database. Thus, both spaces and clustering are not generated for each recommendation, but only when new information about user ratings or choices is received.

Starting from this information, only three steps are necessary to complete the recommendation process:

1. Unknown images projection

We have a corpus of d images unknown by the user. The way to apply our method is to project these images to the personalized subspaces. Therefore, the initial matrix, of dimension $d \times 16$, is transformed into two matrices adapted to user tastes: PA (Acceptance space projection) and PN (Rejection space projection), both of dimension $d \times m$.

2. Distance calculation

For each unknown image, two distances are obtained in parallel, one for each personalized subspace. These distances are calculated from the unknown images to the clusters, and they are represented through the matrix DC (for the Acceptance Space) and the matrix DL (for the Rejection Space). Both matrices have dimension $d \times nc$, where d is the number of unknown images in the corpus and nc is the number of clusters.

The metric chosen for the calculation of the distances, based on the weighted Minkowski distance, similar to other studies about visual perception and quality [36, 37], is shown in (7).

$$d(i, c_k) = \sqrt[p]{\sum_{n=1}^m \alpha_{kn} \cdot [(c_{kn} - \mu_n) - i_n]^p} \quad (7)$$

where the weight coefficient is defined in (8).

$$\alpha_{kn} = \frac{c_{kn} - \mu_n}{\sigma_n} \quad (8)$$

The Eq. (7) expresses the value for the distance from the unknown image i (composed of n dimensions/projected attributes) to the cluster c_k , and μ_n represents the mean value of the dimension n (corresponding to an attribute before personalizing the space) in the corpus of highly-rated images for the user. It should be underlined that after conducting experiments (as will be described in Section 7), the estimated optimum value for the coefficient p was found to be around 2.7. The weighting of the projected attributes is performed so that those which are further apart from the mean, especially in attributes with little dispersion (σ_n), are more relevant in the description of the tastes of the user.

3. Final decision

The final decision, which combines the distances in the Acceptance Space with those in the Rejection Space, and takes into account the weighting of the clusters, is made following the expression of $R(i)$, which represents the final rating predicted per user for the image i :

$$R(i) = \sqrt[m]{\left(\sum_{k=1}^K \frac{w(c_k)}{[d(i, c_k)]^2} \right)^m + \left(\left(\sum_{k'=1}^{K'} \frac{[w'(c_{k'})]^2}{[d'(i, c_{k'})] - 1} \right) / S \right)^m} \quad (9)$$

where K is the number of clusters of the Acceptance Space, K' is the number of clusters of the Rejection Space, $d(i, c_k)$ is the distance from the image i to the cluster k of the Acceptance Space, $w(c_k)$ is the weight of the cluster k in the Acceptance Space, $d'(i, c_{k'})$ is the distance from the image i to the cluster k' of the Rejection Space, $w'(c_{k'})$ is the weight of the cluster k' in the Rejection Space, m is the Minkowski coefficient, and S is the Smoothing parameter.

The expression (9) is also based on the Minkowski distance, and it takes into account the importance of both the positive and negative attributes in the user. The first term of the m -th root characterizes the influence of positive attributes, and it generates a higher rating for images closer to the referential centers representing the favorite aesthetic styles to the user (higher weight). The second term contributes with an additional rating which is higher when images are further from the styles that the user does not like.

S and m are adjustment input parameters of the system, but they can be automatically defined depending on the user's feedback, modeled as a parameter of usefulness of

the generated recommendations. The m parameter is the Minkowski coefficient, used to weight the extreme measures for both small and big distances. The S (Smoothing) parameter expresses the different weights for the accepted or rejected influential characteristics of the images. Users have a different S parameter because the taste of certain users is more influenced by characteristics they like whereas others are more influenced by those they dislike.

5.3 Content-based initialization algorithm

The content-based algorithm prevents the cold start problem associated with the number of users of the system. However, there is still a cold start problem, related to the number of opinions stored from each user. The described recommendation process needs to process a minimum number of opinions to model the taste of each user. Therefore, a Content-Based Initialization Algorithm has been designed with the aim of running while the general algorithm does not have enough information to provide representative results.

This algorithm takes only the reference of a highly rated image (which has just been seen by the user) and recommends other images with similar aesthetics. This information is stored until the PBRA process has enough information to generate the personalized subspaces required to run the general algorithm. The Content-Based Initialization Algorithm works applying (7) and (9) directly over the general space which is used to represent the images in a general way.

6 Implementation

To implement this system, it was decided to develop a main core coded in Java. This core can be integrated with a web interface, accessible and usable from any device supporting an Internet browser. The final interface has been implemented as an Android application. Final user tests have been developed in a tablet computer, as it can be seen in Fig. 3.

Into the Java core, we placed the classification and recommendation processes and algorithms described in the paper. Moreover, this core is the only module that interacts with the databases.

The client interface records information about the last recommendation divided into two different parts: content-based and collaborative recommendation. In doing so, when content-based parameters do not change, the client asks the server for the final results computing only collaborative filtering, thus improving the time responses, as we will see.

Due to the enormous amount of processes running, besides considering classification and recommendation mechanisms separately, we divided the system modules into layers. This allows the interface to communicate only with the upper layer



Fig. 3 System implementation in a tablet

and not with the rest of the algorithms. For convenience, our core gives its results to the interface using XML format.

6.1 Classification

The system offers the user the possibility to add pictures to the corpus. For a successful operation, the main core must have access to this picture corpus. Each new annotation requires an update of global attributes' mean and variance. This information is stored under persistence on the database. When performing this task, the new mean and variance calculation is optimized by using the following algorithms:

To update the mean: If a new item a_{N+1} is added, then the mean μ_N is updated to μ_{N+1} as follows:

$$\mu_{N+1} = \frac{1}{N+1} (N\mu_N + a_{N+1}) \quad (10)$$

Where N is the previous number of items, μ_N is the previous mean, a_{N+1} is the value of the new item and a_i is the value of the removed element.

To update the variance: If a new item is added:

$$\sigma_{N+1}^2 = \frac{1}{N+1} (N\sigma_N^2 + N\mu_N^2 + a_{N+1}^2 + (N+1)\mu_{N+1}^2 - 2N\mu_N\mu_{N+1} - 2\mu_{N+1}a_{N+1}) \quad (11)$$

Where N is the previous number of items, σ_N^2 is the previous variance, μ_N is the previous mean, a_{N+1} is the new item, a_i is the removed item and μ_{N+1} is the updated mean. If an item is removed, equations are analogous.

Similarly, each new user rating related to a picture requires an update of user ratings' mean and variance. A user can create new ratings over the result corpus shown on the interface. Surely, every rated picture is discarded for future recommendations. The use of these formulas for a

dynamic update of the mean and variance represents a great improvement in the mobile scenario regarding resource optimization and system efficiency, since it avoids processing every element: only the previous mean or variance value and the new element value are needed.

6.2 Recommendation

Recommendation process starts at the interface, where a user sets the recommendation parameters. It is important to note that the information about the last response is saved. Therefore, the interface does not request information to the application core for every recommendation. On the contrary, a new request is sent only if new recommendation parameters require doing so and the saved information is not enough. So, the core itself generates new spaces and clusters only when necessary. This is a very important improvement regarding the system timing responses.

The user is allowed to save parameters values as their default ones, by means of an option on the interface. These parameters are sent to the core and, at the same time, to the database, feeding the system for a future recommendation according to the user's taste.

7 Tests and experiments

The testing of the system has been divided into two parts: the testing of the classification process and the testing of the Content-based recommendation process. The collaborative filtering algorithm has not been tested because its design is similar to other current developments and the performance and results are well-known [38].

To evaluate the described modules 200 images were selected from the virtual art gallery *Ciudad de la pintura* (<http://pintura.aut.org>). These images represent paintings from the 15th to the 20th centuries, with different styles, and were scored by 60 users with a rating from 1 (dislike) to 5 (like). These images are size-normalized because this recommender takes only into account the concept of the image, but not its physical representation. However, these physical parameters can be included in further studies in order to determine their influence on the definition of the users' preferences. Several tests have been implemented to evaluate the system. As previously said, we have divided them into two parts: classification testing and recommendation testing.

7.1 Classification testing

The first test is necessary to check that the defined "Image recommendation descriptors" characterizes the tastes of the users in a reliable way. It also establishes a classification from best to worst characterization descriptors.

For each user, a number of $200 \cdot thi_u$ "favorite" images from the corpus were selected. This number of images for the system testing was chosen following the procedure of the first stage of the recommendation algorithm described in Section 5. The value selected for this test was $thi_u = 0,15$. Therefore, the 30 favorite images of each user were chosen for the test.

Starting from these data, a measurement called *Significance* has been defined.

Let a be one of the descriptors defined in Section 4. Therefore, the *Significance* of a is obtained by comparing both the descriptor values variance in the 200 images of the corpus and the descriptor values variance in each user's favorites images. The *Significance* of descriptor a can be defined as follows:

$$\text{Significance}_a = \sum_{\text{user}=1}^{60} \frac{\text{Var}(X_a)}{\text{Var}(Y_{au})} \quad (12)$$

where X corresponds to the values of the descriptor in the 200 images of the corpus, and Y to the values of the descriptor in the user favorite images. This test is analogous to the operation (4) of the recommendation process.

An additional parameter was defined, *Affected Users*, which establishes how many people from the 60 users tested are influenced by each descriptor. The way to obtain the final value is to add all the users who meet the condition $\frac{\text{Var}(X_a)}{\text{Var}(Y_a)} > \text{threshold}$. In Table 1 we can see the test results obtained by applying a threshold value=1.15, where the descriptors are sorted from best to worst, and The Ratio measurement is in charge of establishing a quotient between Significance and Affected users.

An analogous process was developed with the disliked images.

7.2 Recommendation testing

For the second group of tests, D ratings from the 200 ratings of each user were randomly deleted. These D images were the unknown images for the system. The other $N=200-D$ images were used to characterize the system, and the tastes of the users were inferred from them. It is important to consider that not all users have the same general tastes: there can be users with a considerable preference for most paintings, along with users with no preference for hardly any painting. The results of the system performance for a single user are not representative. To deal with this, the mean of the scores among all users was taken into account.

Usually, recommender systems performance is evaluated using the *Mean Absolute Error (MAE)* [39], but our approach does not calculate numeric predictions in the rating scale. The objective of the evaluation should be to estimate

Table 1 Image recommendation descriptors evaluation

Descriptor type	Descriptor	Significance	Affected users	Ratio (Sg/AU)
Luminance descriptors	Entropy variety	213.79	22 (37 %)	9.72
Texture descriptors	Line homogeneity	160.2951	44 (73 %)	3.64
Color descriptors	Chromatic variety (intra)	138.0416	43 (72 %)	3.21
Luminance descriptors	Bit-plane distribution entropy (4)	100.0674	56 (93 %)	1.79
Texture descriptors	Line energy	91.4375	45 (75 %)	2.03
Color descriptors	Color planarity	74.1061	40 (67 %)	1.85
Color descriptors	HSV color space (saturation variance)	73.4121	36 (60 %)	2.04
Luminance descriptors	Bit-plane distribution entropy (1)	68.5153	41 (68 %)	1.67
Color descriptors	HSV color space (saturation mean)	63.7606	44 (73 %)	1.45
Luminance descriptors	Direct luminance (variance)	60.5754	37 (62 %)	1.64
Color descriptors	Chromatic variety (inter)	44.1947	31 (52 %)	1.43
Color descriptors	Spatial coherence	41.7862	26 (43 %)	1.61
Luminance descriptors	Bit-plane distribution entropy (3)	27.8353	21 (35 %)	1.33
Color descriptors	Chromatic variety (N)	20.6617	15 (25 %)	1.38
Luminance descriptors	Bit-plane distribution entropy (2)	17.2227	12 (20 %)	1.44
Luminance descriptors	Direct luminance (mean)	5.3371	4 (7 %)	1.33

to which extent the *NR* recommended images are the preferred images for each user.

To achieve this goal, an Improvement measurement has been defined. The recommendation system selected the first *NR* recommendations and the mean of the real user ratings over these items was calculated. Then, this mean was compared to the mean of the scores of the unknown images for each user. The *Improvement* of the recommendation for each user *u* is considered as the increase of the recommendation mean over the user mean, associated with a number of recommended images, *NR*.

$$\text{Improvement}_{u, NR} = \frac{\text{Mean}(\text{Ratings}_{\text{recommended_images}})}{\text{Mean}(\text{Ratings}_{\text{unknown_images}})} \bigg|_{u, NR} \quad (13)$$

To obtain homogeneous results, the tests were developed taking the input parameters as constants, except for the *Number of Recommendations (NR)*. These values are defined in Table 2, and they were chosen after running a testing collection because they yielded the best average results for the group of testing users.

The first test of this block aims to compare the different variations of the recommendation algorithm described in Section 5 (Table 3). Each technique was performed for the 60 users, and the final Improvement value was the mean value of the 60 individual values. For this test, five recommended images were generated for each user by the system (*NR*=5). The complete algorithm, “Acceptance and Rejection with weighted clustering”, achieved the best result; the second, which does not take into account the weights of the clusters in the rejection branch of the algorithm, yielded similar results. The third technique does not take into

account the cluster weights for any of the branches, and the last one does not use the rejection branch. In this technique, the final recommendation decision comes only from the results of the distances obtained in the “Acceptance Space”.

The next experiment was used to check whether a smaller number of recommended images means that these images will be better for the user and, therefore, that the mean rating will be higher; that is, if decreasing the number of recommendations improves their quality.

As we already used 150 training images, the results for this test were obtained modifying the number of recommended images from 1 to 50, which is the total number of unknown images. Therefore, if the system recommends 50 images, the improvement should be equal to 1, because there is no recommendation, and this value should increase as the number of images recommended by the system decreases. The recommended images should be the best images for the user.

Figure 4 shows the results of the mean Improvement value for the 60 users as the number of recommended

Table 2 Input parameters values chosen for testing

Input parameter	Value
Image Threshold	0.15
Attribute Threshold	1.15
m coefficient (9)	2.5
S coefficient (9)	6
Number of training images	120

Table 3 Content-based technique comparative for NR=5

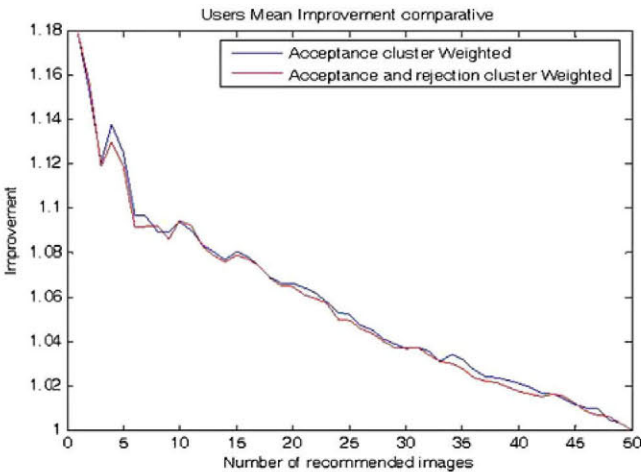
Technique	Mean improvement
Acceptance and Rejection with weighted clustering	1.1255
Acceptance with weighted clustering	1.1205
Acceptance and Rejection with basic clustering	1.1195
Acceptance branch	1.0988

images changes. It can be seen that the results of the “Acceptance cluster Weighted” technique are similar to those of the “Acceptance and rejection cluster Weighted” technique. Therefore, the rejection branch does not offer a benefit in the rating prediction performance. However, the inclusion of this stage is justified to solve the overspecialization problem typical of content-based recommender systems [6].

Nevertheless, if the content-based recommendation system is going to be integrated with a social technique into a hybrid system, the overspecialization problem disappears [6].

Some tests have been performed over the same corpus for applying our algorithm to a hybrid system. On the first place, a collaborative filtering algorithm has been implemented with the best configuration for our use case [16]: the user-based collaborative filtering. Therefore, Pearson correlation (cosine of deviation from the mean) has been used for obtaining the similarities between users (16), and the final predictions have been computed using a weighted sum of deviations from the mean (17):

$$w(a, u) = \frac{\sum_{i \in s_a \cap s_u} (v_{ai} - \bar{v}_a)(v_{ui} - \bar{v}_u)}{\sqrt{\sum_{i \in s_a \cap s_u} (v_{ai} - \bar{v}_a)^2 \sum_{i \in s_a \cap s_u} (v_{ui} - \bar{v}_u)^2}} \quad (14)$$

**Fig. 4** Mean improvement value

Where v_{ui} is the rating of user u on item i , S_u is the set of items rated by user u , and \bar{v}_u is the mean rating of user u .

$$Pai = \bar{v}_a + \frac{\sum_{u \in s_u} w(a, u) \cdot (v_{ui} - \bar{v}_u)}{\sum_{u \in s_u} |w(a, u)|} \quad (15)$$

According to this configuration, the mean improvement has been obtained in the same conditions of the content-based tests, and the result for lists of five recommendations (NR=5) is $I_{CF}=1.445$. Afterwards, two different hybridization techniques have been implemented to join content-based and collaborative algorithms: switching hybridization and cascade hybridization [19].

The switching hybridization is the simplest one. The system decides from past ratings which recommendation technique is better for each user. The selected technique (content-based or collaborative) is applied to generate the predictions for the new items. The improvement result in the previously mentioned scenario is $I_{SW}=1.4678$, which is only a small increase.

In the cascade hybridization, the collaborative filtering algorithm is used to refine the content-based results. The first stage (content-based algorithm) selects the images whose style and formal characteristics are correlated with the user tastes. Starting from these partial ratings for the items, the collaborative algorithm generates a final rating. The weight of each stage is obtained dynamically for each user, computing the results of both algorithms for the items already rated. In this case, a huge improvement is achieved, because the mean user result under the same conditions is $I_{HC}=1.66$. This result approaches to the theoretical limit of the recommendation, computed in this scenario in $I_{TH}=1.8440$. Anyway, the implemented final application also allows the user adjusting manually the weight of both content-based and collaborative techniques. Table 4 summarizes the mean improvement results for the several techniques implemented.

Analogously, inverse cascade technique (collaborative-content) was tested, and the efficiency results were slightly lower than the content-collaborative order, because content algorithm is more reliable and collaborative algorithm more efficient.

Table 4 General Techniques Comparative for NR=5

Technique	Mean improvement
Content-Based Recommendation Improvement	1.1255
Collaborative Filtering Improvement	1.4450
Hybrid Recommendation Improvement (Switching)	1.4678
Hybrid Recommendation Improvement (Cascade)	1.6600
Theoretical Maximum Improvement	1.8440

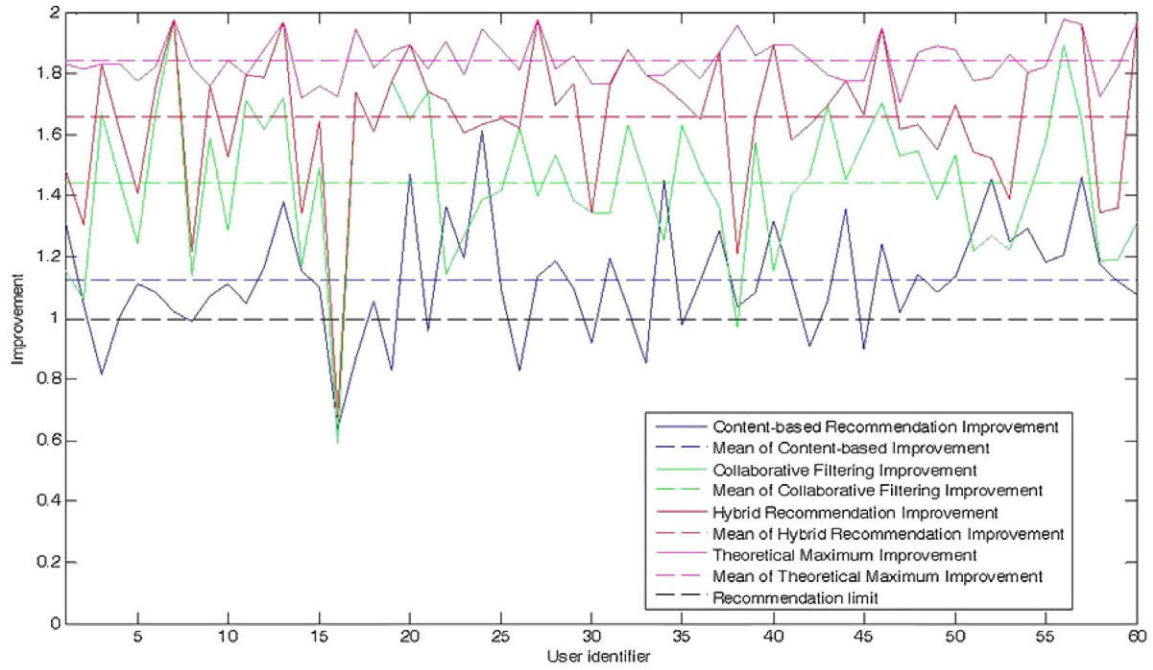


Fig. 5 Comparison between techniques and theoretical maximum improvement for NR=5

Finally, Fig. 5 compares the improvement of the several recommendation techniques for each user with the theoretical bound (magenta line in Fig. 5). The improvement value is different for each user, because it depends on the values of the personal ratings.

To set an upper limit for the recommendation, we consider that theoretically, the maximum value is the mean of the ratings of the NR best images from each user, which implies that the system recommends the preferred images. Therefore, each personal improvement should be compared with its theoretical maximum value.

As seen in Fig. 5, the hybrid algorithm was useful for 59 of the 60 users, and the recommendation achieves the theoretical maximum for 19 of the 60 users. The Root Mean Square Error of each technique (for the 60 users) with regard to the theoretical maximum can be seen in Table 5.

8 Future work

This work opens multiple future research lines. The main challenge is to improve the hybridization of the content-based and the collaborative filtering algorithms [6]. Our system allows a manual adjustment of the weights of both algorithms in the final result, and it also estimates these weights automatically computing the performance of the algorithms with past ratings. Nevertheless, this process could also take into account characteristics inherent to the users (for example, their taste independence, or the peculiarity of their tastes) obtained via the algorithms, or data

about the reliability of collaborative filtering (number of users, degree of affinity with other users, etc.) or content-based methods (aesthetics taste dispersion, undefined visual tastes, etc.). Based on these data, the system could estimate the best weighting of both methods for each user.

Other important characteristics to take into account in the future work come from the user's context, be it physical (e.g. *a user's location*), social (e.g. the social network of a user) or emotional (e.g. user's mood at a given moment) [8, 40]. These characteristics can be used for hybridization weighting, automatic adjusting of the input parameters in the content-based algorithm, or configuration of the collaborative filtering [41].

Another important future research line is related to the inference of implicit information to feed the system as a complement to the explicit ratings [42].

9 Conclusion

In this paper we have proposed a hybrid recommender system for images in mobile social networks, with a novel

Table 5 RMSE of recommendation techniques

Technique	Root Mean Square Error (RMSE)
Content-Based Recommendation	5.7373
Collaborative Filtering	3.5626
Hybrid Recommendation (Cascade)	2.1380

content-based algorithm and hybrid system. The content-based algorithm is based on aesthetics and formal characteristics of the images, which cannot produce a real advance if not hybridized with collaborative filtering to be adapted to the social networks inherent characteristics.

The system has demonstrated to generate better results than other collaborative recommenders, offering 21.8 % better results than collaborative filtering algorithms. This percentage has been obtained by comparing improvement results of both algorithms. These results are produced from the use of the personal aesthetics perception in the recommendation process which is able to distinguish real preferences from users, not able to be covered by other social techniques.

In addition our system offers the user the freedom to select their preferences and has been implemented as an application suitable to mobile terminals and social networks.

Acknowledgments This work was supported in part by the Spanish Ministry of Science and Innovation - CDTI under the contract of the CENIT Program, project "BUSCAMEDIA" (CEN- 20091026) (www.cenit-buscamedia.es). The authors of this paper would like to thank the people who helped to the completion of this paper by providing their ratings of the selected images to test and validate our algorithms. Moreover, the authors thank Álvaro Martínez and Iago Fernández-Cedrón for the help with the implementation of the Android application, and Javier Arróspeide for the English language review.

References

1. Roberts M, Ducheneaut N, Begole B, Partridge K, Price B, Bellotti V, Walendowski A, Rasmussen P (2008) Scalable architecture for context-aware activity-detecting mobile recommendation systems. *International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp 1–6, 23–26 June 2008
2. Gombrich EH (2002) *Art and illusion: a study in the psychology of pictorial representation*. Phaidon Press, London
3. Itten J (2002) *The art of color: the subjective experience and objective rationale of color*. Wiley, New York
4. Davis S (ed) (2000) *Color perception: philosophical, psychological, artistic, and computational perspectives*. Oxford University Press, New York
5. Ren T, Wu G (2010) Automatic image retargeting evaluation based on user perception. *Image Processing (ICIP)*, 2010 17th IEEE International Conference on, vol., no., pp 1569–1572, 26–29 Sept. 2010
6. Adomavicius G, Tuzhilin A (2006) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17(6):734–742
7. Veltkamp RC, Tanase M (2000) Content-based image retrieval systems: a survey. Technical Report UU-CS-2000-34, Dept. of Computing Science, Utrecht University
8. Boutemedjet S, Ziou D (2008) A graphical model for context-aware visual content recommendation. *IEEE Trans Multimed* 10(1):52–62
9. Gudivada VN, Raghavan VV (1995) Content based image retrieval systems. *Computer* 28(9):18–22
10. Godoy D, Amandi A (2008) Hybrid content and tag-based profiles for recommendation in collaborative tagging systems. *Proceedings of the 6th Latin American web congress*. IEEE Computer Society Vila Velha, Brazil, Pages 58–65
11. Hyvönen E, Styrman A, Saarela S (2002) Ontology-based image retrieval. Towards the semantic web and web services. *Proceedings of the XML Finland 2002 Conference*, Helsinki, Finland, 2002, Pages 15–27
12. Sheth B, Maes P (1993) Evolving agents for personalized information filtering. *Proceedings to the Ninth Conference on Artificial Intelligence for Applications*, March 1993, Pages 345–352
13. Balabanovic M, Shoham Y (1997) Fab: content-based, collaborative recommendation. *Communications of ACM* 40, March 1997, Pages 66–72
14. Tkalčič M, Burnik U, Košir A. Using affective parameters in a content-based recommender system for images. *User Model User-Adap* 20(4):279–311
15. Benítez AB, Beigi M, Shih-Fu C (1998) Using relevance feedback in content-based image metasearch. *IEEE Internet Comput* 2(4):59–69
16. Candillier L, Meyer F, Boullé M (2007) Comparing state-of-the-art collaborative filtering systems. *Proceedings of the 5th international conference on machine learning and data mining in pattern recognition*, July 18–20, 2007, Leipzig, Germany
17. Gao Y, Luo H, Fan J (2009) Personalized image recommendation. *15th International Multimedia Modeling Conference, MMM 2009*, volume 5371 LNCS, pp 217–219, Sophia-Antipolis, France, 7–9 Jan 2009
18. Milicevic AK, Nanopoulos A, Ivanovic M (2010) Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, Springer Netherlands, Pages 187–209
19. Burke R (2002) Hybrid recommender systems: survey and experiments. *User modeling and user-adapted interaction*, Springer Netherlands, issue 4, vol. 12, Nov. 2002, pp 331–370
20. Sang Hyun C, Young-Seon J, Jeong MK (2010) A hybrid recommendation method with reduced data for large-scale application. *IEEE Trans Syst Man Cybern C Appl Rev* 40(5):557–566
21. Kazienko P, Musiał K and Kajdanowicz T (2011) Multidimensional social network in the social recommender system. *IEEE Trans Syst Man Cybern Syst Hum* Vol. 41, N° 4, Julio 2011
22. Arazy O, Kumar N, Shapira B (2009) Improving social recommender systems. *Social computing*. IEEE Computer Society Julio/Agosto
23. Granovetter MS (1983) The strength of the weak tie: revisited [PDF]. *Socio Theor* 1:201–233
24. Krackhardt D (1992) The strength of strong ties: the importance of philos in organizations. In: Nohria N, Eccles R (eds) *Networks and organizations: structure, form, and action*. Harvard Business School Press, Boston, pp 216–239
25. Herlocker JL, Constant JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):5–53
26. O'Donovan J, Smyth B (2005) Trust in recommender systems. *IUI'05*, January 9–12, 2005, San Diego, California, USA
27. Manjunath BS, Salembier P and Sikora T (2002) Introduction to MPEG-7: multimedia content description interface. Ed. John Wiley & Sons, Ltd
28. Bastan M, Cam H, Gudukbay U, Ulusoy O (2009) An MPEG-7 compatible video retrieval system with integrated support for complex multimodal queries. *IEEE MultiMedia*
29. Barrilero M, Uribe S, Alduan M, Sanchez F, Alvarez F (2011) Innetwork content based image recommendation system for Contentaware Networks. *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2011, Pages 115–120
30. Shan Z, Hai-tao W (2008) Image retrieval based on bit-plane distribution entropy. *2008 International Conference on Computer Science and Software Engineering*.

31. Ye F, Shi Z, Shi Z (2009) A comparative study of PCA, LDA and kernel LDA for image classification. Ubiquitous virtual reality, 2009, ISUVR '09. International Symposium on, vol., no., pp 51–54, 8–11 July 2009
32. Feng-Cheng C, Hsueh-Ming H (2008) An improved presentation method for relevance feedback in a content-based image retrieval system. Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHHMSP '08 International Conference on, vol., no., pp 91–94, 15–17 Aug. 2008
33. Hadjhamadi AH, Homayounpour MM, Ahadi SM (2008) Robust weighted fuzzy c-means clustering. Fuzzy Systems, 2008, FUZZ-IEEE 2008, (IEEE World Congress on Computational Intelligence), IEEE International Conference on, vol., no., pp 305–311, 1–6 June 2008
34. Sledge II, Bezdek JC, Havens TC, Keller JM (2010) Relational generalizations of cluster validity indices. IEEE Trans Fuzzy Syst 18(4):771–786
35. Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 13(8):841–847
36. Jones BC, Wilkes DM (2001) A new analysis framework for relevance feedback-driven similarity measure refinement in content-based image retrieval. Computer vision and pattern recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol.1, no., pp I-920- I-925 vol. 1, 2001
37. Sugimoto O, Naito S, Sakazawa S, Koike A (2009) Objective perceptual video quality measurement method based on hybrid no reference framework. Image Processing (ICIP), 2009 16th IEEE International Conference on, vol., no., pp 2237–2240, 7–10 Nov. 2009
38. Cho JH, Kwon K, Park Y (2007) Collaborative filtering using dual information sources. IEEE Intell Syst 22(3):30–38
39. Breese JS, Heckerman D and Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In Proc. 14th Conf. Uncertainty Artificial Intell., 1998, pp 43–52
40. Woerndl W, Groh G (2007) Utilizing physical and social context to improve recommender systems. Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on, vol., no., pp 123–128, 5–12 Nov. 2007
41. Widisinghe A, Ranasinghe D, Kulathilaka K, Kaluarachchi R, Wimalawarne KADNK (2010) picSEEK: collaborative filtering for context-based image recommendation. Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on, vol., no., pp 225–232, 17–19 Dec. 2010
42. Sanchez F, Barrilero M, Alvarez F, Cisneros G (2011) Embedded audiovisual recommender system for user terminals based on user social and implicit information. Consumer Electronics (ICCE), 2011 IEEE International Conference on, vol., no., pp 773–774, 9–12 Jan. 2011