Information Sciences 181 (2011) 4943–4958

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Halfway through the semantic gap: Prosemantic features for image retrieval

Gianluigi Ciocca [a,1], Claudio Cusano [a,*,1], Simone Santini [b,1], Raimondo Schettini [a,1]

[a] DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione), Università degli Studi di Milano–Bicocca, Viale Sarca 336, 20126 Milano, Italy
[b] Escuela Politécnica Superior, Universidad Autónoma de Madrid, C/ Tomas y Valiente 11, 28049 Madrid, Spain

ARTICLE INFO

Article history:
Received 4 February 2010
Received in revised form 3 June 2011
Accepted 12 June 2011
Available online 13 July 2011

Keywords:
Image retrieval
Relevance feedback
Image classification
Prosemantic features
Semantic gap
Image indexing

ABSTRACT

We present here, an image description approach based on *prosemantic* features. These features are obtained through a two-level feature extraction process. A first level of features, related to image structure and color distribution, is extracted from the images, and used as input to a bank of classifiers, each one trained to recognize a given category. Packing together the scores, the features that we call prosemantic are obtained, and used to index images in an image retrieval system where searches are performed using relevance feedback. Prosemantic features have been evaluated on a public domain dataset, and compared against two different sets of features. Our experiments show that the use of prosemantic features allows for a more successful and quick retrieval with respect to the other features considered.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Even with the most unrelenting optimism in the world, one must admit that, thus far, the attempt to capture the meaning of an image into a set of computable features has not been satisfactory. In its most ambitious form, the problem is of course insoluble, simply because the meaning of an image is not a function of its contents, but depends on the discursive practices of the environment in which it was produced, the cultural context in which it is read, and so on. Still, the problem of one might call "ophtosemantics" (semantics of the eye, or semantics of perception) exists. There is little doubt (and some measurable evidence) that, even without the cultural and interpretative scaffolding that we need to make full sense of an image, we do, even at a relatively low level of understanding, classify what we see based on a fairly sophisticated classification system [40,31,17]. This kind of classification, which we dub it prosemantics (a term we coined from "towards the meaning") is the subject of this paper. More in detail, we propose here a method for image description based on prosemantic features. We will show how these features can be embedded into a standard image retrieval engine based on relevance feedback. A user study has been conducted to demonstrate the effectiveness of our approach, by comparing prosemantic feature to a set of state of the art low-level features (see Section 5).

The representation solution that we propose is fairly simple and effective but in order to frame it properly it is necessary to draw a conceptual map of the path that leads to it. There is some evidence that pre-attentive classification is based on the similarity of the *perceptum* with a collection of prototypical templates. That is, at a pre-interpretative level, a concept such as "dog" is represented as a set of templates of more or less prototypical dogs, and the determination of the "dogness" of the

[*] Corresponding author.
E-mail addresses: ciocca@disco.unimib.it (G. Ciocca), cusano@disco.unimib.it (C. Cusano), simone.santini@uam.es (S. Santini), schettini@disco.unimib.it (R. Schettini).
[1] The authors contributed equally to this work.

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.06.025

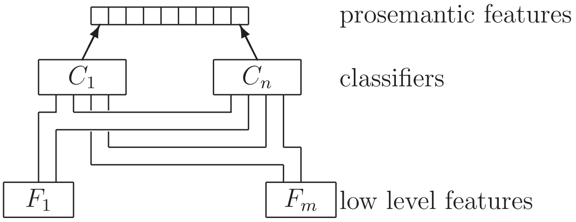

Fig. 1. A schema of the prosemantic feature extraction.

perceptum is a function of its similarity to one or more of these dog templates [29,18]. All this, of course, within the limitations of a semantics that does not depend on any non-visual knowledge: there is no difference (not even in principle) between a wolf and a dog, and whales are fish.

This consideration seems to point out that a possible way of defining semantics is through a set of prototypical templates that acts as "attractors" of relevant semantic categories. Two considerations are relevant here: this *prototypicality* method does not reduce to template matching or to classification, and it does not dispense us from defining features. The method does not reduce to template matching nor to classification because prototypicality does not postulate a one-to-one correspondence between prototypes and categories. There may be several prototypes for one category, and the same prototype may be characteristics of several categories. The identity of an image, therefore, is not given simply by its associations with the closest template, but by its similarity with all the available prototypes. In this sense it is a similarity-based distributed representation. Second, comparison with the prototypes cannot be done simply by matching images with images, but needs to be based on suitable features that isolate the characteristics that are relevant to matching.

Previous studies suggest that when our perceptual system does similarity matching, it does not do so completely independently of classification (see, for instance, the experiment of Maruyama et al. concerning the interpretation of face images [30]). Somehow a partial interpretation of an image is present quite early in understanding, and influences the way templates are matched. We need to find features that give us a perceptual space in which we can apply the prototypicality idea. To come back to the example of the dog, the dog templates represent "dogness", and the images are compared to them to the extent that we are willing to recognize them as dogs. This is tantamount to the previous idea of representing the images by the degree to which they are associated to some prototypical categories. If we try to classify images as dogs, horses, or snakes, we can expect that the actual dogs will cluster around some prototypical dogs, horses around prototypical horses, and snakes around prototypical snakes. The advantages of putting these results into a metric space, instead of just doing classification is that if we start looking at images of, say, pigs, we should not need to add a new category: the category of pigs should emerge alone, based on pigs' similarity to dogs and horses, as a cluster somewhere between the two.

We implemented this through a two-level feature system (see Fig. 1). A first level of features are extracted from the images, and used as inputs to a bank of *n* classifiers, each one trained to recognize a given category. The outputs of the classifiers are vectors in the *prosemantic* feature space in which we operate. While we believe that this class of features has a broad validity and can be applied to the most diverse problems in computer vision, in this paper we concentrate on image retrieval, and in order to better determine and isolate the contribution of prosemantic features we embed them into a fairly standard state-of-the-art retrieval environment. The feature vectors will be embedded into a feature space in which similarity is defined as a function of the distance, and the search is done using relevance feedback.

With respect to other approaches, based on automatic annotation (where keywords are automatically assigned by classifiers [1,4,25]), prosemantic features present a few advantages: (i) they are robust against misclassification (since they rely on soft membership); (ii) they can deal with images whose categorizations do not exactly match the classes on which the classifiers have been trained; (iii) they retain information about low-level visual properties of the images, which sometimes is very useful (e.g. retrieval of gray level images); (iv) their representation is as simple as a real valued vector of a relatively small number of components, therefore they can be effectively embedded in frameworks designed to deal with vectors of feature values.

2. Related work

The literature on content based retrieval has become so vast that any attempt at exhaustivity, or even representativity, in a paper like this would be futile. A survey of some of the most important techniques used in Content-Based Image Retrieval (CBIR) systems can be found in [43]. Here, we can only afford to briefly mention those papers that, in technical development or general philosophy, are related to the work that we are presenting here. To overcome the necessity to manually describe the image content, many of these systems are based on image features derived from computer vision, which can be computed directly and automatically from the images themselves. However, researchers soon realized that simple content-based features could not characterize even the ophtosemantics of images to the degree of generality and sophistication that was required for general-purpose retrieval. In order to come to grips with this problem and to provide satisfactory retrieval performance, hermeneutic solutions were introduced in the retrieval process to take into account the subjectivity of human perception. One of these solutions is *relevance feedback* [61], which relies on the interaction with the user to provide the system with examples of images relevant to the query. The system then refines its result depending on the selected images. The user's feedback provides a way to infer short term and case-specific query semantics. An example of this can be found in [26] where the system learns a non-linear embedding that maps clusters of images into a hidden space of semantic attributes. Long term learning can be achieved by logging the previous user's interactions for further processing [16].

Other systems explicitly extract and embed in the retrieval process semantic information about the image content through the use of automatic classification techniques [19]. These techniques can then be employed to automatically annotate the image content with keywords, which are then used for retrieval. If the underlying annotation is reliable, text-based image retrieval can be semantically more meaningful than other retrieval approaches [16]. Concept detection techniques categorize images into general categories such as city, landscape, sunset, forest, sea, etc., using supervised classification [49,7]. The idea here is that meaning is provided implicitly through the classification of the training set, and it will supplement and integrate the low-level information provided by the features. Note that this technique postulates the existence of a fairly strong correlation between the contents of an image and its meaning. We will return on this kind of integration of high-level semantics further on.

The annotation approaches described above can be considered as an example of *crisp* annotation: if an image is annotated with a given label, then the image expresses that concept and belongs to the corresponding class. An alternative approach is presented in [6], where the authors tested two classification approaches to *soft* image annotation: support vector machines (SVMs) and Bayes point machines (BPMs). At the end of the annotation process, each image is annotated with a label vector, and a confidence coefficient is assigned to each label in the vector. These confidence coefficients can then be used in a text-based search where images are retrieved and ranked according to the confidence coefficient of the matching labels (see also the OFFS technique [48] and its improvement [56]).

The approach in [44] tries to use vector space techniques, typical of query by example, to deal with semantics. Semantic information is learned directly from the image content and forms a vector of semantic weights. Each weight is associated to a concept and is derived from the confidence score obtained by a support vector machine trained to recognize that concept. Retrieval in the semantic space is based on similarity comparisons between two model vectors using the L_2 metric. A similar approach was followed in [28].

With the exception of a few examples, all these techniques deal with the problem of semantic image retrieval from the point of view of indexing, viz. they focus on the accuracy of the indexing scheme. Few have been evaluated in a CBIR scenario or tested on large image databases.

One of the first attempts to integrate and compare semantic keywords and low-level features into a single CBIR framework is the SIMPLIcity system [55]. The semantic classification is used to categorize images so that different semantically-adaptive search methods can be applied to each category. The system is also able to narrow down the subset of images to be searched by selecting those in the same category as the query. The reference categories chosen by the author are textured vs. non textured and graphics vs. photos. A more recent paper [39] defines a new paradigm, denoted query-by-semantic-example (QBSE), which combines a query-by-example approach with semantic retrieval. Using the vector model to describe the image contents, the authors define a vector of semantic multinomial values, where each value is associated to a specific concept. They compared the QBSE and the query-by-visual-examples approaches in a CBIR system within a minimum probability error retrieval framework.

In [9] we presented an experiment where broad high level concepts and low level visual features are combined. In particular, we designed an image description method based on three multi-concept classifiers: day/sunset/night, urban/rural, and mountain/sea. The paper reports the outcome of qualitative experiments performed using the combination of the classifiers' output and low-level features. Preliminary results on a database of more than 46,000 photos, show that this approach effectively improves the accuracy of the image retrieval sessions.

A further possibility, made realistic by the widespread success of "social networks" on the Internet, is to use the classification made by other people as semantic prototypes [15]. This approach is conceptually independent from and complementary to the one presented here, and the two can easily be integrated.

We originally proposed prosemantic features in [12]. Here, we discuss in detail their conceptual background, and we present a thorough experimentation which further explores the effectiveness of a prosemantic representation in content-based image retrieval.

3. Prosemantic image descriptors

As we mentioned in the introduction, in our approach we try to retain the advantages of using classifiers—namely the possibility of introducing semantics implicitly into the classifier through the high level categorization of the training set—without the disadvantages—namely that a system based a on classifier is useful only as long as the queries are made on the categories on which it was trained. To do this, we begin by describing the images with a suitable set of content ("low-level") features. These features are used as input to an array of 56 soft classifiers,[2] trained to recognize a set of 14 par-

[2] The number 56 has nothing magic about it. We use 4 feature descriptions of the images and 14 categories. One classifier per category per feature gives the grand total of 56.

tially overlapping classes. The outputs of the classifiers form a 56-dimensional vector that we place in a suitable metric space in order to index the images using relevance feedback.

It should also be noted that semantic descriptors cannot completely replace pictorial features, as only low-level features can discriminate images with the same semantics, but different visual properties. In fact, properties like color, shape, texture… provide a natural and intuitive way for describing images. Unfortunately, their CBIR typical representation as image statistics is not intuitive at all for the average user.

As a consequence of considerations such as these, there is an increasing interest in "soft" semantic representations of images, that is, on systems that (unlike the ontological approach) either do not rely on external tags for the semantic characterization of images or, if they do, supplement them with semantics inferred from the image content through the use of classifiers. The so-called *semantic spaces*, in which the image contents are classified and represented in a space of concepts have begun to appear in the literature [2,51].

This is part of the reason why we do not just use the results of the classifiers, but put their outputs into the prosemantic space: compared to a "crisp" semantic description of the images (e.g. "sunset on the beach"), prosemantic features provide a richer description of visual content by correlating low-level features to prototypical scenes (e.g. "image with an edge distribution that can easily be found in seaside scenes").

In order to provide semantically meaningful information about the content of the images, several categories in which images may be automatically classified have been proposed [49,50,53,41,46]. One of the problems of these purely categorical representations is that they consider each concept as atomic, so that each possible concept needs a classifier and an axis on which it can be represented. This has led to systems with a very large number of independent concepts, whose number is usually in the hundreds [57,58] and can reach the thousands of concepts [23,24], creating problems not only because of the sheer complexity of the system, but for its brittleness as well: any concept not explicitly included in the system is *ipso facto* impossible to represent.

The main purpose of this paper is to explore an alternative in the representation of the concept space, an alternative that we can call *generative* (we use this term in the connotation in which it is used, for example, in generative grammars). It is a well established psychological fact that visual concepts are not independent, and that concepts combine with each other to form new ones [35]. We have exploited this fact and the geometry of the concept space to see whether a relatively small set of concepts could work as a *base* of the concept space, so that further concepts, not explicitly designed into the system, could be derived from them. This would solve both problems of the current semantic representations since, on the one hand, it will help in keeping the number of concepts manageable and, on the other hand; it will allow for new concepts to be recognized dynamically, as combinations of the existing ones, without having been explicitly learned.

As a proof of this concept, we have considered a very small set of categories, as few as 14: animals, city, close-up, desert, flowers, forest, indoor, mountain, night, people, rural, sea, street, and sunset. Some classes describe the image at a scene level (city, close-up, desert, forest, indoor, mountain, night, rural, sea, street, sunset) other describe the main subject of the picture (animals, flowers, people). While it is unlikely that just 14 categories will suffice to form the basis of a working system, we have decided to test the idea by placing ourselves in a limit situation, a situation in which the validity of the hypothesis might be more evident (a system with too many categories might overfit the data and fail to highlight the possibilities of combination of categories). There is, of course, nothing special about the number 14 or the specific categories that we have chosen, except that they are fairly general and common ones. Many categorization systems work on TRECVid categories, which tend to be haphazard and chosen somewhat *ad hoc*, so the literature is unlikely to offer guidelines in this sense. We have resorted to the psychological literature [36] to see what were the broad divisions operated by people were engaged in categorization (day/night, indoor/outdoor) and selected a set of categories that, on the one hand, contained the essential dichotomies expressed in the psychological literature and, on the other hand, were well-established in the computer vision community, so that it was known beforehand that efficient classifiers for them could be built.

For each class, we trained several classifiers using different low-level features. This decision is not motivated by the need from a more robust classification (which is the most common reason for adopting a multiple classifiers strategy), but by the desire to exploit the relationship between classes and individual features. We allow the retrieval system, through a relevance feedback algorithm [61], to select which features and classes are appropriate on a case by case basis. We use four features, selected from the most common features in the literature about image classification, and divided along a color-shape and a locality axes: two features represent color, and two represent shape; two features are local, and two are global. We tuned the parameters of the features trying to maximize the generalization accuracy of the resulting classifiers.

For their simplicity and satisfactory performance, bag-of-features representations have become widely used for image content classification and retrieval [59,54,47,21]. The basic idea is to select a collection of representative patches of the image, compute a visual descriptor for each patch, and use the resulting distribution of descriptors to characterize the whole image. In our work, the patches are the areas surrounding distinctive key-points and are described using the Scale Invariant Feature Transform (SIFT). SIFT descriptors are invariant to image scale and rotation, and have been shown to be robust across a substantial range of affine distortions, changes in 3D viewpoint, additions of noise, and changes in illumination [27]. We adopted the implementation described in [52] for both key-point detection and description. The SIFT descriptors extracted from an image are then quantized into "visual words", which are defined by clustering a large number of descriptors extracted from a set of training images [34]. To do so we applied the Antipole clustering method [3] to select the visual words from a set of more than 15 millions descriptors extracted from the images in the dataset described in the next section. Note that the clustering method automatically determines the number of clusters: in this case, 1096 clusters have been selected. The final feature vector is the normalized histogram of the occurrences of the visual words in the image (1096 components).

Statistics about the direction of edges may greatly help in discriminating between images depicting natural and man made objects [50]. To describe the most salient edges we used an eight bin edge direction histogram. The image is subdivided into 8 × 8 blocks, and a histogram for each block is computed (for a total of 512 components). Only the points for which the magnitude of the gradient, computed using Gaussian derivative filters, exceeds a set threshold will contribute to the histograms (a pilot study revealed that $\sigma = 1$ and a threshold of 0.5 were suitable values).

Spatial color distribution is one of the most widely used features in image content analysis and categorization. In fact, some classes of images may be characterized in terms of layout of color regions, such as blue sky on the top or green grass on the bottom for country landscape images that contain the horizon. Similarly to Vailaya et al. [49], we divided each image into 9 × 9 blocks and computed the mean and standard deviation of the values of the color channels of the pixels in each block. The LUV color space is used here, since moments in this color space are more discriminant than in other spaces, at least for image retrieval [20]. This feature includes 486 components (six for each block).

Color moments are only marginally useful when the blocks contain heterogeneous color regions. Therefore, a global color histogram has been selected as a second color feature. The RGB color space has been subdivided in 512 bins by a uniform quantization of each component in eight ranges. Different levels of quantization led to comparable results in terms of classification error; we chose the 8 × 8 × 8 quantization since it gives a feature vector of size comparable with the other features considered.

3.1. Learning the prosemantic features

In order to collect suitable training samples, we queried various image search engines on the web with several keywords related to the classes, and downloaded the resulting pictures. The images were then manually inspected in order to remove those that did not belong to the classes, as well as those of poor quality. The final data set consists of 30,084 pictures, divided into 14 sets of more than 2,000 images each. For each class, a set of negative examples was selected by taking pictures from the other classes. Since the classes may overlap, manual inspection was needed to verify that all the selected images were actually negative examples. Note that this data set is completely separated from the one we used for evaluation.

For each combination of low-level feature and class, a Support Vector Machine (SVM) was trained using the implementation described in [5]. The central idea of SVM is to adjust a discriminating function so that it makes optimal use of the separability information of boundary cases [13]. Given a training set of feature vectors $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\},\ \mathbf{x}_i \in \mathbb{R}^d$ together with the class labels $\{y_1,\ldots,y_N\}$, $y_i \in \{-1,+1\}$, the SVM requires the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle_{\mathcal{H}} + C\sum_{i=1}^{N}\xi_i, \\ \text{subject to} \quad & y_i\big(\langle \mathbf{w},\phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b\big) \geqslant 1-\xi_i, \\ & \xi_i \geqslant 0, \quad i \in 1,\ldots,N, \end{aligned} \tag{1}$$

where the function ϕ is used to map the training vectors into a higher (possibly infinite) dimensional space $\mathcal{H}$ characterized by the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. The optimal values of $\mathbf{w} \in \mathcal{H}$ and $b \in \mathbb{R}$ define the hyperplane that maximizes the separation between the two classes. To deal with the case of non linearly separable classes, the slack variables ξ_i are penalized by the coefficient C.

In practice, the dual formulation of the optimization problem is considered [33]:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i, \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, \quad i = 1,\ldots,N, \end{aligned} \tag{2}$$

where the kernel function k is defined as $k(\mathbf{x}',\mathbf{x}'') = \langle\phi(\mathbf{x}'),\phi(\mathbf{x}'')\rangle_{\mathcal{H}}$. The dual formulation has the advantage of not requiring any explicit computation in the high dimensional space $\mathcal{H}$. Since it can be shown that the dual problem is convex, it can be solved by using standard quadratic programming algorithms. Moreover, the structure of the problem encourages a sparse solution: only a (usually small) fraction of training vectors corresponds to non null multipliers α_i. These vectors are called *Support Vectors*, and determine the classifier which is defined as the sign of the score function s:

$$s(\mathbf{x}) = b + \sum_{i=1}^{N}\alpha_i y_i k(\mathbf{x},\mathbf{x}_i). \tag{3}$$

The optimal value of b can be determined by exploiting the complementarity conditions: given any j for which $0 < \alpha_j < C$, we have that $b = y_j - \sum_{i=1}^{N}\alpha_i y_i k(\mathbf{x}_j,\mathbf{x}_i)$.

In this work, we trained multiple SVMs with the widely used Gaussian kernel function $k(\mathbf{x}',\mathbf{x}'') = \exp(-\gamma\|\mathbf{x}' - \mathbf{x}''\|^2)$. Therefore, for each SVM there are two parameters that need to be tuned, the penalization parameter C and the scale of

Table 1
Values of the parameters C and γ used to train the 56 SVMs. The values have been selected by cross validation separately for combination of classes and low-level features (Bag of features (BoF), color histogram in the RGB color space (CH), color moments in the YUV color space (CM), and edge direction histograms (EDH)).

Class	BoF		CH		CM		EDH	
	C	γ	C	γ	C	γ	C	γ
Animals	2	512	128	0.5	32	0.5	2	0.125
City	8	32	8	8	512	0.5	2	0.125
Closeup	8	32	32	2	32	0.125	2	0.125
Desert	32	2	8	8	8	0.125	0.5	0.125
Flowers	2	128	512	0.5	8	0.125	2	0.125
Forest	2	512	32	8	2	0.125	8	0.5
Indoor	128	2	8	8	2	0.125	2	0.125
Mountain	8	128	128	0.5	2	0.125	0.5	0.125
Night	8	32	8	8	2	0.125	8	0.125
People	2	128	8	8	2	0.125	2	0.125
Rural	8	128	2	8	2	0.125	2	0.125
Sea	2	128	8	8	2	0.125	2	0.125
Street	2	128	32	2	2	0.125	8	0.5
Sunset	8	32	32	2	8	0.125	2	0.125

the Gaussian kernel γ, for which we empirically considered a set of candidate values: $C \in \{0.5, 2, 8, 32, 128, 512, 2048\}$, $\gamma \in \{0.03125, 0.125, 0.5, 2, 8, 32, 128, 512, 2048\}$. For each combination of C and γ we estimated the error rate of the classifier by cross-validation: the training set of positive and negative examples is randomly partitioned into five subsets of equal size; then five SVMs are trained, each one on a different choice of four of the five subsets; finally, each SVM is evaluated on the fifth subset and the overall error rate is estimated as the average of the five misclassification rates. The pair of values C, γ corresponding to the lowest error is chosen and used to train the final SVM on the whole training set (see [60], for instance, for an alternative method for selecting the parameters of the SVMs). Table 1 shows the value of the parameters selected for the 56 SVMs.

The classification performance (see Table 2) varies greatly depending on classes and features, ranging from 6.6% of misclassifications for the "night" class using color moments, to a 30% for the class "animals" using the color histogram. There is no clearly superior feature and each feature obtained the lowest classification error for at least one class. Moreover, the difference in classification error between the best feature and the second best (last column of Table 2) shows that there is a considerable redundancy between the features: the second best is always less than 4% away from the best feature. This leaves open the possibility of operating some form of dimensionality reduction in the feature space. However, at the present time, we have made no attempt in this sense.

Better results can probably be obtained by combining the four scores for each class. However, our goal is not to achieve low misclassification rates, but rather to use the classifiers to transform the high-dimensional feature space into a low-dimensional semantic space without losing valuable information about the visual content of the images.

At the end of the training, we have a distinct SVM for each feature and for each class. Given a new image Q, represented by the feature vector $\mathbf{x}_Q^{(f)}$, the SVM provides a score $s^{(c,f)}$:

Table 2
Percentage of classification errors of the classifiers on the 14 classes, using the four low-level features considered (Bag of features (BoF), color histogram in the RGB color space (CH), color moments in the YUV color space (CM), and edge direction histograms (EDH)). The errors have been estimated by a fivefold cross validation on the training sets. For each class, the best result is reported in bold. The last column reports the difference in classification error between the best feature and the second best.

Class	BoF	CH	CM	EDH	ΔE
Animals	**22.5**	30.0	22.9	25.5	0.4
City	**10.1**	20.6	17.1	12.5	2.4
Closeup	17.7	27.3	17.2	**15.0**	2.2
Desert	18.7	15.7	**14.1**	22.0	1.6
Flowers	12.8	**12.0**	12.6	13.3	0.6
Forest	**7.0**	13.6	9.8	9.4	2.4
Indoor	14.7	18.5	18.3	**12.9**	1.8
Mountain	14.1	16.8	**13.7**	20.3	0.4
Night	13.5	8.3	**6.6**	27.5	1.7
People	**17.0**	23.8	20.2	20.5	3.2
Rural	18.5	15.7	**12.2**	22.6	3.5
Sea	23.1	21.9	19.4	**16.7**	2.7
Street	18.6	24.5	18.8	**17.4**	1.4
Sunset	12.5	8.4	**6.6**	16.3	1.8
Average	15.8	18.4	**15.0**	18.0	0.8

$$s^{(c,f)}\left(\mathbf{x}_Q^{(f)}\right) = b^{(c,f)} + \sum_{I \in T^{(c)}} \alpha_I^{(c,f)} y_I^{(c)} \exp\left(-\gamma^{(c,f)} \left\| \mathbf{x}_I^{(f)} - \mathbf{x}_Q^{(f)} \right\|^2\right), \tag{4}$$

where $T^{(c)}$ is the training set for class c, $\mathbf{x}_I^{(f)}$ denotes the feature vectors computed on the image I, $y_I^{(c)}$ is the label in $\{-1,+1\}$ which indicates whether I is a positive or a negative example, $b^{(c,f)}$ and $\alpha_I^{(c,f)}$ are the parameters determined by the training procedure, and $\gamma^{(c,f)}$ is the scale parameter of the kernel. The score is expected to be positive when the image belongs to the class c, and negative otherwise. The higher the score, the more likely is it that the image belongs to the class [38]. Packing together the 56 scores we obtain a compact vector of prosemantic features.

Summing up, an image is described by prosemantic components, each of which is based on a single low-level feature, and is a weighted sum of similarities (given by the kernel function) with respect to a group of representative examples of a class (the support vectors). Each support vector can then be considered as a prototype of the target class (or of its opposite) from the point of view of a given low-level feature. Prosemantic feature extraction works as a non-linear dimensionality reduction scheme where a large low-level representation (more than two thousand components) is shrunk to a vector of 56 components.

4. Image retrieval by relevance feedback

Once one has defined the prosemantic feature vector, one has only to endow the corresponding vector space with a suitable metric in order to do similarity searches. In practice, this highly complex conceptual operation that could conceivably result in the creation of an arbitrary Riemann metric reduces to the selection of the parameters of a suitable Minkowski metric L_k, that is, to the choice of a suitable k and of suitable coefficients that weight the contribution of the different axes. Rather than *a priori* definition of such coefficients, we prefer to have them adapted on a case-by-case basis using relevance feedback. The relevance feedback method we used is derived from that used by the QuickLook^2 CBIR system [10]. The method was designed with low-level image descriptors in mind, but can be extended to prosemantic features as well, since the only assumption it makes is that the feature is a vector whose components can be compared and combined. It is composed of two main steps: a reweighting scheme, which modifies the metric used in the retrieval process, and a query refinement mechanism, which defines a representation of the user's needs.

Let $\mathbf{x}_I$ be the representation of the image I. Images can be described by different features so $\mathbf{x}_I$ is composed of different numerical vectors, each one representing an image characteristic (e.g. color histogram, shape, etc.). We indicate these vectors for image I as $\mathbf{x}_I^{(1)}, \mathbf{x}_I^{(2)}, \ldots, \mathbf{x}_I^{(p)}$. Given a query Q and a image I, the dissimilarity between the two representations is computed as:

$$D(Q,I) = \frac{1}{p} \sum_{f=1}^{p} D^{(f)}\left(\mathbf{x}_Q^{(f)}, \mathbf{x}_I^{(f)}\right) w^{(f)}, \tag{5}$$

where $D^{(f)}$ and $w^{(f)}$ are the dissimilarity metric and the weight associated to the feature f respectively. The weights $w^{(f)}$ determine the importance of each feature in the overall dissimilarity measure depending on the positive (i.e. relevant) and negative (i.e. not-relevant) images selected by the user. The query Q is computed by the query refinement algorithm from the positive images only. The dissimilarities are computed between the query and each image in the database. The images most similar to the query are presented to the user ordered by increasing dissimilarity. The underlying idea for the use of a weighted dissimilarity measure is that if the positive images share the same features, then these features are relevant and thus they must weigh more in the dissimilarity measure (i.e. any deviation from these features must be emphasized). On the contrary, features having different values among the selected images will receive lower weights since they are not distinctive for the query.

4.1. Reweighting scheme

The basic idea of the relevance feedback mechanism is that the distribution, in the feature space, of the images that the user has judged relevant (or not relevant) can be used to determine what features the user has taken into account (and to what extent) in formulating this judgment. With this information, one can accentuate the influence of the relevant features in the overall evaluation of image dissimilarity, as well as in the formulation of a new query. The structure of the relevance feedback mechanism is entirely description-independent, that is, the index can be modified, or extended to include other features without requiring any change in the algorithm as long as the features can be expressed as numerical vectors. The relevance feedback algorithm works as follows: let R_+ the set of relevant images and R_- the set of non relevant images. The feature weights are computed as:

$$w^{(f)} = \begin{cases} \frac{1}{\epsilon}, & \text{if } \|R_+\| < 3, \\ \frac{1}{\epsilon + \mu_+^{(f)}}, & \text{if } \|R_+\| \geqslant 3 \quad \text{and} \quad \|R_-\| = 0, \\ \frac{1}{\epsilon + \mu_+^{(f)}} - \eta \frac{1}{\epsilon + \mu_*^{(f)}}, & \text{otherwise}, \end{cases} \tag{6}$$

where $\mu_{+}^{(f)}$ is the average of the dissimilarities computed on the *f*th feature between each pair of images in R_{+}, $\mu_{*}^{(f)}$ the average of the dissimilarities computed on the *f*th feature between each image in R_{+} and each image in R_{-}. ϵ is a small positive constant that guarantees that the denominators in (6) are never zero ($\epsilon = 10^{-5}$ in the current implementation), η is also a positive constant and is used to adjust the contribution of the negative examples in the computation of the weights (we used the default system's value of 0.6 which experimentally has been found to provide good results on a variety of image databases). Negative weights are set to 0. A weight is large if the corresponding feature is present in all the relevant images, while it is small or dampened if the corresponding feature assumes a broad range of values within the relevant images or if is also present in the non relevant images (viz. it is present in the relevant images but it is not relevant).

4.2. Query refinement

In content-based retrieval images are sometimes considered relevant because they resemble the query image in just some limited sense related to low-level features that are particularly prominent, even if semantically not very significant. Consequently, after an initial query, a given image may be selected by the user as relevant because it has one of the characteristics of the query (e.g. the same color), and another selected for another characteristic (e.g. the shape), although the two are actually quite different from each other. To cope with this problem a method called *query refinement* is used to compute the query vector. On the basis of the images selected by the user, the system formulates a new query that better represents the images of interest to the user, taking into account the features of the relevant images, without allowing any particular feature value to bias the query computation. Let $\mathbf{x}_I^{(f)}(k)$ be the *k*th value of the *f*th feature of image *I*. By considering only the images in the relevant set R_{+}, the query *Q* is computed as:

$$Y_k^{(f)} = \left\{ \mathbf{x}_I^{(f)}(k) : \left| \mathbf{x}_I^{(f)}(k) - \mathbf{x}_{\overline{Q}}^{(f)}(k) \right| \leqslant 3\sigma_k^{(f)} \right\}, \tag{7}$$

$$\mathbf{x}_Q^{(f)}(k) = \frac{1}{\left\| Y_k^{(f)} \right\|} \sum_{\mathbf{x}_I^{(f)}(k) \in Y_k^{(f)}} \mathbf{x}_I^{(f)}(k), \tag{8}$$

where $\overline{Q}$ is the average query and $\sigma_k^{(f)}$ is the standard deviation of the *k*th values in the *f*th feature. The query is thus computed from the feature values that agree with the user selection, while the outliers are removed from the computation.

5. Evaluation

A user study has been conducted to evaluate the performance of our prosemantic features against low-level features. We organized two sessions of experiments. First prosemantic features were compared to the underlying low-level features described in Section 3 (in the following we will call these "pre-classification features"). In the second, we used a set of state-of-the-art low-level features specifically designed for image retrieval. These features were first tested alone and then in combination with prosemantic features.

All the experiments were based on the *target search* approach using relevance feedback. In a target search experiment the subject is asked to search for a specific image, and the experiment terminates when that image is found. We decided to employ target search because, as already conjectured by Cox et al. [14], it might provide more reliable statistical measures than, for instance, category search. This allows us to obtain a fairly precise indication of the performance of the methods even with a limited number of subjects.

For each of the two sessions of experiments we asked 20 subjects to perform ten searches. The subjects participated in the study singularly in the same environment and with the same instructor. Each subject was constrained to retrieve the target image by selecting any number of relevant and not relevant images within the top 60 retrieved images. They were also instructed that they must retrieve the target image in a maximum of 20 operations without a time limit.

Each session of experiments entailed the comparison between two sets of features. In order to minimize user adaptation, the retrieval sessions were conducted alternating the two kinds of features considered. Moreover, each user searched the ten query images in a different order. The subjects did not know what kind of features they were using at any time, but they knew that they were using two different sets. The retrieval sessions were organized in such a way that at the end of the session of experiments, each target image was searched the same number of times by using the two sets of features. Before starting each session, the users were instructed in the use of the system by performing a guided retrieval test.

5.1. The dataset

The choice of a database on which a certain solution is to be evaluated is an extremely delicate one. The use of relevance feedback entails that the database should have the following characteristics:

(i) there should be representatives of well defined classes of images, belonging to certain "semantic" categories (if the data base is largely composed of ambiguous images, then the interpretation of the results will also be ambiguous, and will not give a clear indication of the performance of the methods under evaluation);

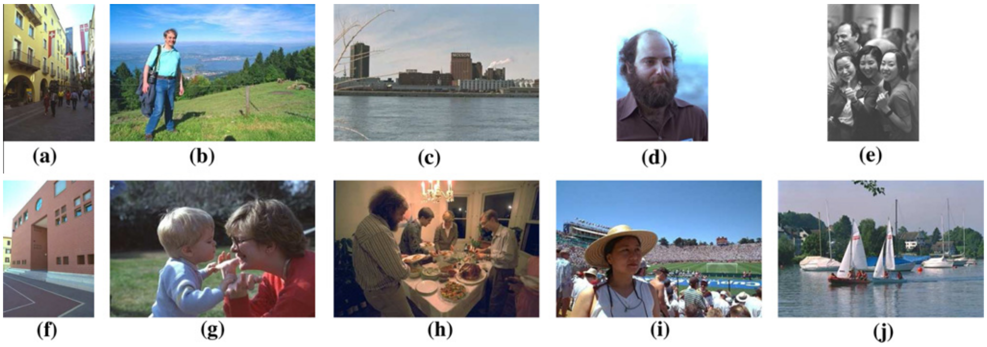

Fig. 2. The ten images used in the target search retrieval sessions.

(ii) there should be no meaningful classes definable in terms of low-level visual features only (e.g. a class "fruits" in which all fruits are red, or a class "buildings" in which all images have blue sky in the upper part);
(iii) there should be no isolated images in the feature space: it should be possible to arrive to every image in the database starting from any other image and giving only "small" jumps;
(iv) at the same time, there should be no large clusters of very similar images.

The last two points are important in light of the experimental method that we are using. It is worth remembering that here we are evaluating the features, not the relevance feedback, and that we must select a database in which all components (with the exception of the features) will work uniformly well. Relevance feedback works by crawling through the feature space based on the indications of the user. It does not typically allow big jumps across the feature space. Requirement (iii) is meant to guarantee that images will not be unreachable because of the moving limitations of relevance feedback. Another problem, at the origin of requirement (iv), is that relevance feedback can get stuck in what we call *feature swamps*. The only information that we can give to the system using relevance feedback derives from the difference between the images that we see displayed: there must be enough variety so that by saying "this image is good, this one is not" we give enough information to allow the system to move briskly in the correct direction. If all the images displayed come from the same region of the feature space, the system movement will be sluggish and haphazard, requiring many iterations to get out of the swamp.

Falling into a swamp is a more or less unpredictable occurrence, and independent of the quality of the features, so the presence of swamps would cause an undesired uncontrolled parameter that could lead to meaningless results.

All these considerations make large, erratically collected databases ill-suited to serious evaluation so, to cut the Gordian knot, we set to create our own. However, rather than starting from scratch, we created our data base as a subset of a collection that already exhibited many desirable characteristics, including the very important ones of not being privately owned and of being freely available: the Benchatlon database [22].

The dataset is composed of typical consumer photographs depicting a very wide range of situations. Compared to other widely used datasets (e.g. Corel[3] and imageCLEF [32]) it is, in our opinion, a challenging dataset for the image retrieval task. In particular, the dataset is composed of uncategorized photographs presenting a wide variability in terms of content, composition and illumination conditions. For instance, the scenes depicted vary from very strong illuminated outdoor shots to very dark indoor ones; the field of view of the shots ranges from panoramas to close-ups. Being a dataset of consumer photographs, the quality of the images is variable including out of focus shots, cluttered scenes, gray scale images, and images with strong color casts. The dataset presents a few large clusters of images of the same event (e.g. a picnic, a birthday party, etc.). Moreover, the dataset shows a very different distribution of concepts with respect to the dataset used to train the classifiers. For instance, very often the image would fall in the "people" class, while very few images can be considered as belonging to the "desert" or "flowers" classes.

We created our data set using 1875 images taken from the whole data base. This subsampling resulted in a reduced density of the images in the feature space, making it much harder for the relevance feedback to navigate the space. In other words, the reduction in the size of the data base was an experimental strategy to increase the discriminative power of the experiments. The size was a compromise in that reducing it further we would have risked losing categories for lack of significant representatives. This is an important point that we cannot stress strongly enough, since it goes against a certain (misled) common wisdom: the relatively small size of the test data base is not a liability but an asset of the experimental procedure, since it allowed the creation of a controlled environment in which the differences between the methods subject to experimentation are more readily evidenced.

The ten target images have been randomly selected and are shown in Fig. 2. Other 60 images have been randomly selected to compose the page from which the users started all their searches. These images are shown in Fig. 3.

[3] http://www.corel.com/.

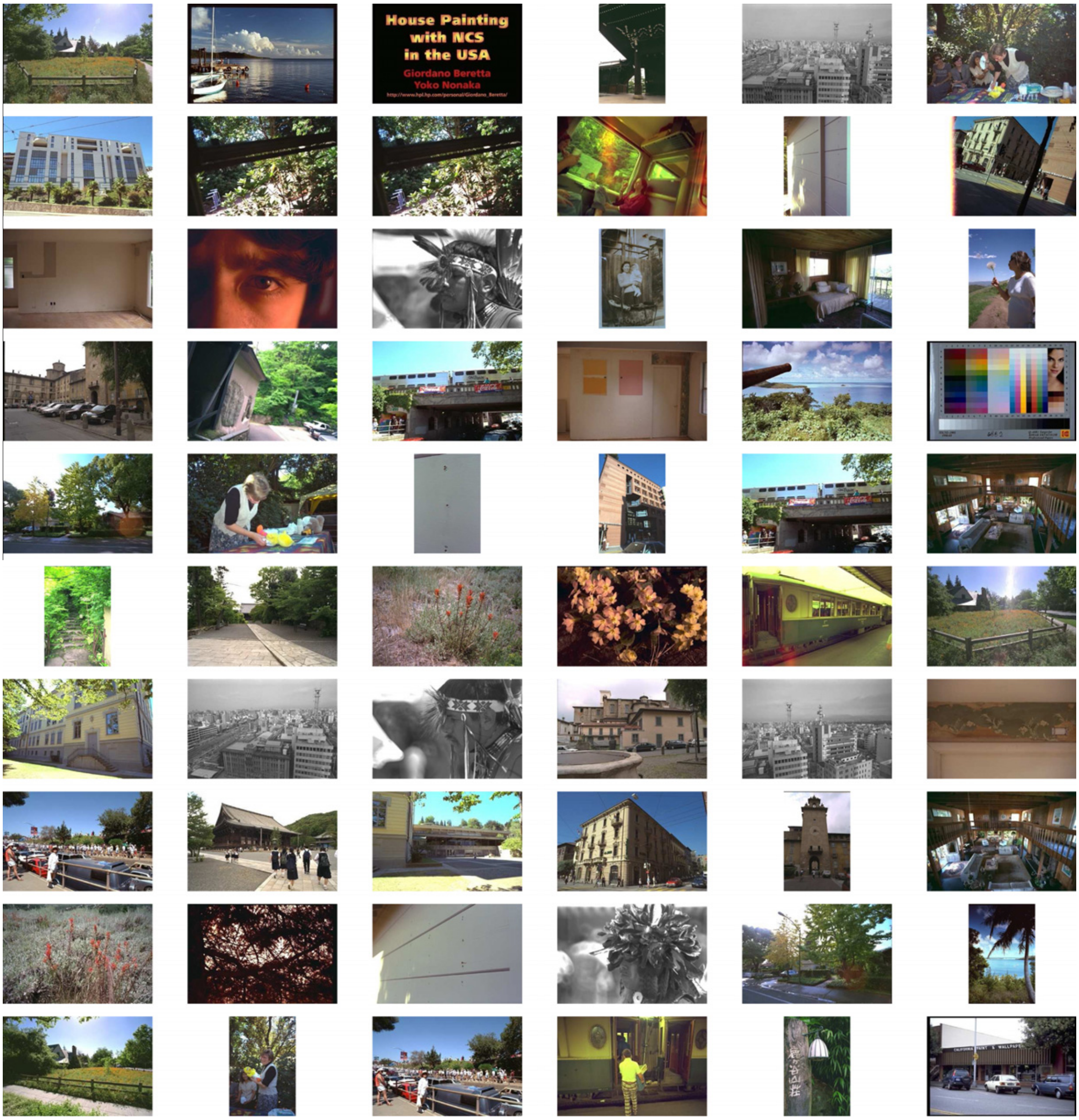

Fig. 3. The 60 images which compose the starting page of the searches.

5.2. First experiment: prosemantic vs. pre-classification features

The first experiment was aimed to determine if, and to what extent, prosemantic features are more suited for image retrieval than the low-level features they are built upon (pre-classification features). In fact, prosemantic features can be considered as a non-linear dimensionality reduction scheme (from more than two thousands to 56 components) that takes pictorial features and maps them into a space where semantic relationships are easier to establish. Prosemantic features do not add any information to pre-classification features, with the exception of the information derived from the training sets and encoded in the classifiers. It should be noted that the pre-classification features were not designed for image retrieval. In fact, they have been tuned to obtain good classification performance.

Twenty users took part in this experiment. All subjects came from the computer science department of the University of Milan-Bicocca: four of them have a background on image processing or computer vision (two Ph.D. students and two post-doctoral fellows), the other 16 are graduate (three) or undergraduate (13) students.

Table 3
Detail of the results obtained on the ten query images using the pre-classification and prosemantic features. For each query image (see Fig. 2) we report the number of successful searches (over 10 attempts for each feature set), the number of iterations needed to retrieve the image (averaged over the successful searches), and the corresponding standard deviation.

Query image	Pre-classification			Prosemantic		
	Succ.	Avg	Std dev	Succ.	Avg	Std dev
a	8	9.75	5.49	10	6.80	4.21
b	5	4.20	4.35	9	4.00	2.11
c	6	3.67	5.09	9	1.11	0.31
d	0	–	–	9	3.33	1.70
e	7	1.29	0.45	10	3.80	3.16
f	0	–	–	10	1.30	0.64
g	9	7.78	4.39	7	8.00	5.63
h	7	5.29	4.40	9	8.11	4.56
i	6	7.50	5.41	10	1.10	0.30
j	3	9.00	5.10	10	1.80	1.60
Total	51	6.06	5.32	93	3.80	3.89

Since each subject searched the ten target images, we have 200 searches: 100 performed using pre-classification features and 100 with prosemantic features. The outcome of the 200 searches clearly demonstrates the effectiveness of prosemantic features with respect to pre-classification features. Table 3 shows the detail of the results obtained. On nine cases out of ten, the use of prosemantic features obtained a higher success rate. The only exception is query (g) which has been quite difficult to find with both the features considered. Two images have never been found using pre-classification features (d and f), while they have been considered among the easiest to find using prosemantic features. There are two cases (queries e and h) which present clearly distinguishable visual characteristics (one is a gray scale image, the other presents a strong color cast). This fact has been recognized by the majority of users which exploited it to quickly find the targets using pre-classification features; however, the few users who have not been able to master how low-level similarity works failed the retrieval task. In these two cases retrieval with prosemantic features required (on average) a higher number of iterations, but with only one failure.

Using the prosemantic features, only seven times were the users not able to retrieve the target images within the limit of 20 retrieval operations. By contrast the limit has been exceeded 49 times in the case of pre-classification features. On the basis of these results we can conclude that prosemantic features provide for a large and consistent boost in retrieval performance with respect to the underlying low-level features.

5.3. Second experiment: prosemantic vs. QuickLook^2 features

For the second experiment we considered the features used by default by the QuickLook^2 image retrieval system. These are low-level features which have been specifically designed for image retrieval: we consider them as representative of the state of the art in image description for this specific task. The aim of this experiment is to verify whether or not prosemantic features allows for a more efficient image retrieval with respect to traditional pictorial features. In this experiment we also considered the combination of prosemantic features with the features of the QuickLook^2 system ("combined features," in the following). Since prosemantic features are based on quite different features than the QuickLook^2 ones, we expect that combined features will provide a more effective description of the images.

The pictorial features in the QuickLook^2 system can be broadly divided into three categories: color features, structural/textual features, and composite features. The last category comprises those features that describe different aspects of the image. The color features include the color histogram, the color coherence vector, color distribution and, color transitions. The histogram of edge's directions, the wavelet statistics, and the edge gradient histogram belong in the structural/textual

Table 4
Low-level pictorial features used in the QuickLook^2 CBIR system.

Feature	Description	Size	Ref.
Color histogram	Color quantized histogram	64	[11]
Color coherence vector	Coherent/non coherent color regions	128	[37]
Color distribution	Moments of the color channel distributions	9	[45]
Color transitions	Color quantized pair occurrences	66	[11]
Wavelet statistics	Energy band statistics	20	[42]
Edge gradient histogram	Histogram of quantized gradient magnitude	2	[11]
Histogram of edge directions	Canny's edge directions	30	[11]
Color regions composition	Spatial distribution of color regions	5	[11]
Spatial chromatic histogram	Color histogram with spatial information	11	[8]

Table 5
Detail of the results obtained on the ten query images using the features in the QuickLook^2 retrieval system and their combination with prosemantic features. To make the comparison easier, the results obtained with the prosemantic features have been repeated. For each query image (see Fig. 2) we report the number of successful searches (over 10 attempts for each feature set), the number of iterations needed to retrieve the image (averaged over the successful searches), and the corresponding standard deviation.

Query image	QuickLook^2			Prosemantic			Combined		
	Succ.	Avg	S. dev	Succ.	Avg	S. dev	Succ.	Avg	S. dev
a	10	5.50	3.50	10	6.80	4.21	10	4.90	4.95
b	9	3.44	3.24	9	4.00	2.11	10	6.30	6.94
c	3	3.67	1.70	9	1.11	0.31	10	1.70	0.90
d	5	4.60	1.85	9	3.33	1.70	10	3.90	2.62
e	10	2.70	1.90	10	3.80	3.16	10	4.20	2.60
f	0	–	–	10	1.30	0.64	10	2.10	1.22
g	6	3.83	4.22	7	8.00	5.63	10	6.90	3.83
h	10	5.20	3.79	9	8.11	4.56	10	4.60	5.16
i	9	2.56	1.17	10	1.10	0.30	9	1.22	0.42
j	10	4.60	5.10	10	1.80	1.60	10	1.30	0.46
Total	72	4.04	3.50	93	3.80	3.89	99	3.74	4.12

category. The composite feature category comprises the color regions composition and the spatial chromatic histogram. Table 4 summarizes the features used in the QuickLook^2 system.

Again, 20 users took part in the experiment: three Ph.D. students, two post-doctoral fellows, one researcher, one graduate student, and 13 undergraduate students. Among these, six also took part in the first experiment. However, since a period of more than four months passed between the two sessions of experiments, we consider all the twenty users as first-time users.

The outcome of the 200 searches is reported in Table 5. The results demonstrate that the combination of prosemantic and QuickLook^2 features clearly outperforms the use of QuickLook^2 features only. More in detail, only in one case a user was not able to retrieve the target image (i) within the limit of 20 iterations. For QuickLook^2 features, failures occurred 28 times. In particular, one target image (f) has never been found using only the features of QuickLook^2. Moreover, images have been found with fewer iterations, on average, using the combination of the two features.

5.4. Discussion

Summing up the results of the two sessions of experiments, we can conclude that: prosemantic features perform significantly better than the two sets of low-level features considered; the combination with low-level features slightly increases the performance of prosemantic features; as expected, the features included in QuickLook^2 performed significantly better than pre-classification features. Fig. 4 shows the cumulative success rate for the four sets of features as a function of the number of iterations. As a baseline, the performance corresponding to a random browsing of the database (where for each iteration 60 new images are shown to the user) is also reported. The plot shows how prosemantic features, alone or combined with low-level features, allows the retrieval of more target images and with less iterations. In particular, in the case of prosemantic features in more than one third (35/100) of the cases the retrieval of the target image required only one iteration (i.e. without really exploiting the relevance feedback algorithm). This happened only in 11 cases for pre-classification

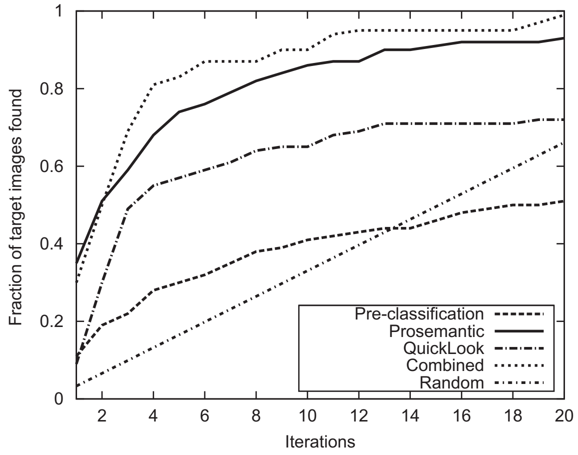

Fig. 4. Fraction of images successfully retrieved as a function of the number of iterations.

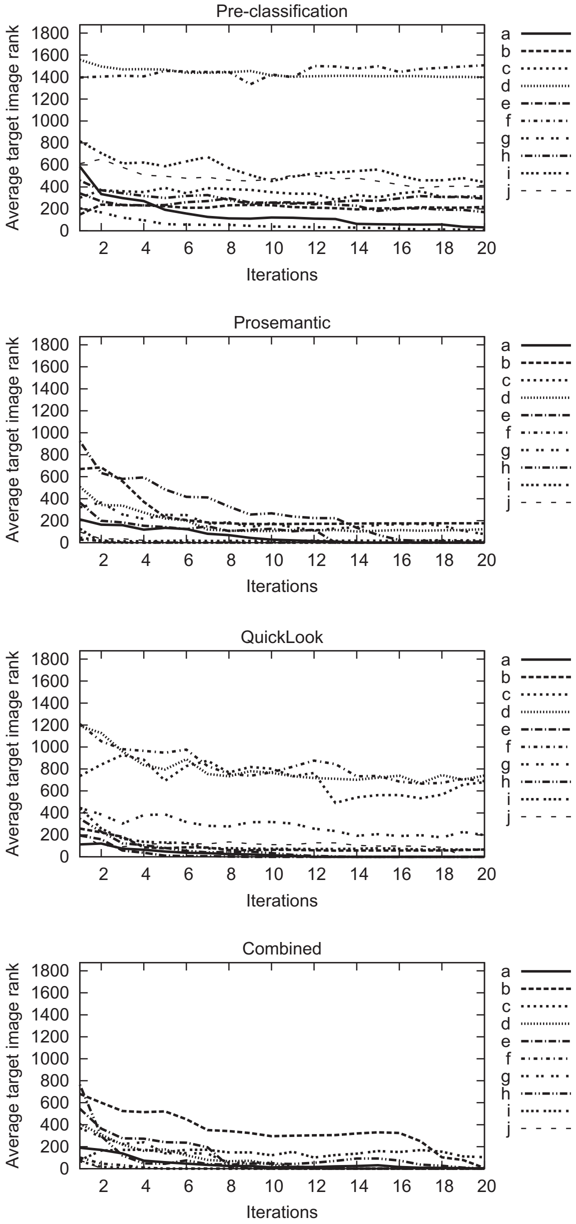

Fig. 5. Average rank of the target image as a function of the number of iterations. Each subfigure reports the results obtained with a different feature set on the ten target images. When an image is found, its rank is considered zero for all the remaining iterations.

features and in 9 cases for the features of QuickLook^2. Retrieval in a single iteration occurred 30 times with the combined features.

When the target image is hard to retrieve, random browsing outperforms pre-classification features and approaches the performance of QuickLook^2 features. This happens when the search falls into a feature swamp where the user is not able to make progress toward the target image. For pre-classification and QuickLook^2 features this happened regularly for some target images such as query (f). Fig. 5 shows the average rank assigned to each target image as a function of the number of iterations performed. With pre-classification features we observe that, for almost all the target images, some users failed to converge. For the prosemantic features, instead, convergence occurred in the majority of cases. The QuickLook^2 features produced an unstable behavior: some target images are found very quickly, but the others completely fail to converge. Using the combined features convergence is always very quick, with the possible exception of query (b).

Observing the users and discussing with them after the experiments, we made the hypothesis that the effectiveness of the prosemantic features derives from their capability of encoding characteristics of the images which allow a better match

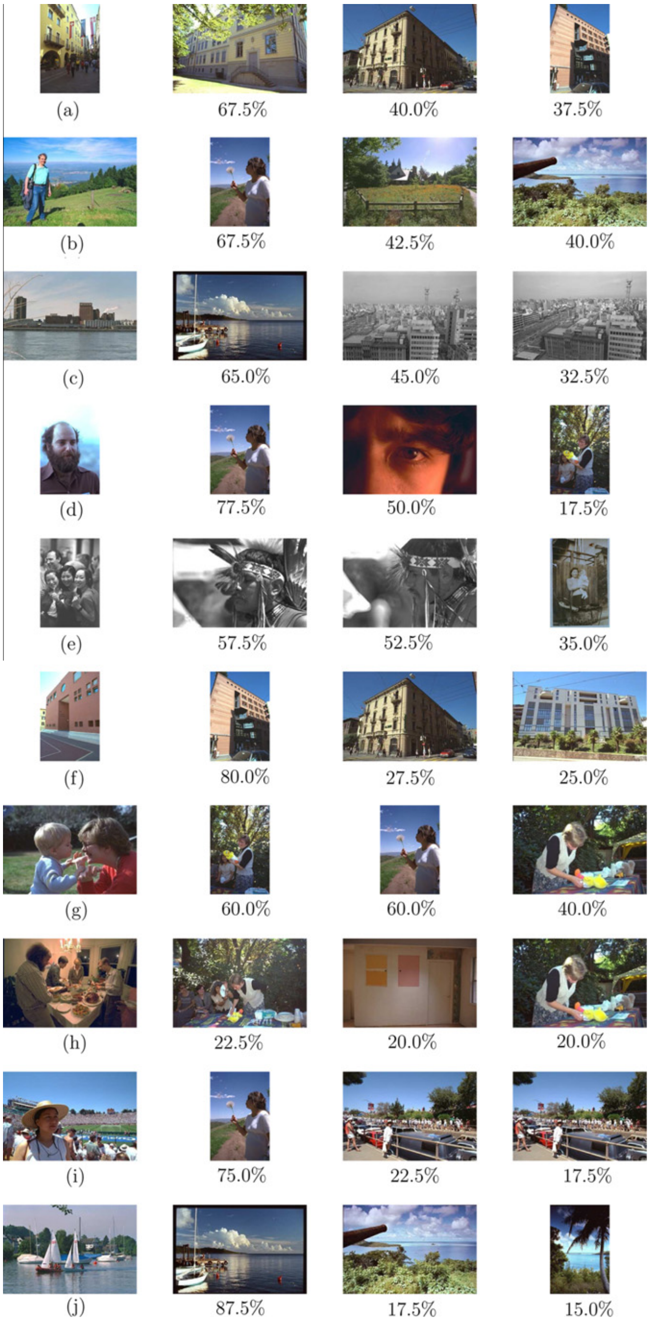

Fig. 6. The most frequent positive examples chosen by the users at the first iteration. For each target image (first column) we report the three most frequently chosen examples (second to fourth columns). Under each example is the percentage of times that the image has been chosen.

against users' intuition about the similarity of the images. In fact, the users often started by selecting pictures with the same "general theme" of the target image (e.g. pictures of people, city shots...). Only a few subjects (among the ones with pre-existing skills in image processing) based their reasoning on pictorial properties.

To understand what similarity criteria the users adopted during the searches we analyzed their first selection of positive examples for each target image. We observed similar choices for all the four sets of features considered, therefore we report the results aggregated. Fig. 6 shows the three most frequent positive examples chosen by the users at the first iteration among the 60 images which compose the starting page (see Fig. 3). In almost all the cases, the users selected positive examples with a similar semantic content with respect to the target images. A notable example is that half of the times they selected a detail of a face to search target image (d), even if the two images show completely different pictorial properties. In several cases the users selected images containing the objects depicted in the target image. For instance, 87.5% of times they selected the image of a boat to search the target image (j). The starting page does not contain images similar (from a semantic point of view) to target (h). In this case there is not a consensus among the examples picked by the users, but we note how some of them selected an indoor image, and pictures of people near a table. Pictorial properties were not completely disregarded, it seems that they were considered only for images with a compatible semantic. For instance, query (a) has been searched by selecting yellow buildings, and target image (e) by selecting gray level pictures of faces.

6. Conclusions

We have presented here an approach to CBIR based on the information encoded in prosemantic features. These are computed on the basis of a set of low-level features that are fed to a battery of image classifiers trained to evaluate the membership of the images with respect to a set of 14 overlapping classes. The output of the classifiers is used to index the images that are searched using relevance feedback.

To verify the effectiveness of the approach we designed a target search experimentation where prosemantic features are compared against two sets of low-level features. The combination of prosemantic and low-level features has been also evaluated.

On the basis of the experimental results, we can conclude that prosemantic features perform significantly better than the two sets of low-level features considered. Moreover, the combination with low-level features slightly increases the performance of prosemantic features.

Since prosemantic features were shown to be very effective in the target search task, we are considering to evaluate them in other retrieval scenarios such as category search and browsing. At the same time, we will verify the scalability of the approach by performing new tests on larger datasets. We are also considering the use of prosemantic features for other imaging applications such as automatic image annotation and classification. For these tasks we cannot rely on the relevance feedback mechanism to select and to weight the components. Therefore, we are currently investigating other strategies for the analysis of the prosemantic feature space, with particular focus on its metric properties. We are also investigating how to assess the relevance of individual prosemantic components and their correlation. We believe that this activity would result in useful insights on how the classes and the low-level features could be chosen to improve image description by prosemantic features. In particular, it would be useful to assess the possible benefits of an extension of the approach with additional classes.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D.M. Blei, M.I. Jordan, Matching words and pictures, The Journal of Machine Learning Research 3 (2003) 1107–1135.

[2] A. Bosch, A. Zisserman, X. Muñoz, Scene classification via pLSA, in: Computer Vision – ECCV 2006, vol. 3954, 2006, pp. 517–530.

[3] D. Cantone, A. Ferro, A. Pulvirenti, D.R. Recupero, D. Shasha, Antipole tree indexing to support range search and *k*-nearest neighbor search in metric spaces, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 535–550.

[4] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 394–410.

[5] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001, Software available at <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

[6] E. Chang, G. Kingshy, G. Sychay, W. Gang, CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines, IEEE Transactions on Circuits and Systems for Video Technology 13 (1) (2003) 26–38.

[7] X. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, Journal of Machine Learning Research 5 (2004) 913–939.

[8] L. Cinque, G. Ciocca, S. Levialdi, A. Pellicanò, R. Schettini, Color-based image retrieval using spatial-chromatic histograms, Image and Vision Computing 19 (13) (2001) 979–986.

[9] G. Ciocca, C. Cusano, R. Schettini, Semantic classification, low level features and relevance feedback for content-based image retrieval, Proceedings of Multimedia Content Access: Algorithms and Systems III 7255 (2009) 72550D–72559D.

[10] G. Ciocca, I. Gagliardi, R. Schettini, Quicklook^2: an integrated multimedia system, Journal of Visual Languages & Computing 12 (1) (2001) 81–103.

[11] G. Ciocca, R. Schettini, The quicklook image search engine, Journal of Image and Graphics (2000) 645–648.

[12] Gianluigi Ciocca, Claudio Cusano, Simone Santini, Raimondo Schettini, Prosemantic features for content-based image retrieval, in: Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User, Lecture Notes in Computer Science, Vol. 6535, 2011, pp. 87–100.

[13] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[14] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, P.N. Yianilos, The bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments, IEEE Transactions on Image Processing 9 (1) (2000) 20–37.

[15] C. Cusano, S. Santini, R. Schettini, On the Coöperative Creation of Multimedia Meaning, in: Proceedings of the Fourth International Conference on Semantic and Digital Media Technologies: Semantic Multimedia, 2009, pp. 28–39.

[16] R. Datta, J. Li, J.Z. Wang, Content-based image retrieval: approaches and trends of the new age, in: Proceedings of the Seventh ACM SIGMM international workshop on Multimedia information retrieval, 2009, pp. 253–262.

[17] G.M. Davies, J.W. Shepherd, H.D. Ellis, Similarity effects in face recognition, American Journal of Psychology 92 (3) (1979) 507–523.

[18] S. Edelman, Representation, Similarity, and the Chorus of Prototypes, Technical report, The Weizman Institute of Science, Rehovot, Israel, 1993.

[19] J. Fan, Y. Gao, H. Luo, G. Xu, Automatic image annotation by using concept-sensitive salient objects for image content representation, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004, pp. 361–368.

[20] B. Furht, Handbook on Multimedia Computing, CRC Press, Inc., 1998. Chapter: Content-based image indexing and retrieval.

[21] K. Grauman, T. Darrell. The pyramid match kernel: discriminative classification with sets of image features, in: Proceedings of the 10th IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1458–1465.

[22] N.J. Gunther, G. Beretta, A Benchmark for Image Retrieval using Distributed Systems over the Internet: BIRDS-I, Technical Report HPL-2000-162, HP Labs, Palo Alto, 2001.

[23] A. Hauptmann, R. Yan, W.H. Lin, How many high-level concepts will fill the semantic gap in news video retrieval?, in: Proceedings of the Sixth ACM International Conference on Image and Video Retrieval, 2007, pp. 627–634.

[24] A. Hauptmann, R. Yan, W.H. Lin, M. Christel, H. Wactlar, Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news, IEEE Transactions on Multimedia 9 (5) (2007) 958–966.

[25] J. Jeon, V. Lavrenko, R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 119–126.

[26] C.S. Lee, W.-Y. Ma, H. Zhang, Information embedding based on user's relevance feedback for image retrieval, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series 3846 (1999) 294–304.

[27] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[28] J. Lu, S.P. Ma, M. Zhang, Automatic image annotation based on model space, in: Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005, pp. 455–460.

[29] B.J. MacLennan, The emergence of the symbolic processes from the subsymbolic substrate, in: International Symposium on New Information Processing Technologies '91, Tokyo, March 1991.

[30] K. Maruyama, K. Masame, M. Endo, F. Cheng, An analysis on the difference of processing mode between upright and inverted faces in their similarity judgment, Tohoku Psychologica Folia 47 (1–4) (1988) 85–94.

[31] C. Mason, E.R. Kandel, Central visual pathways, in: Eric R. Kandel, James H. Schwartz, Thomas M. Jessell (Eds.), Principles of Neural Science, Appleton & Lange, 1991, pp. 420–439. chapter 30.

[32] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, D.M. Squire, A framework for benchmarking in CBIR, Multimedia Tools Application 21 (1) (2003) 55–73.

[33] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (2) (2001) 181–201.

[34] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2161–2168.

[35] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.

[36] S. Park, T.F. Brady, M.R. Greene, A. Oliva, Disentangling scene content from its spatial boundary: complementary roles for the PPA and LOC in representing real-world scenes, Journal of Neuroscience 31 (4) (2011) 1333–1340.

[37] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: MULTIMEDIA '96: Proceedings of the Fourth ACM International Conference on Multimedia, 1996, pp. 65–73.

[38] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, 1999, pp. 61–74.

[39] N. Rasiwasia, P.J. Moreno, N. Vasconcelos, Bridging the gap: query by semantic example, IEEE Transactions on Multimedia 9 (5) (2007) 923–938.

[40] S. Santini, R. Jain, Similarity is a geometer, in: Borko Furht (Ed.), Multimedia Technologies and Applications for the 21st Century, Visions of World Experts, Kluwer Academic Publishers, 1997, pp. 185–214.

[41] R. Schettini, C. Brambilla, C. Cusano, G. Ciocca, Automatic classification of digital photographs based on decision forests, International Journal of Pattern Recognition and Artificial Intelligence 18 (5) (2004) 819–845.

[42] P. Scheunders, S. Livens, G. Van de Wouwer, P. Vautrot, D. Van Dyck, Wavelet-based texture analysis. International Journal of Computer Science and Information Management, Special issue on Image Processing, vol. 1, 1997.

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

[44] J.R. Smith, M. Naphade, A. Natsev, Multimedia semantic indexing using model vectors, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2003, pp. 445–448.

[45] M.A. Stricker, M. Orengo, Similarity of color images, in: Storage and Retrieval for Image and Video Databases III, vol. 2420, 1995, pp. 381–392.

[46] M. Szummer, R.W. Picard, Indoor-outdoor image classification, in: Proceedings of IEEE International Workshop on Content-Based Access of Image and Video Database, 1998, pp. 42–51.

[47] Y. Tang, P. Yan, Y. Yuan, X. Li, Single-image super-resolution via local learning, International Journal of Machine Learning and Cybernetics 2 (2011) 15–23.

[48] E.C.C. Tsang, D.S. Yeung, X.Z. Wang, OFFSS: optimal fuzzy-valued feature subset selection, IEEE Transactions on Fuzzy Systems 11 (2) (2003) 202–213.

[49] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, Hong-Jiang Zhang, Image classification for content-based indexing, IEEE Transactions on Image Processing 10 (1) (2001) 117–130.

[50] A. Vailaya, A. Jain, H.J. Zhang, On image classification: city images vs. landscapes, Pattern Recognition 31 (12) (1998) 1921–1935.

[51] N. Vasconcelos, From pixels to semantic spaces: advances in content-based image retrieval, Computer 40 (7) (2007) 20–26.

[52] A Vedaldi, SIFT++ a lightweight C++ implementation of SIFT, <http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html>.

[53] J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval, International Journal of Computer Vision 72 (2) (2007) 133–157.

[54] C. Wallraven, B. Caputo, A. Graf, Recognition with local features: the kernel recipe, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 257–264.

[55] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.

[56] X.Z. Wang, C.R. Dong, Improving generalization of fuzzy IF–THEN rules by maximizing fuzzy entropy, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 556–567.

[57] X.Y. Wei, C.W. Ngo, Y.G. Jiang, Selection of concept detectors for video search by ontology-enriched semantic spaces, IEEE Transactions on Multimedia 10 (6) (2008) 1085–1096.

[58] A. Yanagawa, S.F. Chang, L. Kennedy, W. Hsu, Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts, Technical Report 222-2006-8, Columbia University, 2007.

[59] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.

[60] S. Zhang, P. McCullagh, C. Nugent, H. Zheng, M. Baumgarten, Optimal model selection for posture recognition in home-based healthcare, International Journal of Machine Learning and Cybernetics 2 (2011) 1–14.

[61] X.S. Zhou, T.S. Huang, Relevance feedback in image retrieval: a comprehensive review, Multimedia Systems 8 (6) (2003) 536–544.