Multimed Tools Appl (2017) 76:13929–13951
DOI 10.1007/s11042-016-3788-1

Multi-index structure based on SIFT and color features for large scale image retrieval

Zied Elleuch[1,2] · **Kirmene Marzouki**[3,2]

Received: 31 October 2015 / Revised: 18 May 2016 / Accepted: 18 July 2016 /
Published online: 27 July 2016

Abstract During the past few years, the Bag-of-Words (BoW) model based on SIFT features has been one of the most adopted approaches by the Content-Based Image Retrieval (CBIR) systems. However, these CBIR systems have shown some weaknesses and shortcomings especially for large scale image collections. This is due to two main causes: First, information is lost in the quantization step and second, the SIFT features describe only the local gradient. To tackle these issues, we proposed to take advantage of the Hamming Embedding, soft assignment and multiple assignment techniques, on the one hand, and to fuse SIFT and color features at the indexing level in a multi-index structure, on the other. In fact, in this paper, generic and non-parametric image retrieval schemes as well as a novel multi-IDF design based on multi-index structure were proposed.

Extensive experiments were conducted on three public datasets (Holidays, Ukbench and MIR Flickr 1 M as distractor). The experimental results are promising and outperform the state-of-the-art CBIR systems. In addition, only 117 bits are needed to represent each key-point which enables us to make our image retrieval schema suitable for large-scale experiments.

Keywords Image retrieval · Multi-IDF · Multi-index · Multiple assignment · Soft-weighting

✉ Zied Elleuch
elleuch.zied@gmail.com

Kirmene Marzouki
kirmene@marzouki.tn

[1] Higher Institute of Applied Sciences and Technology of Gafsa, University of Gafsa, Gafsa, Tunisia

[2] Informatics for Industrial Systems Lab. National Institute of Applied Sciences and Technology of Tunis, LISI- INSAT, University of Carthage, Tunis, Tunisia

[3] Higher Institute of Applied Sciences and Technology of Sousse, University of Sousse, Sousse, Tunisia

1 Introduction

In recent years, the amount of multimedia data has dramatically increased with the advent of broadband Internet and digital TV. Image indexing and retrieval has been the subject of many research works for the computer vision community [13, 34, 40]. The major challenge lies in setting up an efficient system for large-scale image retrieval.

Several image retrieval systems have been proposed [13, 23, 24, 36, 38, 48]. The early ones are based only on textual tags and queries [33, 36]. However, these textual descriptions do not reflect objectively the wealth of the image content. In fact, these early systems have shown many limitations [10]. Therefore, since its appearance, the Content-Based Image Retrieval (CBIR) has been the topic of interest of the computer vision community.

Nowadays, several state-of-the-art CBIR systems are under study [13, 34, 37, 38, 40]. These systems focus on Bag-of-Words (BoW) model [40]. It is one of the most popular and commonly used approaches for image content representation. In fact, it has been adopted in many areas such as object detection, image classification and image retrieval. The BoW manages to segment images into salient local regions or detect the key-points and characterize them through a high-dimensional feature vector (e.g. SIFT [30], PCA-SIFT [19], SURF [4]). These low-level features were then quantified through clustering algorithms [8] and hash methods [1, 7, 11, 18, 35, 46], among others. The motivation behind this is to associate each class or group of feature vectors with a label (or index) also called visual word that identifies it. Quantization based hash method such as locality sensitive hashing (LSH) [11] or its derivates [1, 7, 18] require typically long hash code and several hash tables (typically 100–500 bytes per descriptor) to achieve satisfactory retrieval accuracies [14, 26, 51]. Therefore, the hash methods involve a huge storage overhead which is not tractable over a large database up to 2 billion of local descriptors. Quantization based clustering algorithms such as flat k-means, is the typically employed alternative [31] which requires only 2–3 bytes per descriptor to index the low-level features. All produced labels are considered as a visual dictionary, also called codebook. The content of an image is represented as a frequency histogram or a weight histogram of each visual word. The frequency histogram aims to quantize each feature vector to its closest visual word while the weight histogram aims to assign a weight for each visual word. The most common weighting scheme is the Term Frequency-Inverse Document Frequency (TF-IDF) [40]. Thus, an inverted file data structure is built for an efficient retrieval.

The ground truth shows that, in order to provide a high image retrieval performance, accurate features matching between images is needed. However, two drawbacks hinder this process. First, the discriminative power of the image feature may be reduced due to the quantization step [8]. Indeed, in the quantization step, a high dimensional feature vector (e.g. a 128-D float SIFT feature) is mapped to a single integer value leading to lose of its discriminative power. Second, the color information is often enough neglected because most of the existing local image descriptors are based on the intensity or the gradient information. Thereby, similar regions in the texture space but different in the color space are considered as true matches. Some colored descriptors based on SIFT have been proposed: Opponent-SIFT [42], HSV-SIFT [5], HueSIFT [43], etc., but these lose some invariance properties and are high-dimensional [42].

To address the above drawbacks, many techniques have been proposed. Regarding the first issue, the soft-assignment, the Multiple-Assignments (MA) and the Hamming Embedding (HE) are the most widely used techniques. The soft assignment [38] aims to quantize each feature to more than one visual word with a membership score. Generally, the weight assigned to a feature vector is an exponential function of the distance to the cluster center. However,

tuning the parameters of the exponential function still remains experimental. The multiple assignment [16] aims to assign each feature vector to k-nearest visual words only on the query side. The Hamming Embedding [13] aims to map the feature vectors to the hamming space in order to achieve a tradeoff between the large memory/time cost. However, binary features consume much less memory than the floating ones. Moreover, during the matching step, two binary features are efficiently compared using the Hamming distance via the XOR operations unlike the Euclidean distance between the floating features. Indeed, Hamming Embedding of the binary features is particularly useful especially when the database size is large.

As for the second issue, the fusion of the other characteristics such as color and shape with the texture, which is generally provided by the SIFT descriptor, remains the most suitable and solid choice. In fact, two conventional *early* and *late* fusion approaches have been proposed [21]. The early fusion involves the concatenation of several different features, related to color and texture for example, at an early stage of the quantization step. Thus, the size of the feature vectors becomes larger and the discriminative power of color information degrades. Moreover, the features are different and generally independent. The late fusion overcomes these weaknesses. Actually, the late fusion focuses on the concatenation of several BoWs at a later stage of the quantization step of each feature type. Thus, an inverted file is built for each feature type in addition to the inverted file of the concatenated BoW. In this case, the indexing strategy is inefficient in terms of both time and memory. Therefore, it is interesting to design an indexing structure that takes into account the coupling of different feature types.

In this paper, we proposed an efficient scheme for large scale image retrieval. We coupled the SIFT features as texture features with the RGB, HSV and HSL values as color features. To this end, we suggested a 2-D multi-index structure with the conjunction of a new multi-IDF formula which is more discriminative than the conventional IDF score. The proposed 2-D multi-index structure stores the information of an indexed key-point such as the image ID, the SIFT binary feature, the Term Frequency and other metadata and each entry of the 2-D multi-index structure is the pairing of a SIFT visual word with a color visual word. Moreover, during the matching step, we explored the binary features hamming embedding, the soft-assignment and the multiple-assignment techniques. In fact, the HE is employed only for the SIFT features. In addition, we suggested a non-parametric weighting formula for SIFT and color features which overcomes the tuning problem of the exponential function parameters. As for the multiple assignment, the k-nearest visual word of the SIFT features is determined by Approximate Nearest Neighbor (ANN) whereas for the color features, we explored the spatial and the weight space of the Self-Organizing-Map (SOM) [22].

The remaining of this paper is organized as follows. In Section 2, the related works were presented. In Section 3, the proposed schema for large scale image retrieval was described. In Section 4, we displayed our extensive experimental results conducted on three challenging public benchmarks and provide with a comparative evaluation with respect to the state-of-the-art CBIR systems. The paper was concluded in Section 5.

2 Related works

To date, many popular techniques have been proposed in the computer vision field in order to boost the performance of the standard approaches in terms of efficiency. In the following, we presented a review of related works on codebook generation, feature quantization, feature fusion, indexing method and visual word weighting schema.

Codebook generation The clustering step is an important phase in the image retrieval system. It is usually carried out using one of the most common clustering algorithms, the k-means algorithm [31]. The hierarchical k-means [34], approximate k-means [37] and other variants [17, 45] have been proposed to speed up the clustering step. In [8], a Self-Organizing-Map (SOM) was used to generate a low-level feature codebook.

Feature quantization Typically, in this stage, the features space is quantized into clusters called visual words via approximate near neighbor algorithm. The quantization step may cause the loss of information about the feature vector. In fact, a high dimensional feature vector is quantized to a single integer. Several approaches have been proposed to alleviate the problems caused by descriptor quantization error. In [38], a soft assignment was applied to the whole database where each feature vector is assigned to some visual words. Instead, Jegou et al. [16] recommended to only soft-assign the query feature that only increases the query processing time and no additional storage is required. Liuly et al. [27] softly match features that are neighbors in the vocabulary tree. Another approach named binary encoding scheme has been proposed. In [13], a compact binary signature of every key-point is computed by performing a random projection and thresholding the feature vector with the median value of the corresponding visual word. In [53], a SIFT binary signature is computed by a thresholding feature vector with its median value. In [23] a color feature vector was produced from a local patch around the key-point with the Color Name (CN) descriptor, and quantized to form a binary signature.

Feature fusion In image retrieval tasks, SIFT is the most widely used low-level descriptor. Despite its success, it only describes the local gradient distribution. Thus, feature fusion can be performed to enhance the performance of the local descriptors. In fact, the fusion of different features has shown great efficiency in several tasks, such as face recognition [49], image classification [9], object recognition [20], etc. Two popular approaches have been proposed: early and late fusion. The early fusion aims to combine the low-level features related to the images visual content (color, texture, shape) before performing the clustering step. The late fusion is carried out after the clustering phase in order to combine the outputs of several results. In [47], global and local color feature descriptors are jointly used with a standard SIFT feature to provide color as complementary information. In [50], an undirected graph fusion method is performed to re-rank two different ranked lists produced with BoW and global features, respectively. An improved version is proposed in [29]. In [23], a binary signature was computed for both color and SIFT features and integrated into the inverted file. Also, the features fusion was performed at the indexing level.

Indexing method Inverted files structure [40] has been proven to be the most efficient approach for large-scale image retrieval systems [13]. In fact, the inverted file is an index file used in the retrieval that stores for every visual word a list of documents containing the word. To improve accuracy, some authors propose to incorporate additional information directly into the inverted file, such as binary signature of descriptor [12], coordinates of key-points, spatial contextual information [28], etc. In [3], the authors first split the SIFT features into two blocks and then a multi-index structure was built by the product quantization, PQ. In [24], two different descriptors, SIFT and color name descriptors, are used to quantize features vectors into a visual word tuple. Then, an entry index is determined in the multi-index structure.

Visual word weighting Term Frequency-Inversed Document Frequency (TF-IDF) [40] is a traditional method for weighting features based on their distinctiveness. The inverted file structure allows a quick and efficient calculation of the TF-IDF score for each image in the database. In [38], the authors used a soft weighting by assigning a feature vector to several visual words. As mentioned in [15], the traditional weighting method does not address the problem of the visual word burst. To face this phenomenon, the intra and inter burst were produced by square-rooting the term frequency TF of the visual word in an image. In [25], an Lp-norm IDF weighting strategy is employed. Other approaches focus on the distribution of the local descriptor in the spatial neighborhood such as spatial weighting [6] and spatial contextual weighting [44].

3 Proposed approaches

In this section, we followed the notation and the proposed framework which encapsulates many popular techniques, including bag-of-words (BoW), rootSIFT [2], Hamming Embedding (HE) [13] and burst weighting [15].

Let us assume that an image collection has totally N images, denoted as $IDB = \{I_1, I_2, \ldots, I_N\}$. Each image $I_i$is described by a set $X = \{x_1, x_2, \ldots, x_n\}$ of n key-points. Given a quantized function q that maps a feature x_i into a visual word ID $q(x_i)$, such that, $q(x_i) \in C$, where $C = \{c_1, c_2, \ldots, c_k\}$ is a visual codebook of size k.

During the online query process, given a query image Q, the similarity function between Q and each image I of the database IDB can be formulated as:

$$sim(Q, I) = \frac{\sum_{x \in Q, y \in I} f(x, y)}{\|Q\| * \|I\|} \tag{1}$$

where $f(x, y)$ is a similarity function between the x, y features which are quantized on the same visual word and ‖.‖ is the L2 norm.

3.1 Feature extraction and codebooks generation

In order to derive compact descriptions and efficiently represent the content of an image, two features types were coupled during the indexing process. In fact, the SIFT feature was used as a texture feature and the pixel color value as a color feature. The main motivation was to enhance the discriminative power of the SIFT features. In fact, the SIFT descriptor only captures the gradient distribution of a local region and therefore its discriminative power is lost during the quantization step. Moreover, when relying on the gray level space, it ignores the color values of pixels. These features are then separately quantized into two different codebooks.

SIFT feature extraction Scale-invariant key-points are detected with Hessian-affine region detector and described using a SIFT descriptor. In addition, the hamming embedding technique was adopted as it has been proposed in [13] to inject a SIFT binary feature in our framework.

Color feature extraction Three 3-D vectors were extracted for each detected key-point. Each vector represents the pixel color values in RGB, HSV and HSL color spaces, respectively. Then, they are concatenated, at the early fusion step, in order to produce a 9-D vector. Each bin

is a float ranging from 0 to 1. We note that the saturation value in the HSV space is different from the HSL space. The main objective behind fusing the three spaces was to produce a stable feature vector against illumination change and small variations in the RGB color space. In fact, HSV/HSL spaces are consistent with the human visual perception system, while the RGB space preserves the pixels initial values.

SIFT codebook generation We used the approximate k-means (AKM) for SIFT as in [37]. The codebook was trained using an independent flickr60K dataset [13]. Each SIFT descriptor was quantized to the nearest centroid using an Approximate Nearest Neighbor (ANN) algorithm.

Color codebook generation For color feature, a 2-D Self-Organizing-Map (SOM) [22] was used for quantization. The advantage of using the SOM in the codebook generation lies in the fact that codebooks can be formed in an unsupervised way. SOM attempts to preserve the topological properties of the feature space. Two close to each other features in the feature space are mapped into neighbor classes (neuron) on the map.

Multiple assignments MA As a variant, MA aims to assign a local descriptor to several visual words instead of choosing the nearest neighbor. Indeed, the 2-D inverted file generates a large codebook size that allows achieving a high precision; therefore, using the MA strategy is recommended to improve recall. MA is known to increase the query time and memory overload, thus, it is only performed on the query side. In this paper, we adopted the MA strategy for both SIFT and color descriptors.

3.2 The inverted multi-index illustration

In this section, we explained how an inverted Multi-Index is organized. Relying on this multi-index structure, we proposed a new multi-IDF schema (Sec. 3.4).

First, we presented a brief description of a conventional inverted file (Fig. 1) denoted as $W = \{w_1, w_2, .., w_k\}$ where each entry W_i contains a list of posting. In practice, each entry $W_i$can be represented as a contiguous array in the memory. In the computer vision field, each posting contains normally a pair of image ID and term frequency (TF) scores. Many works [13, 23, 44, 52] have incorporated other metadata associated with the indexed key-point such as a SIFT binary signature, color binary signature, etc.

As shown in Fig. 2, an inverted multi-index file can be seen as a multi-dimensional table which entries correspond to all possible tuples of visual words from the codebooks corresponding to different features. Given M features indexed in the inverted-file, the multi-index dimension is equal to M. In this paper, we used a 2-D inverted index as [23] which is also

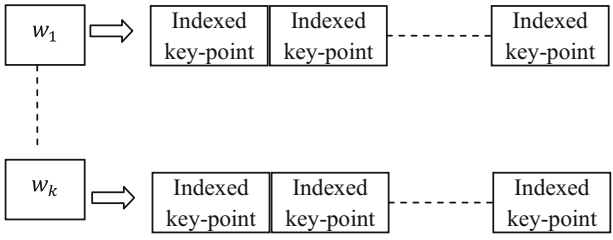

Fig. 1 Structure of conventional inverted file

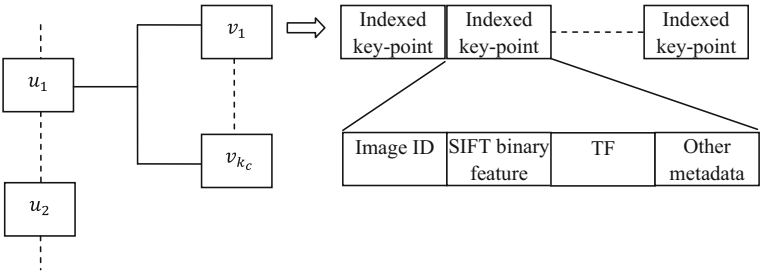

Fig. 2 Illustration of 2-D inverted multi-index

called second-order in [3]. It can be built as follows. First, for each image in the database, key-points are extracted, and for each key-point x_i, two features denoted as $x^s \in R^{D_s}$ and $x^c \in R^{D_c}$ are computed representing SIFT and color descriptor, respectively. Then, x^s and x^c are quantized to their closest corresponding visual words using pre-generated codebooks $U = \{u_1, u_2, \ldots, u_{k_s}\}$ and $V = \{v_1, v_2, \ldots, v_{k_c}\}$, respectively, where k_s and k_c are the codebook sizes. The result is a tuple of visual words (u_i, v_j) , $i = 1 \ldots k_s$, $j = 1 \ldots k_c$, which would be assigned to an entry in the inverted multi-index. A 2-D multi-index inverted file is presented in Fig. 2.

Given a query feature tuple$[x^s, x^c]$, we first quantized it into a visual word pair(u_i, v_j). Then, we determined the correspondent entry in the inverted multi-index and therefore, a posting list of indexed key-points was taken as a candidate list.

3.3 Proposed weighting formula

In soft-assignment, the weight assigned to a query feature is usually an exponential function of the distance between a query feature and a database feature. It is given by the following equation:

$$f(x,y) = \begin{cases} \exp\left(-\frac{{d_h}^2}{\sigma^2}\right), \text{if} |d_h < h_t \\ 0, |\text{otherwise} \end{cases} \tag{2}$$

where x is a query feature, y is an indexed feature which is quantized to the same visual word with x, d_h denotes the hamming distance between binary signature of x and y, σ is the spatial scale and h_t is the hamming threshold.

The bandwidth σ is the most important parameter. In fact, several works [13] try to tune σ in order to find the best value. The purpose is to choose σ so that a substantial weight is assigned to the nearest feature of the query.

Our first contribution consists in alleviating the problem of the parameters tuning. To this end, we proposed a formula that does not induce on any tuning parameters that improves results relative to existing ones. Therefore, we proposed a formula that decreases the weighting in a square linear function passing by two coordinate points (0, 1) and $(h_t + 1, 0)$.

$$f(x,y) = \begin{cases} (m * d_h + b)^2, \text{if} |d_h < h_t \\ 0, |\text{otherwise} \end{cases} \tag{3}$$

where $m = -1/(h_t + 1)$ is the slope of the line, $b = 1$determines the y-intercept, d_h denotes the hamming distance between the binary signature of x and y and h_t is the hamming threshold.

Fig. 3(a). shows the plot of the function $\exp\left(-\frac{d_h^2}{\sigma^2}\right)$ when $h_t = 64$ with different spatial scale σ values. It is clear that the best value is $\sigma = 26$. Conversely, Fig. 3(b) shows a comparison of the proposed weighting formula with $\exp\left(-\frac{d_h^2}{\sigma^2}\right)$ when $\sigma = 26$.

In addition, once the spatial scale σ is determined Eq. (2) assigns the same weights, without taking into account the maximum value of the hamming threshold h_t. Moreover, our weighting formula automatically adjusts, depending on the hamming threshold making it independent from any tuning parameters.

Color weighting It is beneficial to take the color weighting in our system into account. Our second contribution is revealed in this context. The SOM topology map was exploited in the calculation of the color weighting. Two different weighting types were used. The first one involves the neighborhood distance between two features in the map Euclidian space. The second one involves the distance between the weight vectors of the mapped neuron. Given two color features *xc* and *yc* the spatial distance between these features is given by the formula:

$$f_c(xc, yc) = \begin{cases} c_t + 1 - \|nx - ny\|, \text{if} \|nx - ny\| < c_t \\ 0, \text{otherwise} \end{cases} \quad (4)$$

where *nx* and *ny* are the indices of the mapped neuron of *xc* and *yc*, respectively and c_t is a predefined color threshold.

The second weighting distance is given by the following formula:

$$d(nx, ny) = \exp(-\|w(nx) - w(ny)\|), \text{if} \|nx - ny\| < c_t \quad (5)$$

where *w*(.) is the weighting vector of a visual word, in our case, it is the weight vector of one neuron of map.

The proposed Eq. (4) and (5) are very complementary to each other. In fact, assuming that two close features in a feature space are mapped into two classes that are far in the map space, Eq. (4) allows assigning a high value while Eq. (5) allows assigning a low one. Thus, the final weight obtained by the product of the two equations is a low weight.

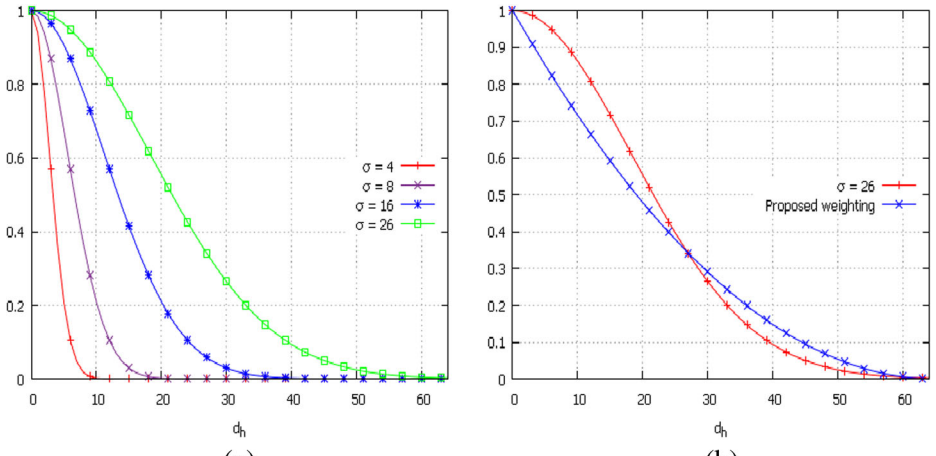

Fig. 3 Comparison of the proposed weighting formula

3.4 Multi-IDF

In this section, we introduced the third contribution of this paper: the multi-IDF formula which is based on a 2-D inverted file structure.

Conventional IDF The TF-IDF (term frequency-inverse document frequency) [34, 40] weighting schema is the most well-known method used in information retrieval and classification. It is used as a metric for measuring the importance of a word in a document. TF represents the number of occurrences of a visual word that appears in an image, whereas the IDF reflects their discriminative abilities of the visual words for an image I in an image collection IDB. Their introduction in the computer vision field was pioneered by Sivic et al. [40] in 2003. Since then, they have been prevalently used in the BoW-based image retrieval. For a conventional inverted file, the IDF value of a visual word c_i is formulated as:

$$IDF(c_i) = \log\frac{N}{n_i} \tag{6}$$

where N denotes the total number of images in IDB, and n_i encodes the number of images where c_i occurs.

Moreover, in case of a 2-D inverted file, the IDF value of a visual word tuple (u_i, v_j) is defined as:

$$IDF\left(u_i, v_j\right) = \log\frac{N}{n_{i,j}} \tag{7}$$

where N denotes the total number of images in the IDB while $n_{i,j}$ encodes the number of images where (u_i, v_j)occurs.

Proposed IDF Let (u_i, v_j)a visual word tuple where u_i (SIFT feature) and v_j (color feature) are the principal and auxiliary visual words, respectively. The drawback of Eq. (7) is that the importance of the IDF value of each visual word is overlooked. However, obviously, it is insufficient to consider only the IDF value of a tuple of visual word when measuring its weight. A reasonable choice to address the above problem involves the integration of the IDF value of the principal visual word in calculating the weight of the visual word tuple. Thus, we proposed, in this paper, a new multi IDF-weighting formula for a 2-D inverted file that takes advantage of multi-index structure defined as:

$$IDF_2\left(u_i, v_j\right) = IDF(u_i) * IDF\left(u_i, v_j\right) \tag{8}$$

An example of our multi IDF weighting is presented in Fig. 4. and evaluated in Sec. 4.3.

According to the above equations, Eq. (1) becomes as follows:

$$sim(Q, I) = \frac{\sum_{(x,xc)\in Q,(y,yc)\in I} f(x,y) * f_c(xc,yc) * d(nx,ny) * IDF_2\left(u_i, v_j\right)}{\|Q\| * \|I\|} \tag{9}$$

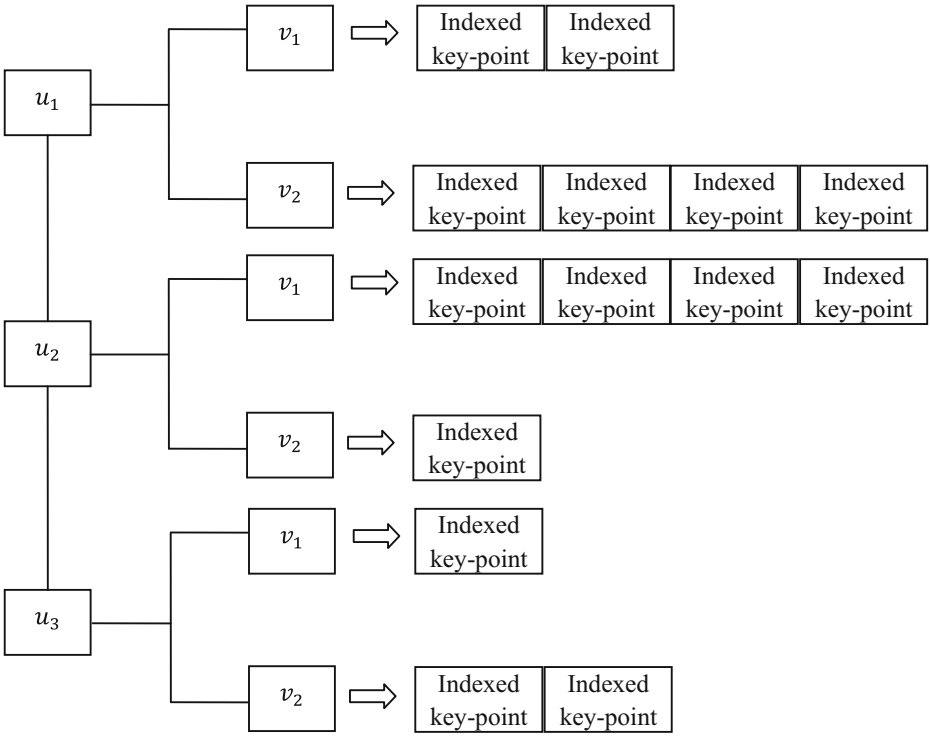

Fig. 4 An example of our multi IDF weighting. For observation convenience we set IDF function as N/n_i. If k_s is 3 and k_c is 2, we can get IDF value of visual word tuple $(u_1, v_1) = 14/2$ equal to tuple (u_3, v_2) likewise for tuple (u_1, v_2) and tuple (u_2, v_1). However, visual word u_3 is more discriminative than u_1, because the total # of key-point quantized to u_3 equal to 3 is lower than u_1 equal to 6. So, if we consider 1-D inverted file composed only of a visual word U, $IDF(u_3) = 14/3$ and $IDF(u_1) = 14/6$. Thus, the new IDF_2 value according Eq. (8) of tuple (u_1, v_1) becomes (14/6)*(14/2) which is lower and different from the value of tuple $(u_3, v_2) = (14/3) * (14/2)$

4 Experiment

This section detailed the experiments that were carried out to evaluate the performance of the proposed framework. The experiments were performed on three public benchmark datasets: Holidays, Ukbench and MIR Flickr 1 M. We first introduced the datasets, analyzed their different parameters and evaluated their impact. Secondly, we provided a comparison with the state-of-the-art. Finally, we provided some measures of memory cost, efficiency and scalability of the proposed framework.

4.1 Datasets

Holidays This dataset contains 1491 personal holiday images undergoing various transformations. The dataset contains 500 image groups where the first image of each group is the query and the correct retrieval results are the other images of the group. The dataset is available at [13].

Ukbench This dataset contains 2550 different objects or scenes collected by [34]. Each object is represented by four images taken from four different viewpoints, giving 10,200 images.

MIR Flickr 1 M This dataset contains one million images randomly crawled from Flickr. It is used to check the scalability of our framework.

Flickr60K This dataset contains 67,714 images downloaded from Flickr and provided with the Holidays dataset [13]. It is used as an independent dataset to determine several parameters of the proposed framework.

To evaluate the performance of our framework, we used the mean average precision (mAP) as a metric. In addition, we also used the average recall of the four top returned images, refereed as to Score@4, only for the experiments on the Ukbench dataset.

4.2 Parameter analysis

Three main parameters are taken into account in the proposed framework: the color codebook size k_c, the hamming threshold h_t and the color threshold c_t, which allow determining the number of color multiple assignments. We set the SIFT codebook size to 20 K and $MA^S = 3$ for SIFT descriptor; so that for each SIFT descriptor the three neighboring visual words are set, in the following experiments unless otherwise stated. Following [13] we set SIFT binary signatures to 128-D and 64-D on Holidays and Ukbench, respectively.

Color codebook size To evaluate the impact of the color codebook size, we tested different settings. First, we randomly extracted the color features. A total of 32 K features vectors were provided for learning. Then, SOM was trained to generate different sized (Table 1) two-dimensional codebooks for different numbers of training epochs of the SOM algorithm. The results are plotted in Fig. 5.

We presented, in Fig. 5, the mAP results on Holidays dataset with different color codebook sizes and hamming distance thresholds. Fig. 5(a) shows results obtained without multiple assignment strategy for both SIFT and color. It is shown that a small color codebook size performs better results than a large one. This is obvious because with a smaller codebook size, visual words are not distinctive enough and more relevant features are checked. In addition, it provides a higher recall rate compared to a large codebook that allows boosting mAP. On the other hand, when multiple assignments were employed, an accuracy peak with a 400 codebook size (square map of 20*20) was reached, as revealed by Fig. 5(b). In fact, a large codebook size provides a lower recall, so MA tends to overcome this drawback and to improve accuracy. The same behavior was noticed over all datasets. In the following experiments, we set the color codebook size $k_c = 400$.

SIFT weighting parameters Only one parameter was used in our weighting formula: the hamming threshold h_t. In fact, h_t is an important parameter of the proposed framework and its setting may depend on the other used techniques.

Table 1 Different sizes of the used color codebooks (features map)

Map X_Y	10_10	16_16	20_20	32_32
Codebook sizek_c	100	256	400	1024

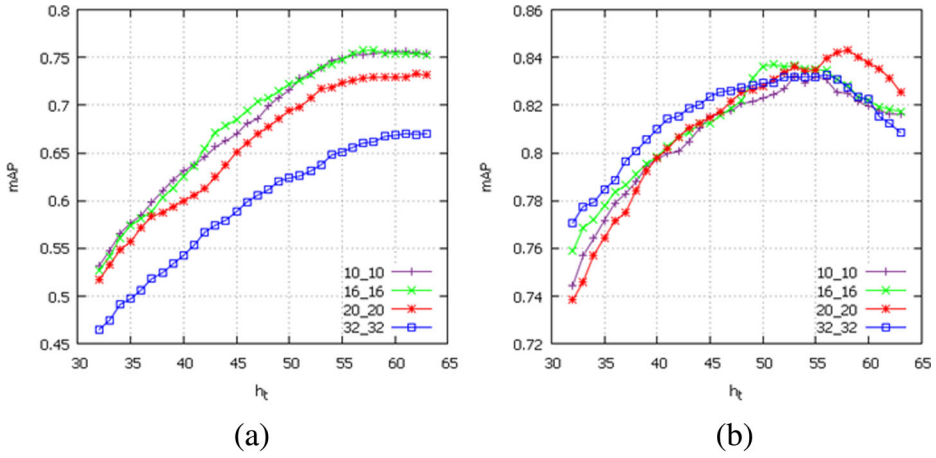

Fig. 5 Impact of color codebook size

As shown in Fig. 6 when the hamming threshold h_t increases, the mAP first increases to a peak, and then drops slightly.

SIFT multiple assignments MA^S Fig. 7 shows the impact of the MA^S parameter by varying its value from 1 to 3. In fact, Fig. 7(a) shows the results obtained using the integration of the color feature with the SIFT feature (we have used the weighting strategies of the SIFT and color features and Hamming Embedding). While, Fig. 7(b) display the results obtained based on the SIFT feature. We note that we did not use the IDF and the burstiness weighting strategies in this experiment. From Fig. 7 it can be seen that the best retrieval performance was achieved with a value of MA^S = 3 and the results obtained varying the value of MA^S are relatively close.

Color parameters Just like the SIFT weighting, only one parameter is involved in color weighting formulas (Eq. (4) and Eq. (5)), i.e., the color threshold c_t. Furthermore, this parameter plays another role: it allows controlling the number of color multiple assignments for each key-point. To check the effectiveness of c_t, we performed several experiments by varying c_t either as a percentage of the diagonal value of the map used in learning, or of all

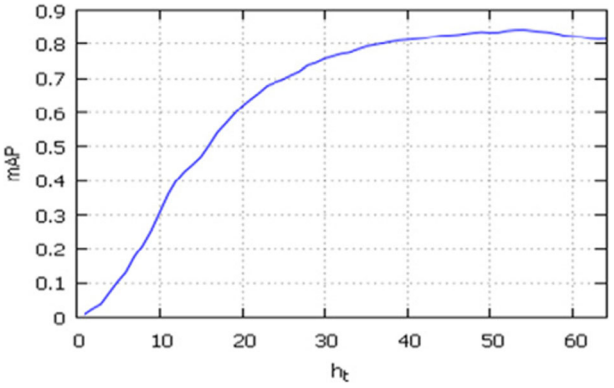

Fig. 6 Impact of hamming threshold on Holidays dataset

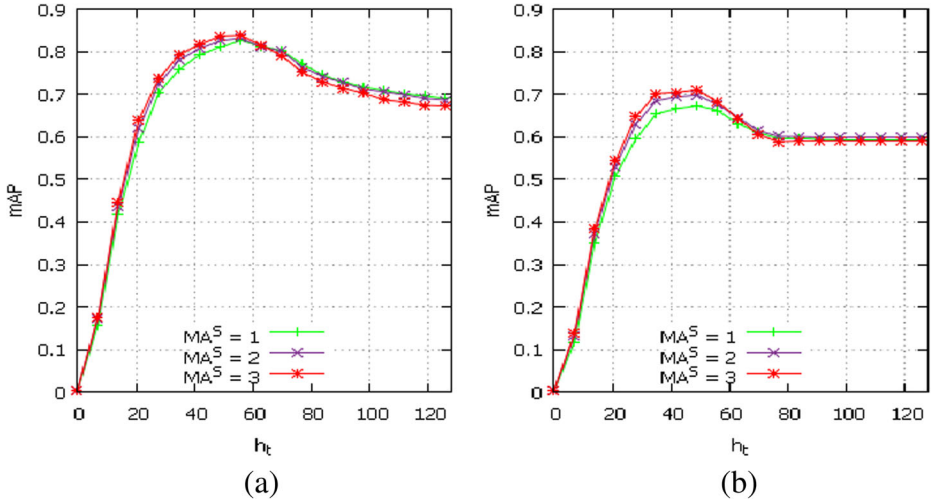

Fig. 7 Impact of MA^S on Holidays dataset

possible values (we used the percentage only for the comparison between different codebook sizes, because for different sizes of k_c, the diagonal value of the map would be different). At this stage, we conducted two experiments to check the influence of the c_t parameter. We first studied the impact of the color threshold $c_t$in multiple assignments and after that we presented the importance of adopting the color weighting in our system.

Color multiple assignments MA^C For each color feature, the number of color MA^C varied. In fact, it is monitored by two parameters; the first is the spatial position of the neuron in the map for which the color feature is mapped and the second is the value of c_t.

Fig. 8 shows the results of the color threshold impact of multiple assignments obtained with color codebooks size equal to 400 learned with SOM algorithm, achieved with a square topology map of dimension 20*20. When c_t increases, the mAP curves rise swiftly and steeply before stabilizing at $c_t = 21$, meaning 75 % of the SOM map diagonal value. This is due to the

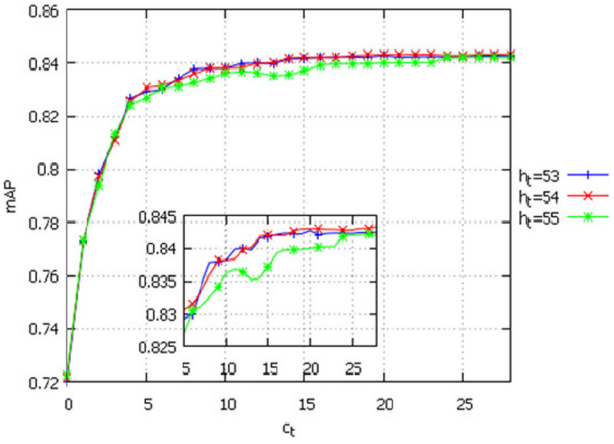

Fig. 8 Color threshold impact on the Holidays dataset. The horizontal axis represents all possible values of c_t. Three plots corresponding to h_t = 53, 54 and 55 respectively. We observe a superior performance at h_t = 54

fact that, when the c_t value is small, i.e. around 4, many relevant features are included. In fact, similar features in the input space are mapped into nearby classes in the feature map. However, when c_t increases more noise features are also included yet, the curve continues to grow but slowly. It is because some relevant features are mapped into classes that are far from the query feature classes. On the other hand, a large value of c_t increases the query time cost. To establish a tradeoff between efficiency and time cost we set $c_t = 21$ in all the experiments.

Conversely, as illustrated in Fig. 9, the best accuracy was achieved with a color codebook size equal to 400. On the other side, when the value of c_t increases almost all the plots increase from a stable status, which is consistent with previous experiments.

Fig. 10 presents mAP and Score@4 results obtained on Ukbench dataset as function of the color threshold. We showed that the profiles of the two plots are quite similar. In fact, mAP first increases and when c_t is larger than 20 % of the value of the diagonal map, the plots are approximately linear with variations of about 0.01 and 0.1 for mAP and Score@4 measures, respectively. This is due to the fact that the illumination of images in Ukbench is less severe compared to that of Holidays. Thus, the benefit of a large value of MA^C is very important on Holidays.

4.3 Impact of color weighting

As shown in Fig. 11 this experiment reveals the impact of our color weighting formulas. In fact, we have adopted, in this paper, two different equations for color weighting: spatial and weighting. To evaluate the benefits of the proposed color weighting formulas, we first tested the impact of each equation separately (Eq. (4) and Eq. (5)). In fact, Fig. 11 shows that the results obtained with Eq. (4) and Eq. (5) are very close with a slight improvement obtained when the color weighting is not used. Conversely, by coupling these two alternatives, retrieval performance is further improved. Therefore, we proved that the integration of the color weighting boosts the retrieval system accuracy.

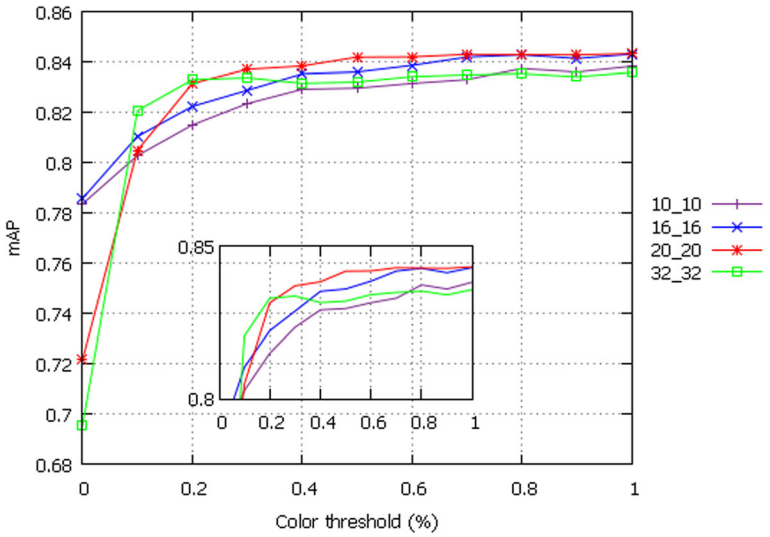

Fig. 9 Impact of color threshold on the Holidays dataset

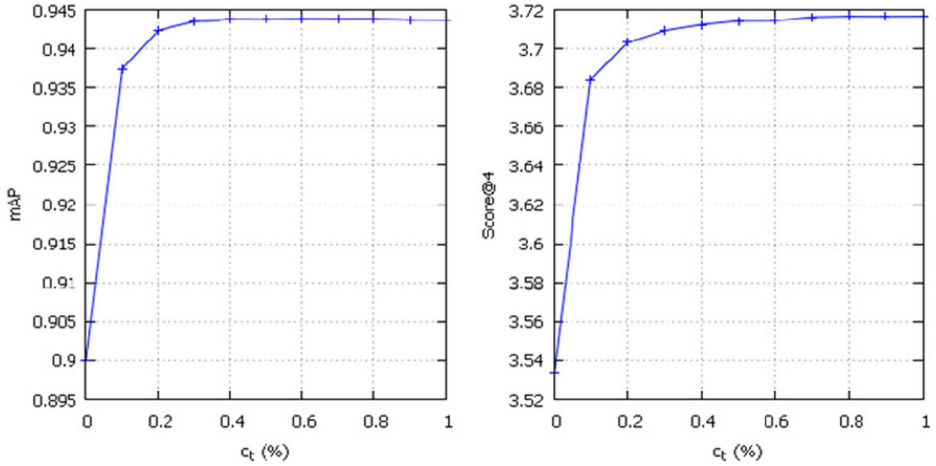

Fig. 10 Impact of color threshold on Ukbench dataset

4.4 Evaluation

Baseline We implemented the standard bag-of-words BoW approach as follows. We applied the rootSIFT [2] for each SIFT features extracted with Hessian-Affine from the interest points [32]. The performance was calculated according to Eq. 1 where $f(x, y)$ is the square IDF value of the visual word of the feature vector x which is quantized to the same visual word with y.

In a first step, we started the evaluation by checking the effect of introducing the color information in our framework. Therefore, we presented mAP Fig. 12(a) and Score@4 Fig. 12(b) results on Holidays and Ukbench datasets, respectively. As shown in Fig. 12, the color feature improves significantly the retrieval performance when it is coupled with the SIFT feature.

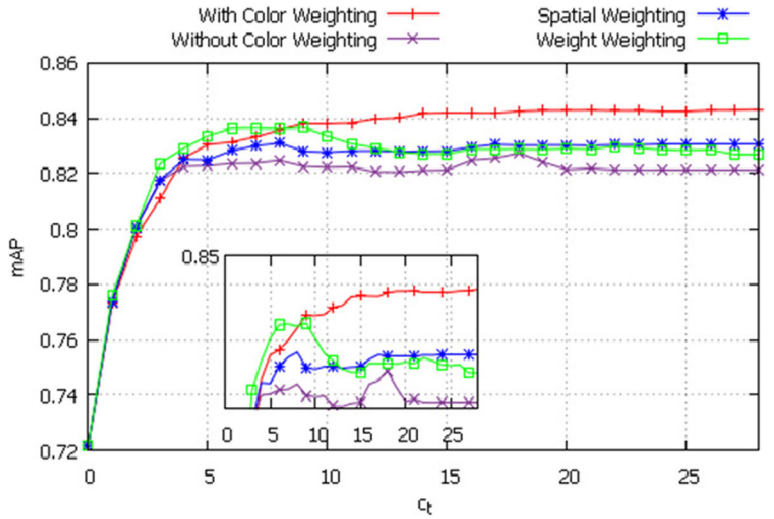

Fig. 11 Impact of color weighting on Holidays dataset

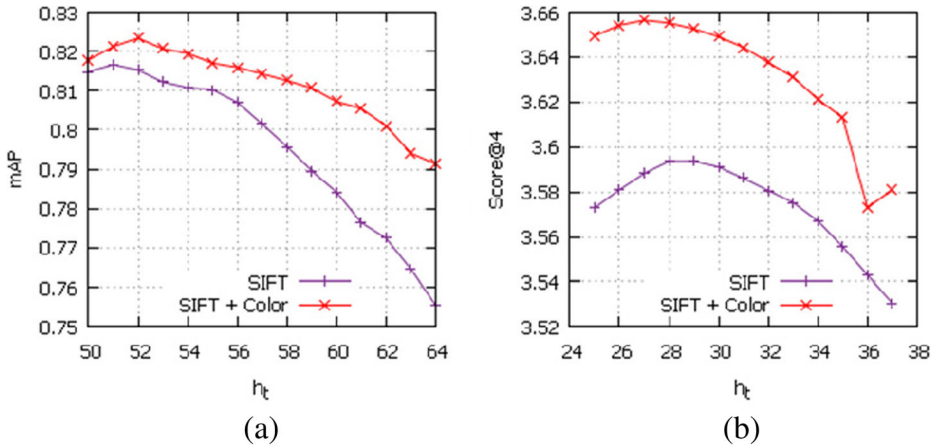

Fig. 12 Impact of color features

Then, we compared the proposed weighting formula (Eq. (3)) and multi-IDF (Eq. (8)) to the conventional weighting formula (Eq. (2)) and baseline IDF, respectively.

In most of the proposed works that used $\exp\left(-\frac{d_h^2}{\sigma^2}\right)$ as a weighting function, the best accuracy was reached with a hamming threshold value ranging between 40 % and 50 % of the number of bits used for the binary signatures and σ is typically chosen to be one quarter of the number of bits [15, 41]. Relying on this observation and to showcase the impact of the proposed weighting formula, the values of σ was varied and mAP was compared on the Holidays dataset. The results of Fig. 13 prove the superior performance of the proposed weighting formula Eq. (3) in terms of mAP. We also show that $\exp\left(-\frac{d_h^2}{\sigma^2}\right)$ is very sensitive to the value of σ.

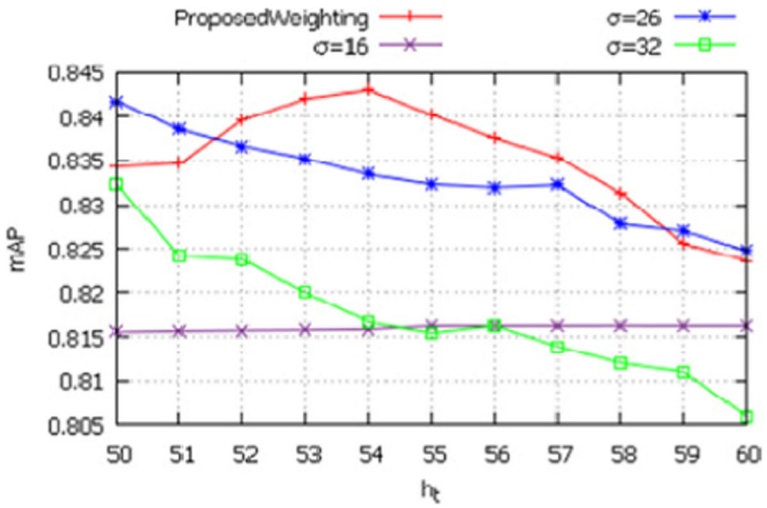

Fig. 13 Impact of the proposed weighting function

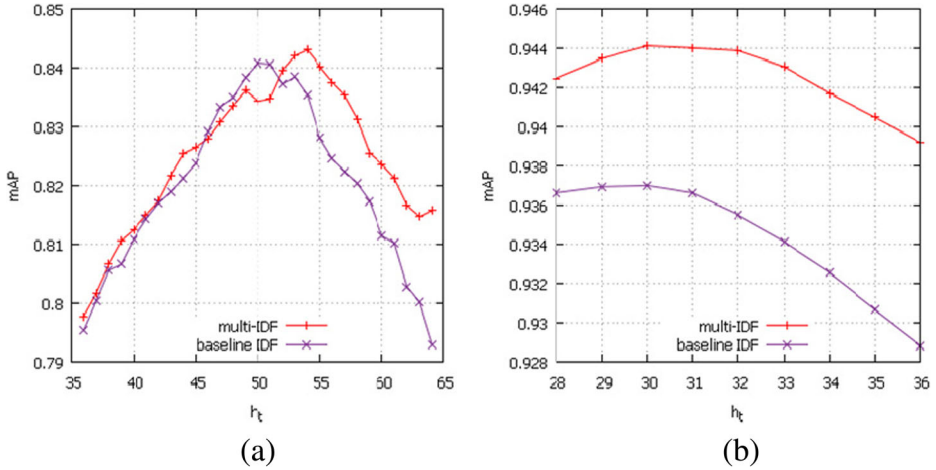

Fig. 14 Comparison of the baseline IDF with the proposed multi-IDF, **a** Holidays dataset and **b** Ukbench dataset

The other important result is the performance of the proposed multi-IDF compared to the baseline IDF. As expected, the multi-IDF outperforms the retrieval performance in terms of mAP given by a baseline IDF almost for different values of h_t on both Holidays and Ukbench datasets. We noticed the same behavior for the variation of the Score@4 plot with Ukbench dataset as well (Fig. 14.).

Impact of different strategies As shown in Table 2 the fusion of different strategies achieved the best retrieval performance across different datasets. In fact, we obtained a mAP

Table 2 Retrieval performance by different variants of the proposed method on Holidays and Ukbench datasets. Multi-Index (MI), Burst weighting (Burst), hamming embedding weighting (W^{HE}), hamming embedding (HE), SIFT multiple assignment (MA^S), color weighting (W^C) and color multiple assignment (MA^C)

Methods	MI	multi-IDF	burst	W^{HE}	HE	MA^S	W^C	MA^C	Ukbench ($h_t = 32$)		Holidays ($h_t = 54$)
									Score@4	mAP (%)	mAP (%)
BoW									3.115	78.14	49.67
BoW					×				3.316	83.91	70.79
BoW				×	×				3.546	90.26	79.04
BoW				×	×	×			3.580	90.92	81.08
BoW			×	×	×				3.612	91.73	79.71
BoW			×	×	×	×			3.644	92.42	81.94
BoW	×								3.222	82.70	59.41
BoW	×	×		×	×				3.436	87.50	67.82
BoW	×	×	×	×	×				3.466	88.03	67.92
BoW	×	×	×	×	×	×			3.533	89.99	72.14
BoW	×	×	×	×	×	×		×	3.633	92.13	76.04
BoW	×	×	×	×	×	×	×	×	3.716	94.38	84.30

Table 3 Comparison with state-of-the-art results without post processing step

Approaches	Holidays (mAP %)	Ukbench (Score@4)
[51] Zhang et al.	80.9	3.60
[39] Philbin et al.	76.2	3.52
[24] Zheng et al.	79.6	3.60
[12] Jain et al.	79.4	-
[16] Jegou et al.	81.3	3.42
[44] Wang et al.	78.0	3.56
[23] Zheng et al.	84.0	3.71
[15] Jegou et al.	83.9	3.54
[26] Lin et al.	77.7	-
Proposed	84.3	3.71

of 84.3 % and a Score@4 of 3.71 on Holidays and Ukbench datasets, respectively. Compared to the baseline, a large improvement of mAP by 34.63 % on Holidays and 16.24 % on Ukbench was observed. As expected, the multi-index scheme improves accuracy over the baseline system by a mAP equal to +9.74 % and a Score@4 equal to +0.10 on Holidays and Ukbench, respectively. A similar situation can be observed when using both of multi-index and the multi-IDF schemas. This is obvious since the multi-IDF schema is based on the multi-index schema. It should be noted that introducing the color feature brings the mAP from 81.94 % up to 84.30 % on Holidays, and, Score@4 from 3.64 up to 3.71 on Ukbench. In addition, we showed again that W^C and MA^C are strongly tied. Indeed, when the W^C was used the best mAP achieved with MA^C and other techniques jumped from 76.04 % to 84.30 % and from 92.13 % to 94.38 % on Holidays and Ukbench, respectively. A similar trend can be observed on Ukbench: Score@4 rises from 3.63 up to 3.71. Likewise, the multiple assignment method for both SIFT and color has achieved a great large improvement over the baseline approach. In addition, the benefit of HE and burst are consistent with previous works [13, 15].

Comparison with state-of-the-art A comparison of our framework results with the state-of-the-art, on the two most widely used dataset namely Holidays and Ukbench was provided in Table 3 without applying a post-processing step. Firstly, our framework exceeds the LSH based frameworks such as [26, 51] in terms of mAP and Score@4. Moreover, the use of color features enhances the CBIR accuracy. In fact, the results provided by our framework and those of Zheng et al. [23, 24] are very close and outperform the results of other frameworks based only on SIFT features [12, 15, 16, 39, 44]. We have reached a **mAP = 84.3** and **Score@4 = 3.71** on Holidays and Ukbench datasets, respectively. The obtained mAP score slightly exceeds those of [23] on Holidays by about +0.3 %.

Table 4 Memory cost per key-point

Mata-data	Image ID	SIFT index	HE	Color Index	TF
Memory per key-point (bit)	21	15	64	9	8

Table 5 Memory cost for different approaches

Mata-data	Image ID	MI	HE	MI + HE
1 M dataset (GB) (ours)	1.2	1.2	3.5	4.7
1 M dataset (GB) [23]	1.7	2.8	5.0	6.1

4.5 Large scale evaluation

Memory cost The memory usage is an important factor in large-scale images retrieval. As shown in Table 4, for each indexed key-point, we need 21 bits to store the image ID (in case of 1 M images datasets), 15 bits to store the quantized index of SIFT features, 64 bits are allocated for the 64-bits binary SIFT features, 9 bits are needed to store the quantized index of pixel key-point color, and 8 bits to store the TF of the key-point. To show the effectiveness of the proposed system, the memory usage of our system was compared with a similar state-of-the-art system [23]. As shown in Table 5, on the 1 M dataset, the visual words for both SIFT and color consumes only 1.2 GB memory, which means −1.6 GB compared to [23]. Also, MI + HE totally consume 4.7 GB which makes our approach suitable for large-scale experiments.

Efficiency The experiments were performed on a PC with 3.4 GHz CPU and 32 GB physical memory. The average extraction time per image for SIFT and color features is 0.431 s and 0.064 s on the 1 M dataset, respectively. Table 6 shows the average retrieval time cost per query for three methods on Holidays dataset + MIR Flick 1 M. The time cost of features extraction and quantization are not included.

Introducing the hamming embedding marginally increases the retrieval time to 0.107 s per query. In contrast, MA is the most time consuming, introducing an additional 0.618 s over the baseline, mainly due to the use of MA for SIFT, color features and the use of the color weighting.

Scalability The scalability of the proposed framework was evaluated on two datasets. We merged Holidays and Ukbench with various fractions of the MIR Flickr 1 M dataset, respectively. The results are plotted against the database size, as shown in Fig. 15.

Fig. 15 shows that the proposed framework works well on a large scale setting. In fact, the accuracy, first, dropped slowly when we merged the query datasets with a small fraction of MIR Flickr (e.g. 100 K), and then, it kept stable regardless of the number of distractor images. Also, from Fig. 15 we can see that when we adopted the different strategies, the addition of 1 M distractor images caused only a 7.9 % and 3.4 % drop in accuracy in terms of mAP on Holidays and Ukbench, respectively.

Table 6 Average query time on Holidays +1 M dataset

Methods	BoW + MI	BoW + MI + HE + W^{HE}	BoW + MI + HE + W^{HE} + MA
Average time cost per query (second)	0.102	0.107	0.720

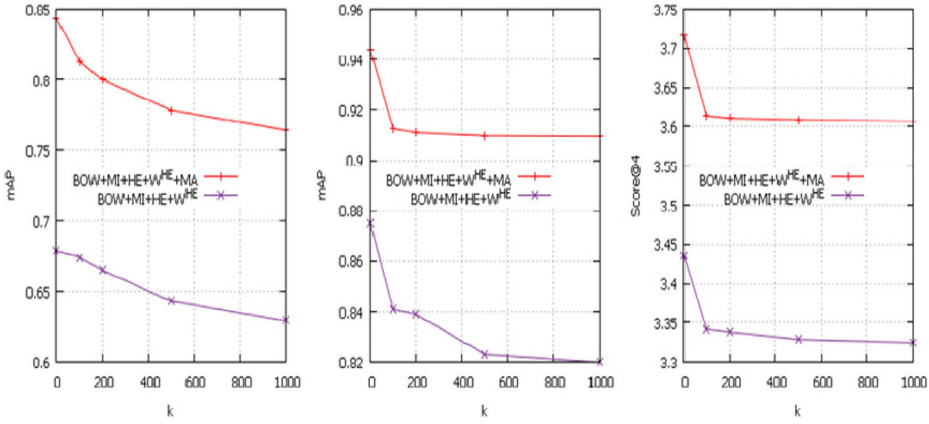

Fig. 15 Large scale experiments results with MIR Flickr 1 M. mAP for **a** Holidays and **b** Ukbench. Score@4 for **c** Ukbench. Note that the multi-IDF is applied in the entire test

We present, in Table 7, a comparison of the baseline system with various state-of-the-art baseline systems results. As shown, for the large scale experiments, we reached a mAP equal to 34.2 % on Holidays dataset and a Score@4 equal to 3.00 on the Ukbench dataset. The obtained results were compared favorably with the state-of-the-art systems.

5 Conclusion

In this paper, an efficient scheme for large scale image retrieval was proposed. It focused on coupling the SIFT features and the color features at the indexing level. To this end, a 2-D multi-index structure together with a new multi-IDF formula were proposed. Moreover, we introduced an effective weighting formula that is independent of any parameters. The effectiveness of our image retrieval scheme was proven. In fact, it consistently outperforms several state-of-the-art methods on two public datasets and consumes an acceptable memory cost.

As a future work, we will investigate the adoption of post-processing techniques such as RANSAC verification [37], k-NN re-ranking [39] and graph fusion [50] to further improve the accuracy of the proposed framework,.

Table 7 Comparison with state-of-the-art baseline systems results

Approaches (baseline)	Holidays (mAP %)		Ukbench (Score@4)	
Dataset distractor size	-	1 M	-	1 M
[24] Zheng et al.	61.0	32.1	3.40	3.05
[16] Jegou et al.	58.0	33.0	2.95	-
[23] Zheng et al.	49.0	23.2	3.02	2.72
[15] Jegou et al.	46.9	24.0	2.99	-
Proposed	59.4	34.2	3.22	3.00

References

1. Andoni A, Indyk P (2008) Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun ACM* 51(1):117–122
2. Arandjelovic R and Zisserman A (2012) Three things everyone should know to improve object retrieval. In *Proceedings of the 2012 I.E. Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2911–2918, Washington, DC, USA, IEEE Computer Society
3. Babenko, A and Lempitsky VS (2012) The inverted multi-index. In *2012 I.E. Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16–21, 2012*, pages 3069–3076
4. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). *Comput Vis Image Underst* 110(3):346–359
5. Bosch A, Zisserman A, Muñoz X (2008) Scene classification using a hybrid generative/discriminative approach. *IEEE Trans Pattern Anal Mach Intell* 30(4):712–727
6. Chen X, Hu, X and Shen X (2009) Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. In Theeramunkong T, Kijsirikul B, Cercone N and Ho TB editors, *PAKDD*, volume 5476 of *Lecture Notes in Computer Science*, pages 867–874. Springer
7. Datar M, Immorlica N, Indyk P and Mirrokni VS (2004) Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, pages 253–262, New York, NY, USA, ACM
8. Elleuch Z and Marzouki K (2013) Optimization of BOW using self organizing map artificial neural network in similar images retrieval systems. In *Pattern Recognition and Image Analysis - 6th Iberian Conference, IbPRIA 2013, Funchal, Madeira, Portugal, June 5–7, 2013. Proceedings*, pages 330–339
9. Fernando B Fromont É Muselet D and Sebban M (2012) Discriminative feature fusion for image classification. In *2012 I.E. Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16–21, 2012*, pages 3434–3441
10. Hua X-S, Wang S, Li S, Lu W, Wang J (2011) Contextual image search. In *ACM Multimedia*
11. Indyk P and Motwani R (1998) Approximate nearest neighbors: Towards removing the curse of dimensionality. In Vitter JS editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23–26, 1998*, pages 604–613. ACM
12. Jain M Jégou H and Gros P (2011) Asymmetric hamming embedding: taking the best of our bits for large scale image search. In K. Selçuk Candan, Sethuraman Panchanathan, Balakrishnan Prabhakaran, Hari Sundaram, Wu-chi Feng, and Nicu Sebe, editors, *Proceedings of the 19th International Conference on Multimedia 2011, Scottsdale, AZ, USA, November 28–December 1, 2011*, pages 1441–1444. ACM
13. Jegou H, Douze M and Schmid C (2008a) Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 304–317, Berlin, Heidelberg, Springer-Verlag
14. Jegou H, Douze, M and Schmid, C (2008b) Recent advances in large scale image search. In Frank Nielsen, editor, *Emerging Trends in Visual Computing, LIX Fall Colloquium, ETVC 2008, Palaiseau, France, November 18–20, 2008. Revised Invited Papers*, volume 5416 of *Lecture Notes in Computer Science*, pages 305–326. Springer
15. Jegou H, Douze M and Schmid C (2009) On the burstiness of visual elements. In 2009 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20–-25 June 2009, Miami, Florida, USA, pages 1169–1176. IEEE Computer Society
16. Jegou H, Douze M, Schmid C (2010) Improving bag-of-features for large scale image search. *Int J Comput Vis* 87(3):316–336
17. Ji, R, Xie, X, Yao, H and Wei-Ying Ma. (2009) Vocabulary hierarchy optimization for effective and transferable retrieval. In *2009 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20–25 June 2009, Miami, Florida, USA*, pages 1161–1168
18. Jiang K, Que Q and Kulis B (2015) Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015*, pages 4933–4941. IEEE
19. Ke K and Sukthankar R (2004) Pca-sift: a more distinctive representation for local image descriptors. In *Proceedings of the 2004 I.E. computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 506–513, Washington, DC, USA. IEEE Computer Society
20. Khan FS, van de Weijer J, Vanrell M (2012) Modulating shape features by color attention for object recognition. *Int J Comput Vis* 98(1):49–64
21. Khan FS, Rao MA, van de Weijer J, Bagdanov A, Lopez A, Felsberg M (2013) Coloring Action Recognition in Still Images. *Int J Comput Vis* 105(3):205–221
22. Kohonen T (1990) The self-organizing map. *Proc IEEE* 78:1464–1480

23. Liang Z, Wang S, Liu Z and Tian Q (2014) Packing and padding: Coupled multi-index for accurate image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2014 I.E. Conference on*, pages 1947–1954. IEEE
24. Liang Z, Wang S, Tian Q (2014a) Coupled binary embedding for large-scale image retrieval. *IEEE Trans Image Process* 23(8):3368–3380
25. Liang Z, Wang S, Tian Q (2014b) Lp-norm idf for scalable image retrieval. *Image Processing, IEEE Transactions On*. doi:10.1109/TIP.2014.2329182
26. Lin J, Morère O, Petta J, Chandrasekhar V and Veillard A (2015) Tiny descriptors for image retrieval with unsupervised triplet hashing. *CoRR*, abs/1511.03055
27. Liu X, Lou Y, Yu AW and Lang B (2011) Search by mobile image based on visual and spatial consistency. In *Proceedings of the 2011 I.E. International Conference on Multimedia and Expo, ICME 2011, 11–15 July, 2011, Barcelona, Catalonia, Spain*, pages 1–6
28. Liu Z, Li H, Zhou W and Tian Q (2012) Embedding spatial context information into inverted file for large-scale image retrieval. In *Proceedings of the 20th ACM Multimedia Conference, MM '12, Nara, Japan, October 29–November 02, 2012*, pages 199–208
29. Liu Z, Wang S, Zheng L and Tian Q (2014) Visual reranking with improved image graph. In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4—9, 2014, pages 6889–3893. IEEE
30. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
31. MacQueeen JB (1967) Some methods for classification and analysis of multivariate observations. In LM Le Cam and J Neyman editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press
32. Mikolajczyk K, Schmid C (2004) Scale & affine invariant interest point detectors. *Int J Comput Vis* 60(1):63–86
33. Niblack W, Barber R, Equitz W, Flickner M, Glasman EH, Petkovic D, Yanker P, Faloutsos C and Taubin G (1993) The qbic project: Querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 173–187
34. Nister D and Stewenius, H (2006) Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 I.E. Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA, IEEE Computer Society
35. Norouzi, M and Fleet, DJ (2011) Minimal loss hashing for compact binary codes. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, 2011*, pages 353–360. Omnipress
36. Ogle VE, Stonebraker M (1995) Chabot: Retrieval from a relational database of images. *IEEE Comput* 28(9):40–48
37. Philbin J, Chum O, Isard M, Sivic J and Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*
38. Philbin J, Chum O, Isard M, Sivic J and Zisserman A (2008) Lost in quantization: Improving particular object retrieval in large scale image databases. In 2008 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24—26 June 2008, Anchorage, Alaska, USA. IEEE Computer Society
39. Shen X, Lin, Z, Brandt, J, Avidan, S and Wu, Y (2012) Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In 2012 I.E. Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16—21, 2012, pages 3013–3020. IEEE Computer Society
40. Sivic, J and Zisserman, A (2003) Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA, IEEE Computer Society
41. Tolias G, Jégou H (2014) Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recogn* 47(10):3466–3476
42. van de Sande K, Gevers T, Snoek C (2010) Evaluating color descriptors for object and scene recognition. *IEEE Trans Pattern Anal Mach Intell* 32(9):1582–1596
43. van de Weijer J, Gevers T, Bagdanov AD (2006) Boosting color saliency in image feature detection. *IEEE Trans Pattern Anal Mach Intell* 28(1):150–156
44. Wang X, Yang M Cour T, Zhu S, Yu K and Han TX (2011) Contextual weighting for vocabulary tree based image retrieval. In DN Metaxas, L Quan, A Sanfeliu and LJ Van Gool, editors, *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6–13, 2011*, pages 209–216. IEEE
45. Wang J, Wang J, Ke Q, Zeng G and Li S (2013) Fast approximate k-means via cluster closures. *CoRR*, abs/1312.3061
46. Weiss Y, Torralba, A and Fergus, R (2008) Spectral hashing. In D Koller, D Schuurmans, Y Bengio, and L Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8–11, 2008*, pages 1753–1760. Curran Associates, Inc.

47. Wengert C, Douze, M and Jégou, H (2011) Bag-of-colors for improved image search. In *Proceedings of the 19th International Conference on Multimedia 2011, Scottsdale, AZ, USA, November 28–December 1, 2011*, pages 1437–1440
48. Yanai K (2005) Image collector ii: A system to gather a large number of images from the web. *IEICE Trans* 88-D(10):2432–2436
49. Yun F, Cao L, Guo G and Huang TS (2008) Multiple feature fusion by subspace learning. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, CIVR '08, pages 127–134, New York, NY, USA, ACM
50. Zhang S, Yang M, Cour T, Yu K and Metaxas DN (2012) Query specific fusion for image retrieval. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part II*, pages 660–673
51. Zhang S, Yang M, Wang X, Lin Y and Tian Q (2013) Semantic-aware co-indexing for image retrieval. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1–8, 2013*, pages 1673–1680. IEEE
52. Zhou W, Li, H, Lu, Y and Tian, Q (2013) Sift match verification by geometric coding for large-scale partial-duplicate web image search. *ACM Trans Multimed Comput Commun Appl*, 9(1):4:1–4:18
53. Zhou W, Li H, Lu Y, Wang M, Tian Q (2015) Visual word expansion and BSIFT verification for large-scale image search. *Multimedia Systems* 21(3):245–254