

Packing and Padding: Coupled Multi-index for Accurate Image Retrieval

Liang Zheng¹, Shengjin Wang¹, Ziqiong Liu¹, and Qi Tian²

¹State Key Laboratory of Intelligent Technology and Systems;

¹Tsinghua National Laboratory for Information Science and Technology;

¹Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

²University of Texas at San Antonio, TX, 78249, USA

zheng-106@mails.tsinghua.edu.cn wsgsj@tsinghua.edu.cn

liuziqiong@ocrserv.ee.tsinghua.edu.cn qitian@cs.utsa.edu

Abstract

In Bag-of-Words (BoW) based image retrieval, the SIFT visual word has a low discriminative power, so false positive matches occur prevalently. Apart from the information loss during quantization, another cause is that the SIFT feature only describes the local gradient distribution. To address this problem, this paper proposes a coupled Multi-Index (c-MI) framework to perform feature fusion at indexing level. Basically, complementary features are coupled into a multi-dimensional inverted index. Each dimension of c-MI corresponds to one kind of feature, and the retrieval process votes for images similar in both SIFT and other feature spaces. Specifically, we exploit the fusion of local color feature into c-MI. While the precision of visual match is greatly enhanced, we adopt Multiple Assignment to improve recall. The joint cooperation of SIFT and color features significantly reduces the impact of false positive matches.

Extensive experiments on several benchmark datasets demonstrate that c-MI improves the retrieval accuracy significantly, while consuming only half of the query time compared to the baseline. Importantly, we show that c-MI is well complementary to many prior techniques. Assembling these methods, we have obtained an mAP of 85.8% and N-S score of 3.85 on Holidays and Ukbench datasets, respectively, which compare favorably with the state-of-the-arts.

1. Introduction

This paper considers the task of near duplicate image retrieval in large scale databases. Specifically, given a query image, our goal is to find all images sharing similar appearance in real time.

Many state-of-the-art image retrieval systems rely on the



Figure 1. Three examples of image retrieval from Ukbench (**Top** and **Middle**) and Holidays (**Bottom**) datasets. For each query (left), results obtained by the baseline (the first row) and c-MI (the second row) are demonstrated. The retrieval results start from the second image in the rank list.

Bag-of-Words (BoW) representation. In this model, local features such as the SIFT descriptor [10] are extracted and quantized to *visual words* using a pre-trained *code-book*. Typically, each visual word is weighted using the *tf-idf* scheme [20, 28]. Then, an *inverted index* is leveraged to reduce computational burden and memory requirements, enabling fast online retrieval.

One crucial aspect in the BoW model concerns visual matching between images based on visual words. However, the reliance on the SIFT feature leads to an ignorance of other characteristics, such as color, of an image. This prob-

lem, together with the information loss during quantization, leads to many false positive matches and thus compromises the retrieval accuracy.

To enhance the discriminative power of SIFT visual words, we present a coupled Multi-Index (c-MI) framework to perform local feature fusion at indexing level. To the best of our knowledge, it is the first time that a multi-dimensional inverted index is employed in the field of image retrieval. Particularly, this paper “couples” SIFT and color features into a multi-index [1], so that efficient yet effective image retrieval can be achieved. The final system of this paper consists of “packing” and “padding” modules.

In the “packing” step, we construct the coupled Multi-Index by taking each of the SIFT and color features as one dimension of the multi-index. Therefore, the multi-index becomes a joint cooperation of two heterogeneous features. Since each SIFT descriptor is coupled with a color feature, its discriminative power is greatly enhanced. On the other hand, to improve recall, Multiple Assignment (MA) is employed. Particularly, to make c-MI more robust to illumination changes, we adopt a large MA value on the side of color feature. Fig. 1 presents three sample retrieval results of our method. We observe that c-MI improves the retrieval accuracy and returns some challenging results.

Moreover, in the “padding” step, we further incorporate some prior techniques to enhance retrieval performance. We show in the experiments that c-MI is well compatible with methods such as rootSIFT [17], Hamming Embedding [4], burstiness weighting [5], graph fusion [25], etc. As another major contribution, we have achieved new state-of-the-art results on Holidays [4] and Ukbench [12] datasets. Namely, we obtained an mAP of 85.8% and N-S score of 3.85 on Holidays and Ukbench, respectively.

The remainder of this paper is organized as follows. After an overview of related work in Section 2, we describe the “packing” of c-MI framework in Section 3. In Section 4, the “padding” methods and results are presented and discussed. Finally, we conclude in Section 5.

2. Related Work

In the image retrieval community, a myriad of works have been proposed to improve the accuracy of image retrieval. In this section, we provide a brief review of several closely related aspects.

Matching Refinement In visual matching, a large codebook [14] typically means a high precision but low recall, while constructing a small codebook (e.g., 20K) [6] guarantees high recall. To improve precision given high recall, some works explore contextual cues of visual words, such as spatial information [14, 19, 31, 22, 2, 27]. To name a few, Shen et al. [19] perform image retrieval and localization simultaneously by a voting-based method. Alternatively, Wang et al. [22] weight visual matching based on the

local spatial context similarity. Meanwhile, the precision of visual matching can be also improved by embedding binary features [4, 23, 32, 9]. Specifically, methods such as Hamming Embedding [4] rebuild the discriminative ability of visual words by projecting SIFT descriptor into binary features. Then, efficient *xor* operation between binary signatures is employed, providing further check of visual matching.

Feature Fusion The fusion of multiple cues has been proven to be effective in many tasks [18, 30, 13]. Since the SIFT descriptor used in most image retrieval systems only describes the local gradient distribution, feature fusion can be performed to capture complementary information. For example, Wengert et al. [23] embed local color feature into the inverted index to provide local color information. To perform feature fusion between global and local features, Zhang et al. [25] combine BoW and global features by graph fusion and maximizing weighted density, while co-indexing [26] expands the inverted index according to global attribute consistency.

Indexing Strategy The inverted index [20] significantly promotes the efficiency of BoW based image retrieval. Motivated from text retrieval framework, each entry in the inverted index stores information associated with each indexed feature, such as image IDs [14, 26], binary features [4, 23], etc. Recent state-of-the-art works include joint inverted index [24] which jointly optimizes all visual words in all codebooks. The closest inspiring work to ours includes the inverted multi-index [1] which addresses NN search problem by “de-composing” the SIFT vector into different dimensions of the multi-index. Our work departs from [1] in two aspects. First, the problem considered in this paper consists in the indexing level feature fusion, applied in the task of large scale image retrieval. Second, we actually “couple” different features into a multi-index, after which the “coupled Multi-Index (c-MI)” is named.

3. Proposed Approach

This section gives a formal description of the proposed c-MI framework.

3.1. Conventional Inverted Index Revisit

A majority of works in the BoW based image retrieval community employ a ONE-dimensional inverted Index [28, 14, 12], in which each entry corresponds to a visual word defined in the codebook of SIFT descriptor. Assume that a total of N images are contained in an image database, denoted as $\mathcal{D} = \{I_i\}_{i=1}^N$. Each image I_i has a set of local features $\{x_j\}_{j=1}^{d_i}$, where d_i is the number of local features. Given a codebook $\{w_i\}_{i=1}^K$ of size K , a conventional 1-D inverted index is represented as $\mathcal{W} = \{W_1, W_2, \dots, W_K\}$. In \mathcal{W} , each entry W_i contains a list of indexed features, in which image ID, TF score, or other metadata [4, 31, 22] are

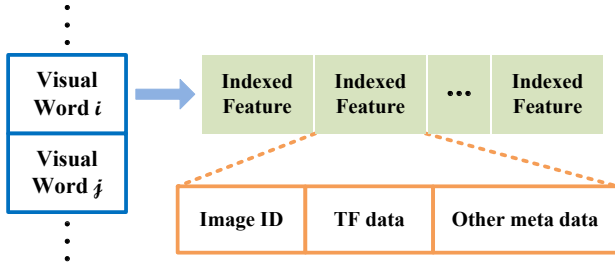


Figure 2. Conventional 1-D inverted Index. Only one kind of feature (typically the SIFT feature) is used to build the inverted Index.

stored. An example of the conventional inverted Index is illustrated in Fig. 2.

Given a query feature, an entry W_i in the inverted index is identified after feature quantization. Then, the indexed features are taken as the candidate nearest neighbors of the query feature. In this scenario, the matching function $f_q(\cdot)$ of two local features x and y is defined as

$$f_q(x, y) = \delta_{q(x), q(y)}, \quad (1)$$

where $q(\cdot)$ is the quantization function mapping a local feature to its nearest centroid in the codebook, and δ is the Kronecker delta response.

The 1-D inverted index votes for candidate images similar to the query in *one* feature space, typically the SIFT descriptor space. However, the intensity-based features are unable to capture other characteristics of a local region. Moreover, due to the quantization artifacts, the SIFT visual word is prone to producing false positive matches: local patches, similar or not, may be mapped to the same visual word. Therefore, it is undesirable to take visual word as the only ticket to feature matching. While many previous works use spatial contexts [31, 22] or binary features [4] to filter out false matches, our work, instead, proposes to incorporate local color feature to provide additional discriminative power via the coupled Multi-Index (c-MI).

3.2. Feature Extraction and Quantization

This paper considers the coupling of SIFT and color features. The primary reason lies in that feature fusion works better for features with low correlation, such as SIFT and color. In feature matching, complementary information may be of vital importance. For example, given two keypoints quantized to the same SIFT visual word, if the coupled color features are largely different, they may be considered to be a false match (see Fig. 3 for an illustration). To this end, SIFT and color features are extracted and subsequently quantized as follows.

SIFT extraction: Scale-invariant keypoints are detected with detectors, e.g. DoG [10], Hessian-affine [14], etc. Then, a 16×16 patch around each keypoint is considered, from which a 128-dimensional SIFT vector is calculated.

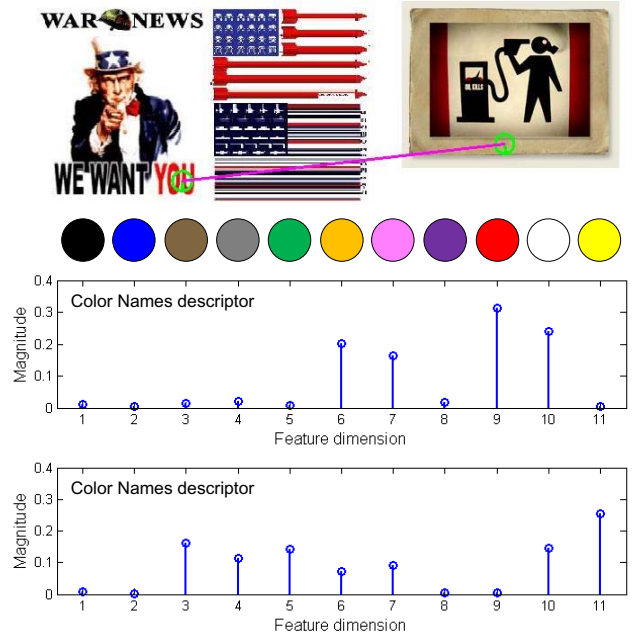


Figure 3. An example of visual match. **Top:** A matched SIFT pair between two images. The Hamming distance between their 64-D SIFT Hamming signatures is 12. The 11-D color name descriptors of the two keypoints in the left (**middle**) and right (**bottom**) images are presented below. Also shown are the prototypes of the 11 basic colors (colored discs). In this example, the two local features are considered as a good match both by visual word equality and Hamming distance consistency. However, they differ a lot in color space, thus considered as a false positive match in c-MI.

Color extraction: we employ the Color Names (CN) descriptor [18]. CN assigns a 11-D vector to each pixel, in which each entry encodes one of the eleven basic colors: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow. Around each detected keypoint, we consider a local patch with an area proportional to the scale of the keypoint. Then, CN vectors of each pixel in this area are calculated. We take the mean CN vector as the color descriptor coupling SIFT for the current keypoint.

Quantization For SIFT and CN descriptors, we use the conventional quantization scheme as in [14]. Codebooks are trained using independent SIFT and CN descriptors, respectively. Each descriptor is quantized to the nearest centroid in the corresponding codebook by Approximate Nearest Neighbor (ANN) algorithm. To improve recall, Multiple Assignment (MA) is applied. Particularly, to deal with the illumination variations, MA is set large for CN feature.

Binary signature calculation In order to reduce quantization error, we calculate binary signatures from original descriptors. For a SIFT descriptor, we follow the method proposed in [4], resulting in a 64-D binary signature.

Nevertheless, on the side of color feature, since each dimension of the CN descriptor has explicit semantic mean-

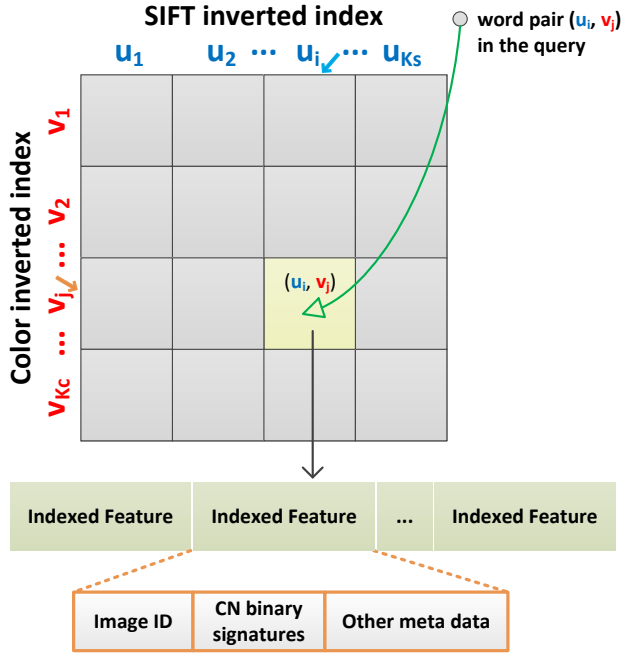


Figure 4. Structure of c-MI. The codebook sizes are K_s and K_c for SIFT and color features, respectively. During online retrieval, the entry of word tuple (u_i, v_j) is checked.

ing, we employ the binarization scheme introduced in [32]. Specifically, given a CN descriptor represented as $(f_1, f_2, \dots, f_{11})^T$, a 22-bit binary feature \mathbf{b} can be produced as follows

$$(b_i, b_{i+11}) = \begin{cases} (1, 1), & \text{if } f_i > \hat{t}h_1, \\ (1, 0), & \text{if } \hat{t}h_2 < f_i \leq \hat{t}h_1, \\ (0, 0), & \text{if } f_i \leq \hat{t}h_2 \end{cases} \quad (2)$$

where $b_i (i = 1, 2, \dots, 11)$ is the i th entry of the resulting binary feature \mathbf{b} . Thresholds $\hat{t}h_1 = g_2$, $\hat{t}h_2 = g_5$, where $(g_1, g_2, \dots, g_{11})^T$ is the sorted vector of $(f_1, f_2, \dots, f_{11})^T$ in descending order.

3.3. Coupled Multi-Index

Structure of c-MI In [1], the 128-D SIFT descriptor is *de-composed* into several blocks produced by product quantization [7]. The multi-index is thus organized around the codebooks of corresponding blocks. Their approach enables more accurate nearest neighbor (NN) search for SIFT features. In our work, however, we consider the task of image retrieval, which differs from pure NN search. Moreover, contrary to [1] we *couple* different features into a multi-index, so that feature fusion is performed at indexing level. In this paper, we consider the 2-D inverted index, which is also called *second-order* in [1].

Let $\vec{x} = [x^s, x^c] \in \mathcal{R}^{D_s+c}$ be a coupled feature descriptor at keypoint p , where $x^s \in \mathcal{R}^{D_s}$, $x^c \in \mathcal{R}^{D_c}$ are

SIFT and color descriptors of dimension D_s and D_c , respectively. For c-MI, two codebooks are trained for each feature. Specifically, for SIFT and color descriptors, codebooks $\mathcal{U} = \{u_1, u_2, \dots, u_{K_s}\}$ and $\mathcal{V} = \{v_1, v_2, \dots, v_{K_c}\}$ are generated, where K_s and K_c are codebook sizes, respectively. As a consequence, c-MI consists of $K_s \times K_c$ entries, denoted as $\mathcal{W} = \{W_{11}, W_{12}, \dots, W_{ij}, \dots, W_{K_s K_c}\}$, $i = 1, \dots, K_s, j = 1, \dots, K_c$, as illustrated in Fig. 4.

When building the multi-index, all feature tuples $\vec{x} = [x^s, x^c]$ are quantized into visual word pairs $(u_i, v_j), i = 1, \dots, K_s, j = 1, \dots, K_c$ using codebooks \mathcal{U} and \mathcal{V} , so that u_i and v_j are the nearest centroids to features x^s and x^c in codebooks \mathcal{U} and \mathcal{V} , respectively. Then, in the entry W_{ij} , information (e.g. image ID, CN binary signatures and other meta data) associated with the current feature tuple \vec{x} is stored continuously in memory.

Querying c-MI Given a query feature tuple $\vec{x} = [x^s, x^c]$, we first quantize it into a visual word pair (u_i, v_j) as in the offline phase. Then, the corresponding entry W_{ij} in c-MI is identified, and the list of indexed features are taken as the candidate images, similar to the classic inverted index described in Section 3.1. In essence, the matching function $f_{q_s, q_c}^0(\cdot)$ of two local feature tuples $\vec{x} = [x^s, x^c]$ and $\vec{y} = [y^s, y^c]$ is written as

$$f_{q_s, q_c}^0(\vec{x}, \vec{y}) = \delta_{q_s(x^s), q_s(y^s)} \cdot \delta_{q_c(x^c), q_c(y^c)}, \quad (3)$$

where $q_s(\cdot)$ and $q_c(\cdot)$ are quantization functions for SIFT and CN features, respectively, and δ is the Kronecker delta response as in Eq. 1. As a consequence, a local match is valid only if the two feature tuples are similar both in SIFT and color feature spaces.

Moreover, the Inverse Document Frequency (IDF) scheme can be applied in the multi-index directly. Specifically, the IDF value of entry W_{ij} is defined as

$$idf(i, j) = \frac{N}{n_{ij}}, \quad (4)$$

where N is the total number of images in the database, and n_{ij} encodes the number of images containing the visual word pair (u_i, v_j) . Furthermore, the l_2 normalization can be also adopted in the 2-D case. Let an image be represented as a 2-D histogram $\{h_{i,j}\}, i = 1, \dots, K_s, j = 1, \dots, K_c$, where $h_{i,j}$ is the term-frequency (TF) of visual word pair (u_i, v_j) in image I , the l_2 norm is calculated as,

$$\|I\|_2 = \left(\sum_{i=1}^{K_s} \sum_{j=1}^{K_c} h_{i,j}^2 \right)^{\frac{1}{2}}. \quad (5)$$

Since our multi-index structure mainly works by achieving high precision, we employ Multiple Assignment (MA) to improve recall. To address illumination variations, we set

a relatively large value to the color feature. In our experiments, we find that l_2 -normalization produces slightly higher performance than Eq. 5, which is probably due to the asymmetric structure of the coupled multi-index.

Furthermore, to enhance the discriminative power of CN visual words, we incorporate color Hamming Embedding (HE^c) into c-MI. Two feature tuples are considered as a match *iff* Eq. 3 is satisfied *and* the Hamming distance d_b between their binary signatures is below a pre-defined threshold κ . The matching strength is defined as $\exp(-\frac{d_b^2}{\sigma^2})$. Therefore, the matching function in Eq. 3 is updated as

$$f_{q_s, q_c}(\vec{x}, \vec{y}) = \begin{cases} f_{q_s, q_c}^0(\vec{x}, \vec{y}) \cdot \exp\left(-\frac{d_b^2}{\sigma^2}\right), & d_b < \kappa, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Then, in the framework of c-MI, the similarity score between a database image I and query image Q is defined as

$$sim(Q, I) = \frac{\sum_{\vec{x} \in Q, \vec{y} \in I} f_{q_s, q_c}(\vec{x}, \vec{y}) \cdot idf^2}{\|Q\|_2 \|I\|_2}, \quad (7)$$

4. Experiments

In this section, we evaluate the proposed method on five public available datasets: the Ukbench [12], Holidays [4], DupImage [31], Mobile [22] and MIR Flickr 1M [11].

4.1. Datasets

Ukbench A total of 10200 images are contained in this dataset, divided into 2550 groups. Each image is taken as the query in turn. The performance is measured by the average recall of the top four ranked images, referred to as N-S score (maximum 4).

Holidays This dataset consists of 1491 images from personal holiday photos. There are 500 queries, most of which have 1-2 ground truth images. mAP (mean average precision) is employed to measure the retrieval accuracy.

DupImages This dataset is composed of 1104 images divided into 33 groups of partial-duplicate images. 108 images are selected as queries, and mAP is again used as the accuracy measurement.

Mobile The Mobile dataset has 400 database images and 2500 queries, captured by mobile devices. The Top-1 (τ_1) and Top-10 (τ_{10}) precision are employed.

MIR Flickr 1M This is a distractor dataset, with one million images randomly retrieved from Flickr. We add this dataset to test the scalability of our method.

4.2. Experiment Settings

Baseline This paper adopts the baseline in [14, 4]. Hessian Affine detector and SIFT descriptor are used for feature extraction. Following [17], rootSIFT is used on every point

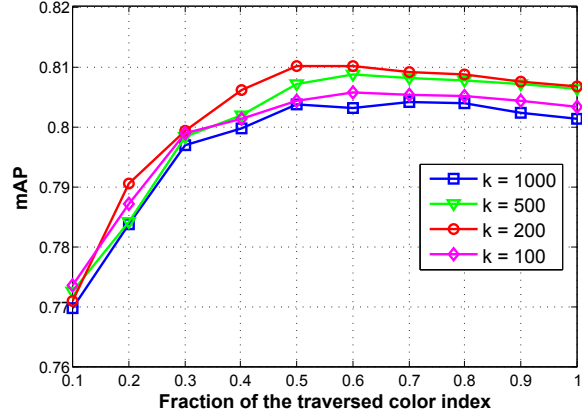


Figure 5. Impact of color codebook size on the Holidays dataset. Color codebooks of size $k = 100, 200, 500$, and 1000 are trained on independent data. The horizontal axis represents the fraction of the codebook traversed during MA^c . We observe a superior performance of the codebook of size 200 , and with $MA^c = 200 \times 0.5 = 100$. Note that the query time is halved accordingly.

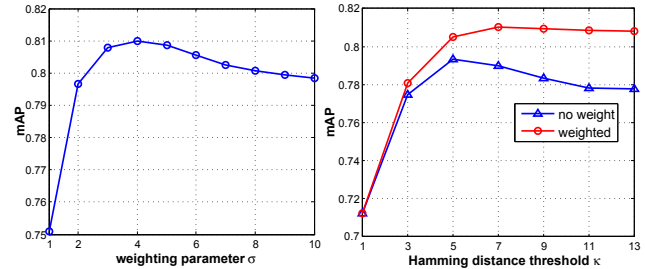


Figure 6. Influence of weighting parameter σ (left) and Hamming distance threshold κ (right) on Holidays dataset. mAP results are presented. We set σ and κ to 4 and 7 , respectively.

since it is shown to be effective under Euclidean distance. We also adopt the average IDF defined in [28] in place of the original IDF. Then, a codebook of size $20K$ is trained using the independent Flickr60k data released in [4].

SIFT Hamming Embedding The 64-bit SIFT Hamming Embedding (HE^s) [5] is used in addition to c-MI. The Hamming threshold is set to 30 , and the weighting parameter is set to 16 . Moreover, we employ the SIFT Multiple Assignment (MA^s) [4] scheme on the query side, in which a SIFT descriptor is assigned to 3 visual words.

Graph Fusion As a post-processing step, we implemented the graph fusion algorithm proposed in [25]. We extract 1000 -D global HSV histogram for each image, followed by L_1 normalization and square scaling, similar to rootSIFT [17]. Rank lists obtained by c-MI and the HSV histogram are merged, yielding new ranking results.

4.3. Parameter Analysis

Color Codebook Size and MA We extract CN descriptors from independent images and train codebooks of var-

Methods	c-MI	Burst ^s	HE ^s	MA ^s	Ukbench		Holidays	DupImage	Mobile	
					N-S	mAP(%)	mAP(%)	mAP(%)	τ_1 (%)	τ_{10} (%)
BoW					3.11	78.89	50.10	48.61	52.88	80.08
BoW	×				3.43	88.45	63.32	56.28	67.20	86.64
BoW	×	×			3.52	90.35	66.38	59.87	71.24	87.76
BoW	×		×		3.64	92.96	81.00	85.10	94.60	98.56
BoW	×	×	×		3.69	94.00	82.14	86.63	96.48	98.64
BoW	×	×	×	×	3.71	94.66	84.02	87.60	96.04	98.76

Table 1. Results for four datasets and for different methods: coupled Multi-Index (c-MI), burstiness weighting (Burst^s), SIFT Hamming Embedding (HE^s), and Multiple Assignment (MA^s). Parameters are selected as in Section 4.2 and 4.3.

ious sizes, i.e., $k = 100, 200, 500, 1000$. During color quantization Multiple Assignment (MA^c) is employed, and we vary the number of assigned words as a percentage of the codebook size. We present the mAP results on Holidays dataset in Fig. 5. It is shown that the codebook of size 200 performs favorably. Moreover, assigning 50% visual words in the codebook has a better performance, so $MA^c = 200 \times 50\% = 100$ for color feature. Intuitively, since large variation in color is often observed due to illumination, a large value of MA^c helps to improve recall. Also note that (HE^s) with default parameters is used in parameter analysis.

Color Hamming Embedding Two parameters are involved in color Hamming Embedding: the Hamming distance threshold κ and weighting factor σ . Fig. 6 demonstrates the mAP results on Holidays dataset obtained by varying the two parameters. In Fig. 6(a), the mAP first rises to the peak at $\sigma = 4$ and then slowly drops. From Fig. 6(b), the best performance is achieved at $\kappa = 7$, no matter the weighted distance is employed or not. Therefore, we set σ and κ to 4 and 7, respectively.

4.4. Evaluation

To what extent does it improve the baseline? Implemented as described in Section 4.2, the baseline results for Ukbench and Holidays are 3.11 in N-S score and 50.1% in mAP, respectively, both higher than the reported results [4, 5]. After expanding inverted index into the 2-D case as c-MI, large improvement over the baseline approach can be seen from Table 1. On Ukbench, we observe a big improvement of +0.32 in N-S score. Similarly on Holidays and Mobile datasets, the improvement is +13.2% in mAP and +14.3% in Top-1 precision. Note that the improvement is less prominent for DupImage (mAP from 48.6% to 56.3%), because the ground truth images in this dataset have a large variety in color (even gray-level images).

Complementarity to some existing methods To test whether c-MI is compatible with some prior arts used in the 1-D inverted index, we further “pad” burstiness weighting (Burst^s) [5], Hamming Embedding (HE^s) [4], Multiple

Methods	Ours	HSV	HSV*	Ours + HSV*
Ukbench, N-S	3.71	2.97	3.40	3.85
Holidays, mAP(%)	84.02	59.43	65.29	85.76

Table 2. Performance of our method combined with graph fusion on Ukbench and Holidays datasets. * denotes results obtained by HSV histogram scaled as described in Section 4.2.

Methods	Baseline	HE ^s	c-MI + HE ^s
Ukbench	2.282	1.933	1.339
Holidays	2.722	2.140	1.413
DupImage	1.900	1.391	0.885
Mobile	1.421	1.185	0.667

Table 3. Average query time (s) on Ukbench, Holidays, DupImage, and Mobile + MIR Flickr 1M datasets.

Assignment (MA^s) [4], etc., into our framework. Note that these techniques are applied on the SIFT side.

It is clear from Table 1 that these methods bring about consistent improvements. Taking Ukbench for example, the combination of Burst^s and HE^s each improves the N-S from 3.43 to 3.52, and from 3.43 to 3.64, respectively. Combining the three steps brings the result to 3.69. Then, the use of MA^s obtains the N-S of 3.71. Similar situation is observed for the other datasets. These demonstrate the feasibility of c-MI as a general framework for image retrieval.

In addition, we add a post-processing step, i.e., the graph fusion of global HSV histogram [25] to the Holidays and Ukbench datasets. Similar to the scaling method applied in rootSIFT [17], we also normalize the HSV histogram by its l_1 norm, and then exert a square root scaling. With the modified HSV histogram (HSV*), we have obtained a much higher result of HSV-based image retrieval (see Table 2). After merging the graphs constructed from the c-MI and HSV rank lists, the final result arrives at 3.85 for Ukbench and 85.8% for Holidays, respectively.

Large-scale experiments To test the scalability of our method, the four benchmark datasets are merged with various fractions of the MIR Flickr 1M images. In what follows, we mainly report three related aspects, i.e., accuracy, time efficiency, and memory cost.

Methods	Ours	[21]	[22]	[8]	[19]	[26]	[23]	[6]	[16]	[29]	[5]
<i>Ukbench</i> , N-S score	3.71	-	3.56	3.61	3.52	3.60	3.50	3.42	-	3.62	3.54
<i>Holidays</i> , mAP(%)	84.0	82.2	78.0	-	76.2	80.9	78.9	81.3	82.1	81.9	83.9

Table 5. Performance comparison with state-of-the-art methods without post-processing

Methods	Ours	[25]	[3]	[8]	[15]	[19]	[6]	[16]	[5]
<i>Ukbench</i> , N-S score	3.85	3.77	84.7	3.68	3.67	3.56	3.55	-	3.64
<i>Holidays</i> , mAP(%)	85.8	84.6	3.75	-	-	-	84.8	80.1	84.8

Table 6. Performance comparison with state-of-the-art methods with post-processing

Methods	Baseline	c-MI	HE ^s	c-MI + HE ^s
Per feature (bytes)	4	6.75	12	14.75
1M dataset (GB)	1.7	2.8	5.0	6.1

Table 4. Memory cost for different approaches.

First, we plot the image retrieval accuracy against the database size in Fig. 7. We note that when applied alone, HE^s and c-MI each brings about a significant improvement over the baseline. The reason is that both methods works by enhancing the discriminative power of visual words. Moreover, the combination of HE^s and c-MI achieves further improvements on all the four datasets. As the database is scaled up, the performance gap between c-MI and the baseline seems to become larger: the feature fusion scheme works better for large databases.

Second, the average query time for the 1M database is presented in Table 3. The experiments are performed on a server with 3.46 GHz CPU and 64GB memory. The feature extraction and quantization takes an average of 0.67s and 0.24s on the 1M dataset, respectively. From Table 3, the baseline approach is the most time-consuming, e.g. 2.28s for a query in the *Ukbench* dataset. HE^s is more efficient than the baseline due to the filtering effect of the Hamming threshold. On *Ukbench*, HE^s reduces the query time to 1.93s. Furthermore, the c-MI + HE^s method proves to be the most time efficient one. On all the four datasets, c-MI cuts the query time to about one half compared to the baseline. The reason lies in that compared with the conventional inverted index, c-MI shortens the list of indexed features per entry. Moreover, since 50% of the color index are traversed, c-MI actually halves the query time. Nevertheless, the query time can be further decreased if fewer entries are visited, i.e., at a cost of lower accuracy.

Third, we discuss the memory cost of c-MI in Table 4. For each indexed feature, 4 bytes are allocated to store image ID in the baseline. In HE^s, 8 bytes are needed to store the 64-bit binary SIFT feature. c-MI adds another 22 bits (2.75 bytes) for the binary CN signature. On the 1M dataset, the c-MI + HE^s method totally consumes 6.1 GB memory.

The above analysis indicates that c-MI is especially suitable for large scale settings: higher accuracy accompanied with less query time, and acceptable memory cost.

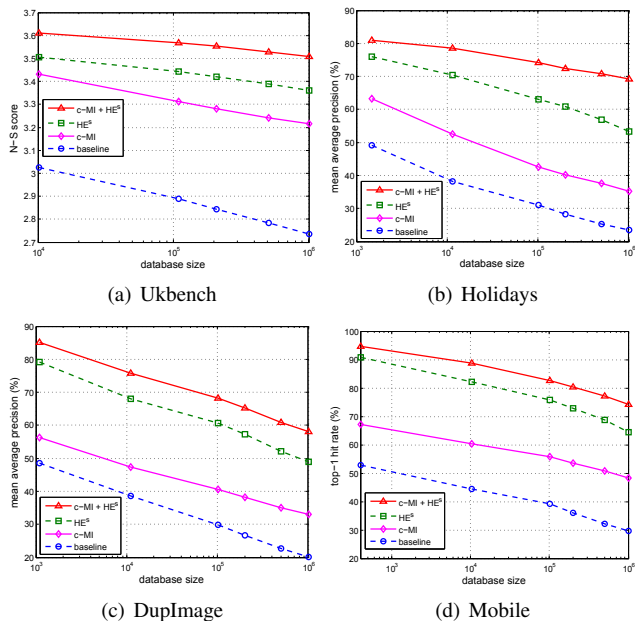


Figure 7. Image retrieval performance against the database size for BoW baseline, c-MI, HE^s, and c-MI + HE^s methods. (a) *Ukbench*, (b) *Holidays*, (c) *DupImage*, and (d) *Mobile* datasets are merged with various fractions of MIR Flickr 1M dataset.

4.5. Comparison with state-of-the-arts

We first compare our results with state-of-the-art methods which do not apply any post-processing procedure. As shown in Table 5, for *Ukbench* dataset, we achieve the best N-S score 3.71, which significantly exceeds the result reported in [8] by +0.10. For *Holidays* dataset, our result (mAP = 84.0%) also outperforms the state-of-the-art approaches. By +0.1% in mAP, our result is slightly higher than [5]. In fact, [5] also employs the inter-image burstiness weighting and weak geometric consistency, which are absent in our retrieval system.

Moreover, in Table 6, we present a comparison with results obtained by various post-processing algorithms, including RANSAC verification [14], kNN re-ranking [19], graph fusion [25], and RNN re-ranking [15], etc. We show that, followed by graph fusion [25] of modified global HSV feature (HSV*), we have set new records on both *Ukbench*

and Holidays datasets. Notably, we achieve an N-S score of 3.85 on Ukbench, and an mAP of 85.8% on Holidays, which greatly exceeds the N-S score of 3.77 [25] and mAP of 84.8% [5], respectively. We envision that other post-processing steps can also benefit from our method.

5. Conclusion

In this paper, we present a coupled Multi-Index (c-MI) framework for accurate image retrieval. Each keypoint in the image is described by both SIFT and color descriptors. Two distinct features are then *coupled* into a multi-index, each as one dimension. c-MI enables indexing-level feature fusion of SIFT and color descriptors, so the discriminative power of BoW model is greatly enhanced. To overcome the illumination changes and improve recall, a large MA is used for color feature. By further incorporating other complementary methods, we achieve new state-of-the-art performance on Holidays (mAP = 85.8%) and Ukbench (N-S score = 3.85) datasets. Moreover, c-MI is efficient in terms of both memory and time (about half compared to the baseline) costs, thus suitable for large scale settings. As another contribution, codes and data are released on our website¹.

In the future, more efforts will be made to explore the intrinsic properties of the coupled multi-index. Moreover, since c-MI can be extended to include other local descriptors, different feature selection strategies and c-MI of higher orders will be investigated.

Acknowledgement First, we would like to thank Dr. Hervé Jégou for discussion and data sharing in both this paper and [29]. This work was supported by the National High Technology Research and Development Program of China (863 program) under Grant No. 2012AA011004 and the National Science and Technology Support Program under Grant No. 2013BAK02B04. This work also was supported in part to Dr. Qi Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, and 2012 UTSA START-R Research Award respectively. This work was supported in part by National Science Foundation of China (NSFC) 61128007.

References

- [1] A. Babenko and V. Lempitsky. The inverted multi-index. In *CVPR*, 2012.
- [2] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR*, 2010.
- [3] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao. Visual reranking through weakly supervised multi-graph learning. In *ICCV*, 2013.
- [4] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.
- [5] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.
- [6] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 2010.
- [7] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128, 2011.
- [8] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *PAMI*, 32(1):2–11, 2010.
- [9] Z. Liu, H. Li, L. Zhang, W. Zhou, and Q. Tian. Cross-indexing of binary sift codes for large-scale image search. *TIP*, 23(5):2047–2057, 2014.
- [10] D. G. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2004.
- [11] B. T. Mark J. Huiskes and M. S. Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *ACM MIR*, 2010.
- [12] D. Niester and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.
- [13] Z. Niu, G. Hua, X. Gao, and Q. Tian. Context aware topic model for scene recognition. In *CVPR*, 2012.
- [14] J. Philbin, O. Chum, M. Isard, and A. Zisserman. Object retrieval with large vocabularies and fast spital matching. In *CVPR*, 2007.
- [15] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011.
- [16] D. Qin and C. W. L. van Gool. Query adaptive similarity for large scale object retrieval. In *CVPR*, 2013.
- [17] A. Relja and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.
- [18] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *CVPR*, 2012.
- [19] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 2012.
- [20] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV*, 2003.
- [21] G. Toliás, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013.
- [22] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV*, 2011.
- [23] C. Wengert, M. Douze, and H. Jégou. Bag-of-colors for improved image search. In *ACM MM*, 2011.
- [24] Y. Xia, K. He, F. Wen, and J. Sun. Joint inverted index. In *ICCV*, 2013.
- [25] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, 2012.
- [26] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian. Semantic-aware co-indexing for near-duplicate image retrieval. In *ICCV*, 2013.
- [27] L. Zheng and S. Wang. Visual phraselet: Refining spatial constraints for large scale image search. *Signal Processing Letters, IEEE*, 20(4):391–394, 2013.
- [28] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm idf for large scale image search. In *CVPR*, 2013.
- [29] L. Zheng, S. Wang, W. Zhou, and Q. Tian. Bayes merging of multiple vocabularies for scalable image retrieval. In *CVPR*, 2014.
- [30] Y. Zheng, Y.-J. Zhang, and H. Larochelle. Topic modeling of multimodal data: an autoregressive approach. In *CVPR*, 2014.
- [31] W. Zhou, H. Li, Y. Lu, and Q. Tian. Sift match verification by geometric coding for large-scale partial-duplicate web image search. *ACM TOMCCAP*, 9(1):4, 2013.
- [32] W. Zhou, Y. Lu, H. Li, and Q. Tian. Scalar quantization for large scale image search. In *ACM MM*, 2012.

¹<http://www.liangzheng.com.cn>