# A New Three-dimensional Integrated Multi-index Method for CBIR System

**Mingzhu Zhang***
School of Technology and Engineering, Xi'an Fanyi University
Xi'an , Shaanxi Province 710105, China
[e-mail: mingzhusee@163.com]
*Corresponding author: Mingzhu Zhang

*Received October 20, 2020; revised December 29, 2020; accepted February 23, 2021; published March 31, 2021*

## *Abstract*

This paper proposes a new image retrieval method called the 3D integrated multi-index to fuse SIFT (Scale Invariant Feature Transform) visual words with other features at the indexing level. The advantage of the 3D integrated multi-index is that it can produce finer subdivisions in the search space. Compared with the inverted indices of medium-sized codebook, the proposed method increases time slightly in preprocessing and querying. Particularly, the SIFT, contour and colour features are fused into the integrated multi-index, and the joint cooperation of complementary features significantly reduces the impact of false positive matches, so that effective image retrieval can be achieved. Extensive experiments on five benchmark datasets show that the 3D integrated multi-index significantly improves the retrieval accuracy. While compared with other methods, it requires an acceptable memory usage and query time. Importantly, we show that the 3D integrated multi-index is well complementary to many prior techniques, which make our method compared favorably with the state-of-the-arts.

***Keywords:*** Multi-index, Content-based Image Retrieval, Feature Fusion, Indexing Strategy

This work is supported in part by the National Natural Science Foundation of China Projects No.71771386 and Scientific Research Scheme of Shaanxi Provincial Education Department No. 17JK0990.

**http://doi.org/10.3837/tiis.2021.03.010** **ISSN : 1976-7277**

## 1. Introduction

Content-based image retrieval (CBIR) is demanding progressively more attention because of its rapid growth in object recognition, agriculture, biomedical domains and so on. The purpose of content-based image retrieval is to obtain similar images from a massive dataset by matching the given query images and the images in the dataset. In many approaches, invariant local features [1-3] or deep features [4, 5] are adopted to represent images, and the bag-of-feature (BOF) model [6] has been popularly used to retrieve images as a group of visual words. The visual words in the BOF model are usually weighted by TF-IDF (Term frequency-inverse document frequency) [7], where the TF is able to indicate the importance of visual words in the query image and the IDF has the capability of reflecting the discriminative abilities of visual words in the image dataset. These approaches have demonstrated excellent retrieval precision and scalability. The **Fig. 1** shows the general image retrieval framework, where the image representation (e.g., BOF model) is one of the key issues for large-scale image retrieval. In this paper, we focus on the image representation step.

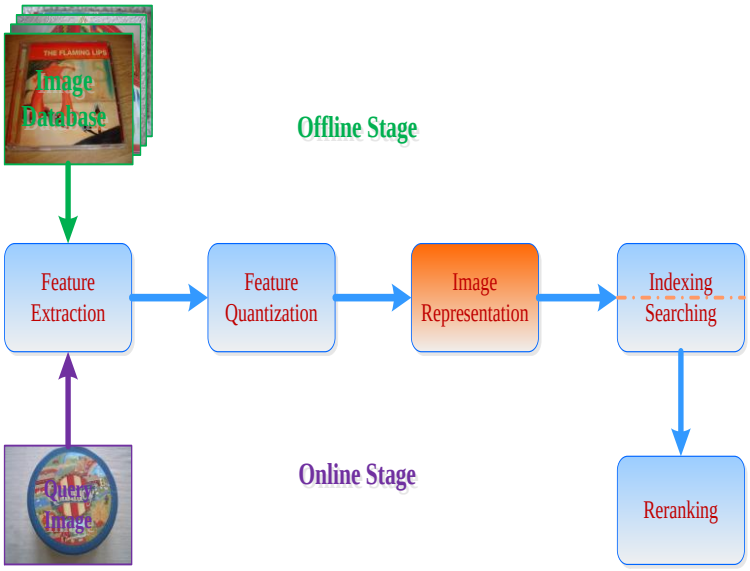


**Fig. 1.** General pipeline of the CBIR system. In this paper, we focus on the image representation step

Feature matching in the conventional BOF model is carried out implicitly by verifying whether the two feature vectors are quantified to the same visual word in the same codebook [6]. In other words, the BOF model first pre-trains numerous unique visual words using clustering algorithms (e.g. K mean, approximate K-means [8], or hierarchical K-means [9]), and then use high-dimensional histograms of visual words to describe the images. Quantification is achieved by classifying descriptor vectors into the closest visual words.

The inverted index structure is often used on BOF model to search large-scale image datasets for fast and stable image retrieval. The purpose of inverted indexes is to efficiently produce a list of database vectors that can be queried quickly. The traditional inverted indexing structure is constructed a codebook that contains a set of visual words. An inverted index stores a list of feature vectors near each codeword [10].

Given a query vector, the nearest visual word or a set of nearest visual words can be quickly identified. Then, the lists that correspond to those visual words are collected to generate a response to the query vector. Searching in inverted index could to avert calculating distances between the query feature and every database feature, which greatly accelerates the speed compared with exhaustive search method. Additionally, the memory usage of each feature point can be reduced substantially, because the inverted index needn't contain the original database vectors (over 128 dimensions) but only useful metadata (e.g., image IDs, which are just a few dimensions) to perform the search. In virtue of these efficiency benefits, inverted index structure is widely utilized in computer vision systems to implement location identification [11] or image and video search [12].

The inverted index structure essentially matches feature vectors by their quantization values. Although the proposed scheme significantly reduces the computational complexity, the codebook with large visual words trained by clustering algorithms has some drawbacks. First, it is difficult to control the error of vector quantization by using a feature quantization approach. K-means or its variants [8] are used for clustering, and the feature space is divided into a set of Voronoi cell units by fitting the allocation of training feature samples. However, the coverage of the generated Voronoi cells is different. Some features located in the edge region of the Voronoi unit will be easily misclassified. And these edge regions are always existed, so it is difficult to control the quantization errors. Second, the efficiency of an inverted index has some limitations when used for a very large database of vectors (e.g., hundreds of millions to billions). In this situation, the fine partitions of the search space are desirable in order to return a short list with vectors that contains better positioned around the query vector. Unfortunately, increasing the amount of code words (e.g., tree codebooks [9]) for finer partitions also increases the index construction time and query time. Third, the BOF model heavily depends on the performance of a local feature (e.g., SIFT). However, the reliance on one feature (e.g., SIFT) leads to an omission of other image characteristics, such as the contour and colour. These problems lead to a large number of incorrect positive matches, thereby reducing the accuracy of retrieval.

To tackle these problems, this paper proposes a new type data organization, 3D Integrated Multiple Index (I-MI) to combine local features at the indexing level. This data structure is similar in many ways to inverted indexes [6] and multiple indexes [13], and it can be used in a similar way for computer vision algorithms (see **Fig. 2**). The motivation is that the fine partition of the search space and the fusion of multiple features will be very useful to improve the performance of image retrieval. The major contributions of this paper can be summarized as follows:

- The advantage of the 3D integrated multi-indices approach is that it can produce more fine subdivisions in search space. More importantly, for large datasets the relative increase of memory usage is also small.
- Particularly, we fused SIFT with contour and colour features into the integrated multi-index. Therefore, effective and efficient image search can be realized.
- In addition, we proposed an efficient and simple algorithm to generate a series of IMI entries ordered by an increasing distance between the centroid of the corresponding entry and the given query vector.

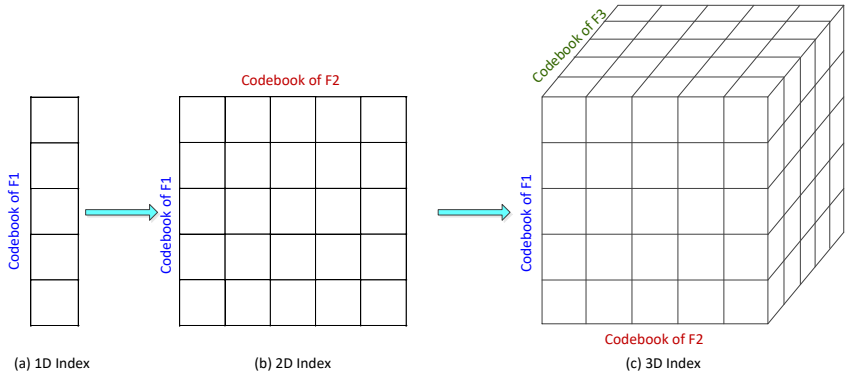


**Fig. 2.** 3D integrated multi-index construction: (a) The classic inverted index, (b) GIST coupled multi-index, (c) 3D integrated multi-index. Three features are denoted as F1, F2 and F3. For each feature, multiple codebooks are trained

Consequently, the I-MI approach results in a more faster and accurate approximate nearest neighbour (ANN) search, especially to keep the storage efficiency of the inverted index when processing large databases.

The remainder of the paper is organized as following. Section 2 discusses related work about the feature fusion, matching refinement and indexing strategy. Then, we describe our 3D integrated multi-indices framework in Section 3. In Section 4, we provide experimental results compared with those recent algorithms in terms of memory cost, efficiency and retrieval accuracy. Finally, we present the conclusions in Section 5.

## 2. Related Work

In computer vision fields, CBIR has become an important issue, having attracted considerable attention for some decades [14, 15]. In recent literatures [16-18], invariant local features are usually used for image representation. The inverted index structure [6] and the model of bag-of-features are adopted for effective large-scale image retrieval. In general, there are three necessary key components in this image-searching framework, which are include feature extraction and fusion, feature quantization and matching, and image indexing and ranking. In this section, we will review related work on these aspects.

### 2.1 Feature Fusion

Generally, one kind of feature on its own cannot adequately describe an image. The fusion of multiple features has been shown that it is very valid for many tasks [19, 20]. As the image data can be represented by a number of different features, the fusion of multiple features has become a popular research topic. However, how to identify the similarity or correlation between two observation points represented by multiple features is the key issue of multiple features fusion. The traditional multi-source fusion methods can be divided into early fusion strategies and late fusion strategies.

In early fusion strategies, the multiple features are combined at the input stage. Many image retrieval systems adopt the SIFT descriptor to present the images, but SIFT features only describe the local gradient distribution. So the early feature fusion can be used to capture complementary information. For instance, Cui et al. [21] present a method that integrates different features into a unified space in which their applications can be performed. To provide local colour information, Wengert et al. [22] embed local colour features in conventional

inverted indexes. Zheng et al. [13] propose a coupled multi-index (c-MI) method to perform feature fusion at the indexing level. Zhang et al. [15] combine BOF and global features by maximizing weighted density and graph fusion to perform feature fusion between local and global features, while according to global attribute consistency, they expand the conventional inverted index in the literature [23]. Jain et al. [24] connect all the features vector using feature-level fusion which forms a large feature vector. However, early fusion of multiple cues increases the computational complexity heavily, and the structural information of each individual feature cannot be well preserved.

In the later fusion strategies, different fusion results are separated from different features, and then the results are fused together by special algorithms. In the literature [25], the kernel-level fusion method is adopted to combine different features. In the literature [26], the authors proposed an adaptive query late fusion method for person re-identification and image search, this is an effective post fusion technique at the score level. Another methods, for example in literature [27], utilize tensors to integrate multi-features, however these approaches concentrate on transformational learning. In general, these late fusion techniques do not fully consider the correlation among the multiple features, and they are more computationally expensive for training.

### 2.2 Matching Refinement

The retrieval accuracy of vocabulary-based techniques is limited by the discriminative ability of the traditional BOF model, which can be further improved by verifying the additional distance in order to encode spatial configurations or to decrease the quantization error between matched visual words.

In general, the feature vectors are quantized as the closest visual words with the smallest distance in the feature space. If two features be quantified from different images for the same visual words, they are regarded as a match. The SIFT descriptor quantization can be performed with spatial context quantization separately or jointly to improve the retrieval accuracy. In the literature [28], a new quantization approach is presented to jointly optimize indexing to improve the accuracy of the feature match. To alleviate the quantization error, Philbin et al. [29] propose a soft quantization method, in which the soft decision strategy is used to quantify the SIFT function into several closest visual words. This soft quantization method greatly reduces the chance of false positive matches. However, to accurately find multiple nearest visual words from a large visual codebook is also an expensive computation.

Some algorithms explore contextual cues of visual words to improve the precision and recall, for example spatial information [30, 31]. To mention a few examples, Wang et al. [30] use the local spatial context similarity to weight visual matching. Meanwhile, Shen et al. [31] utilize a voting-based method to perform image localization and retrieval simultaneously. On the other hand, the Hamming embedding (HE) [32] use each of the feature points to generate a binary features to verify the feature matching, and the precision of image matching and retrieval can also be improved by embedding the binary features [22, 32]. Generally, these improvements make retrieval methods capable of finding candidate images with more reliable matches to the query images (e.g., more consistent spatial neighbours or less matching errors).

### 2.3 Indexing Strategy

Indexing and hashing are often used to improve the retrieval efficiency. For example, Snoek et al. [33] uses geometric hashing to build dataset indices, while Cha GuangHo [34] perform indexing using a tree structure of matching criteria. In literature [6], Sivic and Zisserman proposed a video Google system , which uses BOF retrieval method to retrieve videos from the

dataset. In literature [35], Liu et al. reviewed earlier efforts in CBIR, mostly relying on local feature-based matching methods.

Recently, BOF based matching method has been widely used in image retrieval system. The inverted index and modified inverted indices significantly promote the efficiency of BOF-based image retrieval. From a methodological point of view, we can divide these inverted indices into four categories, IDF (Inverse Document Frequency), M-IDF (Multi-index Document Frequency), C-MI (Couple multi-index) and G-MI (Global multi-index). The IDF methods use TF-IDF [7] to weight each visual word (e.g., [6, 32, 36]). The M-IDF method uses product quantization instead of vector quantization in the inverse index [28], such as the methods [10, 37]. To name a few, Babenco et al. [10] address the ANN problem by the inverse multiple index based on product quantization. In the C-MI category methods [13, 38], they couple color and SIFT features into multiple indexes to perform feature fusion implicitly at the index level, which greatly improves retrieval efficiency. For the G-MI category methods (e.g., [23, 39]), according to the consistency of the global feature space extend inverted index. In our experiments, we will compare the performance of the proposed methods with the modified reversed indices [10, 13, 32].

## 3. Integrated Multi-index

The section gives a formal description of the presented 3D integrated multi-index framework. The construction of the 3D integrated multi-index can be seen in **Fig. 2**. Concisely, the 3D integrated multi-index is consists of two stages. First, a two-dimensional GIST (Spatial envelope feature) [40] coupled multi-index is constructed, in which each dimension corresponds to a distinct codebook. In the second stage, the colour feature (CN) [41] is integrated into the 2D index. This article uses the Colour name feature have two reasons. First, it is a pure colour feature, which is useful for late fusion with SIFT and GIST features. Second, it is shown in [42] that the Colour names feature has superior performance compared with several frequently used colour features.

### 3.1 Structure of the GIST-coupled Multi-index (G-MI)

Let $x_i = \left[ x_i^S, x_i^G \right] \in R^{D_{S+G}}$ be a combined descriptor vector at feature point $p$, where $x_i^S \in R^{D_S}$ , $x_i^G \in R^{D_G}$ are SIFT and GIST descriptors of dimension $D_S$ and $D_G$ , respectively. Compared with the inverted multi-index and other BOF-based image retrieval systems, the GIST-coupled multi-index performs better because the correlation between $D_S = \left\{ x^S \right\}$ and $D_G = \left\{ x^G \right\}$ is lower and the amount of variance within $D_S$ and $D_G$ are very close. For GIST and SIFT vectors, using them directly into 64 dimensions and 128 dimensions appears to be a near-optimal policy, and in other situations the dimensions may be rearranged to reduce the correlation to balance the variances between the two descriptors.

The codebooks used for the GIST-coupled multi-index are generated independently by K-means algorithm clustering of the datasets $D_S$ and $D_G$ , producing the codebooks $U = \left\{ u_1, u_2, ..., u_k \right\}$ for the SIFT vectors and $V = \left\{ v_1, v_2, ..., v_k \right\}$ for the GIST vectors. Then, the feature quantization steps are put into effect on the database vectors, with the result that the $K^2$ lists which correspond to all possible pairs of codewords $(u_i, v_j)$ , $i$=1 ,…, $K$, $j$=1 ,…, $K$

are obtained. Given the closest point $[u_i, v_j]$, the 2-D inverted index (GIST-coupled multi-index) is denoted as $W = \{W_{11}, W_{12}, ..., W_{ij} ..., W_{KK}\}$, in which each entry $W_{ij}$ is defined as:

$$W_{ij} = \{x = \left[ x^S, x^G \right] \in R^{D_{S+G}} \mid i = \arg\min_k d_1(x_i^S, u_k) \wedge j = \arg\min_k d_2(x_i^G, v_k)\} \tag{1}$$

Note that the data in each list $W_{ij}$ is a Cartesian set of the two Voronoi cells in $R^{D_S}$ and $R^{D_G}$ that are induced by $d$, consequently

$$\forall a, b \in R^{D_{S+G}} : d(a,b) = d_1(a^1, b^1) + d_2(a^2, b^2) \tag{2}$$

The most important and simplest case is setting $d, d_1$ and $d_2$ as Euclidean distances in the respective spaces, consequently the GIST coupled multi-index can be applied to match feature vectors with the Euclidean distance from the query.

Querying on the GIST-coupled multi-index proceeds as follows. Given a query feature vector $x = \left[ x^S, x^G \right]$, we first quantize it into a codeword pair ($u_i$, $v_j$). Then, the corresponding entry $W_{ij}$ in the GIST-coupled multi-index is found, while the list of database feature vectors are taken as the candidate features, which is similar to the conventional inverted index method. Therefore, the matching function $f^0_{q_1,q_2}(\cdot)$ of two local feature vectors $x = \left[ x^S, x^G \right]$ and $y = \left[ y^S, y^G \right]$ is defined as

$$f^0_{q_1,q_2}(x, y) = \delta_{q_1(x^S), q_1(y^S)} \cdot \delta_{q_2(x^G), q_2(y^G)} \tag{3}$$

where $\delta$ is the Kronecker delta response, and $q_1(\cdot)$ and $q_2(\cdot)$ are quantization functions.

### 3.2 Constructing 3D Integrated Multi-index (I-MI)

In multi-index [10], the 128 dimensional SIFT descriptor is decomposed into two equal blocks. Then, around the corresponding blocks of codewords to organize 2D inverted index. This method enables more accurate search results for SIFT features. In contrast to the literature [10], we integrate different features into a multi-index (see **Fig. 3**) to perform the feature fusion at the indexing level. In this article, a three-dimensional inverted index based on GIST coupled multi-index is considered.

In **Fig. 3(a)** Inverted multi-index, the 128-D SIFT descriptor is divided into two segments to generate a visual word pair again. In **Fig. 3(b)** 3D Integrated multi-index, a key point is in the image by the SIFT, GIST and CN descriptors co-describing together, and then use the three yards to quantify a triple visual words.

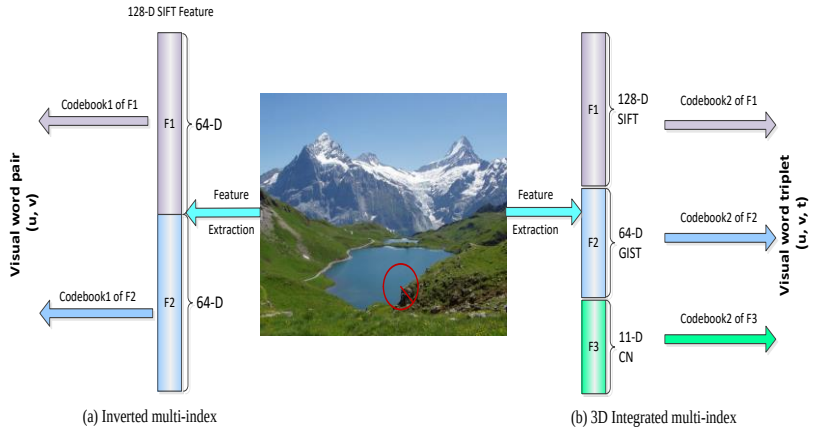


**Fig. 3.** Feature extraction and quantization for the 2D Inverted multi-index and the 3D integrated multi-index

Let $X = \left[ x^S, x^G, x^C \right] \in R^{D_{S+G+C}}$ be an integrated feature descriptor at feature point $p$, where $x^S \in R^{D_S}, x^G \in R^{D_G}, x^C \in R^{D_C}$ are SIFT, GIST and colour (CN) descriptors of dimensions $D_S$, $D_G$ and $D_C$, respectively. For I-MI, three codebooks are trained for the integrated feature. Specifically, for SIFT, GIST and colour descriptors, the SIFT codebook $U = \{u_1, u_2, ..., u_{K_S}\}$, the GIST codebook $V = \{v_1, v_2, ..., v_{K_G}\}$ and the colour codebook $T = \{t_1, t_2, ..., t_{K_C}\}$ are generated, in which $K_S$, $K_G$ and $K_C$ are codebook sizes, respectively. As a consequence, I-MI consists of $K_S \times K_G \times K_C$ entries, denoted as $W = \{W_{111}, W_{112}, ..., W_{ijk}, ..., W_{K_S K_G K_C}\}$, $i = 1, ..., K_S$, $j = 1, ..., K_G$, $k = 1, ..., K_C$, as illustrated in **Fig. 2**.

In the establishment of a three-dimensional comprehensive index, all feature vectors $X = \left[ x^S, x^G, x^C \right]$ are quantized into codeword triplets ($u_i$, $v_j$, $t_k$), $i = 1, ..., K_S$, $j = 1, ..., K_G$, $k = 1, ..., K_C$ using codebooks $U$, $V$ and $T$ so that $u_i$, $v_j$ and $t_k$ are the closest centroids to the features $x^S$, $x^G$ and $x^C$ in their respective codebooks $U$, $V$ and $T$. Like that, in the entry $W_{ijk}$, informations associated with the current feature vector $\vec{x}$, such as the image ID, TF-IDF value and other metadata, are stored continuously in memory.

### 3.3 Querying the 3D Integrated Multi-index

Given a query feature vector $X = \left[ x^S, x^G, x^C \right]$, it must be quantized into a visual word triplet ($u_i$, $v_j$, $t_k$) as in the offline stage. Then, the corresponding codebook entry $W_{ijk}$ is found in I-MI, and use the database feature vector list as candidate features at the same time, which is similar to the conventional inverted index method presented in Section 3.1. So that the matching function $f^0_{q_S, q_G, q_C}(\cdot)$ of two local feature vectors $X = \left[ x^S, x^G, x^C \right]$ and $Y = \left[ y^S, y^G, y^C \right]$ can be written as

$$f^0_{q_S, q_G, q_C}(X, Y) = \delta_{q_S(x^S), q_S(y^S)} \cdot \delta_{q_G(x^G), q_G(y^G)} \cdot \delta_{q_C(x^C), q_C(y^C)} \tag{4}$$

where $\delta$ is the Kronecker delta response and $q_S(\cdot)$, $q_G(\cdot)$ and $q_C(\cdot)$ are quantization functions. Consequently, if only the two feature vectors are similar in SIFT, GIST and CN feature spaces, the local match is valid.

Moreover, the IDF scheme (the reciprocal of document frequency) can be directly applied to 3D integrated multi-index. Specifically, we can define the IDF value of entry $W_{ijk}$ as the following

$$idf(i, j, k) = \frac{N}{n_{i\,jk}} \tag{5}$$

where $n_{ijk}$ is the number of images which contain the visual word triplet ($u_i$, $v_j$, $t_k$), and $N$ means the total number of database images. Furthermore, we can use the $l_2$ normalization in the 3D integrated multi-index. Let an image $I$ be represented as a 3D histogram $\{h_{i,j,k}\}$, $i = 1, ..., K_S$, $j = 1, ..., K_G$, $k = 1, ..., K_C$ where $h_{i,j,k}$ is the TF value (term-frequency) of the visual word triplet ($u_i$, $v_j$, $t_k$) in the image $I$, the $l_2$ norm can be calculated as,

$$\|I\|_2 = \left( \sum_{i=1}^{K_S} \sum_{j=1}^{K_G} \sum_{k=1}^{K_C} h_{i,j,k}^2 \right)^{\frac{1}{2}} \tag{6}$$

In our experiments, we find that the $l_2$ norm can yield an improvement in a slightly higher performance, which is likely because of the asymmetric structure of the integrated multi-index. Because the 3D integrated multi-index method mainly used to achieve higher accuracy, we utilize the multiple assignment (MA) [28] to further improve the recall rate. Furthermore, in order to solve the problem of illumination or contour changes, we set relatively large values for CN and GIST descriptors. Two feature vectors are considered to be matched if (5) is satisfied and their Hamming distance $d_b$ is lower than the predetermined threshold $\varepsilon$. The strength of matching is denoted as $\exp(-\frac{d_b^2}{\sigma^2})$. In this way, the matching function in (5) can be rewritten as follows.

$$f_{q_S, q_G, q_C}(X, Y) = \begin{cases} f^0_{q_S, q_G, q_C}(X, Y) \cdot \exp(-\frac{d_b^2}{\sigma^2}), & d_b < \varepsilon, \\ 0, & otherwise. \end{cases} \tag{7}$$

Then, in the framework of the 3D integrated multi-index, the similarity score between a query image $Q$ and database image $I$ is defined as

$$sim(Q, I) = \frac{\sum_{\vec{x} \in Q, \vec{y} \in I} f_{q_S, q_G, q_{C^c}}(X, Y) \cdot idf^2}{\|Q\|_2 \|I\|_2} \tag{8}$$

Here, we propose a fast query algorithm to perform such a retrieval process in the 3D integrated multi-index (see Algorithm 1). The priority queue of index triplets $(i, j, k)$ is the core of the fast query algorithm, in which the priority of each triplet is denoted as

$$q(i) + r(j) + s(k) = d(q, [u_i, v_j, t_k]) \tag{9}$$

where the q(:), r(:) and s(:) are the three input sequences. The queue is initialized by a special triplet (1, 1, 1). At each subsequent step *m*, the triplet $(i_m, j_m, k_m)$ with top priority is considered traversed and popped from the queue. The triplets $(i_m + 1, j_m, k_m)$, $(i_m, j_m + 1, k_m)$ and $(i_m, j_m, k_m + 1)$ are then considered to insert into the priority queue. The triplet $(i_m + 1, j_m, k_m)$ is inserted into the queue if its previous preceding triplets have also been traversed. Similarly, the triplets $(i_m, j_m + 1, k_m)$ and $(i_m, j_m, k_m + 1)$ are inserted into the queue if their previous preceding triplets have also been traversed. The main idea of this algorithm is that every triplets are inserted into the queue only once when all of its preceding triplets have also been traversed. The fast query algorithm generates a set of triplets $(i, j, k)$, whose lists $W_{i,j,k}$ are accumulated into the query's response.

**Algorithm 1:** Fast query algorithm()

INPUT: Three feature vectors q(:), r(:), s(:) ;
OUTPUT: The sequence of index triplets out(:);
Initialization:
Set vector out to Ø;
Set vector traversed (1, 1, 1) to **false;**
Set stack pqueue to new PriorityQueue;
Push the current features q(1)+r(1)+s(1) into the stack pqueue;
Traversal:
**repeat**
Pop the value of the current stack pqueue into ((i, j, k), d);
Set Traversed(i, j, k) to **true;**
**Set** out ∪ {(i, j, k)} to out;
**if** i < length(q) **and** ((j=1 **or** traversed(i+1, j-1, k))
**and** (k=1 **or** traversed(i+1, j, k-1)))
**then** push the current features q(i+1)+r(j) +s(k) into the stack pqueue (i+1, j, k);
**if** j < length(r) **and** ((i=1 **or** traversed(i-1, j+1,k))
**and** (k=1 **or** traversed(i, j+1, k-1)))
**then** push the current features q(i)+r(j+1)+s(k) into the stack pqueue (i, j+1, k);
**if** k < length(s) **and** ((i=1 **or** traversed(i-1, j,k+1))
**and** (j=1 **or** traversed(i, j-1, k+1)))
**then** push the current features q(i)+r(j)+s(k+1) into the stack pqueue (i, j, k+1);

**until** (enough traversed);

### 3.4 Discussion

Now let's discuss three types of index structure (conventional inverted index, inverted index and the suggestion of 3D integrated multiple index) with their relative efficiency, given the

SIFT codebook size $K_1$, the GIST codebook size $K_2$ and the CN codebook size $K_3$. In this case, the feature space of the proposed 3D integrated multi-index is quite different from inverted multi-index and traditional inverted index (see **Fig. 2** and **Fig. 3**). Especially the conventional index requires $K_1$ lists, and the multi-index requires $K_1^2$ lists that correspond to the spatial subdivision into $K_1$ cells and $K_1^2$ cells, meanwhile the proposed 3D multi-index requires $K_1K_2K_3$ ($K_1$>> $K_2$, $K_3$) lists which correspond to a much finer subdivision of the feature space. In the traditional inverted index, the distribution of the list length tends to be balanced, while the lengths of feature lists within the multi-indices (the inverted multi-index and the proposed 3D integrated multi-index) are highly non-uniform. However, although there is a highly non-uniform distribution of list lengths in the experiments, the 3D integrated multi-index has a great improvement in the retrieval accuracy because of the higher sampling density.

In addition, for the 3D integrated multi-index, even though there is an the increase of the subdivision density, matching a query with codebooks requires approximately the same number of operations as that of the other index structures. In the conventional inverted index case, one needs to calculate the distances $K_1$ times between M-dimensional feature vectors, and in the case of inverted multi-index, distances between M/2-dimensional feature vectors are calculated $2K_1$ times, while in the 3D integrated multi-index case $K_1+K_2+K_3$ ($K_1$>> $K_2$, $K_3$) distances are computed. So, we can see that in the case of conventional inverted index the matching is moderately faster. Due to the use of three-dimensional index structure, the query of 3D integrated multi-index also brings computational overhead. However, in our experiments (in Section 4.4), we observed that the overhead cost was small compared with other index structures.

Using 3D integrated multi-index can also lead to memory overhead, because it must maintain $K_1K_2K_3$ ($K_1$>> $K_2$, $K_3$) rather than $K_1$ or $K_1^2$ lists. However, the total length of all lists remains unchanged because the total number of feature vectors N is unchanged. Therefore, supposing that every list $W_{ijk}$ is stored continuously as a large array, to effectively maintain each list $W_{ijk}$ actually requires only an integer containing the starting position of the list. Such memory overhead is also smaller than several compressed vector or metadata usually stored for each instance.

For higher-dimensional multi-index, the experiments show that they do lead to smaller quantification times, but their memory overheads increases very fast with the increase of K, and the non-uniformity of the number of empty cells in the index and the length of the list is the same. The memory overhead limits the use of this kind of high-dimensional multi-index, and also limits the accuracy of retrieval. In conclusion, the 3D integrated multi-index is proved to be a optimum position between conventional inverted indices (large quantization times, low memory overhead) and higher dimensional multi-indices (low quantization time, high memory overhead) in our experiments. The use of the 3D integrated multi-index requires a small preprocessing time because it takes much shorter time to quantify all database vectors (for the K is lower).

The common property of inverted multi-index and the 3D integrated multi-index is that a feature vector is described by a set of visual words , rather than a single visual word in the conventional inverted index. With this representation, if two feature vectors are quantified to an identical visual word triplet, they are matched. On the other hand, the difference between the inverted multi-index and the 3D integrated multi-index is that inverted multi-index produces more fine division on one feature space, while 3D integrated multi-index is a feature fusion method. For the inverted multi-index, if two SIFT feature vectors in the first and second half part are similar, they are viewed as a match. However, for the 3D integrated multi-index, a

local area is jointly described by CN, GIST and SIFT descriptors, and all of the descriptors have to be similar to make a valid match.

## 4. Experiments

The experiment is designed to evaluate the proposed 3D integration multi-index structure and, especially, its usability in the task of content-based image retrieval. In the experiments, we compare the proposed method (I-MI) with the traditional inverted index. We also compare different variations of the inverted index, including the inverted multi-index (M-IDF) [10], the coupled multi-index (C-MI) [13]. In addition, we also evaluate the new improvements of the CBIR system suggested by other researchers.

### 4.1 Datasets

In this paper, we evaluate the proposed approach on four challenging image databases, namely, Holidays, UKbench, Paris6k and Oxford5k. We also employ the large image database ImageNet-L for large-scale retrieval experiments. MAP is used to evaluate the performance of these databases.

*Holidays:* It contains 1,491 images from personal holiday photo collections that consist of different locations and objects with various transformations. There are 500 image groups in the database, where the first image of each group is the query and the others are used as the database images for training. Most of the queries have 1–2 ground truth images.

*Ukbench:* This database contains 2 550 groups of 10 200 images. Each group has four images that contained the same object but are taken from different viewpoints and illuminations. In our experiments, all of the 10 200 images are used as both database and queries images. The expected retrieval result is the four most similar images of the same object for each image query.

*Oxford5K*: This database consists of 5062 images of Oxford's famous buildings, which are collected from Flickr. Fifty-five images of 11 Oxford landmarks are selected as the query images, and their ground truth retrieval results are provided. For each query image, a bounding box is provided to represent the region of the query image that describes a landmark building. In our experiments, we only use the local features inside the bounding box to construct a compact image representation.

*Paris6k*: It is composed of 6412 images, which are all downloaded from Flickr by searching the associated text tags for famous Paris landmarks. Similar to the Oxford5k Building database, 55 query images annotated by a rectangular of region interest are selected from the Paris landmarks, and the ground truths of images are provided as in the Oxford5K database.

*ImageNet-L*: The ImageNet is an image dataset organized according to the WordNet hierarchy where each node of the hierarchy is described by thousands of images. Since the ImageNet dataset contains sufficiently large variations and is publicly available, it is well suited to benchmark the memory usage, computation, and retrieval accuracy for the large-scale image retrieval. To further evaluate the performance on the large-scale image datasets, we merge Ukbench with the ImageNet dataset fo use as distractors. We use approximately 1.3 million images of 2,550 categories as a large-scale image database, denoted by ImageNet-L.

### 4.2 Features and Baseline

For the feature points of each image, we use Hessian affine detector to detect and use the SIFT, GIST and CN descriptors to describe the features. We calculate a 64-dimensional GIST [40] descriptor for each image. The image patches are restricted to 32×32 and then orientation histograms are computed on a 2×2 grid. We use an 11-dimensional colour names (CN) descriptor [41] computed from 16 × 16 sized image patches.

This paper uses the baseline method (named IDF) proposed in [32]. We also use the conventional 1D inverted index to produce a higher baseline, and the baseline results for Ukbench and Holidays are 70.72% and 48.58% in MAP (Mean Average Precision), respectively. The baseline scores are similar to those reported in [32].

### 4.3 Parameter Selection

*Size of codebooks:* In the experiment, We found that retrieval performance is greatly affected by the codebook. For the colour codebook and SIFT codebook, we follow the settings in [13] and set their sizes to 200 and 20 K respectively. For different GIST vocabulary sizes, **Fig. 4** shows that the retrieval accuracy increases as the size of the GIST codebook increases, and growth is relatively slow after the vocabulary exceeds 100. Since a smaller codebook leads to a lower memory cost, we set GIST codebooks to 100.

*Hamming embedding:* Two parameters are very importent in Hamming embedding (HE): the weighting parameter $\sigma$ and the Hamming threshold $\varepsilon$. For the SIFT and CN codebooks, we use the same parameter setting as in [32]. We set $\varepsilon = 22$ and $\sigma = 16$ for SIFT codebooks. Moreover, we set $\varepsilon = 7$ and $\sigma = 4$ for a CN codebook of size 200 in MIS−*C*, to obtain satisfactory performance in the experiments. In addition, for a GIST codebook of size 100 in MIS−*G*, we set $\varepsilon = 7$ and $\sigma = 4$, to obtain satisfactory performance in the experiments.

*Multiple assignment:* For M-IDF, C-MI, G-MI and I-MI, MA is used for both the query and reference images. To make MA do a good job, we set MA to 3 in M-IDF and set it to 50% for C-MI, which is also suggested in [28]. For G-MI and I-MI, the influence of MA is demonstrated in **Fig. 5**. From this figure, we can find that as the parameter MA increases for the GIST feature, the MAP increases first and then decreases. Here, we set MA to 40% for G-MI and I-MI, because this setting produces better performance in our preliminary experiments.

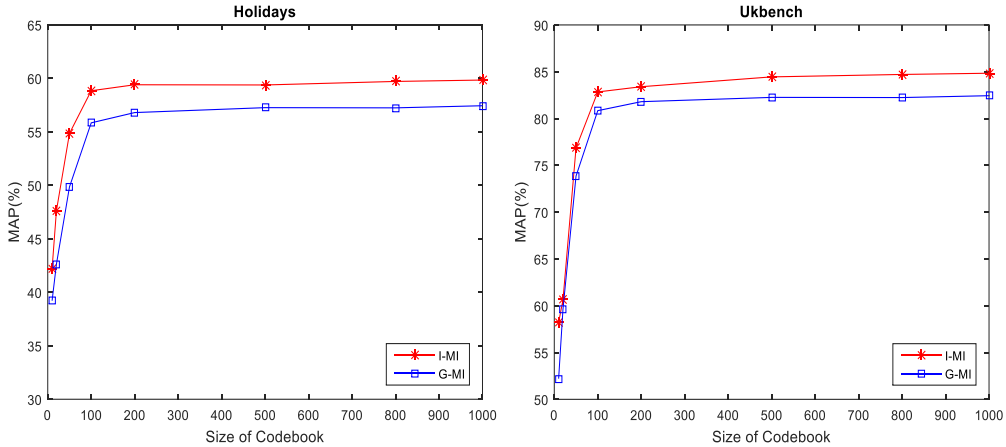


**Fig. 4.** Retrieval performance with different codebook size on two datasets

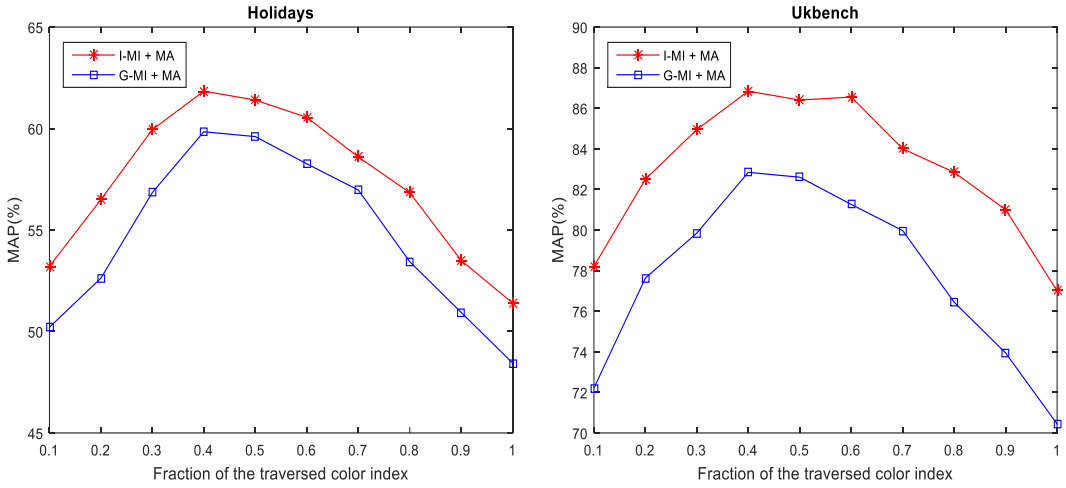


**Fig. 5.** The influence of MA for G-MI, I-MI on Holidays and Ukbench datasets. The codebook sizes for SIFT, GIST and CN features are 20K, 100 and 200 respectively

## 4.4 Evaluation

### 4.4.1 Comparison Among Different Inverted Indices

The performance of the conventional inverted index (IDF) and four variants, M-IDF [10], C-MI [13], and the proposed indices (G-MI and I-MI) can be observed in **Table 1**. From these results, different working mechanisms can be observed. In **Table 1**, IDF is an one-dimensional index structure. M-IDF, C-IDF and G-IDF are two-dimensional index structures and I-MI is a three-dimensional index structure. From the experimental results in **Table 1**, it can be seen that the three-dimensional I-MI is better than the other results, especially in comparison with the one-dimensional result. These results show that I-MI has better performance than other inverted indices. The main reason is that I-MI employs complementary information (contour and colour features) to provide stronger recognition. On the Holidays, Ukbench, Oxford5K and Paris6K datasets, the colour features C-MI and I-MI are good discriminators. However, if the illumination changes drastically, it might be the case that M-IDF, G-MI and I-MI work better. In M-IDF with two 20K codebooks, the total number of visual word pairs is equal to 400M. In our experiments, we find that many of the entries in the M-IDF are empty, which means many entries are wasted. Therefore, if the above problems are solved, the performance of M-IDF can be further improved. I-MI is essentially derived from the concept of multiple indexes. However, it takes into account the contributions of all SIFT, GIST, and CN features, which can alleviate some problems, while M-ID only considers the main functions.

**Table 1.** The performance of various methods on Holidays, Ukbench, Oxford5K and Paris6K datasets

| Methods | Holidays | Ukbench | Oxford5K | Paris6K |
|---|---|---|---|---|
| IDF | 48.58 | 70.72 | 51.87 | 56.62 |
| M-IDF | 54.41 | 79.21 | 57.29 | 68.09 |
| C-MI | 57.56 | 81.92 | 61.82 | 70.27 |
| G-MI | 57.66 | 82.52 | 61.79 | 70.84 |
| I-MI | 59.84 | 84.26 | 64.74 | 73.57 |

### 4.4.2 Complementarity to Some Existing Methods

To test whether I-MI is compatible with some prior techniques used in the conventional inverted index, we further combine Multiple Assignment (MA)[28], Hamming Embedding (HE) [32] , etc., into our framework.

It is clear from the **Table 2** that these techniques introduce consistent improvements on the four datasets. For example, in the Holidays dataset, the combination of HE improves the MAP approximately 22%. In addition, the MA method improves the accuracy moderately. As can be seen from **Table 2**, the accuracy of G-MI+MA is approximately 4% higher than that of G-MI in **Table 1**, and I-MI+HE+MA is better, with an increase of approximately 25%. So, the use of MA improves the accuracy and combining these methods adds more improvement to the accuracy. Similar results are obtained from other datasets, which demonstrates the feasibility of I-MI as a general index structure for image retrieval.

Compared to SIFT, the CN or GIST descriptor is not a very good discriminator in itself. Because of their low dimensions and sensitivity to contour and illumination changes, many more error matches may be produced than the SIFT descriptor. Therefore, the CN and the GIST features are more effective if they are used complementarily to SIFT, which will improve the matching precision, as implemented in this paper. **Fig. 6** illustrates some results of image retrieval on Ukbench using the IDF and I-MI methods. We can see that the I-MI method can improve the retrieval accuracy by fusing the representation of the SIFT, CN and GIST features.

**Table 2.** The performance of the proposed methods combined with HE and MA on Holidays, Ukbench, Oxford5K and Paris6K datasets

| Methods | Holidays | Ukbench | Oxford5K | Paris6k |
|---|---|---|---|---|
| G-MI +HE | 79.43 | 89.27 | 80.35 | 82.62 |
| I-MI+HE | 80.98 | 90.75 | 82.94 | 85.41 |
| G-MI + MA | 63.58 | 84.57 | 65.39 | 74.24 |
| I-MI+ MA | 64.69 | 86.36 | 68.55 | 77.44 |
| G-MI +HE+MA | 83.96 | 92.73 | 84.25 | 85.81 |
| I-MI+HE+MA | 84.86 | 93.85 | 86.73 | 88.69 |

The results in **Table 1-3** show the complementary properties of SIFT with GIST, CN, MA and HE. The application of MA,HE,GIST and CN alone can improve the accuracy of the SIFT-based IDF. When they are combined into the I-MI framework, further improvements can be made. We also found that the performance of HE was better than that of MA, CN and GIST when they are integrated with the SIFT-based IDF separately. In fact, HE comes directly from the SIFT descriptors, which maintains more discrimination than other methods. In addition, CN and GIST are auxiliary features to the SIFT-based IDF, so the effect may be indirect.

**Table 3.** The performance of various methods combined with HE and MA on Holidays, Ukbench, Oxford5K and Paris6K datasets

| Methods | Holidays | Ukbench | Oxford5K | Paris6k |
|---|---|---|---|---|
| IDF+HE+MA | 69.23 | 78.35 | 70.46 | 74.83 |
| M-IDF + HE + MA | 74.36 | 87.16 | 75.66 | 78.55 |
| C-MI + HE + MA | 83.19 | 92.69 | 83.77 | 85.27 |
| G-MI + HE + MA | 83.96 | 92.73 | 84.25 | 85.81 |
| I-MI + HE + MA | 84.86 | 93.85 | 86.73 | 88.69 |

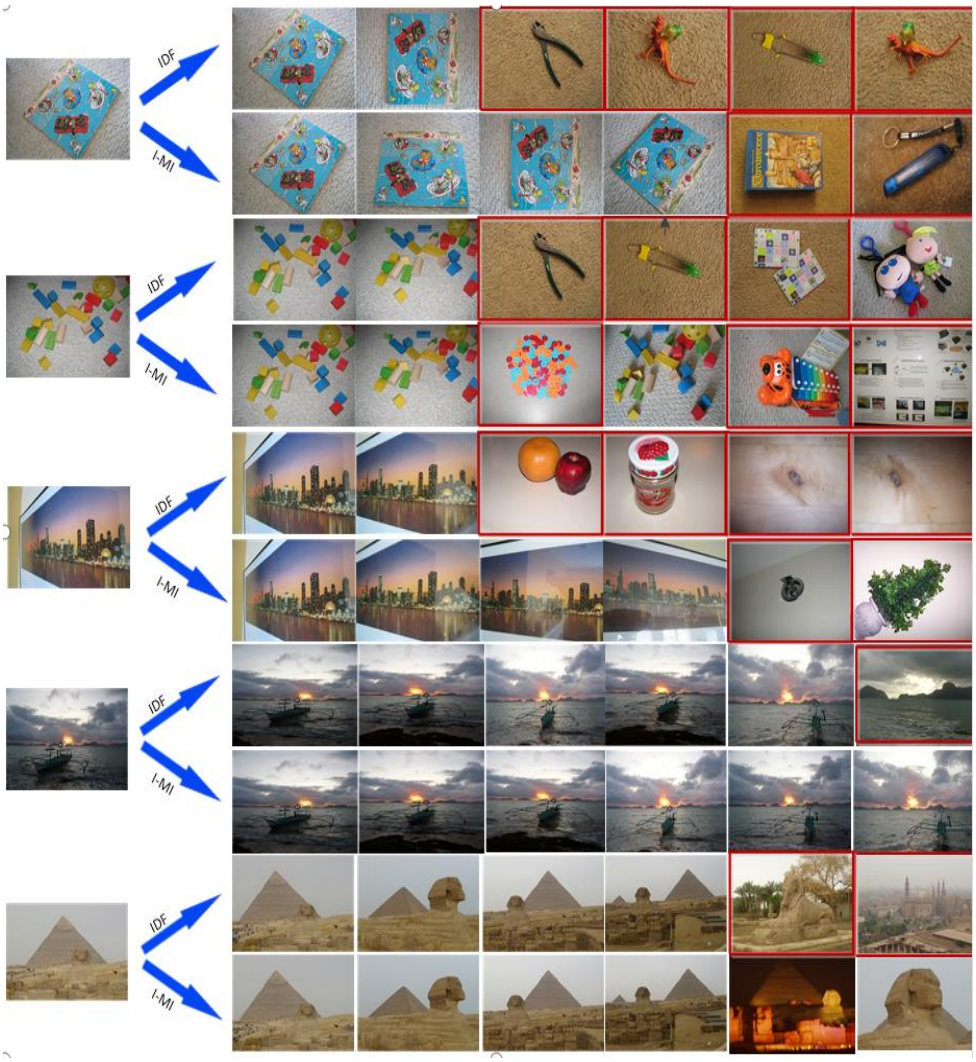


**Fig. 6.** Example results on Ukbench dataset. The first of each query shows IDF retrieval results, and the second row shows results of I-MI method. False positives are marked with red bounding boxes

### 4.4.3 Large-Scale Experiments

We use the ImageNet-L dataset to further evaluate the performance of IDF, M-IDF, C-MI, G-MI and I-MI on large-scale image retrieval, where the Ukbench dataset merged with the ImageNet dataset which are employed as distractors. We utilize different numbers of distractors, including 100, 1 000, 10 000, 120 000, 600 000, and the entire ImageNet-L dataset, to test the scalability of the proposed I-MI framework. **Fig. 7** shows the large scale image retrieval accuracies of IDF, M-IDF, C-MI, G-MI and I-MI with different numbers of distractor images.

**Fig. 7** displays that the performance of IDF, M-IDF, C-MI, G-MI and I-MI representations gradually decreases with the increase of distractor images in the datasets. When the number of distractor images is 100, all methods obtain their best results. When the entire ImageNet-L dataset is added into the Ukbench database, which totals 1 320 000 distractor images, the MAP values of IDF, M-IDF, C-MI, G-MI and I-MI are significantly reduced. The decline of MAP is

mainly due to the inverted file containing more items with the increase of the distractor images. Although the MAPs of all the methods decrease with the increasing number of distractor images, we can learn from the **Fig. 7** that the MAP of the I-MI structure reduces more moderately than the others. Thus, compared with IDF, M-IDF, C-MI and G-MI, the I-MI structure can maintain its discriminative ability more effectively.

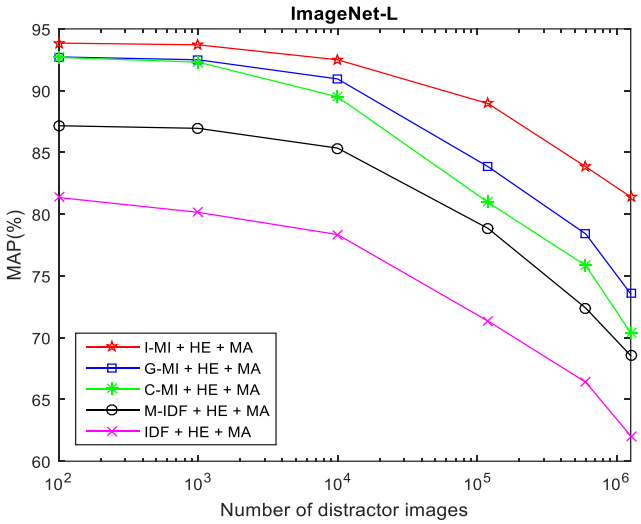


**Fig. 7.** Large-scale image retrieval performance of IDF, M-IDF, C-MI, G-MI and I-MI methods with different numbers of distractor images

**Fig. 8** show the precision vs. recall curvers of IDF, M-IDF, C-MI, G-MI and I-MI methods on ImageNet-L datasets. We observe that I-MI always perform the best. The performance of G-MI and C-MI are also very promising, compared to M-IDF, because they both utilize an additional visual information.

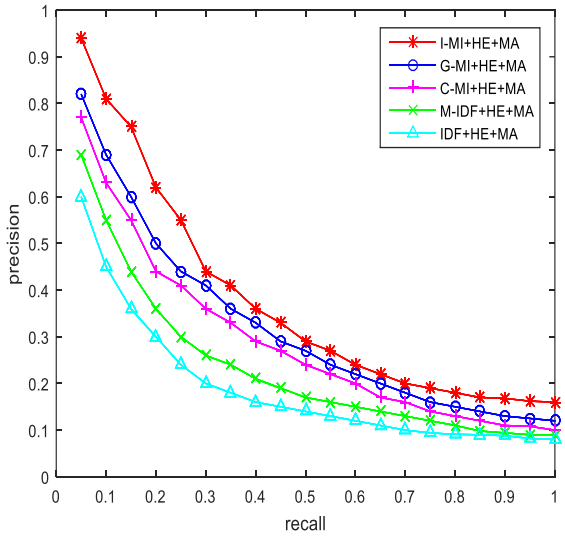


**Fig. 8.** The precision vs. recall curvers on ImageNet-L datasets

### 4.4.4 Time Efficiency

In all kinds of computer vision applications, efficiency has always been a very important problem. In our experiment, the results are obtained on a server with 20 CPUs of Intel Xeon processors at 2.0 GHz and 128 GB of memory. The computation of the BOF-based retrieval methods can be divided into two parts: the first is local feature extraction and quantization, and the second is voting of TF-IDFs with the inverted indices. On ImageNet-L datasets, the average feature extraction time (SIFT, GIST and CN features) is 0.96s, and the average time of feature quantification using 20K codebook is 0.98s. The major computational demand in BOF-based retrieval methods is at the off-line stage, which can be easily parallelized. **Table 4** compares different methods in ImageNet-L dataset on average query time. Note that the time cost of feature extraction and quantization are not included. This experiment only focuses on the online retrieval process. It uses the baseline method 0.106s to perform a search on the ImageNet-L dataset. Due to the use of matching weights, the fusion of CN features increases the retrieval time of each query to 0.135s and the fusion of GIST features slightly increases the retrieval time to 0.128s. The HE method filters out a large number of mismatches, so it is slightly more efficient than CN and GIST. At the same time, it takes 0.184 s to combine CN,GIST and HE to complete an online search, which is acceptable given the significant improvement in accuracy.

**Table 4.** Average query time (s) on the ImageNet-L dataset

| Methods | BOF | CN | GIST | HE | CN+ GIST +HE |
|---|---|---|---|---|---|
| | 0.104 | 0.135 | 0.128 | 0.122 | 0.184 |

### 4.4.5 Memory Cost

In the BOF-based retrieval, the total memory footprint of inverted indices is proportional to the total number of feature vectors in the dataset. This paper compares the memory usage of all fusion methods on **Table 5**. Here, we list the experimental results of the memory cost for each indexed feature vector, and recalculate the memory cost on the ImageNet-L dataset as in our case. For each indexed feature vector, the baseline IDF uses 4 bytes to store the image ID, whereas in HE, 8 bytes are used to store the 64-bit binary SIFT feature. In comparison, the fusion of CN costs approximately 3 bytes to store the binary colour signature, and the fusion of GIST costs 2 bytes to store the binary GIST signature; however, they attain a competitive performance improvement.

On the ImageNet-L dataset, the fusion of CN, GIST and HE requires a total of 6.9 GB of memory .

**Table 5.** Memory cost for different approaches

| Methods | BOF | CN | GIST | HE | CN+HE+GIST |
|---|---|---|---|---|---|
| Per feature(bytes) | 4 | 7 | 6 | 12 | 17 |
| Per image(KB) | 1.7 | 2.8 | 2.5 | 5.0 | 6.9 |
| 1M dataset(GB) | 1.7 | 2.8 | 2.5 | 5.0 | 6.9 |

### 4.5 Comparison with State-of-the-arts Approaches

In this section, we compare the proposed I-MI method against some state-of-the-art approaches in the literature. **Table 6** demonstrates the comparison of the proposed method with other state-of-the-art systems [4, 13, 23, 31, 43-47] in the Oxford5K and Holidays

datasets. The literature [4] is a CNNs-based CBIR method. Here, we have used our own implementation of CRB-CNN-M [4], and this method also has a good experimental results. Most of these approaches employ additional techniques such as multiple assignment [13], query expansion [43, 44], and k-NN2 re-ranking [31, 46]; however, the proposed I-MI method compares favourably to these state-of-the-arts methods.

From **Table 6** we can see that our final result is MAP = 84.86 % for Holidays, and this result exceeds the result in [47], at 0.16 % in MAP for the Holidays dataset. Although the MAP=86.2% in [31] is close to our result on the Oxford5K dataset, the result on the Holidays dataset is only 76.2%. These comparisons confirm the effectiveness of our framework. Importantly, we note that the techniques such as spatial constraints [32], burstiness weighting [28], etc., may also contribute to the proposed framework. Furthermore, various post processing steps, such as the query expansion [48], RANSAC (Random sample consensus) verification [8] and graph fusion [15], can be directly used on top of our method.

**Table 6.** Performance MAP (%) comparison with state-of-the-art methods. Note that the MAP in [4, 13] is obtained from our own implementation

| **Methods** | **Ours** | [13] | [43] | [23] | [44] | [45] | [31] | [46] | [4] | [47] |
|---|---|---|---|---|---|---|---|---|---|---|
| Oxford5K | 86.73 | - | 83.50 | 72.71 | 75.70 | 81.40 | 86.20 | 85.00 | 86.32 | 84.30 |
| Holidays | 84.86 | 83.19 | 84.00 | 81.16 | 84.10 | 42.30 | 76.20 | 84.22 | 84.67 | 84.16 |

## 5. Conclusions

In this paper, a 3D integrated multi index framework is proposed to fuse features at the index level. For very similar preprocessing time and retrieval complexity, the proposed framework achieves a denser search space segmentation while maintaining memory efficiency compared to conventional inverted index and inverted multi-index approaches. Additionally, we exploit the fusion of local contour and colour features in the 3D integrated multi-index method. Each dimension of the 3D integrated multi-index corresponds to a feature space, and the query process is similar in the SIFT feature space. The combination of SIFT, colour and contour features significantly reduces the impact of false positive matches. Consequently, the 3D integrated multi-index approach yields more accurate retrieval and search tasks. A large number of experiments on five benchmark datasets show that compared with other methods, 3D integrated multi index can significantly improve the retrieval accuracy while requiring acceptable memory usage and query time.

## References

[1] Z. A. Abduljabbar, A. Ibrahim, M. A. Hussain, Z. A. Hussien, M. A. Sibahee, and S. Lu, "EEIRI: Efficient Encrypted Image Retrieval in IoT-Cloud," *KSII Transactions on Internet And Information Systems*, vol. 13, no. 11, pp. 5692-5716, Nov. 2019. Article (CrossRef Link)

[2] B. K. Iwana, and S. Uchida, "Time series classification using local distance-based features in multi-modal fusion networks," *Pattern Recognition*, vol. 97, Jan. 2020. Article (CrossRef Link)

[3] S. Y. Jeong, and W. H. Kim, "Thermal Imaging Fire Detection Algorithm with Minimal False Detection," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2156-2170, May 31, 2020. Article (CrossRef Link)

[4] K. Liao, H. Lei, Y. Zheng, G. Lin, C. Cao, M. Zhang, and J. Ding, "IR Feature Embedded BOF Indexing Method for Near-Duplicate Video Retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3743-3753, Dec, 2019. Article (CrossRef Link)

[5] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Unified Binary Generative Adversarial Network for Image Retrieval and Compression," *International Journal of Computer Vision*, vol. 128, no. 8, pp. 2243-2264, Sep. 2020. Article (CrossRef Link)
[6] J. Sivic and A. Zisserman, "Efficient Visual Search of Videos Cast as Text Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591-606, 2009. Article (CrossRef Link)
[7] Q. Thuy, Q. Huu, C. P. Van, and T. N. Quoc, "An efficient semantic - Related image retrieval method," *Expert Systems with Applications*, vol. 72, pp. 30-41, 2017. Article (CrossRef Link)
[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007. Article (CrossRef Link)
[9] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2161-2168, 2006. Article (CrossRef Link)
[10] A. Babenko and V. Lempitsky, "The Inverted Multi-Index," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1247-1260, 2015. Article (CrossRef Link)
[11] X. Qian, H. Wang, Y. Zhao, X. Hou, R. Hong, M. Wang, and Y. Y. Tnag, "Image Location Inference by Multi-Saliency Enhancement," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 813-821, 2017. Article (CrossRef Link)
[12] X. Yang, X. Qian, and Y. Xue, "Scalable Mobile Image Retrieval by Exploring Contextual Saliency," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1709-1721, 2015. Article (CrossRef Link)
[13] L. Zheng, S. Wang, and Q. Tian, "Coupled Binary Embedding for Large-Scale Image Retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3368-3380, 2014. Article (CrossRef Link)
[14] K. Liao, F. Zhao, Y. Zheng, C. Cao, and M. Zhang, "Parallel N-Path Quantification Hierarchical K-Means Clustering Algorithm for Video Retrieval," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 9, Sep, 2017. Article (CrossRef Link)
[15] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query Specific Rank Fusion for Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 803-815, 2015. Article (CrossRef Link)
[16] Y. Rao, and W. Liu, "Region Division for Large-scale Image Retrieval," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 10, pp. 5197-5218, Oct. 2019. Article (CrossRef Link)
[17] K. Liao and G. Liu, "An efficient content based video copy detection using the sample based hierarchical adaptive k-means clustering," *Journal of Intelligent Information Systems*, vol. 44, pp. 133-158, Feb. 2015. Article (CrossRef Link)
[18] J. Lu, V. E. Liong, and J. Zhou, "Simultaneous Local Binary Feature Learning and Encoding for Homogeneous and Heterogeneous Face Recognition," *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 40, no. 8, pp. 1979-1993, Aug. 2018. Article (CrossRef Link)
[19] K. T. Ahmed, S. Ummesafi, and A. Iqbal, "Content based image retrieval using image features information fusion," *Information Fusion*, vol. 51, pp. 76-99, Nov. 2019. Article (CrossRef Link)
[20] C. Wu, H. Zhang, J. Hua, S. Hua, Y. Zhang, X. Lu, and Y. Tang, "A Novel Least Square and Image Rotation based Method for Solving the Inclination Problem of License Plate in Its Camera Captured Image," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 12, pp. 5990-6008, Dec. 31, 2019. Article (CrossRef Link)
[21] C. Bin, A. thung, X. Zhang, and Z. Zhao, "Multiple feature fusion for social media applications," in *Proc. of International Conference on Management of Data*, pp. 435-446, 2010. Article (CrossRef Link)
[22] C. Wengert, M. Douze, and H. Jegou, "Bag-of-colors for improved image search," in *Proc. of the 19th ACM International Conference on Multimedia ACM Multimedia*, pp. 1437-1440, 2011. Article (CrossRef Link)

[23] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-Aware Co-Indexing for Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2573-2587, 2015. Article (CrossRef Link)
[24] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Proc. of Computer Vision and Pattern Recognition*, pp. 1-8, 2008. Article (CrossRef Link)
[25] P. Li, M. Wang, J. Cheng, C. Xu, and H. Lu, "Spectral Hashing With Semantically Consistent Graph for Image Indexing," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 141-152, 2013. Article (CrossRef Link)
[26] L. Zheng, S. Wang, L. Tian, H. Fei, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1741-1750, 2015. Article (CrossRef Link)
[27] Z. Wu, S. Jiang, and Q. Huang, "Near-duplicate video matching with transformation recognition," in *Proc. of the 17th ACM International Conference on Multimedia*, pp. 549-552, 2009. Article (CrossRef Link)
[28] H. Jegou, M. Douze, and C. Schmid, "Improving Bag-of-Features for Large Scale Image Search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316-336, 2010. Article (CrossRef Link)
[29] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008. Article (CrossRef Link)
[30] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual Weighting for Vocabulary Tree based Image Retrieval," in *Proc. of IEEE International Conference on Computer Vision*, pp. 209-216, 2011. Article (CrossRef Link)
[31] 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 *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3013-3020, 2012. Article (CrossRef Link)
[32] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. of European Conference on Computer Vision*, vol. 5302, pp. 304-317, 2018. Article (CrossRef Link)
[33] C. G. M. Snoek and M. Worring, "Multimedia event-based video indexing using time intervals," *IEEE Transactions on Multimedia*, vol. 7, no. 4, pp. 638-647, 2005. Article (CrossRef Link)
[34] S. Park, W. Jeong, and Y. S. Moon, "X-ray Image Segmentation using Multi-task Learning," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 3, pp. 1104-1120, Mar. 2020. Article (CrossRef Link)
[35] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys(CSUR)*, vol. 45, no. 4, 2013. Article (CrossRef Link)
[36] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "L-p-norm IDF for Large Scale Image Search," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1626-1633, 2013. Article (CrossRef Link)
[37] L. Gao, X. Zhu, J. Song, Z. Zhao, and H. T. Shen, "Beyond product quantization: Deep progressive quantization for image retrieval," in *Proc. of IJCAI International Joint Conference on Artificial Intelligence*, pp. 723-729. Article (CrossRef Link)
[38] L. Zheng, S. J. Wang, Z. Q. Liu, and Q. Tian, "Packing and Padding: Coupled Multi-index for Accurate Image Retrieval," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1947-1954, 2014. Article (CrossRef Link)
[39] P. Gehler and S. Nowozin, "On Feature Combination for Multiclass Object Classification," in *Proc. of IEEE 12th International Conference on Computer Vision*, pp. 221-228, 2009. Article (CrossRef Link)
[40] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145-175, 2001. Article (CrossRef Link)

[41] J. Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning Color Names for Real-World Applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512-1523, 2009. Article (CrossRef Link)
[42] F. S. Khan, R. M. Anwer, J. Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3306-3313, 2012. Article (CrossRef Link)
[43] Y. Gao, M. Shi, D. Tao, and C. Xu, "Database Saliency for Fast Image Retrieval," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 359-369, 2015. Article (CrossRef Link)
[44] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian, "Democratic Diffusion Aggregation for Image Retrieval," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp.1661-1674, 2016. Article (CrossRef Link)
[45] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Gool, "Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 777-784. 2011. Article (CrossRef Link)
[46] D. Qin, C. Wengert, and L.Gool, "Query Adaptive Similarity for Large Scale Object Retrieval," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1610-1617, 2013. Article (CrossRef Link)
[47] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao, "Visual Reranking through Weakly Supervised Multi-Graph Learning," in *Proc of IEEE International Conference on Computer Vision*, pp. 2600-2607, 2013. Article (CrossRef Link)
[48] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), pp. 2911-2918, 2012. Article (CrossRef Link)

**Mingzhu Zhang** received the B.S. degree in computer science from the XIDIAN University, Xi'an, China, in 2004, the M.S. degree in Management science and Engineering from the Xi'an Technological University, Xi'an, China, in 2011. She is currently a Full associate professor with the Department of Public Courses, Xi'an Fanyi University, Xi'an China. Her research interests include data mining, pattern recognition, video analysis and retrieval.