Codebook-free Compact Descriptor for Scalable Visual Search

Yuwei Wu, Feng Gao, Yicheng Huang, Jie Lin *Member, IEEE*, Vijay Chandrasekhar *Member, IEEE*, Junsong Yuan *Senior Member, IEEE*, and Ling-Yu Duan *Member, IEEE*

***Abstract*—The MPEG compact descriptors for visual search (CDVS) is a standard towards image matching and retrieval. To achieve high retrieval accuracy over a large scale image/video dataset, recent research efforts have demonstrated that employing extremely high-dimensional descriptors such as the Fisher Vector (FV) and the Vector of Locally Aggregated Descriptors (VLAD) can yield good performance. Since the FV (or VLAD) possesses high discriminability but small visual vocabulary, it has been adopted by CDVS to construct a global compact descriptor. In this paper, we study the development of global compact descriptors in the completed CDVS standard and the emerging compact descriptors for video analysis (CDVA) standard, in which we formulate the FV (or VLAD) compression as a resource-constrained optimization problem. Accordingly, we propose a codebook-free aggregation method via dual selection to generate a global compact visual descriptor, which supports fast and accurate feature matching free of large visual codebooks, fulfilling the low memory requirement of mobile visual search at significantly reduced latency. Specifically, we investigate both sample-specific Gaussian component redundancy and bit dependency within a binary aggregated descriptor to produce compact binary codes. Our technique contributes to the scalable compressed Fisher vector (SCFV) adopted by the CDVS standard. Moreover, the SCFV descriptor is currently serving as the frame level hand-crafted video feature, which inspires the inheritance of CDVS descriptors for the emerging CDVA standard. Furthermore, we investigate the positive complementary effect of our standard compliant compact descriptor and deep learning based features extracted from convolutional neural networks (CNNs) with significant mAP gains. Extensive evaluation over benchmark databases shows the significant merits of the codebook free binary codes for scalable visual search.**

***Index Terms*—Visual Search, Compact Descriptor, CDVS, CDVA, Codebook free, Feature Descriptor Aggregation.**

I. Introduction

With the advent of the era of Big Data, huge body of data resources come from the multimedia sharing platforms such as *YouTube*, *Facebook* and *Flickr*. Visual search has attracted considerable attention in multimedia and computer vision [1], [2], [3], [4]. It refers to the discovery of images/videos contained within a large dataset that describe the same objects/scenes as those depicted by query terms. There exists a wide variety of emerging applications, *e.g.,* location recognition and 3D reconstruction [5], searching logos for estimating brand exposure [6], and locating and tracking criminal suspects from massive surveillance videos [7]. This aera involves a relatively new family of visual search methods, namely the Compact Descriptors for Visual Search (CDVS) techniques standardized from the ISO/IEC Moving Pictures Experts Group (MPEG) [8], [9], [10], [11], [12]. Generally speaking, the MPEG CDVS aims to define the format of compact visual descriptors as well as the feature extraction and visual search process pipeline to enable interoperable design of visual search applications. In this work, we focus on how to generate an extremely low complexity codebook-free [1] global descriptor for CDVS. Moreover, the proposed descriptor contributes to the emerging compact descriptors for video analysis (CDVA) standard [13].

The basic idea of visual search is to extract visual descriptors from images/videos, then perform descriptor matching between the query and dataset to find relevant items. Towards effective and efficient visual search, the visual descriptors need to be discriminative and resource-efficient (*e.g.,* low memory footprint). The discriminability relates to the search accuracy, while the computational resource cost impacts the scalability of a visual search system. Taking mobile visual search as an example, we may extract and compress the visual features of query images at the mobile end and then send compact descriptors to the server end. This is expected to significantly reduce network latency and improve user experience of wearable or mobile devices with limited computational resources or bandwidth. Hereby, we would like to tackle the problem of high performance and low complexity aggregation of local feature descriptors from the perspective of the resource constrained optimization. This is well aligned with the goal of developing an effective and efficient compact feature descriptor representation technique with low memory cost in the "analyze then compress" infrastructure [14].

More recently, various vision tasks are carried out over large scale datasets, and visual search is no exception. To achieve high retrieval accuracy, it is useful to employ high-dimensional hand-crafted descriptors (such as the Fisher Vector (FV) [15],

Yuwei Wu, Feng Gao, Yicheng Huang and Ling-Yu Duan are with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. Yuwei Wu is also with Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology (BIT), Beijing, China.

Jie Lin and Vijay Chandrasekhar are with the Institute for Infocomm Research, Singapore. Vijay Chandrasekhar is also affiliated with Nanyang Technological University (NTU), Singapore.

Junsong Yuan are with School of Electrical and Electronics Engineering at Nanyang Technological University (NTU), Singapore.

Yuwei Wu and Feng Gao are joint first authors. Ling-Yu Duan is the corresponding author.

[1]The typical compression scheme, Product Quantization (PQ), often requires tens of sub-codebooks while conventional Hashing algorithms involve thousands of hash functions, thereby incurring heavy memory cost. Extensive experiments have shown that our compact descriptor consumes extremely low memory of 0.015 MB with promising search performance. This merit of extremely low memory footprint is referred to as "codebook free".

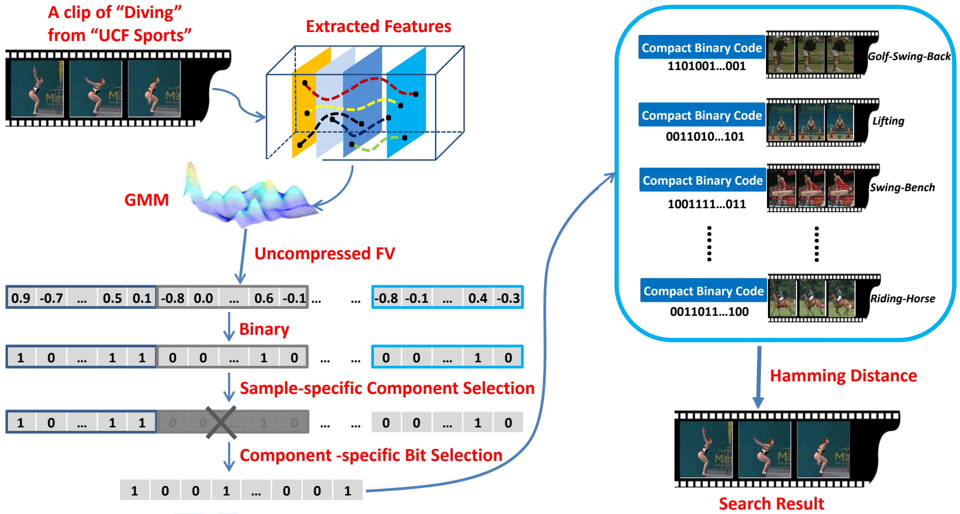

Fig. 1. The flow diagram of visual search by applying the proposed codebook-free compact descriptor. Taking the Fisher vector as an example, we first binarize the FV by a sign function, leading to a binarized Fisher Vector (BFV). Then discriminative bits are selected from the BFV, where both sample-specific Gaussian component redundancy and bit dependency within a binary aggregated descriptor are introduced to produce optimized compact Fisher codes.

the Vector of Locally Aggregated Descriptors (VLAD) [16]), as well as deep learning based features [17], [18]. In particular, CNNs based deep learning features perform remarkably well on a wide range of visual recognition tasks. Moreover, the concatenation of hand-crafted shallow features and deep features proves to be more distinctive and reliable to characterize object appearances. Recently, Lou *et al.* [19] demonstrated that the combination of deep invariant descriptors and hand-crafted descriptors not only fulfills the lowest bitrate budget of CDVA but also significantly advances the video matching and retrieval performance of CDVA. Therefore, how to effectively and efficiently compress the hand-crafted features is practically useful, no matter from the perspective of CDVS and CDVA standardization, or the emerging applications like memory lightweight augmented reality on wearable or mobile devices. In retrieval and matching tasks, short binary codes can alleviate the issues of limited computational resources or bandwidth. In augmented reality scenarios, the compact descriptors of moderate-scale image database can be stored locally, in which image matching can be performed on local devices.

Hashing [3], [20], [21], [22], [23] and product quantization (PQ) [24], [25], [26], [27], [28], [29] are popular compression techniques to obtain binary descriptors. The hashing methods first project the data points into a subspace (or hyperplanes), and then quantize the projection values into binary codes. Learning methods have been involved in projection stage, in which the linear or non-linear projection are applied to convert high dimensional features (*e.g.,* feature $\mathbf{f} \in \mathbb{R}^N$) into binary embeddings and then learn a hashing function in the low-dimensional space. However, the computational complexity of the projection is $\mathcal{O}(N^2)$ [30], [23]. For instance, given that a 426-dimensional dense trajectory feature is extracted from a video and PCA is employed to project each trajectory feature to the 213 dimension, we then obtain a $54,528$-dimensional FV when the number of Gaussian components $K = 128$, as depicted in Figure 1. In the case of $N = 54,528$, a projection matrix alone may take more than 10 GB memory footprint and projecting one vector would cost ~ 800 ms on a single core. On the other hand, the PQ generally divides a data vector into shorter subvectors and uses time-consuming K-means algorithm to train the sub-codebooks. Each subvector is quantized by its own smaller sub-codebooks. However, as K (*e.g.,* the number of Gaussian components in FV aggregation) increases, binarizing high-dimensional descriptors using existing hashing or PQ approaches is intractable or infeasible due to the demanding computational cost and memory requirements. Hence, notwithstanding the effectiveness of preserving data similarity structure by the aforementioned hashing or PQ methods, important academic and industrial efforts like CDVS and CDVA have paid more attentions to generate compact binary codes of high-dimensional descriptors subject to the limited computational resources.

In this work, we focus on discriminative and compact aggregated descriptor for visual search at very low computational resources. To this end, we consider the following observations on the state-of-the-arts aggregation techniques, *i.e.,* FV and VLAD. (1) As introduced in [15], [9], whenlocal features in an image are aggregated w.r.t Gaussian components to form a residual vector, not all the components are of equal importance to distinguish a sample. (2) In addition to the component level redundancy, there exists the bit-wise dependency within each selected component yet to be removed for better compactness.

Motivated by empirical observations mentioned above and the challenges of developing compact feature descriptors for visual search and analysis (*i.e.,* CDVS and CDVA), we formulate the FV (or VLAD) compression as a resource-constrained optimization problem, and propose a codebook-free binary coding method for scalable visual search. A sample-specific component selection scheme is introduced to remove redundant Gaussian components, and a global structure preserving sparse subspace learning method is presented to suppress the bit-wise dependency within each selected component. Figure 1 takes video search as an example and shows the flow diagram of generating compact global descriptors. Specifically, given a raw Fisher vector with K Gaussian components, we first binarize the FV by a sign function, leading to a binarized Fisher vector (BFV). Both sample-specific Gaussian component redundancy and bit dependency within a component are removed to produce compact binary codes.

The proposed approach to compressing aggregated descriptor has been adopted by MPEG CDVS standard [9], [10], [11], [12]. Furthermore, our method serves as the frame level video hand-crafted feature representation towards large-scale video analysis, which leverages the merits of CDVS features for the emerging compact descriptors for video analysis (CDVA) standard. In summary, our contributions are three-fold.

1) We formulate a light-weight compression of aggregated descriptors as a resource constrained optimization problem, which investigates how to improve the descriptor performance in terms of both descriptor compactness and computational complexity. To this end, we come up with a codebook-free global compact descriptor via dual selection for visual search. We circumvent the time-consuming codebook training and avoid the heavy memory cost of storing large codebooks, which is particularly beneficial to mobile devices with limited memory usage.
2) The SCFV descriptor derived from our dual selection scheme, has been adopted by the completed MPEG CDVS standard as a normative aggregation technique to produce a compact global feature representation with much lower complexity. Both sample-specific component redundancy and bit dependency within a high-dimensional binary aggregated descriptor have been removed significantly. Fast matching with extremely low memory footprint is allowed. Extensive experiments over a wide range of benchmark datasets have shown promising search performance.
3) Furthermore, our work has provided a groundwork of compact hand-crafted features for the development of emerging MPEG CDVA standard, which is an important extension for large-scale video matching and retrieval. We have shown that the proposed global descriptors and the state-of-the-art CNN descriptors are positive complementary to each other, and their combination can significantly improve the performance for video retrieval. We argue that, especially towards real-time (mobile) visual search and augmented reality applications, how to harmoniously leverage the merits of highly efficient and low complexity handcrafted descriptors, and the cutting edge CNNs based descriptors, is a competitive and promising topic, which has been validated by CDVA.

The rest of the paper is organized as follows. Section II reviews the related work. In section III, we introduce the problem statement of our work. Section IV presents the details of our compact binary aggregated descriptor via the dual selection scheme. We report and discuss the experimental results in Section VI, and conclude the paper in Section VII.

II. Related work

Our work focuses on high performance and low complexity compact binary aggregated hand-crafted descriptors for CDVS and CDVA. Recent study has shown that there is great complementarity between CNNs and hand-crafted descriptors for better performance in visual search [19], which has been well leveraged in the emerging CDVA standard.. In this section, we review typical techniques of compact representation. Readers are referred to [8], [13] for more comprehensive review.

A. Compact Representation of Hand-crafted Features

Hashing methods aim to map data points into Hamming space, which benefits fast Hamming distance computation and compact representation. In general, there are two categories of mainstream hashing approaches, *i.e.*, data-independent methods versus data-dependent methods. Data-independent methods do not rely on any training data, which is flexible, but often require long codes for satisfactory performance. Local Sensitive Hashing [31] (LSH) adopts a random projection which is independent of training data. Shift Invariant Kernel Hashing [32] (SIKH) chooses random projection and applies shifted cosine function to generate binary codes.

Different from data-independent methods, data-dependent methods learn hashing functions from training data and significantly outperform data-independent methods. Weiss *et al.* [33] proposed Spectral Hashing (SH) by spectral partitioning for the graph constructed from the data similarity relationships. Gong and Lazebnik [20] proposed Iterative Quantization (ITQ) which figures out an orthogonal rotation matrix to refine the initial projection matrix. Wang *et al.* [34] proposed Minimal Reconstruction Bias Hashing (MRH) to learn similarity preserving binary codes that jointly optimize both projection and quantization stages. Liu *et al.* [35] introduced a novel unsupervised hashing approach called min-cost ranking for large-scale data retrieval by learning and selecting salient bits. Many other representative data-dependent hashing methods include k-means hashing [36], graph hashing [37], kernel-based supervised discrete hashing [38], and ordinal constrained binary code [39] etc.

The other alternative approaches of learning binary codes include clustering-based quantization which quantizes the space into cells via k-means. Product quantization [24] divides the data space into multiple disjoint subspaces, where a number of clusters are obtained by k-means over each subspace. Then a datapoint is approximated by concatenating the nearest cluster center of each subspace, yielding a short code comprising the indices of the nearest cluster centers. The distance between two vectors is computed by a precomputed look-up table. Optimized Product Quantization [40] and its equivalent Cartesian k-means [29] improve Product Quantization by finding out an optimal feature space rotation and then performing product quantization over the rotated space. Composite Quantization [41] and its variants [42] further improves the quantization method by approximating a data point using the summation of dictionary words selected from different dictionaries. It is demonstrated that quantization methods usually yield better search performance with comparable code length than hashing algorithms. However, the retrieval of quantization methods is often less efficient as Hashing methods apply the fast Hamming matching. Besides, quantization methods require a large codebook of k-means centroids and a look-up table for distance computation. In contrast, our codebook-free method does not involve memory cost for quantization which can be readily applied in those resource limited scenarios, such as mobile visual search.

B. Compact Representation of Deep Features

While the aforementioned methods have certainly achieved great success, most of them work on hand-crafted features, which do not capture the semantic information and thus limit the retrieval accuracy of binary codes. The great success in deep neural network for representation learning has inspired deep feature coding algorithms. In deep hashing methods, the similarity difference minimization criterion is adopted [43]. The hash codes are computed by minimizing the similarity difference where image representation and hash function are jointly learnt through deep networks. Lin *et al.* [44] developed a deep neural network to learn binary descriptors in an unsupervised manner with three criterions on binary codes, *i.e.,* minimal loss quantization, evenly distributed codes and uncorrelated bits. Liong *et al.* [45] defined a loss to penalize the difference between binary hash codes and real values, and introduced the bit balance and bit uncorrelation conditions.

Undoubtedly, deep learning based representations are becoming the dominant approach to generate compact descriptors for instance retrieval. The major concern is that deep neural network models may require a large storage of up to hundreds of megabytes, which makes it inconvenient to deploy in mobile applications or memory lightweight hardware. Gong *et al.* [46] proposed to leverage deep learning technique to compress FV features, while our method considers the statistics of the FV or VLAD feature to derive a high performance compact aggregated descriptor at extremely low resources. Different from the deep learning based hashing methods that use a weakly-supervised fine-tuning scheme to update network parameters, our unsupervised dual selection scheme possesses superior computational efficiency. In addition, we have shown that the combination of the proposed hand-crafted compact descriptor and CNNs based descriptor can significantly improve the retrieval performance in the emerging CDVA standard.

C. Compression of High Dimensional versus Low Dimensional Descriptors

In the aforementioned methods, an intermediate embedding is generated in certain ways, then the projected values are quantized into binary codes. However, embedding matrices incur considerable memory and extra computational cost. In addition, most of them work on relatively low-dimensional descriptors (*e.g.,* SIFT and GIST). Few work is dedicated to high-dimensional descriptors such as FV and VLAD.

Perronnin *et al.* [47] proposed a ternary encoding method to compress FV. Ternary encoding is fast and memory free, however, it will result in long codesize. Sivic *et al.* [48] binarized the BoW histogram entries over large vocabularies. Chen *et al.* [49] presented a discriminative projection to reduce the dimension of uncompressed descriptors before binarization, leading to comparable accuracy with the original descriptor at smaller codes, but causing additional memory cost because of the projection matrices. Gong *et al.* [30] employed compact bilinear projections instead of a single large projection matrix to get similarity-preserving binary codes. Parkhi *et al.* [50] considered face tracks as the unit of face recognition in videos and developed a compact and discriminative Fisher vector.

The above-mentioned methods have shown that a high-dimensional binary code is often necessary to preserve the discriminative power of aggregated descriptors. Hence, this paper concerns a compact binary aggregated descriptor coding approach for fast approximate nearest neighbor (ANN) search. The proposed method significantly reduces the codesize of raw aggregated descriptors, without degrading the search accuracy or introducing additional memory footprint.

III. Problem Statement

To compress FV (or VLAD) into a compact descriptor for light storage and high efficiency, we formulate the FV (or VLAD) compression as a resource-constrained optimization problem. Let $A(\cdot)$ denotes search accuracy, $R(\cdot)$ denotes descriptor size and $C(\cdot)$ denotes complexity of compression. Our goal is to design a quantizer $q(\cdot)$ to quantize Fisher vector g that is able to maximize search performance $A(q(g))$ subject to the constraints of descriptor compactness R_{budget}, compression complexity C_{budget} in terms of memory and time:

$$\max_q A(q(g)) \quad s.t. \quad R(q) \leq R_{budget} \quad and \quad C(q) \leq C_{budget}. \tag{1}$$

Here, a problem arises: how to solve the abstract problem defined in Eq. (1) in a concrete implementation. To this end, we develop a codebook-free compact binary coding method via dual selection to compress the raw FV (or VLAD). Given a raw FV [2] $\mathbf{g} = [\mathbf{g}(1), ..., \mathbf{g}(K)]$ with K Gaussian components, where $\mathbf{g}(i)(1 \leq i \leq K)$ denotes the i-th sub-vector. Firstly, to fulfill the constraint of compression complexity, we binarize the FV by a sign function and get a binary aggregated descriptor $\mathbf{b} = \{\mathbf{b}(1), ..., \mathbf{b}(K)\}$, where each binary sub-vector code $\mathbf{b}(i) = sgn(\mathbf{g}(i))$. Secondly, we propose to select discriminative bits from the binary aggregated descriptor $\mathbf{b}$ to maximize search performance, subject to the constraint of descriptor compactness. A dual selection scheme, *i.e.,* the sample-specific component selection and the component-specific bit selection, is introduced to obtain the discriminative bits towards a low complexity and high performance FV binary representation. In our work, the sample-specific component selection is derived by the variance of each sub-vector $\mathbf{g}(i)$ of the raw FV, and the component-specific bit selection is applied to further reduce the dimensionality of components, while maintaining search performance as well. In next sections, we will present the key ideas mentioned above in detail.

Remark: This work is tightly related to the MPEG CDVS standard [9], [10], [11], [12]. CDVS adopts the scalable compressed Fisher Vector (SCFV) representation, in which the quantization procedure was briefly reported in [9]. To

[2] Since the VLAD is a simplified non-probabilistic version of FV, in what follows, we take the FV as an example to elaborate how to obtain the compact binary codes for convenient discussion.

compress the high dimensional FV, a subset of Gaussian components in the Gaussian Mixture Model (GMM) are selected by ranking the standard deviation of each sub-vector. The number of selected Gaussian components relies on the bit budget, and then one-bit scalar quantizer is applied to generate binary codes. SCFV derived from the proposed dual selection scheme concentrates on the scalability of compact descriptors. Beyond [9], [12], this work will systematically investigate how to leverage both sample-specific Gaussian component redundancy and bit dependency within a binary aggregated descriptor to produce the codebook-free compact descriptor. A component-specific bit selection scheme is introduced by global structure preserving sparse subspace learning. In addition, we study how to further improve the performance by combining the proposed compact descriptors with CNNs descriptors, which has been adopted by the emerging CDVA standard. Our technique benefits the aggregated descriptor in improving compactness, and removing bits redundancy to ameliorate discriminative power of the descriptor.

IV. Compact Binary Aggregated Descriptor via Dual Selection

Our goal is to compress aggregated descriptors, without incurring considerable loss of search accuracy. The coarse binary quantization would degrade search performance. We propose a dual selection model which utilizes both sample-specific component selection and component-specific bit selection to collect the informative bits. In the stage of sample-specific component selection, a subset of Gaussian components is adaptively determined, which can significantly boost search performance. Furthermore, the component-specific bit selection focuses on the compactness to reduce the dimensionality of components, while maintaining search performance.

A. Sample-specific Component Selection

For FV aggregation, a Gaussian Mixture model (GMM) $p_\lambda(x) = \sum_{i=1}^{K} \omega_i p_i(x)$ with K Gaussians (visual words) is to estimate the distribution of local features over a training set. We denote the set of Gaussian parameters as $\lambda_i = \{\omega_i, \mu_i, \sigma_i^2, i = 1, ..., K\}$, where ω_i, μ_i and σ_i^2 are the weight, mean vector and variance vector of the i-th Gaussian p_i, respectively. Let $\mathbf{F} = \{\mathbf{f}_1, ..., \mathbf{f}_T\}$ denote a collection of T SIFT local features extracted from an image. Projecting the dimensionality of each local SIFT feature to dimension D is beneficial to the overall performance [16], [9]. The gradient vectors of all local features w.r.t. the mean μ_i are aggregated into a D-dimensional vector

$$\mathbf{g}(i) = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma(\mathbf{f}_t, i)\sigma_i^{-1}(\mathbf{f}_t - \mu_i), \tag{2}$$

where $\gamma(\mathbf{f}_t, i) = \omega_i p_i(\mathbf{f}_t)/\sum_{j=1}^{k} \omega_j p_j(\mathbf{f}_t)$ denotes the posterior probability of local feature $\mathbf{f}_t$ being assigned to the i-th Gaussian. By concatenating the sub-vector $\mathbf{g}(i)$ of all the K components, we form the FV $\mathbf{g} = [\mathbf{g}(1), ..., \mathbf{g}(K)] \in \mathbb{R}^{KD}$, where $\mathbf{g}(i) \in \mathbb{R}^D$. Note that the gradient vectors can be extended to the deviates w.r.t. variance as well, which was adopted in CDVS standard to improve the performance at high bitrates [8]. Similarly, the VLAD can be derived from FV by replacing the GMM soft clustering with k-means clustering, *i.e.,*

$$\mathbf{g}(i) = \sum_{t:NN(\mathbf{f}_t)=\mu_i} (\mathbf{f}_t - \mu_i), \tag{3}$$

where $NN(\mathbf{f}_t)$ indicates $\mathbf{f}_t$'s nearest neighbor centroids.

Furthermore, we appy a one-bit quantizer to binarize the high dimensional $\mathbf{g} \in \mathbb{R}^{KD}$, such that nearly zero memory footprint can be achieved. We generate binary aggregated descriptors by quantizing each dimension of FV or VLAD into a single bit $0/1$ by a sign function. Formally speaking, $sgn(\mathbf{g})$ is used to map each element g_j of the descriptor $\mathbf{g}$ to 1 if $g_j > 0$, $j = 1, 2, \cdots, KD$; otherwise, 0, yielding a binary aggregated descriptor $\mathbf{b} = \{\mathbf{b}(1), ..., \mathbf{b}(K)\}$ with KD bits, where $\mathbf{b}(i) \in \mathbb{R}^D$.

FV exhibits a natural 2-D structure, as shown in Figure 2. Referring to Eq. (2), the aggregated FV is formed by concatenating residual vectors of all Gaussian components, while each residual vector is aggregated from local features being assigned to the corresponding Gaussian component. In other words, not all Gaussian components equally contribute to the discriminative power. The role of a Gaussian component relates to the number of local features quantized to that component. The occurrence of different Gaussian components may vary for different image samples. Those Gaussian components with low occurrence are supposed to be less discriminative. If none of local features is assigned to a Gaussian component i, then all the elements of the corresponding subvector $\mathbf{g}(i)$ in Eq. (2) are zero. Accordingly, the proposed sample-specific component selection leverages local statistics of each individual sample to discard the redundant Gaussian components.

Specifically, we process the FV at the granularity of Gaussian components and select all the bits of those components with high importance. Here, the importance measured by the amplitude of their responses determines which Gaussian components are activated. In particular, we adopt the soft assignment coefficients $\gamma(\mathbf{f}_t, i)$ of local features to indicate the importance of Gaussian i, which is employed to adaptively select part of discriminative components for each sample. The importance, *i.e.*, $I(\lambda_i)$, is defined as the sum of posterior probabilities of local features $\{\mathbf{f}_1, ..., \mathbf{f}_T\}$ being assigned to the i-th Gaussian component in the continuous FV model, which is given by

$$I(\lambda_i) = \sum_{t=1}^{T} \gamma(\mathbf{f}_t, i). \tag{4}$$

As each local feature can be soft quantized to multiple Gaussian components. We rank all the Gaussian components based on the accumulated soft assignment statistics of local features to Gaussian components. The Gaussian component ranked at the t-th position is called the t-th nearest neighbor Gaussian component. That is, sample-specific component selection can be implemented by sorting the set $\{I(\lambda_i), i = 1, 2, \cdots, K\}$, and then the subvector of the binary Fisher vector (BFV) $\mathbf{b}(i)$ with the largest $I(\lambda_i)$ is first selected to generate Fisher codes, followed by the $\mathbf{b}(j)$ with the second largest one. In this way,

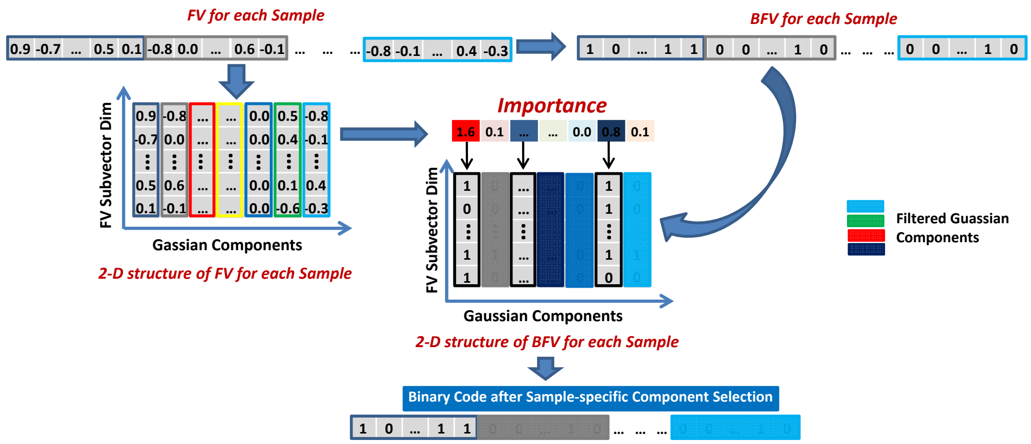

Fig. 2. Sample-specific component selection. The aggregated FV is partitioned into disjoint components by Gaussian components. We select all the bits of Gaussian components with high *importance* and the rest of components are discarded. In this work, the *importance* is defined as the sum of posterior probabilities of local features being assigned to the i-th Gaussian component.

we compute the importance of individual components for each sample using Eq. (4) and select a subset of components with high importance. The rest of components are discarded, as shown in Figure 2. It is necessary to maintain a Gaussian selection mask for each sample. Accordingly, the sample-specific component selection is nearly memory free, and the computational cost of Gaussian selection is $\mathcal{O}(N)$.

B. Component-specific Bit Selection

After sample-specific component selection, there probably exists bit-level redundancy within each selected component. Therefore, we may further improve the compactness of each selected component, while maintaining search performance. A naive solution is to apply random bit selection, however, it ignores the bit correlation within a component. In this work, we introduce a component-specific bit selection scheme by global structure preserving sparse subspace learning to select the most informative bits.

Without loss of generality, let $\mathcal{B} \in \{0,1\}^{\Theta \times D}$ denote a subset of the binary aggregated descriptor (after component selection) associated with the i-th Gaussion component over the whole dataset. Here Θ denotes how many samples select the i-th component as an *important* one, as shown in Figure 3. $\Theta \leq L$, where L is the number of training samples. The goal of bits selection is to find a small set of bits that can capture most useful information of $\mathcal{B}$. One natural way is to measure how close the original data samples are over the learned subspace $\mathbf{H}$ spanned by the selected bits. Mathematically, the component-specific bit selection is formulated as

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} ||\mathcal{B} - \mathcal{B}\mathbf{W}\mathbf{H}||_F^2. \\ &s.t. \mathbf{W} \in \{0,1\}^{D \times D'} \\ &\quad \mathbf{W}^\top \mathbf{1}_{D\times 1} = \mathbf{1}_{D' \times 1} \\ &\quad ||\mathbf{W}\mathbf{1}_{D'\times 1}||_0 = D' \end{aligned} \tag{5}$$

Here, $\mathbf{W}$ is a selection matrix with entries of 0 or 1. $\mathbf{W}^\top \mathbf{1}_{D\times 1} = \mathbf{1}_{D'\times 1}$ enforces that each column of $\mathbf{W}$ has a single 1, *i.e.,* at most D' bits are selected for each Gaussian

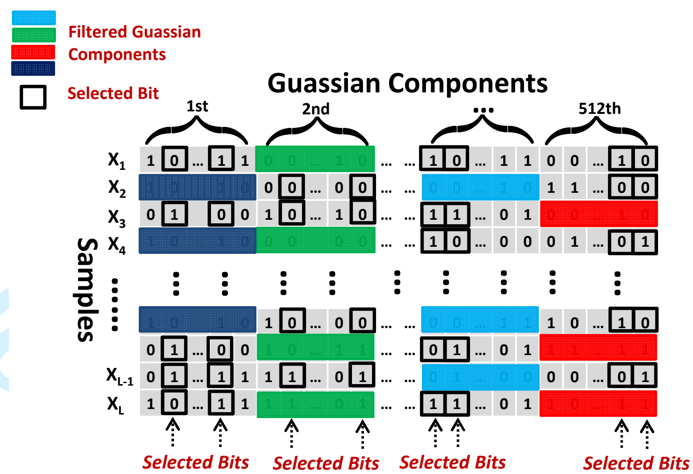

Fig. 3. Component-specific bit selection. The bits that carry as much information as possible are selected based on the global structure preserving sparse subspace learning. For each sample, the bits in black bounding boxes are the final compact Fisher codes. L denotes the number of training samples.

component. $||\mathbf{W}\mathbf{1}_{D'\times 1}||_0 = D'$ guarantees that $\mathbf{W}$ has the D' nonzero rows, and thus D' bits will be selected. Hence, Eq.(5) denotes the distance of $\mathcal{B}$ to the learned subspace $\mathbf{H}$.

A major difficulty in solving Eq. (5) lies in handling the discrete constraints imposed on $\mathbf{W}$. In this work, we relax the 0-1 constraint of $\mathbf{W}$ by introducing nonnegativity constraint and reduce the hard constraints of both $\mathbf{W}^\top \mathbf{1}_{D\times 1} = \mathbf{1}_{D'\times 1}$ and $||\mathbf{W}^\top \mathbf{1}_{D'\times 1}||_0 = D'$ to a $L_{2,1}$ norm constraint. Therefore, optimizing Eq. (5) is equivalent to solving

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} ||\mathcal{B} - \mathcal{B}\mathbf{W}\mathbf{H}||_F^2 + \beta ||\mathbf{W}||_{2,1}, \\ &s.t. \mathbf{W} \in \mathbb{R}_+^{D \times D'} \end{aligned} \tag{6}$$

where $\mathbb{R}_+^{D\times D'}$ denotes a set of $D \times D'$ nonnegative matrices and β is a parameter. Based on the resulting solution $\mathbf{W}$, we choose the bits corresponding to the D' rows of $\mathbf{W}$ with the largest norms.

To solve this optimization, we employ the accelerated block coordinate update (ABCU) method [51] to alternatingly update

$\mathbf{W}$ and $\mathbf{H}$ with

$$\begin{cases} f(\mathbf{W},\mathbf{H}) = ||\mathcal{B} - \mathcal{B}\mathbf{W}\mathbf{H}||_F^2 \\ g(\mathbf{W}) = \beta||\mathbf{W}||_{2,1}. \end{cases} \tag{7}$$

At the τ-th iteration, we need to solve the following optimization problems

$$\begin{cases} \mathbf{W}^{\tau+1} = \arg\min_{\mathbf{W}\geq 0} \left\langle \nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau}), \mathbf{W} - \widehat{\mathbf{W}}^{\tau} \right\rangle \\ \quad + \frac{\mathbf{L}_{\mathbf{W}}^{\tau}}{2}||\mathbf{W} - \widehat{\mathbf{W}}^{\tau}||_F^2 + \beta||\mathbf{W}||_{2,1} \\ \mathbf{H}^{\tau+1} = \arg\min_{\mathbf{H}} f(\mathbf{W}^{\tau+1}, \mathbf{H}), \end{cases} \tag{8}$$

where $\mathbf{L}_{\mathbf{W}}^{\tau} = ||\mathbf{H}^{\tau}(\mathbf{H}^{\tau})^{\top}||_s||\mathcal{B}^{\top}\mathcal{B}||_s$ is the Lipschitz constant of $\nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau})$ with respect to $\mathbf{W}$. Note that $||\Psi||_s$ denotes the spectral norm and equals the largest singular value of Ψ. $\widehat{\mathbf{W}}^{\tau}$ is given by

$$\widehat{\mathbf{W}}^{\tau} = \mathbf{W}^{\tau} + \omega_{\tau}\left(\mathbf{W}^{\tau} - \mathbf{W}^{\tau-1}\right), \tag{9}$$

where

$$\begin{cases} \omega_{\tau} = \min\left(\widehat{\omega}_{\tau}, \theta_{\omega}\sqrt{\frac{\mathbf{L}_{\mathbf{W}}^{\tau-1}}{\mathbf{L}_{\mathbf{W}}^{\tau}}}\right) \\ \widehat{\omega}_{\tau} = \frac{\delta_{\tau-1} - 1}{\delta_{\tau}} \\ \delta_{\tau} = \frac{1 + \sqrt{1 + 4\delta_{\tau-1}^2}}{2}. \end{cases} \tag{10}$$

ω_{τ} has been widely applied to the accelerate proximal gradient method for the convex optimization problem [51].

In Eq. (8), $\mathbf{W}^{\tau+1}$ is obtained by equivalently solving

$$\min_{\mathbf{W}\geq 0} \left\|\mathbf{W} - \left(\widehat{\mathbf{W}}^{\tau} - \frac{1}{\mathbf{L}_{\mathbf{W}}^{\tau}}\nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau})\right)\right\|_F^2 + \frac{\beta}{\mathbf{L}_{\mathbf{W}}^{\tau}}||\mathbf{W}||_{2,1}. \tag{11}$$

Eq. (11) can be decomposed into D smaller independent problems, each of which involves one row of $\mathbf{W}$ and is then solved by a Proximal operator

$$\prod(\mathbf{y}) = \min_{\mathbf{w}\geq 0} \|\mathbf{w} - \mathbf{y}\|_2^2 + \eta\|\mathbf{w}\|_2. \tag{12}$$

where $\mathbf{y}$ is the i-th row of $\widehat{\mathbf{W}}^{\tau} - \frac{1}{\mathbf{L}_{\mathbf{W}}^{\tau}}\nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau})$. $\prod(\mathbf{y})$ is able to be implemented efficiently according to the Theorem 1 [52], [53], which is described in Algorithm 1.

Theorem 1: *Given $\mathbf{y}$ and $\mathcal{I}$ the index set of positive components of $\mathbf{y}$, i.e., $\mathcal{I} = \{i : \mathbf{y}_i > 0\}$, the solution of Eq.(12) is expressed as*

$$\begin{cases} w_i = 0, \quad \forall i \notin \mathbf{I}; \\ \mathbf{w}_{\mathcal{I}} = \mathbf{0}, \quad ||\mathbf{y}_{\mathcal{I}}||_2 \leq \eta; \\ \mathbf{w}_{\mathcal{I}} = \left(||\mathbf{y}_{\mathcal{I}}||_2 - \eta\right)\frac{\mathbf{y}_{\mathcal{I}}}{||\mathbf{y}_{\mathcal{I}}||_2}, \quad otherwise. \end{cases} \tag{13}$$

Readers are referred to [52], [53] for the detailed proof. In addition, $\mathbf{H}^{\tau+1}$ can be obtained in a closed form, *i.e.,*

$$\mathbf{H}^{\tau+1} = \left[(\mathbf{W}^{\tau+1})^{\top}\mathcal{B}^{\top}\mathcal{B}\mathbf{W}^{\tau+1}\right]^{-1}(\mathbf{W}^{\tau+1})^{\top}\mathcal{B}^{\top}\mathcal{B}. \tag{14}$$

The accelerated block coordinate update method for solving Eq.(6) is described in Algorithm 2. The main cost lies in the update of $\mathbf{W}$ and $\mathbf{H}$. For updating $\mathbf{W}$, we should compute the first order partial derivative of $f(\mathbf{W},\mathbf{H})$ with respect to $\mathbf{W}$, *i.e.,* $\nabla_{\mathbf{W}} f(\mathbf{W},\mathbf{H}) = \mathcal{B}^{\top}(\mathcal{B}\mathbf{W}\mathbf{H} - \mathcal{B})\mathbf{H}^{\top}$. Computing $\mathcal{B}\mathbf{W}$, $\mathbf{H}\mathbf{H}^{\top}$, $\mathcal{B}\mathbf{H}^{\top}$, $\mathcal{B}\mathbf{W}(\mathbf{H}\mathbf{H}^{\top})$ and left multiplying $\mathcal{B}^{\top}$ to $\mathcal{B}\mathbf{W}(\mathbf{H}\mathbf{H}^{\top}) - \mathcal{B}\mathbf{H}^{\top}$ will take about $3\Theta DD' + DD'^2 + \Theta D'^2$ flops. Similarly, updating $\mathbf{H}$ will take $2\Theta DD' + DD'^2 + \Theta D'^2 + D'^3$ flops. Therefore, the computational complexity of Algorithm 2 is $\mathcal{O}(\Theta DD')$. Since $D' < min(\Theta, D)$, the ABCU algorithm is scalable. In summary, since the Gaussian components are independent of each other, the component-specific bit selection can be implemented in a parallel fashion. The set of selected bits D' is fixed and applied to all samples, and thus both memory footprint and computational cost of bits selection are module $\mathcal{O}(N)$.

Algorithm 1: Proximal operator for solving $\mathbf{W} = \mathbf{Prox}(\mathbf{Y}, \eta)$

Input: $\mathbf{Y} = \widehat{\mathbf{W}}^{\tau} - \frac{1}{\mathbf{L}_{\mathbf{W}}^{\tau}}\nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau})$

for $i = 1 \rightarrow D$ **do**

  Let $\mathbf{y}$ is the i-th row of $\mathbf{Y}$ and $\mathcal{I}$ the index set of positive components of $\mathbf{y}$;

  Set $\mathbf{w} = \mathbf{0}$;

  if $||\mathbf{y}_{\mathcal{I}}||_2 > \eta$ **then**

    $\mathbf{w}_{\mathcal{I}} = \left(||\mathbf{y}_{\mathcal{I}}||_2 - \eta\right)\frac{\mathbf{y}_{\mathcal{I}}}{||\mathbf{y}_{\mathcal{I}}||_2}$;

    Set the i-th row of $\mathbf{W}$ to $\mathbf{w}$.

  end

end

Algorithm 2: Accelerated block coordinate update for solving Eq.(6).

Input: $\mathcal{B} \in \mathbb{R}^{L\times D}$, the number of selected bits D'.

Output: Index set of selected bits $\mathcal{I} \subseteq \{1, 2, \cdots, D\}$ with $|\mathcal{I}| = D'$

Initialize $\tau = 1$, $\delta_0 = 1$, $\theta_{\omega} = 0.5$

for $\tau = 1, 2, \cdots$ *until convergence* **do**

  Compute $\delta_{\tau} = \frac{1+\sqrt{1+4\delta_{\tau-1}^2}}{2}$;

  Compute $\widehat{\omega}_{\tau} = \frac{\delta_{\tau-1}-1}{\delta_{\tau}}$;

  Compute $\omega_{\tau} = min\left(\widehat{\omega}_{\tau}, \theta_{\omega}\sqrt{\frac{\mathbf{L}_{\mathbf{W}}^{\tau-1}}{\mathbf{L}_{\mathbf{W}}^{\tau}}}\right)$;

  Let $\widehat{\mathbf{W}}^{\tau} = \mathbf{W}^{\tau} + \omega_{\tau}\left(\mathbf{W}^{\tau} - \mathbf{W}^{\tau-1}\right)$;

  Update $\mathbf{W}^{\tau+1} \leftarrow \mathbf{Prox}\left(\widehat{\mathbf{W}}^{\tau} - \frac{1}{\mathbf{L}_{\mathbf{W}}^{\tau}}\nabla_{\mathbf{W}} f(\widehat{\mathbf{W}}^{\tau}, \mathbf{H}^{\tau}), \frac{\beta}{\mathbf{L}_{\mathbf{W}}^{\tau}}\right)$ according to Algorithm 1;

  Update $\mathbf{H}^{\tau+1}$ according to Eq. (14);

  if $f\left(\mathbf{W}^{\tau+1}, \mathbf{H}^{\tau+1}\right) \geq f\left(\mathbf{W}^{\tau}, \mathbf{H}^{\tau}\right)$ **then**

    $\widehat{\mathbf{W}}^{\tau} = \mathbf{W}^{\tau}$;

    else $\tau = \tau + 1$;

  end

end

Normalize each column of $\mathbf{W}$;

Sort $||\mathbf{W}_{i,\cdot}||_2$, $i = 1, 2, \cdots, D$ and select bits corresponding to the D' largest ones.

V. Hamming Distance Matching

In online search, we apply the dual selection scheme to query samples as well and perform search using the selected bits. As the collection of selected components may vary in different samples, we have to solve the issue of matching descriptors across different Gaussian components. That is, given a query $\mathbf{x}_q$ and a dataset sample $\mathbf{x}_r$, if the selected

Gaussian components are different, the similarity cannot be computed directly using standard Hamming distance. So we adopt a normalized cosine similarity score Sc in [11] to calculate the distance, given by

$$Sc = \frac{\sum_{i=1}^{K} s_i^q s_i^r \Big(D' - 2 * h\big(sgn(\mathbf{g}_i^{\mathbf{x}_q}), sgn(\mathbf{g}_i^{\mathbf{x}_r})\big)\Big)}{D' \sqrt{||s^q||_0 ||s^r||_0}}, \quad (15)$$

where s_i^q and s_i^r denote whether the i-th Gaussian component is chosen for $\mathbf{x}_q$ and $\mathbf{x}_r$, respectively, and $h(\cdot,\cdot)$ is the Hamming distance between binarized Fisher subvectors. In practice, Sc is computed based on the overlapping Gaussians $s^q \bigcap s^r$ between $\mathbf{x}_q$ and $\mathbf{x}_r$.

VI. Experiments

In this section, extensive experiments are conducted to evaluate the proposed method in both computational efficiency and search performance. Our approach is implemented in C++. The experiments are performed on an Dell Precision workstation 7400-E5440 with 2.83GHz Intel XEON processor and 32GB RAM in a mode of single core and single thread.

A. Image Retrieval

To compare with baselines, we carry out retrieval experiments on MPEG CDVS datasets [54] and the publicly available INRIA Holidays dataset [55]. The CDVS datasets consists of five data sets used in the MPEG-CDVS standard: Graphics, Paintings, Frames, Landmarks and Common Objects. (1) The *Graphics* dataset depicts CD/DVD/book cover, text document and business card. There are $1,500$ queries and $1,000$ dataset images. (2) The *Painting* dataset contains 400 queries and 100 dataset images of paintings (say history, portraits, etc.) (3) The *Frame* dataset contains 400 queries and 100 dataset images of video frames captured from a range of video contents like movies and news. (4) The *Landmark* dataset contains $3,499$ queries and $9,599$ dataset images from building benchmarks, including the ZuBuD dataset, the Turin buildings, the PKUbench, etc. (5) The *Common Object* dataset contains $2,550$ objects,each containing four images taken from different viewpoints. For each object, the first image is query and the rest are dataset images. (6) The *Holidays* dataset is a collection of $1,491$ holiday photos, there are 500 image groups where the first image of each group is used as a query. To fairly evaluate the performance over a large-scale dataset, we use *FLICKR1M* as the distractor dataset, containing 1 million distractor images collected from Flickr. The retrieval performance is measured by mean Average Precision (mAP).

We evaluate the proposed method against several representative baselines including LSH [31], BP [30], and ITQ [20]. For the LSH, BP and ITQ methods, we randomly chose 20K images from Flickr1M dataset to train the projection matrices. The SIFT features from each image are extracted and then the dimensionality of SIFT is reduced to 32 by PCA. We employ FV encoding to aggregate the dim-reduced SIFT features with 512 Gaussian components. Therefore, the total dimensionality of FV is $512 \times 32 = 16384$. Then, we apply power law ($\alpha = 0.5$) followed by $L2$ normalization to the raw FV feature.

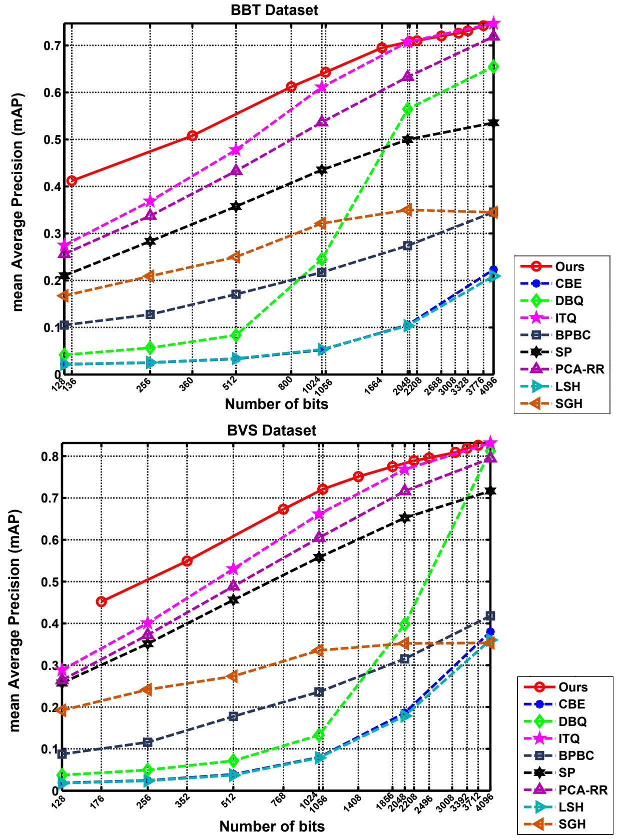

Fig. 5. Comparisons of the mean average precision on BBT and BVS datasets (best viewed in color).

Fig. 4 presents comparison results between the proposed method and both the product quantization approaches and hashing algorithms, in terms of mAP vs. different codesize over different datasets. As it is computationally expensive for $L2$ distance between uncompressed FV, we present the retrieval accuracy of uncompressed FV on the INRIA Holidays dataset only. The proposed method achieves superior accuracy than other methods, especially for small codesizes. Note that ITQ yields better mAP than other baseline methods. This is reasonable since ITQ is fine tuned for projection learning to minimize the mean square error. However, ITQ suffers from huge memory footprint and computational cost because of the projection matrix computation explained in Section I.

B. Video Retrieval

For the extension of video retrieval, we perform evaluation on the face retrieval task with two challenging TV-Series [57], *i.e.,* the Big Bang Theory (BBT), and Buffy the Vampire Slayer (BVS). The 3341 face videos are collected from the first six episodes from the first season of the BBT, and 4779 face videos of the first six episodes are acquired from BVS. As discussed in [7], we simply treat the video as a set of frames. Each face frame is represented with a 240-dimensional discrete cosine transformation feature. To reduce the dimensionality of original features, in this experiment, we employ FV encoding to aggregate the discrete cosine transformation features with 128 Gaussian components. Therefore, the $240 \times 128 = 30,720$

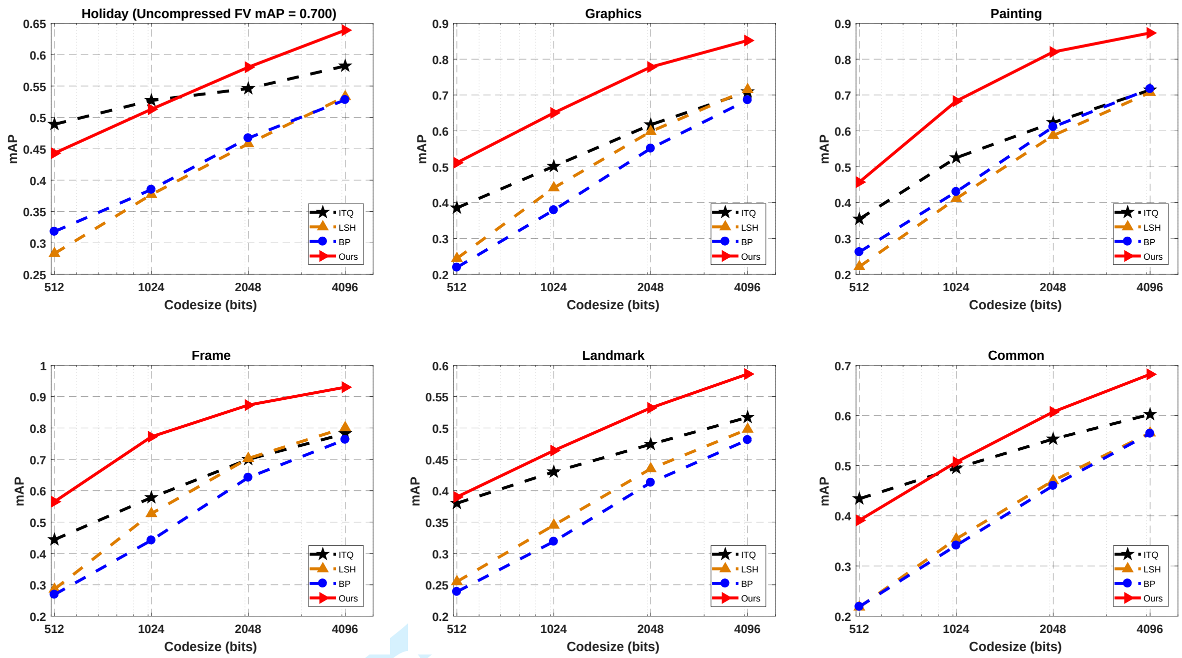

Fig. 4. Retrieval performance in terms of mAP vs. descriptor codesize over various benchmark datasets. (best viewed in color).

TABLE I

COMPARISON OF DESCRIPTOR COMPRESSION RATIO, ADDITIONAL COMPRESSION TIME AND MEMORY FOOTPRINT FOR FV COMPRESSION BETWEEN THE PROPOSED METHOD AND THE BASELINE SCHEMES, WITH COMPARABLE RETRIEVAL MAP, *e.g.,* ABOUT 51% ON THE INRIA HOLIDAYS DATASET. NOTE THAT N, P, Q DENOTE THE DIMENSIONALITY OF FV, THE TARGET CODESIZE AND THE SIZE OF VECTOR QUANTIZATION CODEBOOKS FOR THE PQ METHOD, RESPECTIVELY.

Method	Compression Ratio	Memory Cost		Compression Time	
		Theoretical	Practice (MB)	Theoretical	Practice (ms)
LSH [31]	93.6	$\mathcal{O}(NP)$	91.8	$\mathcal{O}(NP)$	151
BPBC [30]	154.2	$\mathcal{O}(\sqrt{N}\sqrt{P})$	0.031	$\mathcal{O}(N\sqrt{P}+P\sqrt{N})$	17
PQ [24]	65.5	$\mathcal{O}(NQ)$	8.4	$\mathcal{O}(NQ)$	25
RR+PQ [24], [40], [56]	65.5	$\mathcal{O}(NQ+N^2)$	277	$\mathcal{O}(NQ+N^2)$	278
ITQ [20]	256	$\mathcal{O}(N^2)$	254	$\mathcal{O}(N^2)$	257
Ours	512	$\mathcal{O}(N)$	0.015	$\mathcal{O}(N)$	< 1

dimensional aggregated FV is used to represent each face video. We treat video retrieval as an ANN search problem. Given a query video, we find the videos that are the nearest neighbors of the query based on the Euclidean distance. The ground-truth is defined by the 50 nearest neighbors of each query video in the Euclidean space. We repeat the experiments 10 times and take the average mAP as the final result.

Although supervised binary coding methods achieve higher accuracy than those unsupervised and semi-supervised ones [7], [58], we compare our unsupervised method with several state-of-the-art unsupervised methods including LSH [31], BPBC [30], SGH [59], ITQ [20], SP [23], DBQ [60], CBE [61] and PCA-RR [20]. We utilize the published codes and suggested parameters of these methods from the corresponding authors. Performance comparisons are shown in Figure 5. Our method clearly outperforms other approaches on the long codes. Although our method yields comparable performance as ITQ especially when the codesize is larger than 2048, the proposed method runs much faster than ITQ.

C. Effectiveness of Dual Selection

In this section, we further analyze the impact of dual selection on retrieval performance on the MPEG CDVS datasets and the INRIA Holidays dataset, *i.e.*, sample-specific component selection and component-specific bit selection. Note that the proposed method degenerates to two basic models: Ours_S denotes that only the sample-specific component selection scheme is employed. Ours_C denotes that only the component-specific bit selection is applied. From Figure 6, the Ours_S performs consistently better than Ours_C at the same codesize over all datasets. In contrast, the proposed full-fledged method fusing Ours_S and Ours_C significantly outperforms both Ours_S and Ours_C (especially at small codesize). For example, at codesize of 2048 bits, Ours_S and Ours_C yield mAP 74.0% and 33.9% on the Graphics dataset, which is inferior to 77.8%. This gain shows the dual selection scheme is complementary to each other, which can select more informative bits. With more bits, Ours_S, Ours_C and the dual selection scheme can progressively improve the retrieval accuracy. In particular, Ours_S and our full-fledged method approach to the retrieval mAP of uncompressed FV at large codesize. They obtain mAP 65.8% at codesize of 8192 bits on the Holidays dataset, while the original FV yields mAP 70.0%.

Noted that we only show the results of Ours_C at the codesizes from 2048 to 8192, rather than presenting the results at the codesize of both 512 and 1024. This is because our FV encoding aggregates SIFT features with 512 Gaussian components. If the codesize is set to 512, just one bit is chosen from each Gaussian component, which is not reasonable. Moreover, Ours_C is notably worse than Ours_S and our full-fledged method. The main reason can be explained as follows. The original FV signal presents a sort of Gaussian basis sparsity. Indeed, if none of local feature was assigned to Gaussian i, then all the elements of the corresponding Fisher sub-vector in Eq. (2) and Eq. (3) are supposed to be zeros. Although Ours_C alone does not work well, we may yield satisfactory improvements by combining Ours_C with Ours_S.

D. Computational Complexity:

Table I compares the compression ratio, memory and time complexity of the proposed method and other baselines. With comparable retrieval mAP, the compression ratio of our method is 2 to 6 times larger than the baseline schemes, resulting in much smaller compact codes. The memory footprint of our method is extremely low, *i.e.*, 0.015 MB (16 KB) for the globally bit selection mask. By contrast, RR+PQ and LSH cost over hundreds of megabytes to store the projection matrix. In addition, our method is super fast, as only binarization and selection operations are involved, while hashing methods and PQ often involve heavy floating point multiplications.

E. Discussion

Performance Impact of Combining Deep Features and Hand-crafted Features: To further improve the performance of our model, we incorporate CNNs and our hand-crafted compact aggregated descriptors into the well-established CD-VS evaluation framework to study the effectiveness of our method. In particular, we use a Nested Invariance Pooling (NIP) method [19] to derive the compact and robust CNNs descriptor. It is shown that the combination descriptors, referred to as Ours+NIP, can greatly improve performance, as depicted in Figure 6. For example, we have achieved more than 20% mAP improvement over Ours on the Holidays dataset. However, the performance growth trend of Ours+NIP is less apparent on the Holidays and Common datasets. This is because the NIP can get remarkable results with 512 codesize, *e.g.,* 0.885 mAP on the Holidays dataset, 0.964 mAP on the Common dataset. In this case, the incremental performance improvements of our scalable descriptor is inhibited. However, the performance growth trend of Ours+NIP is comparable with Ours as the number of codesize grows on the Graphics, Painting and Frame datasets.

Furthermore, we extend experiments in the evaluation framework of emerging MPEG CDVA to validate the effectiveness of Ours+NIP. The MPEG CDVA dataset [3] includes

[3]http://www.cldatlas.com/cdva/dataset.html

TABLE II
PERFORMANCE OF VIDEO RETRIEVAL ON THE MPEG CDVA DATASET.

	mAP	Precisian@R	Descriptor Size
Pool5 [62]	0.587	0.561	4KB
R-MAC [63]	0.713	0.681	2KB
NIP [19]	0.768	0.736	2KB
Ours+NIP	0.826	0.803	4KB

TABLE III
COMPARISONS OF THE COMPACT BINARY CODES DERIVED FROM BOTH VLAD AND FV ON THE "HOLIDAYS + 1M FLICKR" DATASET [55]. THE CODE LENGTH IS 4096.

Feature	Method	INRIA Holidays		
		mAP	STM	time (s)
Binary codes obtained from FV	Linear search	63.59	69.85	2.23
	HKM [64]	45.20	47.80	0.12
	PQ [24]	62.53	68.15	2.98
	LSH [31]	62.69	68.88	1.06
	EWH [65]	61.23	67.99	0.43
	MIH [66]	63.50	69.80	0.53
	Ours	63.90	69.50	0.14
Binary codes obtained from VLAD	Linear search	62.23	68.2	2.41
	HKM [64]	43.70	45.60	0.11
	PQ [24]	60.15	65.23	2.95
	LSH [31]	61.16	67.50	1.25
	EWH [65]	60.46	67.10	0.47
	MIH [66]	62.20	68.20	0.50
	Ours	62.05	67.90	0.08

9974 query and 5127 reference videos. For video retrieval, the performance is evaluated by the mean Average Precision (mAP) as well as the precision at a given cut-off rank R for a single query (Precisian@R). Table II shows that NIP achieves the promising retrieval performance over both pool5 (CNNs features) and R-MAC. Furthermore, Ours+NIP has significantly improved the retrieval performance against Pool5 (CNNs features) by 23.9% in mAP and 24.2% in Precisian@R. Compared with the state-of-the-art R-MAC, 11.3% in mAP and 12.1% in Precisian@R are achieved. Overall, it is shown that a combination of our global CDVS descriptor and NIP CNNs descriptors can greatly improve performance, in which the positive complementary effects of hand-crafted descriptors and deep invariant descriptors have been well demonstrated.

Performance impact of Codebook Size: We evaluate our method for different numbers of Gaussian components (*i.e.,* the codebook size) and present performance in Figure 7. For raw FV, the longer codesize is, the better performance is. Moreover, the compressed descriptors of different lengths yield the best results over benchmarks when the number of Gaussian components is 512 and the performance is further boosted as the code length grows. However, performance drops when the codebook size is set to 1024, which is mainly attributed to the fewer or even none overlapping Gaussians between the query and database images. In addition, we simply use $sgn(\mathbf{g})$ to binarize each element g_j in Eq. (2) and Eq. (3), which may introduce more noise as the dimensionality of the original FV or VLAD grows. So ours is worse than the original uncompressed FV.

Comparisons with Binary VLAD Code: VLAD encoding is regarded as a simplified version of FV encoding [16]. We can obtain the compact binary VLAD code using the proposed

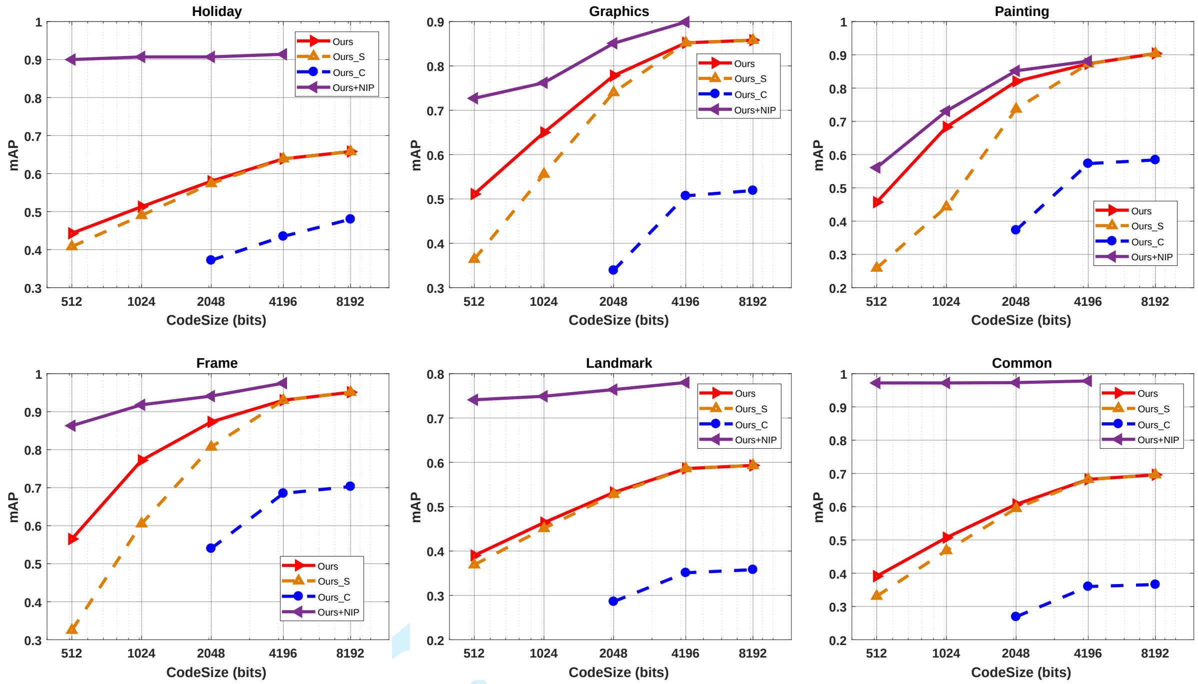

Fig. 6. Effectiveness of Dual Selection. Retrieval performance in terms of mAP vs. descriptor codesize over various benchmark datasets (best viewed on high-resolution display).

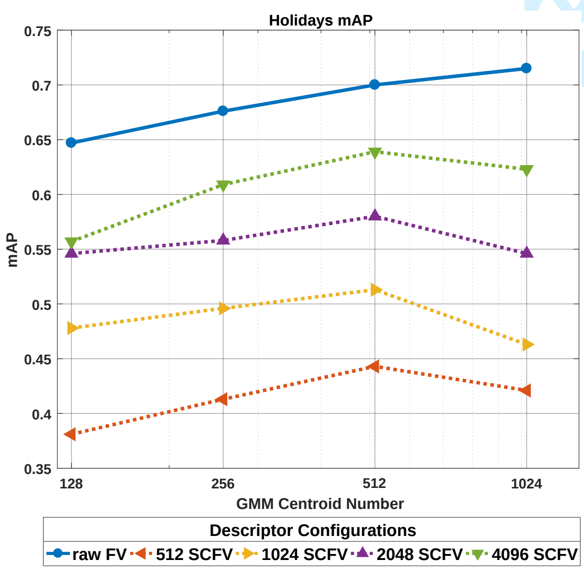

Fig. 7. Impact of the number of Gaussian components (*i.e., codebook size*) measured on the Holidays dataset.

scheme as well. Table III compares the retrieval accuracy in terms of mAP, STM [4] and search time on the "Holidays + 1M Flickr" dataset [55] mentioned above. The proposed method is compared against several state-of-the-art methods including LSH [31], PQ [24], HKM [64], EWH [65] and MIH [66]. The compact descriptors derived from both VLAD and FV are evaluated. Both are significantly faster than linear search with comparable retrieval accuracy. For instance, our method obtains around 30 (*resp.* 16) times speedup than linear search at the cost of a minor mAP drop, *i.e.,* 0.18% (*resp.* 0.09%) for the VLAD feature (*resp.* FV feature). The proposed method performs significantly better than HKM (*i.e.,* +18% mAP) with comparable search time. Although the mAP of oursis slightly worse than MIH, search is faster. Overall, the FV codes outperform VLAD codes.

Impact of Key Parameters: We evaluate the impact of key parameters of the proposed algorithm, including the number of selected Gaussian components M and the number of selected bits D', in terms of retrieval mAP on BBT and BVS datasets, as shown in Table IV. In our experiments, different numbers of selected bits $D' = \{8, 16, 32, 64\}$ are applied to different configuration settings. For example, to obtain a binary code with about 1024 bits, we employ the cross-validation algorithm to obtain the combination $\{M, D'\}$ that produces the best performance on the evaluation dataset. From Table IV, $M = 66$ and $D' = 16$ provides the best results. In practice, we employ cross-validation to determine the best parameters.

VII. Conclusion

We have proposed an effective solution for learning compact binary codes to address the fast ANN problem with high-dimensional aggregated descriptor. Both the sample-specific Gaussian component selection and the component-specific bit selection are proposed to produce a codebook-free compact descriptor, which exhibits extremely low compression memory and time complexity, and supports fast Hamming distance matching. Moreover, our approach has been validated in the evaluation framework of the MPEG CDVS standard, and provided a groundwork of compact hand-crafted features for the emerging MPEG CDVA standard. Extensive experimental

[4]STM denotes the Success rate for Top Match to measure the precision at rank 1, which is defined as (number of times the top retrieved image is relevant)/(number of queries).

TABLE IV
IMPACT OF KEY PARAMETERS OF THE PROPOSED ALGORITHM, INCLUDING THE NUMBER OF SELECTED COMPONENTS M, THE NUMBER OF SELECTED BITS D'. ASSUME THAT THE CODE LENGTH IS ABOUT 1024.

Feature	$M \times D'$						
	128×8	64×16	65×16	66×16	32×32	33×32	16×64
BBT	0.580	0.617	0.629	**0.643**	0.616	0.632	0.591
BVS	0.638	0.674	0.683	**0.721**	0.697	0.705	0.644

results demonstrate superior retrieval performance against the state-of-the-art methods such as hashing and PQ.

REFERENCES

[1] H. Mller and D. Unay, "Retrieval from and understanding of large-scale multi-modal medical datasets: A review," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2093–2104, Sept 2017.

[2] F. Shen, Y. Yang, L. Liu, W. Liu, D. Tao, and H. T. Shen, "Asymmetric binary coding for image search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2022–2032, Sept 2017.

[3] Z. Chen, J. Lu, J. Feng, and J. Zhou, "Nonlinear sparse hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 1996–2009, Sept 2017.

[4] L. Liu, M. Yu, and L. Shao, "Learning short binary codes for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1289–1299, March 2017.

[5] R. Ji, Y. Gao, W. Liu, X. Xie, Q. Tian, and X. Li, "When location meets social multimedia: A survey on vision-based recognition and mining for geo-social multimedia analytics," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 1, p. 1, 2015.

[6] R. Tao, A. W. Smeulders, and S.-F. Chang, "Attributes and categories for generic instance search from one example," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 177–186.

[7] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen, "Face video retrieval with image query via hashing across euclidean space and riemannian manifold," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4758–4767.

[8] L.-Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao, "Overview of the mpeg-cdvs standard," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 179–194, 2016.

[9] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao, "Rate-adaptive compact fisher codes for mobile visual search," *Signal Processing Letters*, vol. 21, no. 2, pp. 195–198, 2014.

[10] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, "Compact descriptors for visual search," *IEEE MultiMedia*, vol. 21, no. 3, pp. 30–40, 2014.

[11] L.-Y. Duan, J. Lin, Z. Wang, T. Huang, and W. Gao, "Weighted component hashing of binary aggregated descriptors for fast visual search," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 828–842, 2015.

[12] "Text of ISO/IEC DIS 15938-13 compact descriptors for visual search," *ISO/IEC JTC1/SC29/WG11/W14392*, 2014.

[13] L.-Y. D. Duan, V. Chandrasekhar, S. Wang, Y. Lou, J. Lin, Y. Bai, T. Huang, A. C. Kot, and W. Gao, "Compact descriptors for video analysis: the emerging mpeg standard," *arXiv preprint arXiv:1704.08141*, 2017.

[14] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2013, pp. 278–282.

[15] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[16] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[17] H. F. Yang, K. Lin, and C. S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[18] V. E. Liong, J. Lu, Y. P. Tan, and J. Zhou, "Deep video hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, June 2017.

[19] Y. Lou, Y. Bai, J. Lin, S. Wang, J. Chen, V. Chandrasekhar, L. Duan, T. Huang, A. C. Kot, and W. Gao, "Compact deep invariant descriptors for video retrieval," in *In Proceedings of the Data Compression Conference (DCC)*, 2017, pp. 73–82.

[20] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[21] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph pca hashing for similarity search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2033–2044, Sept 2017.

[22] Q. Ning, J. Zhu, Z. Zhong, S. C. H. Hoi, and C. Chen, "Scalable image retrieval by sparse product quantization," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 586–597, 2017.

[23] Y. Xia, K. He, P. Kohli, and J. Sun, "Sparse projections for high-dimensional binary codes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3332–3339.

[24] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[25] L. Jin, X. Lan, W. Jiang, W. Jiang, N. Zheng, and W. Ying, "Online variable coding length product quantization for fast nearest neighbor search in mobile retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 559–570, 2017.

[26] S. Liu, J. Shao, and H. Lu, "Generalized residual vector quantization and aggregating tree for large scale search," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1785–1797, Aug 2017.

[27] T. Zhang, G.-J. Qi, J. Tang, and J. Wang, "Sparse composite quantization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4548–4556.

[28] Y. Cao, M. Long, J. Wang, and S. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1328–1337.

[29] J. Wang, J. Wang, J. Song, X.-S. Xu, H. T. Shen, and S. Li, "Optimized cartesian k-means," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 180–192, 2015.

[30] Y. Gong, S. Kumar, H. Rowley, S. Lazebnik *et al.*, "Learning binary codes for high-dimensional data using bilinear projections," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 484–491.

[31] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[32] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 1509–1517.

[33] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems (NIPS)*, 2009, pp. 1753–1760.

[34] Z. Wang, L.-Y. Duan, J. Yuan, T. Huang, and W. Gao, "To project more or to quantize more: minimizing reconstruction bias for learning compact binary codes," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 2016, pp. 2181–2188.

[35] L. Liu, M. Yu, and L. Shao, "Learning short binary codes for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1289–1299, 2017.

[36] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2938–2945.

[37] Y. Guo, G. Ding, L. Liu, J. Han, and L. Shao, "Learning to hash with optimized anchor embedding for scalable retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1344–1354, March 2017.

[38] X. Shi, F. Xing, J. Cai, Z. Zhang, Y. Xie, and L. Yang, "Kernel-based supervised discrete hashing for image retrieval," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 419–433.

[39] H. Liu, R. Ji, Y. Wu, and F. Huang, "Ordinal constrained binary code learning for nearest neighbor search," *arXiv preprint arXiv:1611.06362*, 2016.

[40] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization for approximate nearest neighbor search," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 2946–2953.

[41] T. Zhang, C. Du, and J. Wang, "Composite quantization for approximate nearest neighbor search." in *Proceedings of the International Conference on Machine Learning (ICML)*, no. 2, 2014, pp. 838–846.

[42] X. Wang, T. Zhang, G.-J. Qi, J. Tang, and J. Wang, "Supervised quantization for similarity search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2018–2026.

[43] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning." in *Proceedings of the AAAI Conference on Artificial Intelligence(AAAI)*, vol. 1, 2014, p. 2.

[44] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1183–1192.

[45] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2475–2483.

[46] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 392–407.

[47] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3384–3391.

[48] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2003, pp. 1470–1477.

[49] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Processing*, vol. 93, no. 8, pp. 2316–2327, 2013.

[50] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face track descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1693–1700.

[51] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[52] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[53] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, "Global and local structure preserving sparse subspace learning: an iterative approach to unsupervised feature selection," *Pattern Recognition*, vol. 53, pp. 87–101, 2016.

[54] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, M. Makar, and B. Girod, "Feature matching performance of compact descriptors for visual search," in *Data Compression Conference (DCC)*. IEEE, 2014, pp. 3–12.

[55] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 304–317.

[56] M. Norouzi and D. J. Fleet, "Cartesian k-means," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 3017–3024.

[57] M. Bauml, M. Tapaswi, and R. Stiefelhagen, "Semi-supervised learning with constraints for person identification in multimedia data," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 3602–3609.

[58] Y. Li, R. Wang, S. Shan, and X. Chen, "Hierarchical hybrid statistic based video binary code and its application to face retrieval in tv-series," in *In Proceedings of IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE, 2015, pp. 1–8.

[59] Q.-Y. Jiang and W.-J. Li, "Scalable graph hashing with feature transformation," in *Proceedings of the Twenty-Fourth international joint conference on Artificial Intelligence (IJCAI)*, 2015, pp. 2248–2254.

[60] W. Kong and W.-J. Li, "Double-bit quantization for hashing," in *AAAI Conference on Artificial Intelligence*, 2012.

[61] F. X. Yu, S. Kumar, Y. Gong, and S.-F. Chang, "Circulant binary embedding," in *Proceedings of the 31th international conference on machine learning (ICML)*, 2014, pp. 946–954.

[62] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–45.

[63] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879V2*, 2016.

[64] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2006, pp. 2161–2168.

[65] M. M. Esmaeili, R. K. Ward, and M. Fatourechi, "A fast approximate nearest neighbor search algorithm in the hamming space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2481–2488, 2012.

[66] M. Norouzi, A. Punjani, and D. J. Fleet, "Fast search in hamming space with multi-index hashing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3108–3115.