9-2016

Fast covariant VLAD for image search

Wan-Lei ZHAO

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Hanzi WANG

Fast Covariant VLAD for Image Search

Wan-Lei Zhao, Chong-Wah Ngo, and Hanzi Wang, *Senior Member, IEEE*

***Abstract*—Vector of locally aggregated descriptor (VLAD) is a popular image encoding approach for its simplicity and better scalability over conventional bag-of-visual-word approach. In order to enhance its distinctiveness and geometric invariance, covariant VLAD (CVLAD) is proposed to pool local features based on their dominant orientations/characteristic scales, which leads to a geometric-aware representation. This representation achieves rotation/scale invariance when being associated with circular matching. However, the circular matching induces several times of computation overhead, which makes CVLAD hardly suitable for large-scale retrieval tasks. In this paper, the issue of computation overhead is alleviated by performing the circular matching in CVLAD's frequency domain. In addition, by operating PCA on CVLAD in its frequency domain, much better scalability is achieved than when it is undertaken in the original feature space. Furthermore, the high-dimensional CVLAD subvectors are converted to dozens of very low-dimensional subvectors, which is possible when transforming the feature into its frequency domain. Nearest neighbor search is therefore undertaken on very low-dimensional subspaces, which becomes easily tractable. The effectiveness of our approach is demonstrated in the retrieval scenario on popular benchmarks comprising up to 1 million database images.**

***Index Terms*—Circular matching, covariant pooling, covariant vector of locally aggregated descriptor (CVLAD), similar image search.**

I. Introduction

As the bandwidth accessible to average users is increasing, multimedia data, in particular images and videos, become the fastest growing data type in Internet. Especially with the popularity of social media, there has been exponential growth in images and videos available on the Web. Among these huge volumes of images and videos, there exist large amount of similar images or exact copies [1]. The need of instant search for these similar contents arises from several contexts such as copyright reinfringement detection [2], e-commerce and land-mark identification [3]. In addition, similar image/video search also plays very important role in data-driven image/video annotation [4]–[6].

Manuscript received October 10, 2015; revised March 16, 2016 and May 05, 2016; accepted June 20, 2016. Date of publication June 24, 2016; date of current version August 12, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61572408 and Grant 61472334, and by the Research Grants Council of the Hong Kong Special Administrative Region, China under Grant CityU 118812. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Marco Bertini. *(Corresponding author: Hanzi Wang.)*

W.-L. Zhao and H. Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Fujian 361005, China (e-mail: wlzhao@xmu.edu.cn; hanzi.wang@xmu.edu.cn).

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: cscwngo@cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2585023

Thanks to the introduction of discriminative image local features such as SIFT [7] and SURF [8], large scale image retrieval has witnessed a sequel of breakthroughs in the last decade. Bag-of-words (BoW) framework [9], [10] is popularly adopted for quantizing local features extracted from images into vectors of visual words for image search. Large size vocabulary is trained for quantization, while inverted file [11], [12] is employed to index visual words for efficient image similarity comparison. Due to the side effect of quantization, two types of typical errors, namely mismatch and false match of visual words, will be introduced during comparison.

Several attempts have been made to improve the performance of BoW [13]–[17], mostly focusing on pruning false matches of visual word by post-processing such as visual [13], [17], geometric [13]–[15] verifications or query expansion [16], [17]. However, post-processing results in heavy burden of large memory requirement when scaling up the size of reference set to billions of images. For instance, BoW representation alone could occupy around 6G bytes of memory for one million images. If Hamming embedding signatures [13] for post-processing is further incorporated, the memory consumption increases to 18G bytes.

By using different feature encoding schemes than vector quantization, better performance in scalability has been reported recently. Representative approaches include the Fisher Vector [18]–[20], VLAD [21] and its variants such as VLAT [22] and VLAD$\otimes$ [23], [24]. VLAD can be viewed as a simplified representation of Fisher vector [21]. The basic idea of these approaches is to aggregate local features of an image into a lengthy dense vector followed by dimensionality reduction. By doing so, each feature can be compressed into few dozens of bytes, scaling up efficient search to few hundred millions of images. In addition to retrieval, superior performance is also reported in [20] for visual object classification.

The advantages of aforementioned approaches over BoW are three folds. First, relatively small vocabulary is required, which significantly cuts short the time for quantization. Second, the generated features are compatible with tools such as linear SVM [18] and PCA [21], as Fisher kernel is built upon generative model [25]. Finally, these approaches transform a variable size of features into a fixed length vector. This is especially interesting when local features are densely extracted to achieve better coverage over image content. In contrast, dense feature is hardly compatible with BoW in image retrieval as the inverted files become no longer efficient for dense BoW vector.

In brief, Fisher Vector approaches aggregate image local features into different visual words. When comparing two images, features aggregated into the same visual word are compared. As revealed in [26], [27], this amounts to cross-matching features from two images that are quantized into the same visual word. It is therefore imaginable that correct feature matches

are mixed with large amount of false matches since visual or geometric verification as [13] are no longer applicable. Recent approaches [23], [24], [28]–[30] intend to incorporate orientation coherence constraint to reduce undesired feature matches. In these works, the dominant orientation/characteristic scale is treated as a pooling variable to reduce irrelevant matches. In [28], [29], the dominant orientation of feature points has been quantized on different granularity at the feature representation stage. To achieve rotation/scale invariance, during the matching, the optimization procedure in the classifier searches for the best match between features pooled across different orientation bins. Similarly in [30], the proposed covariant VLAD (or CVLAD) achieves rotation/scale invariance by allowing features that are pooled into different orientation/scale bins to be best matched.

This paper aims at addressing the scalability issue that most of current image/video retrieval systems face. The main contribution of this paper is on improving CVLAD in terms of feature robustness, scalability and generation speed. First, CVLAD [30] is sped up by carrying out circular matching directly in the frequency domain than its original feature space. Second, by operating in frequency domain, PCA can be more effectively applied to Discrete Cosine Transformed (DCT) feature than in the original feature space. Third, CVLAD in frequency domain can be readily converted into a series of sub-vectors in very low-dimensional space. This property is appealing because large-scale nearest neighbor search (NNS) become easily tractable on low dimensional data. Furthermore, parallel computing framework, such as Mapreduce [31], can be employed to distributedly process these sub-vectors for large-scale retrieval.

The remaining of the paper is organized as follows. Section II reviews state-of-the-art works in image search. Section III gives a brief overview about VLAD* [32] and CVLAD [30], serving as a basis for discussions in later sections. Section IV presents the efficient operation of circular matching in frequency domain, and more effective way of applying PCA on the transformed features. Section V further details feature indexing with *product quantizer* (PQ) [33], the state-of-the-art data structure for NNS. Finally, Section VI presents intensive experiments to validate the improvements on several benchmark datasets.

II. Related Work

Global image signatures are known to be less tolerant to geometric and photometric transformations. Local features such as SIFT [7] and SURF [8], which are robust to various content changes such as rotation, scaling and even occlusion and background clutter, are demonstrated to be well-suitable for visual matching. To speed up matching between local features of two images, BoW [9], [13] is popularly adopted for feature representation despite heavy consumption of memory space. BoW takes more than 1K bytes to represent a medium-size image, and is therefore hardly scalable to image set of more than few millions. Several attempts have been made to reduce the memory consumption [34], [35], whereas it is still hard to achieve a good trade-off between search quality and memory cost.

Apart from BoW approach, recent works [18]–[20], [36] propose more concise but still discriminative representation, namely Fisher Kernel approach. In this approach, each visual word is represented by a GMM (Gaussian mixture model) component. The set of features (e.g., SIFT or SURF), denoted as $\mathcal{X}$, from an image are characterized by the following gradient vector:

$$\bigtriangledown \log p(\mathcal{X}|\lambda) \tag{1}$$

where $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ has n feature points and λ is the parameter set characterizing GMM. Typically, this gradient vector can be divided into three sub-vectors by its parameter types: weight (w_i), mean ($\mathbf{c}_i$) and variance (σ_i) ($i = 1 \cdots k$), where k is the number of Gaussians or visual words. For instance, given $\gamma_t(i)$ as the soft assignment of descriptor $\mathbf{x}_t$ to ith Gaussian, the gradient on $\mathbf{c}_i$ is given as

$$\frac{\partial L(\mathcal{X}|\lambda)}{\partial \mathbf{c}_i^d} = \frac{1}{n\sqrt{w_i}} \sum_{t=1}^{n} \gamma_t(i) \frac{\mathbf{x}_t - \mathbf{c}_i}{\sigma_i} \tag{2}$$

where d is the dimension of feature $\mathbf{x}_t$. In the complete Fisher Kernel approach, three sub-vectors defined on w_i, $\mathbf{c}_i$ and σ_i are concatenated as the feature representation.

Interestingly, VLAD [36] is shown to achieve similar or even better performance [21], despite being a simplified version of (2) as following:

$$\frac{\partial L(\mathcal{X}|\lambda)}{\partial \mathbf{c}_i^d} \approx \sum_{\mathbf{x} \in \mathcal{X}: q(\mathbf{x}) = \mathbf{c}_i} \mathbf{x} - \mathbf{c}_i \tag{3}$$

where $q(\mathbf{x})$ is a vector quantizer. The advantage of VLAD over Fisher vector is that it leads to more compact encoding on feature set $\mathcal{X}$. PCA and the compression-based indexing structure [33] jointly reduce the feature dimension further, such that VLAD requires at least one order of magnitude lower memory than that of BoW, while achieving even better search quality. Several pre-processing and post-processing steps [32], [37], [38] have been recently proposed to boost the performance.

Due to its superior performance, VLAD encoding scheme has also been applied on sub-image level, which allows fast object search/classification and localization [27], [39]. Particularly, in [27] sub-vector of VLAD is further quantized to support efficient dot products calculation between VLAD vectors. Alternatively, the matching of sub-image level pooled VLAD can also be sped-up by the branch-and-bound search framework [40]. In our paper, the focus is on the enhancement of the scalability of VLAD encoding scheme and the design of suitable indexing structure for large-scale search task. The schemes proposed in this paper could be complementary for visual instance search in large-scale.

As a new trend, similar image search has been addressed recently by using features trained with convolutional networks (ConvNets) [41], [42]. An image is encoded by a high-dimensional (e.g., 256 dimensions) vector which is extracted from the deep layer of ConvNets. These approaches demonstrate very promising results on several evaluation benchmarks. The focus of these works is on the design of distinctive features. While the NNS indexing on this well-designed feature, which is another indispensable part of an image search system, remains unaddressed.

In this paper, the problems of feature representation and indexing in large-scale are jointly addressed. Firstly, CVLAD [30]

is sped-up by performing circular matching in its frequency domain. Moreover, taking the advantage of the special structure of DCT transformed CVLAD, the NNS indexing is operated on very low-dimensional sub-vectors, which leads to high scalability for the proposed fast CVLAD.

III. Background: VLAD* and Covariant VLAD

This section starts with a brief review of VLAD* [32] and CVLAD [30], which serves as a basis for subsequent Sections IV and V. The schemes introduced by VLAD* will be fully incorporated into the design of CVLAD.

A. VLAD* Baseline

VLAD* is proposed in [32] mainly for addressing the burstiness issue in VLAD. The proposal includes several pre and post processing operations as following.

1) RootSIFT, which performs square-rooting on the the (positive) components of the SIFT feature, is adopted for its superior performance in retrieval [43].
2) ℓ_2-normalization on the residue before feature aggregation, i.e., (3) is rewritten as

$$\mathbf{v}_i = \sum_{\mathbf{x}\in\mathcal{X}:q(\mathbf{x})=\mathbf{c}_i} \frac{\mathbf{x}-\mathbf{c}_i}{\|\mathbf{x}-\mathbf{c}_i\|_2}. \tag{4}$$

   Note that ℓ_2-normalization is not originally proposed in [32]. We employ the normalization in this paper for it leads to better performance.
3) Power-law normalization is applied to scale $\mathbf{v}_i$ as

$$\mathbf{v}_{i,j} := \mathsf{sign}(\mathbf{v}_{i,j})\times|\mathbf{v}_{i,j}|^{\alpha} \tag{5}$$

   where α is a constant in the range of $(0, 1]$, which is fixed to 0.2 in all our experiments. The normalization is shown to be particularly effective in reducing the negative effect of visual bursts [21].
4) PCA is employed for rotating feature descriptor prior to feature aggregation. Note that dimensionality reduction could be detrimental and thus is not performed during feature rotation.

Other schemes that aim to boost the performance of VLAD have also been proposed, such as using multiple vocabularies to reduce the quantization noise [44] or introducing a per-cell normalization strategy instead of power-law [37]. We do not consider these complementary schemes in the paper, although we mention that they cover other aspects of VLAD and should be complementary with the approach introduced in our paper.

B. Covariant VLAD

Measuring similarity between two VLADs is comparable to a series of cross-matching between two sets of aggregated features [26]. As features with different dominant orientation are aggregated together, estimation of geometric transformation between two features becomes impossible. As a result, these globalized features introduce either too much invariance when using orientation-invariant or scale-invariant features, or there is no invariance if non-oriented features are densely extracted. This is in contrast with matching techniques such as weak geometric constraint (WGC) [13], which incorporates feature-point-level geometrical information.

Covariant VLAD (CVLAD) [30] was proposed to address this problem by pooling features according to their characteristic geometrical quantities. The quantities are characteristic scales and dominant orientations [7] which are obtained as byproducts of the feature extraction.

Take pooling on dominant orientation as an example, given θ as the dominant orientation associated with a given feature x, and let

$$b_B(\theta) = \left\lfloor B\frac{\theta}{2\pi} \right\rfloor \tag{6}$$

be the quantization function used to quantize angles with B equally sized bins. The pooling strategy modifies (3) as

$$p_{b,i} = \sum_{\mathbf{x}\in\mathcal{X}:q(\mathbf{x})=\mathbf{c}_i \wedge b_B(\theta)=b} \mathbf{x}-\mathbf{c}_i. \tag{7}$$

In (7), the pooling of the feature $\mathbf{x}$ is controlled by both its quantization index $q(\mathbf{x})$ and its quantized dominant orientation $b_B(\theta)$. The resulting CVLAD, $P = [\mathbf{p}_1, \ldots, \mathbf{p}_B]$ can be viewed as a concatenation of B numbers of VLAD vectors. Each sub-vector encodes the features having the same quantized dominant orientation, producing a vector B times longer. VLAD* is incorporated here by performing the series of pre and post processing on each of the sub-vector separately. Pooling on characteristic scale can be performed similarly. However, as reported in [28], [30], [45], the performance is not as effective as pooling on orientation, and therefore is not considered in this paper.

C. Naive Circular Matching

The similarity $\mathcal{S}(\cdot,\cdot)$ between two CVLAD vectors P and Q is defined on the basis of VLAD* sub-vectors as

$$\mathcal{S}(P,Q) = \max_{\Delta t\in 0\ldots B-1} \sum_{t=0}^{B-1} \{\mathbf{p}_t, \mathbf{q}_{\mathrm{mod}(t+\Delta t,B)}\}. \tag{8}$$

Equation (8) amounts to selecting the orientation maximizing the similarity between the two vectors. This process is comparable to estimating the dominant rotation transformation between two feature sets in WGC [46]. The major difference is that CVLAD performs the estimation on aggregated vectors rather than on histograms of dominant orientations. Note that the method also allows us to restrict the comparison to a subset of possible rotations. Fig. 1(a) visualizes this matching process. Compared to VLAD*, circular matching incurs more computational overhead depending on the number of quantization bins B. As depicted in Fig. 1(b), the CVLAD descriptor Q circularly shifts for eight times to search for the best match, implying eight times overhead in computation.

Circular matching also gives rise to two possible ways of applying PCA, either holistically to the CVLAD vector or partially to its sub-vectors. The former has the deficiency that the resulting rotation might not be optimal, as features pooled from different orientation bins will jam into each other. For the latter, applying PCA separately to each sub-vector is not possible,

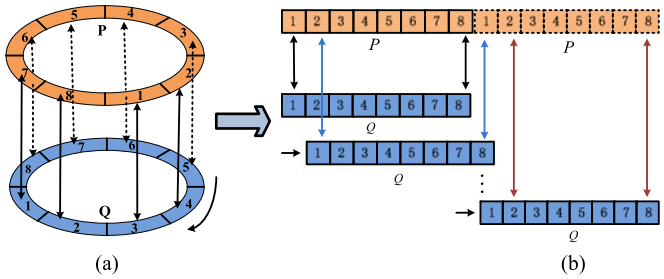


Fig. 1. Illustration of naive circular matching between two CVLAD vectors. The given vector on the upper side is fixed to its position, while the vector on the lower side shifts one sub-vector forward each time to search for the best match between two CVLAD vectors. Circular matching in (a) can be interpreted as 1D correlation on sub-vector level as shown in (b).

because sub-vectors of different orientation is not comparable after the projection. Instead, a universal PCA mapping has to be trained for all the sub-vectors in different orientations. Although feasible, the solution is also sub-optimal as the structure latent in each orientation will be diluted by the universal mapping. In the next section, we will show that performing PCA directly on the frequency domain is a more realistic solution.

IV. Fast Covariant VLAD: CVLAD$^+$

A. Transform CVLAD Into Frequency Domain

As discussed in Section III, the CVLAD vector is a concatenation of VLAD* sub-vectors. Since the similarity for CVLAD is defined on sub-vector basis, to facilitate our discussion, CVLAD vector is viewed as a $B \times n$ matrix, viz $P = [\mathbf{p}_1, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_B]^T \in \mathbb{R}^{B \times n}$, where n is the dimension of sub-vector $\mathbf{p}_i$. In order to find the best match between two CVLAD vectors, (8) performs a circular matching. This circular matching is nothing more than correlation that is undertaken between two series of vectors. Based on the relation between convolution and correlation [47], given that two CVLAD vectors $P, Q \in \mathbb{R}^{B \times n}$ and their column notations $P = [P^T_{(:,1)}, \ldots, P^T_{(:,n)}]^T$ and $Q = [Q^T_{(:,1)}, \ldots, Q^T_{(:,n)}]^T$, (8) is rewritten as cyclic convolution form [48], [49]

$$\mathcal{S}(P, Q) = \max[s_1, \ldots, s_B]$$
$$\text{where } [s_1, \ldots, s_B] = \sum_{i=1}^{n} P_{(:,i)} \circledast Q_{(:,i)}. \quad (9)$$

$\circledast$ is the cyclic convolution operator.

In one round of correlation [illustrated in Fig. 1(b)], $Q_{(:,i)}$ shifts one dimension forward each time and repeats B times, which results in a vector of B dimensions.

According to [49], cyclic convolution between two vectors of equal length can be converted to element wise multiplication if $P_{(:,i)}$ and $Q_{(:,i)}$ have been transformed into frequency domain. Before the transformation, CVLAD vector (consists of B sub-vectors) is reorganized into n sub-vectors [Step 1 in Fig. 2(a)]. This amounts to transposing P and Q. DCT or *Discrete Fourier Transform* (DFT) is performed on each of these B-dimensional sub-vectors. The resulting vector is composed of n numbers of B-dimensional sub-vectors. This process is visualized as Step 1 and 2 in Fig. 2(a). CVLAD after DCT/DFT is updated to a new name: CVLAD$^+$.

As will be seen in Section VI-B, the performance of DCT-transformed CVLAD is close to the performance of DFT-transformed CVLAD. Employing DCT instead of DFT is due to the consideration of computational cost and convenience. In DFT, keeping complex vectors and performing Hermitian product will basically double the overall cost of speed and space for similarity computation. Furthermore, only adhoc solution available to index complex vector with product quantizer [48], in which a d-dimensional complex vector is treated as 2d-dimensional real vector. In constrast, the use of DCT could avoid such complexity. Moreover, DCT actually causes no information loss during the transformation [50]. For convenience, only DCT is referred in the following discussion. However, to the same context, DFT fits as well.

After the transformation, naive circular matching in (8) is rewritten as element wise multiplication in the frequency domain

$$\mathcal{S}(P, Q) = \max[s_1, \ldots, s_B]$$
$$\text{where } [s_1, \ldots, s_B] = \sum_{i=1}^{n} \mathcal{T}^{-1}(\mathcal{T}(P_{(:,i)}) \odot \mathcal{T}(Q_{(:,i)})). \quad (10)$$

$\mathcal{T}(\cdot)$ and $\mathcal{T}^{-1}(\cdot)$ in (10) are 1D DCT and inverse DCT respectively. $\odot$ performs element wise multiplication between $\mathcal{T}(P_{(:,i)})$ and $\mathcal{T}(Q_{(:,i)})$. Due to the linearity of Fourier operator, (10) is rewritten as following such that only one inverse operation is required

$$\mathcal{S}(P, Q) = \max\left(\mathcal{T}^{-1}\left(\sum_{i=1}^{n} \mathcal{P}_{(:,i)} \odot \mathcal{Q}_{(:,i)}\right)\right)$$
$$\text{where } \mathcal{P}_{(:,i)} = \mathcal{T}(P_{(:,i)}), \mathcal{Q}_{(:,i)} = \mathcal{T}(Q_{(:,i)}), \mathcal{P}, \mathcal{Q} \in \mathbb{R}^{B \times n}. \quad (11)$$

Given $\{\mathcal{P}_{(j,:)}\}_{j=1\cdots B}$ and $\{\mathcal{Q}_{(j,:)}\}_{j=1\cdots B}$ are the row notations of P and Q respectively after DCT, (11) actually performs inner-product between $\mathcal{P}_{(j,:)}$ and $\mathcal{Q}_{(j,:)}$. As a result, above equation is rewritten as

$$\mathcal{S}(\mathbf{P}, \mathbf{Q}) = \max(\mathcal{T}^{-1}(\{\mathcal{P}_{(j,:)}, \mathcal{Q}_{(j,:)}\}_{j=1\cdots B})). \quad (12)$$

In (12), $\max(\cdot)$ operator is on B scalar values, which are the result of inner products on B pairs of sub-vectors. Comparing (12) to (8), no sub-vector shifting is required. As a result, the computation overhead due to B times shifting has been alleviated. However, one sub-vector $\mathcal{P}_{(j,:)}$ is still in several thousand dimensions, when doing NSS, the dimensional complexity is still very high even after dimension reduction. To address this issue, we propose an alternative similarity measure between P and Q, which performs inner-product directly between $\mathcal{P}_{(:,i)}$ and $\mathcal{Q}_{(:,i)}$. In addition, the $\max(\cdot)$ operator is replaced by $\Sigma\cdot$.

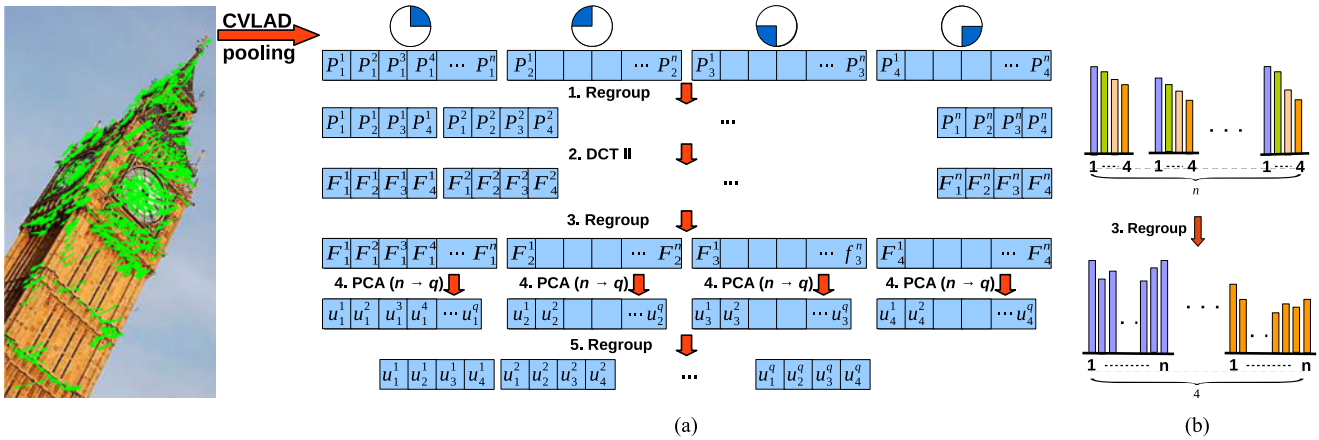


Fig. 2. Illustration on the flow of transforming CVLAD to CVLAD$^+$. (a) shows the general steps for the transformation. Step 1 and Step 2 transform CVLAD to CVLAD$^+$ by DCT, while Steps 3–5 perform PCA mapping on CVLAD$^+$. (b) details the regroup operation of Step 3 in (a). Signals from the same frequency are regrouped into one sub-vector. For the sake of clarity, we use four pooling orientations, while in practice, eight orientations are used.

This new similarity measure is given in (13)

$$\hat{\mathcal{S}}(P,Q) = \sum_{i=1}^{n} \{\mathcal{P}_{(:,i)}, \mathcal{Q}_{(:,i)}\}$$
$$\text{where } \mathcal{P}, \mathcal{Q} \in \mathbb{R}^{B\times n}. \tag{13}$$

As one can see, the similarity score in (13) is defined in the frequency domain. Basically, the inner-product in (13) measures how similar the energy distributions of two signals are across different frequencies. Although $\hat{\mathcal{S}}(P,Q)$ returns different similarity score from (12), it is sufficient for retrieval task as long as the rankings produced by them are similar. Notice that the full rotation invariance that is held by (12) is undermined in (13). However, as emprically revealed later, (13) still shows satisfactory performance as the rotation transformation is in presence. Although (12) and (13) involve the same number of operations, the latter leads to much lower dimensional complexity in viewing the fact that $B \ll n$.

Notice that when measuring similarity between CVLAD$^+$s in their frequency domain, the comparison between two high-dimensional vectors is converted to a series of comparisons between pairs of low-dimensional vectors. With such a low dimensionality, the NNS search problem becomes easily tractable. Furthermore, NNS can be potentially undertaken in parallel as the similarity score in (13) is the summation of hundreds of inner-products.

B. PCA on CVLAD$^+$

After transforming CVLAD to its frequency domain, the computation overhead have been largely alleviated. However, the memory complexity remains considerably high considering that CVLAD$^+$ is a high dimensional vector. For this reason, the dimension reduction is still necessary.

An intuitive way of doing this in frequency domain is by applying PCA to each $\mathcal{P}_{(:,i)}$ and $\mathcal{Q}_{(:,i)}$ in (13). Nevertheless, the effect will not be prominent, given that $\mathcal{P}_{(:,i)}$ comprises B elements from different frequencies. Instead, we re-group the vector elements such that PCA is applied to vectors composed of elements from the same frequency. Concretely, PCA is applied on column vector $\mathcal{P}_{(j,:)}$ of DCT transformed P. Accordingly, given column vector $\mathcal{P}_{(j,:)}$ has been projected from n to q dimensions, (12) is re-written as

$$\mathcal{S}(P,Q) = \max\left(\mathcal{T}^{-1}(\{PCA(\mathcal{P}_{(j,:)}), PCA(\mathcal{Q}_{(j,:)})\}_{j=1\cdots B})\right) \tag{14}$$

which performs inner products B times between two series of sub-vectors from P and Q. Note that (14) and (12) are equivalent if PCA is not considered. Similarly, (13) remains largely the same except that the inner products are conducted on q instead of n numbers of sub-vectors.

Discussion: While implementing circular matching in frequency domain is straightforward, the careful design of CVLAD$^+$ enables two missing peculiarities in other variants of VLAD. First, PCA is performed in the frequency domain, and keeps the significant components of each frequency channel individually, regardless of whether a channel is in high or low frequency. This design is in contrast to conventional way of compression that simply strips away high frequency components. Second, a fundamental difference from other variants of VLAD is that CVLAD$^+$ has B different PCA mapping matrices each for a frequency channel (versus one matrix only in CVLAD), ensuring that the structures latent in each orientation bin could be fully explored by PCA. In short, CVLAD$^+$ has additional plus over CVLAD in the way that special design is taken into account such that PCA can be suitably implemented and fully exploited. This design is critically significant in turning CVLAD$^+$ a highly scalable feature than CVLAD as will be empirically shown in the experimental section.

V. Indexing CVLAD$^+$ With Product Quantizer

In the large-scale image search task, to deploy the features like VLAD and CVLAD$^+$, which are dense and in high dimension, it is necessary to employ the multi-dimensional NNS structure for indexing. There are several off-the-shelf techniques, such as ANN [51], E2LSH [52], FLANN [53] and product quantizer

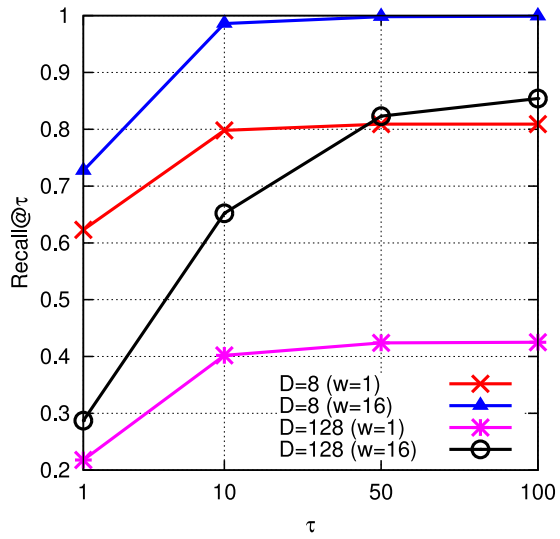


Fig. 3. NNS performance of PQ on 8-dimensional and 128-dimensional data of 1 million. The IVFPQ for both is constructed by using 8192 coarse quantizers and 256 product quantizers. The performances of considering the first nearest coarse quantizer ($w = 1$) and of considering top-16 nearest coarse quantizers ($w = 16$) are presented. Following the convention in [33], the recalls at top-τ ($\tau = 1, 10, 50, 100$) of NNS results are studied.

TABLE I
STATISTICS ON THE FOUR EVALUATION DATASETS

Dataset	Size	Number of queries
Holidays [13]	1492	500
Paris [54]	6692	55
Oxford5K [12]	5063	55
Oxford105K [12]	105,063	55

TABLE II
PERFORMANCE (MAP) OF CVLAD ON HOLIDAYS, OXFORD5K, AND PARIS DATASETS

Methods	Holidays		Oxford5K		Paris	
	HesAff	ODns	HesAff	ODns	HesAff	ODns
Fisher	66.0	73.9	33.0	28.7	38.7	34.0
VLAD*	70.6	75.2	41.5	42.6	44.1	43.5
CVLAD	78.6	82.3	49.6	55.0	**52.1**	52.1
CVLAD$\odot^+$	77.9	80.9	**49.8**	**56.0**	51.7	53.0
CVLAD$\circledast^+$	**79.3**	**83.2**	48.9	55.4	50.9	52.3
CVLAD$^+$	74.7	77.9	49.5	55.2	51.7	**53.6**

(PQ) [33], that could be employed for indexing CVLAD$^+$. This paper adopts PQ for its simplicity and superior performance in retrieval. In PQ, a vector is encoded by multiple product quantizers in lower dimensional space. Note that PQ is lossy and indexing subjects to an error function as following. Let D be the dimension of a vector, m and k be the number and vocabulary size of product-quantizer respectively, the error ε induced by PQ is bounded by

$$\varepsilon \propto \frac{D}{m \cdot k}. \tag{15}$$

Usually k is set to few hundreds (typically $k = 256$) to trade-off between NNS quality and efficiency in memory and speed. As seen in (15), if the value of k is fixed, the lower the ratio D/m, the smaller the error will be. However, in order to achieve high memory efficiency, m cannot be set to arbitrarily large, which corresponds to the number of bytes to encode a vector. As a result, similar to other indexing structures, PQ also shows better performance on low-dimensional data than that of high-dimensional cases, as k and m are fixed to certain range.

Fig. 3 verifies our analysis by showing NSS on SIFT1M dataset [33]. One million of vectors are used for testing the performance difference when the vectors are in 8 and 128 dimensions. As shown in the figure, when all other parameters are fixed, PQ demonstrates much better performance on low-dimensional data.

Based on above analysis, similarity measure introduced in (13) is favored over (14) since the inner-product is performed between B-dimensional sub-vectors in (13). This is in contrast to (14), which performs inner product between high dimensional sub-vectors based on the fact $B \ll q$.

After PCA, CVLAD$^+$ feature is composed of q numbers of sub-vector, i.e., $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^i, \ldots, \mathbf{u}^q\}$ [as shown in Fig. 2(a)]. According to (13), the similarity between two CVLAD$^+$s are the aggregation of inner product between sub-vectors. IVFADC for CVLAD$^+$ is therefore built on sub-vector level. In other words, IVFADC is constructed in each sub-vector space for CVLAD$^+$. One can imagine that q numbers of IVFADCs are constructed in total. Given that a series of product quantizers ϕ_j^i ($j = 1 \cdots m$) are trained in sub-space i and $d(.,.)$ is the l_2 distance measure, the inner product [in (13)] between one sub-vector v^i from query and one sub-vector $\mathbf{u}^i$ from reference image is approximated as

$$\{\mathbf{v}^i, \mathbf{u}^i\} \approx 1 - \frac{\sum_{j=1}^{m} d^2(v_j^i, \phi_j^i(u_j^i))}{2}$$
$$\text{where } \|\mathbf{v}^i\|_2 = 1 \text{ and } \|\mathbf{u}^i\|_2 = 1. \tag{16}$$

As a result, the overall similarity between query and a reference image is the summation of q inner products between the sub-vectors. Notice that the dimension of sub-vector $\mathbf{u}^i$ is very low (typically *8*), according to the analysis presented above, good approximation to the true similarity between $\mathbf{v}^i$ and $\mathbf{u}^i$ is expected. The advantage of using (13) instead of (14) is not only limited to its low dimension complexity but also makes it possible to conduct the NNS in high parallel. As the query with (16) is conducted on each sub-space independently, the whole search process fits very well to the popular MapReduce framework [31].

In the paper, PQ indexing is also adopted for Fisher vector and VLAD*, which are to be compared with. In particular, asymmetric distance computation (ADC) with the support of inverted files (IVFADC) is adopted for all these features. Following common practice, IVFADC is applied on these features after they have been undergone dimension reduction by PCA. As to Fisher vector and VLAD*, only one IVFADC is constructed for each.

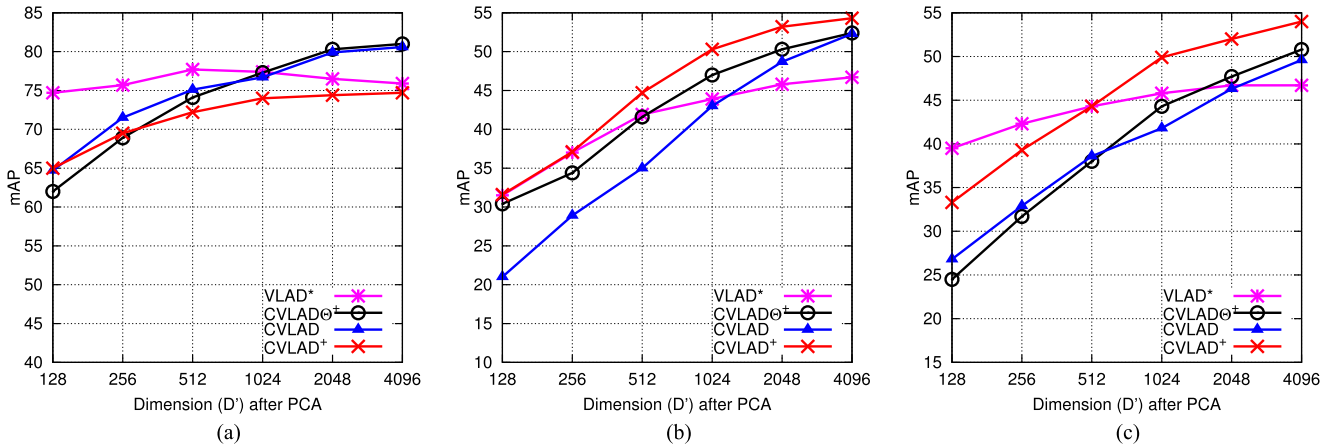


Fig. 4. Effect of dimensionality reduction by PCA on mAP performance with ODns features. Similar performance trends are observed with HesAff features. (a) Holidays. (b) Oxford5K. (c) Paris.

TABLE III
PERFORMANCE (MAP) OF CVLAD VARIANTS WHEN PCA IS APPLIED TO COMPRESS THE FEATURE FROM 65, 536 TO 512 DIMENSIONS

Methods	Holidays		Oxford5K		Paris	
	HesAff	ODns	HesAff	ODns	HesAff	ODns
VLAD*	**66.4**	**77.7**	**35.9**	41.9	39.1	44.3
CVLAD	64.8	75.1	31.5	35.0	33.7	38.6
CVLAD$\odot^+$	64.1	73.2	33.9	40.8	34.6	36.9
CVLAD$^+$	65.1	72.2	35.4	**44.7**	**39.8**	**44.7**

VI. EXPERIMENTS

This section evaluates the performance of CVLAD$^+$ on three benchmark datasets, in comparison to several baselines and state-of-the-art techniques. The empirical studies examine the degree of performance fluctuations in response to dimensionality reduction, increase in database size, and the employment of PQ.

A. Datasets and Evaluation Protocol

The evaluation is conducted on four datasets: Holidays [13], Paris [54], Oxford5K [12] and Oxford105K [12]. Table I lists the number of images and queries on each dataset. For scalability test, the one million images in Flickr1M [55] are used as distractors for the former three datasets. As for preprocessing, each image is resized to 512 pixels on the longer side without altering the aspect ratio. Two local features are employed: Hessian-Affine (HesAff) [56] and Oriented-Dense (ODns). On average, each image has 1519 features extracted by HesAff. The sampling rate for ODns is one feature per five pixels along the x and y directions.[1] The sampling is conducted on one scale only. The dominant orientation is estimated with the way proposed in [7]. Both types of features are represented with RootSIFT descriptors [43]. In the evaluation, all the Fisher approaches such as Fisher Vector, VLAD*, CVLAD, CVLAD$^+$ share the same vocabulary for the same feature. The vocabularies are trained on Flickr60K [13]. The vocabulary size is 64 and is fixed for all the experiments. We adopt mAP (mean Average Precision) as the evaluation measure.

[1]The sampling rate in [30] is 7 pixels. The denser sampling rate and the employment of (4) make the results of VLAD* and CVLAD in this paper slightly better.

B. Performance Comparison

We compare four different variants of CVLAD: CVLAD$^+$ (13), CVLAD$\odot^+$ (12), CVLAD$\circledast^+$ (12) and CVLAD [30], and two baselines: VLAD* [32] and Fisher vector [19]. CVLAD$\odot^+$ is the same as CVLAD$\circledast^+$ except that the former transforms CVLAD with DCT, while the latter uses DFT. Note that PCA is not applied for feature compression and PQ is not adopted for indexing in this experiment. Same as [30], the pooling orientation is set to 8. Table II summarizes the performances of these approaches on three datasets across two types of features. CVLAD and its variants consistently outperform VLAD and Fisher vector. The performance of CVLAD$\odot^+$ and CVLAD$\circledast^+$ is fairly close to that of CVLAD, though not exactly the same. More importantly, CVLAD$^+$, which derives matching score directly in the frequency domain, only suffers slight performance drop. On Paris dataset, the performances are even better than CVLAD when using ODns for feature extraction. As discussed in [30], dense sampling does not perform better than region detector if the extracted features are aggregated with VLAD. However, using CVLAD$^+$ with a fairly tiny vocabulary size (64) achieves result close to the best reported result in [46] on Holidays (mAP = 81.3) which uses a large vocabulary of size 20 000 and region detector.

In terms of speed efficiency, CVLAD is slower than CVLAD$^+$ and CVLAD$\odot^+$ by about 7.5 times, based on brute-force search on Holidays+Flickr1M dataset. CVLAD$^+$ and CVLAD$\odot^+$, nevertheless, do not significantly differ in speed. The 'max' operator in (12) is computationally cheaper than $\sum \cdot$ operation in (13), which compensates the time cost for inverse DCT. As a result, calculating similarity for CVLAD$\odot^+$ is as efficient as CVLAD$^+$. Due to costly Hermitian product

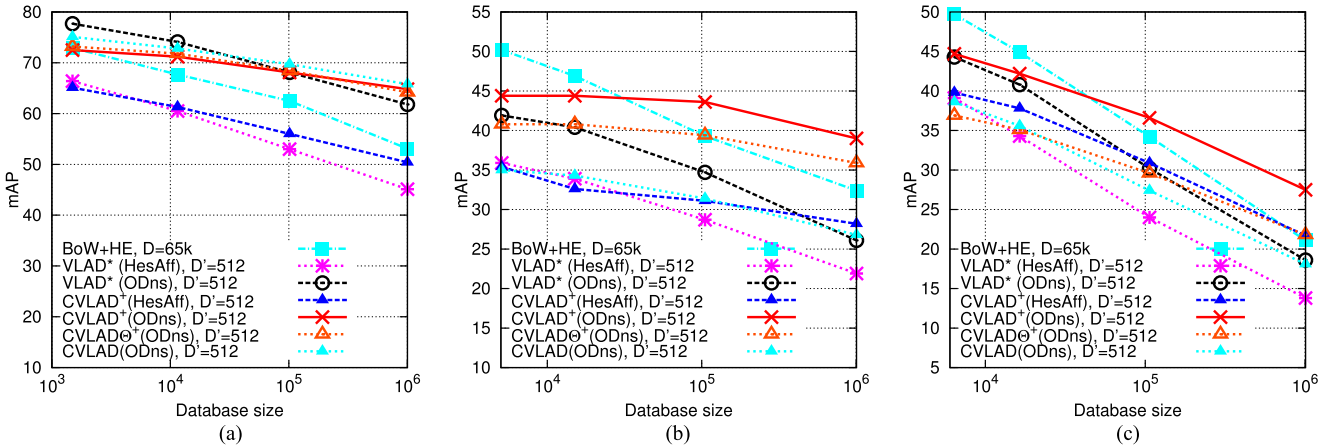


Fig. 5. Scalability peformance when data size is increased to 1 million images. Note that all features (except BoW+HE) are compressed to 512 dimensions. (a) Holidays+1M. (b) Oxford5K+1M. (c) Paris+1M.

on complex vectors, the retrieval cost of CVLAD$\circledast^+$ is tripled over that of CVLAD$\odot^+$ and CVLAD$^+$.

Next, we study the effect of PCA on CVLAD, CVLAD$^+$ and CVLAD$\odot^+$. Note that only one mapping matrix is learnt respectively for VLAD* and CVLAD when applying PCA. In contrast, there are 8 matrices being learnt for CVLAD$^+$ and CVLAD$\odot^+$, where each corresponds to a sub-vector in the frequency domain. The only difference between CVLAD$^+$ and CVLAD$\odot^+$ lies in the similarity measure. Fig. 4 shows the performance difference, where CVLAD$^+$ outperforms CVLAD with large margin on two datasets. Taking ODns feature on Oxford5k as example, CVLAD$^+$ in 256 dimensions already exhibits better mAP than CVLAD with 512 dimensions. We attribute the performance gain of CVLAD$^+$ to the way that features are organized such that PCA can be separately applied on each subspace to achieve the overall less information loss but with higher compression rate. Note that the compression rate of VLAD* (e.g., from 8, 192 to 512 dimensions) is eight times lower than CVLAD and CVLAD$^+$ (from 8×8192 to 512 dimensions). When these features are mapped to very low dimension, VLAD* shows better performance. Until the compression reaches to certain ratio, the performance of VLAD* is bypassed by other approaches. For example, CVLAD$^+$ outperforms VLAD* when dimension exceeds 256 and 512 respectively on Oxford5k and Paris datasets. On Holidays dataset, the performance gap gets closer with the increase of dimension between CVLAD$^+$ and VLAD*. Table III lists the detailed result when the feature is reduced to 512 dimensions.

C. Scalability Test

This section studies the change in mAP with respect to the increase of dataset size from few thousands to one million images. We compare the performance of CVLAD$^+$, CVLAD$\odot^+$ with CVLAD, VLAD* and Hamming Embedding (BoW+HE) [46]. For fairer comparison, CVLAD$^+$, CVLAD$\odot^+$, CVLAD and VLAD* are all compressed to 512 dimensions, since their performances are closer when $d = 512$ as observed in Fig. 5(a)–(c). A visual vocabulary of size 65 K is learnt for BoW+HE, using HesAff as feature extractor and RootSIFT as descriptors. No indexing structure is used in the experiment.

Fig. 5 compares their mAPs across three datasets, where Flickr1M images are gradually included as distractors. As the data size increases, the mAPs of all approaches drop. The degree of performance degradation is observed to be less severe for CVLAD$^+$ and CVLAD$\odot^+$ compared to other approaches, especially on Oxford5k+1M and Paris+1M datasets. The result clearly shows the advantage of applying PCA in frequency domain, despite undergoing eight times higher compression rate than VLAD*. Meanwhile, CVLAD$^+$ demonstrates clearly much better performance over CVLAD$\odot^+$ on two datasets. This indicates that inner-product defined on shorter sub-vector level is able to reflect more minor differences between two vectors, which is particularly helpful when feature loses its distinctiveness after PCA. With much higher dimension, BoW+HE exhibits the best performance when data size is small. Similar to VLAD*, nevertheless, BoW+HE degrades rapidly with larger data size and performs significantly worse than CVLAD$^+$ on dataset with one million images.

Fig. 6 displays the similarity distribution of similar and dissimilar image pairs from four different features. As shown in the figure, the similarity distributions of similiar image pairs are actually similar for different features. The advantage of CVLAD$\odot^+$ and CVLAD$^+$ is that they make similarities between the dissimilar images more concentrating to 0. This is more apparent for CVLAD$^+$, which makes it more robust to noises.

D. Feature Indexing

This section experiments the performance of CVLAD$^+$ in terms of retrieval precision and speed efficiency when PQ, more specifically the indexing structure IVFADC, is employed. We compare CVLAD$^+$ against VLAD*, Fisher vector, BoW+HE and BoW. IVFADC is also applied for VLAD* and Fisher vector. While for BoW+HE and BoW, inverted index is used. Similar to the previous experiment, CVLAD$^+$, VLAD* and Fisher are compressed to 512 dimensions. For CVLAD$^+$, there are 64 IV-

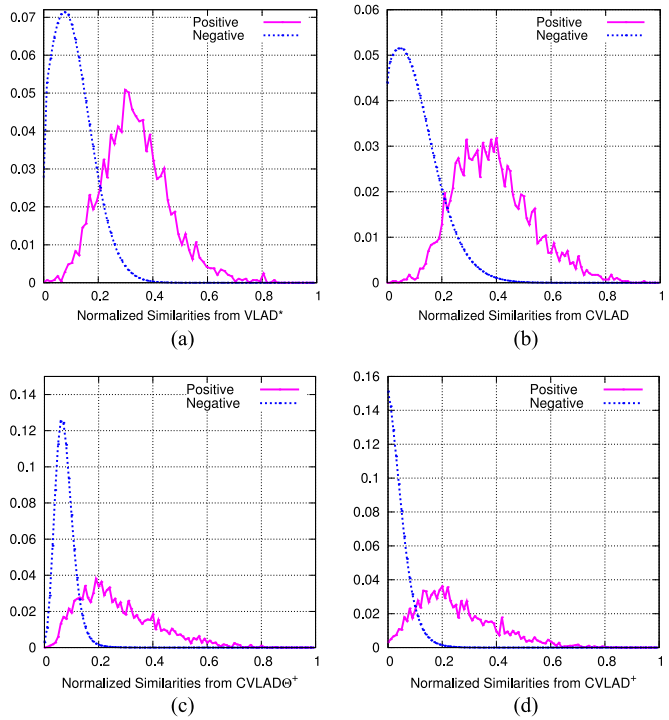


Fig. 6. Similarity distribution of similar (positive) and dissimilar (negative) image pairs from VLAD*, CVLAD, CVLAD$\odot^+$, and CVLAD$^+$. Distances are collected when retrievals on Holidays and Oxford5k are conducted with 100K distractors. Note that features are PCA mapped to 512 dimensions in this experiment. (a) VLAD*. (b) CVLAD. (c) CVLAD$\odot^+$. (d) CVLAD$^+$.

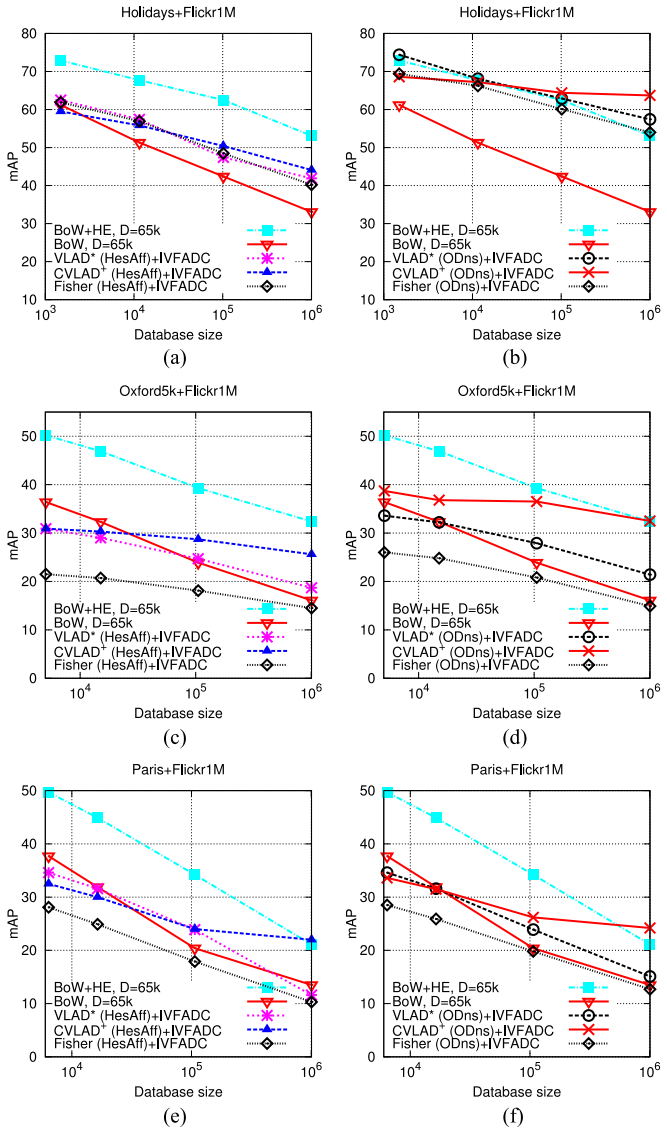


Fig. 7. Effect of indexing structure. CVLAD$^+$, VLAD*, and Fisher use IVFADC, while BoW and BoW+HE use inverted file. (a) HesAff. (b) ODns. (c) HesAff. (d) ODns. (e) HesAff. (f) ODns.

FADCs constructed, each for a sub-vector of 8 dimensions. While for VLAD* and Fisher, only one IVFADC structure is built respectively. For these three approaches, the numbers of coarse and PQ quantizers are set to 8192 and 256 respectively. Note that the complexity of querying 64 8-dimensional IVFADCs is almost identical to that of querying one 512-dimensional IVFADC, since similar number of operations is involved.

Fig. 7 shows the performance comparison on three datasets. CVLAD$^+$ maintains fairly similar performance when the scale of data size increases to one million. The degree of performance drop is less than BoW and BoW+HE, which do not suffer from lossy compression as three other approaches due to the use of PQ. The scalability of CVLAD$^+$ is not achieved by VLAD* or Fisher vector. We attribute the superior performance of CVLAD$^+$ to its feature discriminability and low complexity in NNS. More concretely, NNS is performed in 8-dimensional space versus VLAD* and Fisher in 512-dimensional space.

Table IV lists the average speed of querying one image using different approaches on Holidays+Flickr1M dataset. All the approaches are implemented in C++ and experimented on a standard PC with 2.4 GHz CPU and 32G memory. As shown in the table, the querying time is about 1 second for all the approaches except for BoW+HE. Denote $\bar{n}$ as the average number of local features per image, k as vocabulary size and N as the number of reference images, the complexity of query processing is$O(\frac{\bar{n}^2}{k}\cdot N)$ for BoW and BoW+HE. While for Fisher approaches, the dimension of the aggregated feature plays a critical role. With the support of IVFADC, for VLAD*, Fisher and CVLAD$^+$, the complexity of processing one image query is $O(\frac{D}{k}\cdot N)$, where D is the feature dimension after PCA. As the three approaches use the same number of dimensions and size of vocabulary, the difference in querying time is indeed insignificant. CVLAD$^+$ has an additional advantage, nevertheless, as

TABLE IV
AVERAGE TIME COST (S) PER QUERY ON HOLIDAYS+FLICKR1M

Method	BoW	BoW+HE	VLAD*	Fisher	CVLAD$^+$
Time cost	1.060	1.867	0.992	0.992	1.029

Approaches such as VLAD*, Fisher and CVLAD$^+$ are supported by IVFADC indexing structure.

TABLE V
PERFORMANCE (MAP) OF CVLAD$^+$ VERSUS COMPARABLE APPROACHES OF THE STATE OF THE ART

Method	k	D	Hol.	Ox5k	Ox105k
BoW [46]‡	200k	200,000	58.3	39.1	-
VLAD [21]‡	64	4,096	55.6	37.8	-
Fisher [21]‡	64	4,096	59.5	41.8	-
VLAD*+LCS+RN [32]‡	64	8,192	65.8	51.7	45.6
VLAD*+LCS+RN [32]‡	64	→128	-	32.2	26.2
VLAD-intra [37]‡	256	32,536	65.3	55.8	-
VLAD-intra [37]‡	256	→128	62.5	44.8	-
PVLAT [22]‡	64	8,192	66.4	54.2	-
PVLAT [22]‡	64	→256	60.6	-	-
VLAD⊗ [23], [24]‡	32	28,672	**81.0**	**61.8**	53.9
VLAD⊗ [23], [24]‡	32	→1024	-	-	40.7
CVLAD$^+$ (HesAff)	64	65,536	74.7	49.5	47.1
CVLAD$^+$ (HesAff)	64	→512	65.5	35.5	33.0
CVLAD$^+$ (ODns)	64	65,536	77.9	55.2	**54.9**
CVLAD$^+$ (ODns)	64	→512	72.2	44.7	42.9

‡: numbers are cited directly from the referred papers.

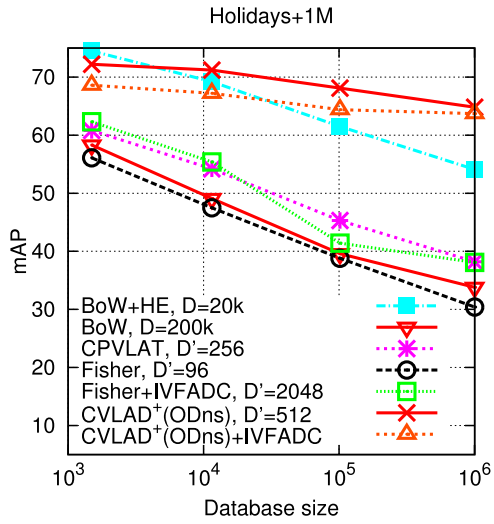


Fig. 8. Performance of CVLAD$^+$ in comparison with the state-of-the-art approaches in the large-scale image search task. The performance of state-of-the-art approaches are cited directly from the papers referred.

querying of 64 IVFADC structures can be run in parallel, which can potentially introduce further speed-up.

E. Comparison With State-of-the-Art Approaches

Table V compares the performance of CVLAD$^+$ with the best reported results in the literature. CVLAD$^+$ clearly outperforms most of the approaches including Fisher [21], VLAD* [32], VLAD-intra [37] and PVLAT [22]. The performance of CVLAD$^+$ is comparable to VLAD⊗, where the former achieves the best mAP on Holidays and Oxford105k datasets, while the latter attains the best result on Oxfor5K dataset. However, VLAD⊗ turns out to be incompatible with PCA rotation and power-law normalization (RN) scheme as discussed in Section III. In order to integrate RN with VLAD⊗, the VLAD⊗ query has to be rotated eight times. As a result, similar as CVLAD, eight more computational overhead is incurred. Fig. 8 further shows the scalability of some approaches. In contrast to other approaches, CVLAD$^+$ is the only approach that can maintain performance without noticeable drop in mAP when increasing the data size from few thousand to one million images.

Table VI details the memory consumption of different approaches. CVLAD$^+$ needs 256 bytes of storage space per image. Among them, 64 bytes are used for feature encoding, while the remaining 3×64 bytes are for PQ. Specifically, each image is indexed by its ID which takes 3 bytes. Since CVLAD$^+$ maintains 64 IVFADC structures, a total of 192 bytes is required per image. VLAD* needs 64 + 3 bytes per image since using only one IVFADC for indexing. The extra space requirement of CVLAD$^+$, however, is traded off by nearly 7% mAP improvement than VLAD*. Furthermore, compared to other approaches such as BoW+HE, it consumes 65 times less storage space while achieving significantly better mAP. Overall, CVLAD$^+$ demonstrates excellent compromise between storage space and retrieval performance.

TABLE VI
MEMORY EFFICIENCY OF CVLAD$^+$ IN COMPARISON WITH STATE-OF-THE-ART APPROACHES ON HOLIDAYS+FLICKR1M

Method	k	D	Mem. cost per image (bytes)	mAP
BoW [46]‡	200k	200k	8,885	33.8
BoW+HE [46]	65k	65k	18,228	51.7
miniBoF [35]‡	1k	1k	640	24.4
Fisher+IVFADC [21]‡	256	4,096	384	38.1
CPVLAT [22]‡	64	512	67	33.4
VLAD*(HesAff)+IVFADC	64	512	67	41.9
VLAD*(ODns)+IVFADC	64	512	67	57.4
CVLAD$^+$ (HesAff)+IVFADC	64	512	256	44.1
CVLAD$^+$ (ODns)+IVFADC	64	512	256	63.7

‡: numbers are cited directly from the referred papers.

VII. CONCLUSION

We have presented fast CVLAD feature representation, viz CVLAD$^+$, for similar image search, on which the circular matching associated with CVLAD has been replaced by a series of inner products between sub-vectors. In such a way, the computation overhead caused by circular matching is alleviated. In the meantime, good discriminativeness of CVLAD feature is still well preserved. CVLAD$^+$ feature particularly shows satisfactory performance when being combined with Oriented-Dense feature.

Moreover, the careful design of CVLAD$^+$ enables two missing peculiarities in other variants of VLAD. First, PCA is performed in the frequency domain, and keeps the significant components of each frequency channel individually. This design is different from conventional way of compression that simply strips away high frequency components. Second, a fundamental difference from other variants of VLAD is that CVLAD$^+$ has B different PCA mapping matrices each for a frequency channel (versus one matrix only in CVLAD), ensuring that the

structures latent in each orientation bin could be fully explored by PCA. In the experiments, high scalability has been observed in the large-scale image search task. CVLAD$^+$ outperforms BoW+HE considerably on million level reference sets while using 65 times less memory. To the best of our knowledge, this is the highest scalability that is ever achieved in recent works. With such a good trade-off between search quality and memory efficiency, CVLAD$^+$ is naturally extensible to large-scale content-based video retrieval and visual object classification.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks to Dr. H. Jégou from INRIA, Rocquencourt, France, for his insightful comments on the paper. Great thanks are extended to Prof. Z. Liu from Shanghai University, Shanghai, China, for many valuable discussions about DCT.

REFERENCES

[1] X.-J. Wang, L. Zhang, M. Liu, Y. Li, and W.-Y. Ma, "ARISTA—image search to annotation on billions of web photos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 2987–2994.

[2] J. Law-to *et al.*, "Video and image copy detection demo," in *Proc. 6th ACM Int. Conf. Image Video Retrieval*, 2007, pp. 97–100.

[3] D. M. Chen, G. Baatz, and K. Koser, "City-scale landmark identification on mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 737–744.

[4] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of web videos by efficient near-duplicate search," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 448–461, Aug. 2010.

[5] Y.-H. Kuo, W.-H. Cheng, H.-T. Lin, and W. Hsu, "Unsupervised semantic feature discovery for image object retrieval and tag refinement," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1079–1090, Jul. 2012.

[6] E. Moxley, T. Mei, and B. S. Manjunath, "Video annotation through search and graph reinforcement mining," *IEEE Trans. Multimedia*, vol. 12, no. 3, pp. 183–193, Apr. 2010.

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[9] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[10] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop Statist. Learn. Comput. Vis.*, 2004, pp. 1–22.

[11] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2006, pp. 2161–2168.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.

[13] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 304–317.

[14] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial duplicate web image retrieval," in *Proc. 18th ACN Int. Conf. Multimedia*, 2010, pp. 511–520.

[15] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 25–32.

[16] H. Xie, Y. Zhang, J. Tan, L. Guo, and J. Li, "Contextual query expansion for image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1104–1114, Feb. 2014.

[17] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 648–659, Mar. 2015.

[18] F. Perronnin and C. R. Dance, "Fisher Kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.

[19] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 143–156.

[20] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 76.1–76.12.

[21] H. Jégou *et al.*, "Aggregating local descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[22] R. Negrel, D. Picard, and P.-H. Gosselin, "Web scale image retrieval using compact tensor aggregation of visual descriptors," *IEEE Multimedia*, vol. 20, no. 3, pp. 24–33, Jul./Sep. 2013.

[23] G. Tolias, T. Furon, and H. Jégou, "Orientation covariant aggregation of local descriptors with embeddings," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 382–397.

[24] G. Tolias, A. Bursuc, T. Furon, and H. Jegou, "Rotation and translation covariant match kernels for image retrieval," *Comput. Vis. Image Understanding*, vol. 33, no. 20, pp. 1–12, Jul. 2015.

[25] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, pp. 487–493.

[26] Y. Avrithis, G. Tolias, and H. Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1401–1408.

[27] R. Tao, E. Gavves, K. van de Sande, C. Snoek, and A. Smeulders, "Locality in generic instance search from one example," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 2099–2106.

[28] P. Koniusz and K. Mikolajczyk, "Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 661–664.

[29] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Comput. Vis. Image Understand.*, vol. 17, no. 5, pp. 479–492, 2013.

[30] W.-L. Zhao, H. Jégou, and G. Gravier, "Oriented pooling for dense and non-dense rotation invariant features," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 99.1–99.11.

[31] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[32] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 653–656.

[33] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[34] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 17–24.

[35] H. Jégou, M. Douze, and C. Schmid, "Packing bag-of-features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2357–2364.

[36] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 3304–3311.

[37] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1578–1585.

[38] E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "A comprehensive study over VLAD and product quantization in large-scale image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1713–1728, Oct. 2014.

[39] Z. Li, E. Gavves, K. van de Sande, C. Snoek, and A. Smeulders, "Codemaps-segment, classify and search objects locally," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2136–2143.

[40] C. H. Lampert, M. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2129–2142, Dec. 2009.

[41] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 584–599.

[42] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual instance retrieval with deep convolutional networks," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6574

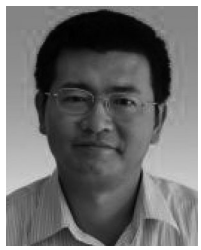
[43] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2911–2918.
[44] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 774–787.
[45] J. Sánchez, F. Perronnin, and T. de Campos, "Modeling the spatial layout of images beyond spatial pyramids," *Pattern Recog. Lett.*, vol. 33, no. 16, pp. 2216–2223, 2012.
[46] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 316–336, Feb. 2010.
[47] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.
[48] J. Revaud, M. Douze, C. Schmid, and H. Jegou, "Event retrieval in large video collections with circulant temporal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2459–2466.
[49] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*, 2nd ed. NEw York, NY, USA: Springer-Verlag, 1997.
[50] G. J. Miao and M. A. Clements, *Digital Signal Processing and Statistical Classification*. Norwood, MA, USA: Artech House, Jun. 2002.
[51] D. M. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching," Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, Jan. 2010. [Online]. Available: https://www.cs.umd.edu/mount/ANN/
[52] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. Cambridge, MA, USA: MIT Press, Mar. 2006, ch. 3.
[53] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. VISAPP Int. Conf. Comput. Vis. Theory Appl.*, Feb. 2009, pp. 331–340.
[54] 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. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[55] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inform. Retrieval*, 2008, pp. 39–43.
[56] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, Oct. 2004.

Wan-Lei Zhao received the B.Eng. and M.Eng. degrees in computer science and engineering from Yunnan University, Yunnan, China, in 2002 and 2006, respectively, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, China, in 2010.

He is currently with Xiamen University, Xiamen, China, as an Associate Professor. Before joining Xiamen University, he was a Postdoctoral Scholar with INRIA, Rocquencourt, France. His research interests include multimedia information retrieval and video processing.

Chong-Wah Ngo received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University, Singapore, in 1994 and 1996, respectively, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, China.

He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. Before joining City University, he was a Postdoctoral Scholar with the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL, USA. He was also a Visiting Researcher with Microsoft Research Asia, Beijing, China. His research interests include large-scale multimedia information retrieval, video computing, multimedia mining, and visualization.



Hanzi Wang (SM'05-SM'09) received the Ph.D. degree in computer vision from Monash University, Melbourne, VIC, Australia, in 2004.

He is currently a Distinguished Professor of Minjiang Scholars and a Founding Director of the Center for Pattern Analysis and Machine Intelligence, Xiamen University, Xiamen, China. His research interests include computer vision and pattern recognition, including visual tracking, robust statistics, and object detection.