A Compact Binary Aggregated Descriptor via Dual Selection for Visual Search

Yuwei Wu[1], Zhe Wang[1], Junsong Yuan[1], Lingyu Duan[2]
[1]School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
[2]School of Electronics Engineering and Computer Science, Institute of Digital Media, Peking University, Beijing, China
wuyuwei@bit.edu.cn; wangzhe@ntu.edu.sg;jsyuan@ntu.edu.sg;lingyu@pku.edu.cn

ABSTRACT

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. To enable fast search, the FV (or VLAD) is usually compressed by product quantization (PQ) or hashing. However, compressing high-dimensional descriptors via PQ or hashing may become intractable and infeasible due to both the storage and computation requirements for the linear/nonlinear projection of PQ or hashing methods. We develop a novel compact aggregated descriptor via dual selection for visual search. We utilize both sample-specific Gaussian component redundancy and bit dependency within a binary aggregated descriptor to produce its compact binary codes. The proposed method can effectively reduce the codesize of the raw aggregated descriptors, without degrading the search accuracy or introducing additional memory footprint. We demonstrate the significant advantages of the proposed binary codes in solving the approximate nearest neighbor (ANN) visual search problem. Experimental results on extensive datasets show that our method outperforms the state-of-the-art methods.

Keywords

Aggregated descriptors;Visual search; Dual selection; Compact binary code

1. INTRODUCTION

With the advent of the era of Big Data, huge body of resources can be easily found on the multimedia sharing platforms such as *YouTube*, *Facebook* and *Flickr*. Visual search is attracting considerable attention in the multimedia and computer vision literature. It refers to the discovery of images contained within a large dataset that describes the same objects as those depicted by query terms. There exists a wide variety of emerging applications, *e.g.,* searching buildings for location recognition and 3D reconstruction [13], searching logos for the estimation of brand exposure [22], and rapidly locating and tracking of criminal suspects from masses of surveillance videos [15]. 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 an effective and efficient visual search system, the visual descriptors need to be discriminative and resource-efficient (*e.g.*, low memory footprint). The discriminability determines the search accuracy, while the resource involves the scalability of a visual search system. In this paper, our goal is to design a new compact binary coding method which can achieve satisfactory search accuracy, search efficiency, and memory cost simultaneously.

More recently, various visual tasks are being performed on the larger and larger dataset, and visual search is no exception. To achieve high retrieval accuracy on such datasets, it is necessary to employ extremely high-dimensional descriptors such as the Fisher Vector (FV) [21] and the Vector of Locally Aggregated Descriptors (VLAD) [12]. In retrieval tasks, the FV (or VLAD) is usually compressed to binary aggregated descriptors because the binarized FV (or VLAD) allows for fast Hamming distance computation as well as light storage of visual descriptors [20, 2, 19, 16, 30].

Product quantization (PQ) [10, 18, 5, 29, 1, 23] and hashing [15, 6, 7, 17, 27, 26, 24] are popularly used techniques to obtain binary aggregated descriptors. In PQ and Hashing methods, the linear (or nonlinear) projection is usually employed to convert high dimensional features (*e.g.,* feature $\mathbf{f} \in \mathbb{R}^N$) into binary embeddings or product codes. However, both the space and computational complexity of the projection is $\mathcal{O}(N^2)$ [14, 6, 27]. For instance, assume that a 426-dimensional dense trajectory feature is extracted from a video and then PCA is employed to project it to the dimensionality 213, we will obtain a $54,528$-dimensional FV when the number of Gaussion components $K = 128$. In the case of $N = 54,528$, projection matrix alone will take more than 10 GB memory footprint and projecting one vector would spend 800 ms on a single core. As the number of Gaussion components K increases, binarizing high-dimensional descriptors directly using PQ or hashing may become intractable and infeasible due to the computational cost and memory requirements.

It is thus desirable that a discriminative and compact aggregated descriptor should be established for visual search over a large scale dataset when only limited hardware resources are available. To achieve this goal, we consider the following observations: (1) As discussed in [21], since each residual vector of the FV (or VLAD) is aggregated from local features being assigned to the corresponding Gaussian component, not all the components are of equal importance to distinguish a sample. (2) Although the redundancy from the component-level is removed, the bit-wise dependency within each selected component may be further reduced.

Motivated by these observations, we propose a novel compact binary coding method for visual search. A sample-specific component selection scheme is introduced to remove the redundant Gaus-

MM '16, October 15-19, 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2964284.2967256

sian components, and a global structure preserving sparse subspace learning method is presented to suppress the bit-wise dependency within a selected component. Specifically, given a raw Fisher vector with K Gaussian components, to fulfill the constraint of compression complexity, 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 then introduced to produce compact binary codes. The obtained binary codes, as required, are discriminative, and yet compact, capable of efficiently handling the large scale dataset, as validated in Section 5.

2. BINARY AGGREGATED DESCRIPTOR

In the FV method, a Gaussian Mixture model (GMM) $p_\lambda(x) = \sum_{i=1}^{K} \omega_i p_i(x)$ with K Gaussians 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. Employing PCA to project the dimensionality of SIFT or trajectory to dimension D is beneficial to the overall performance [11, 12]. By concatenating of 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 $\mathbf{g}(i) \in \mathbb{R}^{D}$. Similarly, the VLAD can be derived from FV by replacing the GMM soft clustering with k-means clustering [12].

Following the compressed fisher vectors[20], we choose a one-bit quantizer to binarize the high dimensional $\mathbf{g} \in \mathbb{R}^{KD}$, such that superior retrieval performance with 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 based on 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 (BAD) $\mathbf{b} = \{\mathbf{b}(1), ..., \mathbf{b}(K)\}$ with $N = KD$ bits, where $\mathbf{b}(i) \in \mathbb{R}^{D}$.

3. COMPACT BINARY AGGREGATED DESCRIPTOR VIA DUAL SELECTION

In this paper, our goal is to compress aggregated descriptors into binary codes, without sacrificing considerable loss of search accuracy. In addition, the encoding process should satisfy small memory footprint and low computational complexity. We formulate the compact binary coding as a resource-constrained (*e.g.,* bit rate, memory footprints and computational complexity) feature compression problem. Although the BAD method supports ultra-fast Hamming distance computation (XOR operation and bit count), it has a fixed codesize and thus cannot meet the given constraints. Moreover, the coarse binary quantization would probably drop the search accuracy of uncompressed aggregated descriptors. To address these issues, we propose a dual selection model which utilizes both sample-specific component selection and component-specific bit selection to explore the informative bits. 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 Fisher codes via dual selection.

3.1 Sample-specific Component Selection

We note that FV exhibits the natural 2-D structure due to the Gaussian component based feature aggregation, as shown in Figure 1. The aggregated FV is formed by concatenating residual vectors computed for 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 describing a sample. Assume that the occurrence of a Gaussian component is the number of local features quantized to that component. The occurrence of different Gaussian components may vary for each sample. Gaussian components with low occurrence are supposed to be less discriminative for describing the sample. In an extreme case, if none of local features is assigned to a Gaussian component i, then all the elements of the corresponding subvector $\mathbf{g}(i)$ are zero. Our method thus considers the sample-specific component selection within an individual sample, which uses local statistics to discard the redundant Gaussian components.

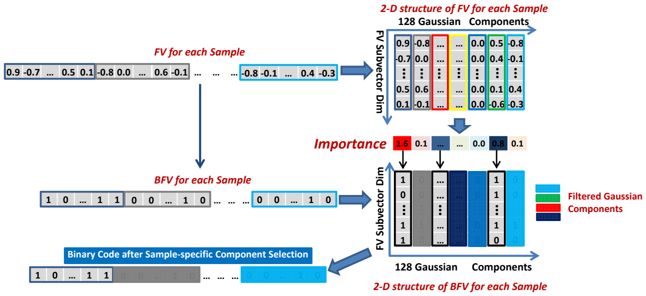

Figure 1: Sample-specific component selection. The aggregated FV is partitioned into disjoint components by Gaussian components. We select all the bits of Gaussian components with high *importance* and the rest of components are discarded. In this work, the *importance* is defined as the sum of posterior probabilities of local features being assigned to the i-th Gaussion component.

Specifically, we can partition the FV into disjoint components by Gaussian components and select all the bits of those with high importance. Here, the importance means that which Gaussian components are activated and how large the amplitudes of their responses are. In particular, we adopt the soft assignment $\gamma(\mathbf{f}_t, i)$ of local features as the importance of Gaussian i 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, given by

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

Sample-specific component selection can be implemented efficiently by applying a sorting algorithm to the set $\{I(\lambda_i), i = 1, 2, \cdots, K\}$, *i.e.,* 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, we can compute the importance of components for each sample using Eq. (1) and select a subset of components with high importance. The rest of components are discarded, as shown in Figure 1. The number of the selected components will be discussed in Section **??**. Since noisy components are removed from the original aggregated FV, the required number of shortlisted candidates can be largely reduced. Meanwhile, it is not necessary to maintain a Gaussian selection mask for all samples. Accordingly, the sample-specific component selection is memory free, and the computational cost of Gaussian selection is $\mathcal{O}(N)$.

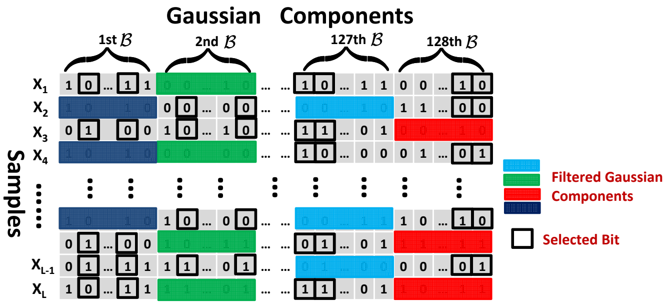

Figure 2: Componet-specific bit selection. The bits that carry as much information as possible are selected by the global structure preserving sparse subspace learning. For each sample, the bits in black bounding boxes are the final compact Fisher codes. L denotes the number of training samples.

3.2 Component-specific Bit Selection

After sample-specific component selection, although the redundancy from a component-level is removed, there probably exists bit-level redundancy within each selected component. Therefore, we need to further reduce the dimensionality of each component, while maintaining search performance. Inspired by [25], in this paper, we introduce a component-specific bit selection scheme by global structure preserving sparse subspace learning to select bits that carry as much information as possible.

Without loss of generality, let $\mathcal{B} \in \{0,1\}^{\Theta \times D}$ denote a subset of the BAD from the i-th Gaussian component corresponding to the whole dataset. Here Θ represents how many samples treating the i-th component as an *important* one, as shown in Figure 2. $\Theta \leq L$, 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 to the learned subspace spanned by the selected bits. Mathematically, the component-specific bit selection is formulated as

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

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 only one 1, *i.e.,* at most the D' bits are selected for each Gaussian component. $||\mathbf{W}\mathbf{1}_{D'\times 1}||_0 = D'$ guarantees that $\mathbf{W}$ has the D' nonzero rows, and thus exact D' bits will be selected. $\mathbf{H}$ is the optimal subspace. Hence, Eq.(2) denotes the distance of $\mathcal{B}$ to the learned subspace $\mathbf{H}$.

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

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} ||\mathcal{B} - \mathcal{B}\mathbf{W}\mathbf{H}||_F^2 + \beta||\mathbf{W}||_{2,1}, \\ &s.t.\mathbf{W} \in \mathbb{R}_+^{D\times D'} \end{aligned} \tag{3}$$

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

In this work, we employ the accelerated block coordinate update (ABCU) method [28] to alternatingly update $\mathbf{W}$ and $\mathbf{H}$ with

$$\begin{cases} f(\mathbf{W},\mathbf{H}) = ||\mathcal{B} - \mathcal{B}\mathbf{W}\mathbf{H}||_F^2 \\ g(\mathbf{W}) = \beta||\mathbf{W}||_{2,1}. \end{cases} \tag{4}$$

Due to the space limit, interested readers refer to [28] for details.

Normalizing each column of $\mathbf{W}$, we sort $||\mathbf{W}_{i,\cdot}||_2, i = 1, 2, \cdots, D$ and select bits corresponding to the D' largest ones . Since the Gaussian components are independent of each other, the component-specific bit selection can be implemented in a parallel fashion. In addition, the number of selected bits D' is fixed and applied to all samples, so both the memory footprint the computational cost of bits selection are $\mathcal{O}(N)$.

4. 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 selected components may vary in different samples, we have to address a challenging issue of matching descriptor 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 compared directly using standard Hamming distance. We adopt a normalized cosine similarity score Sc to calculate the distance matching, given by

$$Sc = \frac{\sum_{i=1}^{K} s_i^q s_i^r \Big(D' - 2 * h\big(sgn(\mathbf{g}_i^{\mathbf{x}_q}), sgn(\mathbf{g}_i^{\mathbf{x}_r})\big)\Big)}{D'\sqrt{||s^q||_0||s^r||_0}}, \tag{5}$$

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 $\mathbf{x}_r$.

5. 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.

5.1 Comparison with the state-of-the-art

We carry out retrieval experiments on MPEG-7 CDVS datasets and the publicly available INRIA Holidays dataset [9]. The MPEG-7 CDVS benchmark data set consists of 5 classes: graphics, paintings, video frames, landmarks, and common objects [4].

To fairly evaluate the performance over a large-scale dataset, we use *FLICKR1M* [8] as the distractor dataset, containing 1 million distractor images collected from Flickr. In the training phase, the codebook of GMM and component-specific bit selection matrix $\mathbf{W}$ are typically learned on the *FLICKR1M*. The retrieval performance is measured by mean Average Precision (mAP). We evaluate the proposed method against LSH [3], BPBC [6], PQ [10], RR+PQ [5, 18], ITQ [7] and BFV [1]. The SIFT from each image is extracted and then the dimensionality of SIFT is reduced to 64 by PCA. We employ FV encoding to aggregate the dim-reduced SIFT features

[1] In the BFV method, the raw FV is directly sign binarized to produce binary codes whose dimensionality is the same as the original FV, as described in Section 2.

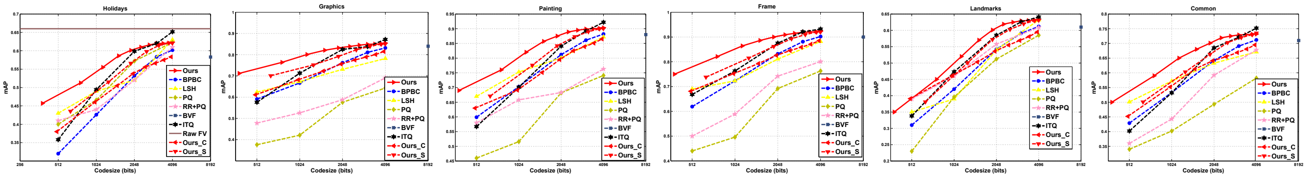

Figure 3: Retrieval performance in terms of mAP vs. descriptor codesize over various benchmark datasets. Note that we evaluate the uncompressed FV on the INRIA Holidays dataset. For MPEG CDVS datasets, we perform large scale retrieval by combining them with the 1 million FLICKR distractor dataset. In the BFV method, the raw FV is directly binarized by a sign function and thus the length of BFV is only 8192 (best viewed on high-resolution display).

Table 1: 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	93.6	$\mathcal{O}(NP)$	91.8	$\mathcal{O}(NP)$	151
BPBC	154.2	$\mathcal{O}(\sqrt{N}\sqrt{P})$	0.031	$\mathcal{O}(N\sqrt{P}+P\sqrt{N})$	17
PQ	65.5	$\mathcal{O}(NQ)$	8.4	$\mathcal{O}(NQ)$	25
RR+PQ	65.5	$\mathcal{O}(NQ+N^2)$	277	$\mathcal{O}(NQ+N^2)$	278
ITQ	256	$\mathcal{O}(N^2)$	254	$\mathcal{O}(N^2)$	257
Ours	374.5	$\mathcal{O}(N)$	0.015	$\mathcal{O}(N)$	< 1

with 128 Gaussian components. Therefore, the total dimensionality of FV is $128 \times 64 = 8192$. Note that for compared methods, we apply power law ($\alpha = 0.5$) followed by $L2$ normalization to the raw FV feature.

Figure. 3 presents an extensive comparison 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 computational expensive for $L2$ distance between uncompressed FV, we only present the retrieval accuracy of uncompressed FV on the INRIA Holidays dataset. We can see that RR+PQ obtains better mAP than PQ, but both of them perform significantly worse than the hashing algorithms (*e.g.,* LSH, ITQ and BPBC). The proposed method achieves superior accuracy than the other schemes, especially for small codesizes. Compared with the BFV, the codesize of our method can save about 3 times, while obtaining better retrieval performance. It should be emphasized that ITQ obtain best mAP for all datasets with large codesize such as 4096 bits. This is reasonable since ITQ is fine tuned for learning a projection to minimize the mean square error. However, ITQ suffers from the huge memory and computational footprints because of the projection matrix computation explained in Section 1.

5.2 Effectiveness of Dual Selection

In this section, we analyze two aspects of our method that are important for good retrieval results, *i.e.*, sample-specific component selection and component-specific bit selection. Note that the proposed method can be degraded to two simple models: Ours_S denotes an algorithm in which only the sample-specific component selection scheme is employed described in Section 3.1. Ours_C only uses the component-specific bit selection presented in Section 3.2.

From Figure 3, we observe that the Ours_S performs consistently better than Ours_C at the same codesize over all datasets (except the Landmarks at small codesize). In contrast, the proposed method significantly outperforms both Ours_S and Ours_C. For example, At codesize of 1920 bits, Ours_S and Ours_C yield mAP 79.2% and 75.6% on Graphics dataset, while our method achieves 83.2%. This gain may be attributed to the fact that the dual selection scheme is complementary to each other, which can select more informative bits. With more bits, Ours_S, Ours_C and the proposed schemes progressively improve the retrieval accuracy. In particular, Ours_S and our methods approach to the retrieval mAP of uncompressed FV at large codesize. They obtain mAP 62.1% at codesize of 3840 bits on the Holidays dataset, which is comparable with the mAP 66% of uncompressed FV.

5.3 Computational Complexity

We evaluate the compression complexity on the "Holidays + 1M Flickr" dataset [9]. Table 1 compares the compression ratio, memory and time complexities of the proposed method and other baselines, with comparable retrieval accuracy. With comparable retrieval mAP, we observe that the compression ratio of the proposed method is 2 to 6 times larger than the baseline schemes, resulting in much smaller compact codes. The memory footprint introduced by the 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, compression time of the proposed method is ultra fast, as there are only binarization and selection operations, while hashing methods and PQ involve a large amount of floating point multiplications.

6. CONCLUSION

In this paper, we have proposed an efficient and effective solution for learning compact binary codes to address the ANN problem with the long binary aggregated descriptor. We utilize both the sample-specific Gaussian component selection and the component-specific bit selection to produce a compact binary code. The proposed method exhibits extremely low compression memory and time complexity, and supports fast Hamming distance matching. Extensive experimental results demonstrate that our method has superior retrieval accuracy against the state-of-the-art methods such as hashing and PQ algorithms, while with fewer bits.

7. ACKNOWLEDGMENTS

This work was supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114 and the National Basic Research Program of China (973 Program): No. 2015CB351806.

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Ministers Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

8. REFERENCES

[1] A. Babenko and V. Lempitsky. Tree quantization for large-scale similarity search and classification. In *CVPR*, pages 4240–4248, 2015.

[2] 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. *SP*, 93(8):2316–2327, 2013.

[3] 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*, pages 253–262. ACM, 2004.

[4] L.-Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao. Overview of the mpeg-cdvs standard. *TIP*, 25(1):179–194, Jan 2016.

[5] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization for approximate nearest neighbor search. In *CVPR*, pages 2946–2953. IEEE, 2013.

[6] Y. Gong, S. Kumar, H. Rowley, S. Lazebnik, et al. Learning binary codes for high-dimensional data using bilinear projections. In *CVPR*, pages 484–491. IEEE, 2013.

[7] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *TPAMI*, 35(12):2916–2929, 2013.

[8] M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *ACM MIR*, pages 527–536, 2010.

[9] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317. Springer, 2008.

[10] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128, 2011.

[11] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311. IEEE, 2010.

[12] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.

[13] 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 TIST*, 6(1):1, 2015.

[14] P. Li, A. Shrivastava, J. L. Moore, and A. C. König. Hashing algorithms for large-scale learning. In *NIPS*, pages 2672–2680, 2011.

[15] 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 *CVPR*, pages 4758–4767, 2015.

[16] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao. Rate-adaptive compact fisher codes for mobile visual search. *SPL*, 21(2):195–198, 2014.

[17] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete graph hashing. In *NIPS*, pages 3419–3427, 2014.

[18] M. Norouzi and D. J. Fleet. Cartesian k-means. In *CVPR*, pages 3017–3024. IEEE, 2013.

[19] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *CVPR*, pages 1693–1700. IEEE, 2014.

[20] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391. IEEE, 2010.

[21] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[22] R. Tao, A. W. Smeulders, and S.-F. Chang. Attributes and categories for generic instance search from one example. In *CVPR*, pages 177–186, 2015.

[23] J. Wang, J. Wang, J. Song, X.-S. Xu, H. T. Shen, and S. Li. Optimized cartesian k-means. *TKDE*, 27(1):180–192, 2015.

[24] Q. Wang, L. Si, and B. Shen. Learning to hash on structured data. In *AAAI*, 2015.

[25] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu. Subspace learning for unsupervised feature selection via matrix factorization. *PR*, 48(1):10–19, 2015.

[26] Z. Wang, L.-Y. Duan, J. Lin, X. Wang, T. Huang, and W. Gao. Hamming compatible quantization for hashing. In *AAAI*, pages 2298–2304, 2015.

[27] Y. Xia, K. He, P. Kohli, and J. Sun. Sparse projections for high-dimensional binary codes. In *CVPR*, pages 3332–3339, 2015.

[28] 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*, 6(3):1758–1789, 2013.

[29] T. Zhang, G.-J. Qi, J. Tang, and J. Wang. Sparse composite quantization. In *CVPR*, pages 4548–4556, 2015.

[30] Y. Zhang, J. Wu, and J. Cai. Compact representation for image classification: To choose or to compress? In *CVPR*, pages 907–914. IEEE, 2014.