Efficient Binary Coding for Subspace-based Query-by-Image Video Retrieval

Ruicong Xu
Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China
ranranxu95@gmail.com

Yang Yang*
Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China
dlyyang@gmail.com

Fumin Shen
Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China
fumin.shen@gmail.com

Ning Xie
Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China
seanxiening@gmail.com

Heng Tao Shen
Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China
shenhengtao@hotmail.com

ABSTRACT

Subspace representations have been widely applied for videos in many tasks. In particular, the subspace-based query-by-image video retrieval (QBIVR), facing high challenges on similarity-preserving measurements and efficient retrieval schemes, urgently needs considerable research attention. In this paper, we propose a novel subspace-based QBIVR framework to enable efficient video search. We first define a new geometry-preserving distance metric to measure the image-to-video distance, which transforms the QBIVR task to be the Maximum Inner Product Search (MIPS) problem. The merit of this distance metric lies in that it helps to preserve the genuine geometric relationship between query images and database videos to the greatest extent. To boost the efficiency of solving the MIPS problem, we introduce two asymmetric hashing schemes which can bridge the domain gap of images and videos properly. The first approach, termed Inner-product Binary Coding (IBC), achieves high-quality binary codes by learning the binary codes and coding functions simultaneously without continuous relaxations. The other one, Bilinear Binary Coding (BBC) approach, employs compact bilinear projections instead of a single large projection matrix to further improve the retrieval efficiency. Extensive experiments on four real-world video datasets verify the effectiveness of our proposed approaches, as compared to the state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**;

*Corresponding author.

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123392

KEYWORDS

query-by-image; video retrieval; asymmetric hashing; geometry-preserving distance metric

1 INTRODUCTION

In recent years, we have witnessed a tremendous explosion of multimedia data (e.g., images, videos) on the Web driven by the advance of digital camera, high-speed Internet, massive storage, etc. Especially, as a significant role, videos expand their application value in reshaping the ways of recording daily life, self-expression and communication, which undoubtedly draw considerable research attention [37, 38]. Query-by-image video retrieval (QBIVR), which is an important type of video retrieval applied to a variety of real-life applications, ranges from searching video lectures using a slide, recommending relevant videos based on images, to searching news videos using photos, etc [3].

Undeniably, video representation is crucial in the QBIVR task. Considerable research endeavors [4, 11, 25, 28] have been dedicated to developing effective schemes for improving the global signature of the whole video. Promising methods in subspace representations such as single or mixture of linear subspaces [33], affine subspace [10], and covariance matrix [32], have demonstrated their superiorities in underpinning a range of multimedia applications. Subspace-based video representations are able to preserve the rich structural properties of videos such as viewpoint, location, spatial transformation, and movement, which are expected to be superior to a single point representation in a high-dimensional feature space. Therefore, this paper focuses on the QBIVR task based on the subspace representations of videos, which can also be seen as a *point-to-subspace retrieval* problem.

However, this task is highly challenging due to the difficulties in designing effective distance functions between images and videos and efficient retrieval schemes for big video datasets. Considering different representation types of images and videos (i.e., a single high-dimensional point for an image and a subspace representation for a video), the similarity measurement between an image and a video becomes non-trivial. One naive method is to compute the

similarity between the query image and each frame of the video and then integrate these similarities by averaging or taking the maximum. Obviously, this measurement suffers from high computational cost and massive storage, as well as ignores the correlations among video frames. Also, other popular angular measurements perform well only on linear subspace representations[12, 13], which are limited to a small scope of application.

Fortunately, aiming at large-scale datasets, powerful hashing-based Approximate Nearest Neighbor (ANN) techniques provide fast retrieval ideas [19, 42]. However, traditional hashing approaches cannot be directly applied due to the different representations of videos and images in the QBIVR task. By projecting each video into a single data point in the same high-dimensional space as images, Basri et al. [1] proposed an Approximate Nearest Subspace method to solve the problem. Then the point-to-subspace retrieval problem is reduced to the well-known point search problem which can be addressed by approximate nearest neighbor (ANN) techniques. Nonetheless, the performance is far from ideal due to the inevitable loss of geometric structural relationships between images and videos, resulting from the aggressive projection of videos. Hence, it is urged to have an effective retrieval strategy preserving the similarity of two distinct representations between a high-dimensional point (i.e., a query image) and a subspace representation (i.e., a database video).

In this paper, we propose a novel QBIVR framework, termed Binary Subspace Coding (BSC), which can fully explore the genuine geometric relationship between query images and database videos as well as provide significant efficiency improvements. In particular, we measure image-to-video similarity by calculating the ℓ_2 distance of a query image and its orthogonal projection in the subspace representing a database video, which transforms the point-to-subspace retrieval problem to be the Maximum Inner Product Search (MIPS) problem. To further accelerate search process and simplify the optimization, our framework employs the asymmetric learning strategy to generate different hash functions for images and videos, which can narrow their domain gap in the common Hamming space effectively and are efficient for high-dimensional computations. Specifically, two asymmetric hashing models are designed. We first propose an Inner-product Binary Coding (IBC) approach, which can preferably preserve image-to-video inner-relationships and guarantee high-quality binary codes by a tractable discrete optimization method. Moreover, we also devise a Bilinear Binary Coding (BBC) approach to significantly lower the computational cost by exploiting compact bilinear projections instead of a single yet large projection matrix.

We illustrate the flowchart of our BSC framework in Figure 1. The main contributions are summarized as follows:

- We devise a novel subspace-based QBIVR framework, termed *Binary Subspace Coding* (BSC). The image-to-video geometry-preserving distance metric we define can preferably preserve geometric relationships between images and videos.
- We propose two asymmetric hashing schemes to achieve efficient QBIVR. By effectively mapping images and videos into a common Hamming space, the domain gap between images and videos can be reduced and efficient retrieval is supported due to fast binary code comparisons.
- Superior experiment results on four datasets, i.e., BBT1, UQE50, FCVID and a micro video Vine dataset collected by ourselves demonstrate the effectiveness of our approaches, as compared to the state-of-the-art methods.

The reminder of this paper is organized as follows. Section 2 gives an introduction to the related work. In Section 3, we elaborate the details of our BSC framework and algorithm analysis. Section 4 demonstrates the experimental results and analysis, followed by the conclusion of this work in Section 5.

2 RELATED WORK

In this section, we give a brief view of previous literatures that are closely related to our work.

Recently, the QBIVR task, a special video retrieval type, is urgently in a need of performance boosting for its widespread applications. An informative video representation is undoubtedly necessary for the retrieval quality. Typically, shots indexing, sets aggregation and global signatures are three prevailing video representation methods. The first two suffer from the severe scarcity of intra-frame relationships and sensitivity of opting training sets respectively. However, global signatures could reserve pretty rich structure information, such as inter-frame and intra-frame relationships which contribute to video modelling pertinently and accurately. Also, the problems of high computational cost and unsatisfactory retrieval speed are accompanied. In [3], an integral representation method which reduces retrieval latency and memory requirements by Scalable Compressed Fisher Vectors (SCFV) is proposed. However, it also sacrifices a plethora of original spatial-temporal features which are crucial for integral representations. To measure the similarity between images and videos for the subspace-based QBIVR task, an existing method [1] is to project the affinity matrix of a video representation into a single data point so that the point-to-subspace retrieval problem can be easily solved by approximate nearest neighbors (ANN) search methods.

Research on hashing has been very active, to support fast ANN search [8, 9, 18, 35, 36]. Some supervised hashing methods such as Supervised Hashing with Kernels (KSH) [22], Minimal Loss Hashing (MLH) [24], Deep Hashing [16, 18, 21] have demonstrated promising performance in some applications with semantic labels, but it's cost-prohibited or even impossible to get semantic labels in many real-life applications. Besides, the learning process could be more complex and time-consuming than unsupervised learning techniques especially dealing with high-resolution videos or images. Some classical unsupervised methods include Iterative Quantization (ITQ) [8] focusing on minimizing quantization error during unsupervised training, Inductive Manifold Hashing (IMH) [31] adopting manifold learning techniques to better model the intrinsic structure embedded in the feature space, Anchor graph hashing (AGH) [23], and scalable graph hashing with feature transformation (SGH) [14] directly exploiting the similarity to guide the learning procedure.

Although the above hashing methods deal with the complexity of computational cost and storage efficiently, neglected modality problem may result in a severe domain gap. Cross-modal or cross-media hashing methods bring a new idea to the research.

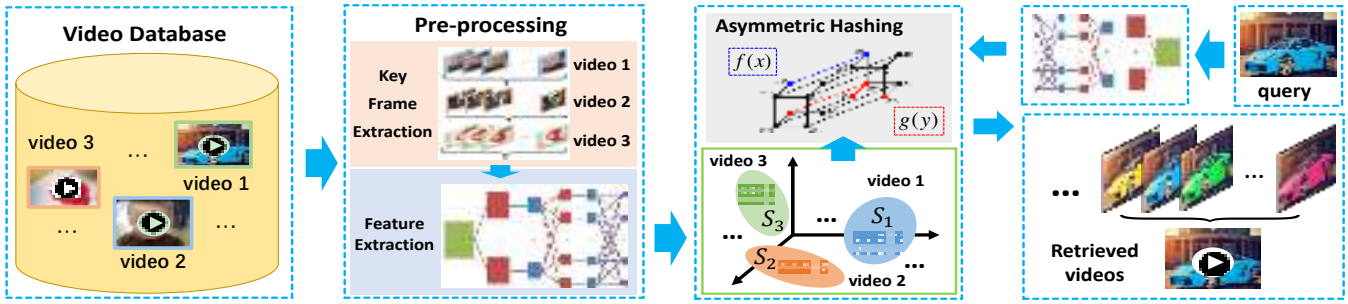


Figure 1: The flowchart of BSC framework. We first extract keyframes from each video and represent each video by its subspace representation (i.e., a matrix composed of its keyframes). Given a query image, it is also mapped into the same Hamming space as videos. Hash functions $f(x)$ and $g(y)$ are asymmetrically learned for database videos and query images respectively, which help to preserve the geometric relationships between images and videos based on the geometry-preserving distance metric.

They are more relevant to ours and successful in some applications, including supervised methods such as Inter-Media Hashing [31], Linear Cross-modal Hashing [42], Multimodal Latent Binary Embedding (MLBE) [40], Semantic Correlation Maximization (SCM) [39], Semantic Preserved Hashing (SePH) [20]. And other unsupervised methods also have outstanding performance such as Latent Semantic Sparse Hashing (LSSH) [41], Collective Matrix Factorization Hashing (CMFH) [5], Predictable Dual-view Hashing (PDH) [26], etc. Shirvastava and Li [30] proposed an Asymmetric Locality-Sensitive Hashing (ALSH) which performs a simple asymmetric transformation on data pairs for different learning. However, they all neglect the structure information of videos which affects retrieval performance greatly. To solve this problem, Yan [19] proposed Hashing across Euclidean Space and Riemannian Manifold (HER) which learns hash functions in a max-margin framework across Euclidean space and Riemannian manifold. However, it becomes unsuitable for large-scale databases owing to unaffordable time when feature dimensionality and data size grow. Inspired by their work and a dimensionality reduction bilinear projection method [6], we aim to seek a more efficient asymmetric binary learning framework to support the QBIVR task based on video subspace learning.

3 BINARY SUBSPACE CODING

3.1 Problem Formulation

Given a database of k videos, denoted as $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$, where S_j $(1 \le j \le k)$ represents the subspace covering all the keyframes for the j-th video.

Given a query image $q \in \mathbb{R}^{d\times 1}$, the main objective of the query-by-image video retrieval (QBIVR) task can be formulated as below:

$$S^* = \arg\min_{S_j \in \mathcal{S}} d(q, S_j). \tag{1}$$

As shown in (1), the major objective of the QBIVR task is to find the subspace S^* whose distance from the query point q is the shortest, where the distance function $d(\cdot,\cdot)$ is very crucial for the task.

3.2 Geometry-Preserving Distance Metric

The QBIVR task is essentially a point-to-subspace search problem in which the query is represented as a point, and the database contains videos represented by their subspaces. Recall that existing solutions to the above problem either aggregate/project all the frames into a single data point compatible with the given query or average all the distances between each frame and the query, which may cause serious information loss or high computation. To compensate such drawbacks, we propose to measure the image-to-video similarity by the distance between the query and its corresponding projection on the subspace plane. In this way, the geometric property and structural information of the subspace can be preserved. Then the new image-to-video distance metric is

$$d(q, S) = \min_{v \in S} \|q - v\|_2^2, \tag{2}$$

where $\|\cdot\|_2$ is the ℓ_2 norm. It is easy to see that the nearest point is the orthogonal projection of q on S, which is calculated as follows:

$$v^* = p(q) = \tilde{S}q, \tag{3}$$

where $\tilde{S} = S(S^\top S)^{-1}S^\top$, and $p(q)$ is the orthogonal projection of q on S. Note that $\tilde{S} \in \mathbb{R}^{d\times d}$ can be computed offline to increase efficiency. Substituting Eq.(3) into Eq.(2), we can obtain the distance of point-to-subspace:

$$d(q, S) = \|q - p(q)\|_2 = \|(I_d - \tilde{S})q\|_2, \tag{4}$$

where I_d is an identity matrix of size $d \times d$. Denoting $D = I_d - \tilde{S}$, given that $\tilde{S}^\top\tilde{S} = \tilde{S}$, we have

$$D^\top D = (I_d - \tilde{S})^\top(I_d - \tilde{S}) = I_d - \tilde{S} = D. \tag{5}$$

Therefore, we can obtain the further conclusion:

$$\begin{aligned} d^2(q, S) &= \|Dq\|_2^2 = Tr(D^\top Dqq^\top) \\ &= Tr((I_d - \tilde{S})qq^\top) = q^\top q - Tr(\tilde{S}qq^\top), \end{aligned} \tag{6}$$

where $Tr(\cdot)$ is the trace of a matrix. Based on Eq.(6), our objective is equivalent to the following problem:

$$S^* = \arg\max_{S \in \mathcal{S}} Tr(\tilde{S}qq^\top). \tag{7}$$

Note that $Tr(\tilde{S}qq^\top) = q^\top\tilde{S}q = \langle p(q), q\rangle$, where $\langle\cdot,\cdot\rangle$ denotes the inner product of two vectors. Then Eq.(7) can be seen as the Maximum Inner Product Search (MIPS) problem w.r.t. the query q and its orthogonal projection $p(q)$ in the subspace. However, all the $p(q)$ in the dataset have to be preprocessed every time a new query is provided, which is apparently computationally expensive. To

bypass this issue, according to the linear algebra manipulation $Tr(\tilde{S}qq^\top) = \mathbf{vec}(\tilde{S})^\top \mathbf{vec}(qq^\top)$, we rewrite the problem (7) as

$$S^* = \arg\max_{S\in\mathcal{S}} \langle \mathbf{vec}(\tilde{S}), \mathbf{vec}(qq^\top)\rangle, \tag{8}$$

where $\mathbf{vec}(\cdot)$ is the function of transforming a matrix of size $d \times d$ to a column vector of size $d^2 \times 1$ by performing column-wise concatenation of the matrix. In this way, we obtain an equivalent MIPS problem w.r.t. $\mathbf{vec}(\tilde{S})$ and $\mathbf{vec}(qq^\top)$ from the original QBIVR problem. The process is in Figure 2.

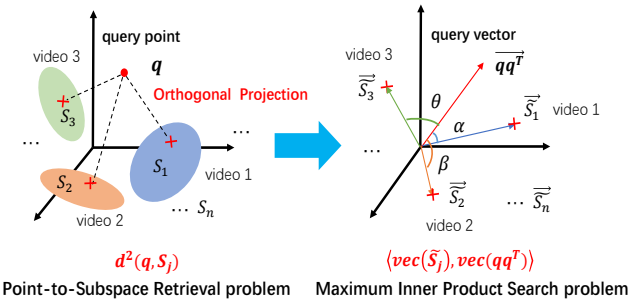


Figure 2: Illustration of the transformation from the point-to-subspace retrieval problem to the MIPS problem based on our geometry-preserving distance metric.

Considering the unaffordable computations of $\mathbf{vec}(\tilde{S})^\top \mathbf{vec}(qq^\top)$ and $qq^\top$ when d is large, i.e., $O(d^2k)$, we employ hashing approaches to binarize the representations of query images and database videos. Different properties of query images and database videos are unnegligible factors for accurate binary codes. Therefore, we learn two different asymmetric hash functions for query images $\mathbf{vec}(qq^\top)$ and database videos $\mathbf{vec}(\tilde{S})$ respectively, then the MIPS problem is reformulated as follows:

$$S^* = \arg\max_{S\in\mathcal{S}} \langle f(\mathbf{vec}(\tilde{S})), g(\mathbf{vec}(qq^\top))\rangle, \tag{9}$$

where $f : \mathbb{R}^{d^2\times 1} \to \{-1,1\}^{r\times 1}$ and $g : \mathbb{R}^{d^2\times 1} \to \{-1,1\}^{r\times 1}$ are hash functions for videos and images, respectively.

3.3 Inner-product Binary Coding

In this part, we present the Inner-product Binary Coding (IBC) approach for asymmetric learning. Firstly, we construct video data V and image data U for training:

$$\begin{cases} V = [\mathbf{vec}(\tilde{S}_1), \mathbf{vec}(\tilde{S}_2), \ldots, \mathbf{vec}(\tilde{S}_k)] \in \mathbb{R}^{d^2\times k}, \\ U = [\mathbf{vec}(x_1x_1^\top), \mathbf{vec}(x_2x_2^\top), \ldots, \mathbf{vec}(x_nx_n^\top)] \in \mathbb{R}^{d^2\times n}, \end{cases}$$

where $\{x_i\}|_{i=1}^n$ are n images randomly sampled from video frames for training. Let A be the correlation matrix of U and V. We choose to use the inner product to represent the similarity, i.e., $A = U^\top V$. Following [27], we consider the following optimization problem:

$$\min_{f,g} \left\| g(U)^\top f(V) - A \right\|_F^2, \tag{10}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $f(V) = [f(\mathbf{vec}(\tilde{S}_1)), \ldots, f(\mathbf{vec}(\tilde{S}_k))]$, $g(U) = [g(\mathbf{vec}(x_1x_1^\top)), \ldots, g(\mathbf{vec}(x_nx_n^\top))]$. For simplicity, we choose to learn linear hash functions, i.e., $f(x) = \mathbf{sgn}(P^\top x)$ and $g(z) = \mathbf{sgn}(Q^\top z)$, where $P \in \mathbb{R}^{d^2\times l}$ and $Q \in \mathbb{R}^{d^2\times l}$ are the two mapping variables for binarizing videos and images, respectively.

In practice, to further speed up the optimization, we deliberately discard the quadratic term $\|g(U)^\top f(V)\|_F^2$, in view of the quadratic term with no help in leveraging the ground-truth similarity. In fact, the term $\|g(U)^\top f(V)\|_F^2$ can be treated as a regularization in the magnitude of the learned inner product. Hence, we arrive at the new objective as follows:

$$\max_{f,g} Tr(g(U)Af(V)^\top), \tag{11}$$

which can be optimized by alternatingly updating g and f. In particular, when learning g with f fixed, we have

$$\max_{P} Tr(\mathbf{sgn}(P^\top U)Af(V)^\top). \tag{12}$$

When updating f with g fixed, we arrive at

$$\max_{Q} Tr(\mathbf{sgn}(Q^\top V)A^\top g(U)^\top). \tag{13}$$

Both of the above sub-problems are of the same form. So we can solve (12) and the sub-problem (13) in the same way.

It is non-trivial to optimize the sub-problem (12) due to the existence of the sign function $\mathbf{sgn}(\cdot)$. To bypass the obstacle, we introduce an auxiliary variable $B \in \{-1,1\}^{r\times n}$ to approximate $\mathbf{sgn}(P^\top U)$, and thus we have

$$\begin{aligned} &\max_{B,P} Tr(BAf(V)^\top) - \lambda \left\| B - P^\top U \right\|_F^2, \\ &\text{s.t. } B \in \{-1,1\}^{r\times n}, \end{aligned} \tag{14}$$

where $\lambda > 0$ is a balance parameter. Setting the derivative of the above objective w.r.t. P to zero, we have

$$P = (UU^\top)^{-1}UB^\top. \tag{15}$$

Fixing P, then we can update B with

$$B = \mathbf{sgn}(f(V)A^\top + 2\lambda P^\top A). \tag{16}$$

The analytical solution of B at the above significantly reduces the training cost, which similarly makes the algorithm easily performed on the large-scale databases.

3.4 Bilinear Binary Coding

Note that in IBC, hashing the vectored images and videos data $\mathbf{vec}(\tilde{S})$ and $\mathbf{vec}(xx^\top)$ with a full projection matrix may cause high computational cost. In this part, we propose a Bilinear Binary Coding (BBC) approach to further accelerate the efficiency of the subspace-based QBIVR task.

We first present a bilinear rotation to maintain matrix structure instead of a single large projection matrix, denoted as $H(X) = \mathbf{sgn}(R_1^\top X R_2)$, which is remarkable successful in lowering running time and storage for code generation. It also has been proved by [6] that a bilinear rotation to $X \in \mathbb{R}^{d_1\times d_2}$ is equivalent to a $d_1d_2 \times d_1d_2$ rotation to $\mathbf{vec}(X)$, denoted as $R = R_2 \bigotimes R_1$, where $\bigotimes$ is the Kronecker product [17]. Now, we can equivalently learn asymmetric hash functions for images and videos as follows:

$$\begin{cases} f(\mathbf{vec}(\tilde{S})) = \mathbf{sgn}(P^\top \mathbf{vec}(\tilde{S})) = \mathbf{sgn}(\mathbf{vec}(P_1^\top \tilde{S} P_2)), \\ g(\mathbf{vec}(xx^\top)) = \mathbf{sgn}(Q^\top \mathbf{vec}(xx^\top)) = \mathbf{sgn}(\mathbf{vec}(Q_1^\top xx^\top Q_2)). \end{cases} \tag{17}$$

Note that $P_1, Q_1 \in \mathbb{R}^{d\times c_1}$, $P_2, Q_2 \in \mathbb{R}^{d\times c_2}$, $c_1, c_2 < d$. Then, we can generate binary codes for vectored images and videos with code length $c = c_1 \times c_2$ to perform an efficient retrieval.

Following [8], a feasible objective is to learn a bilinear rotation which minimizes the angle between $\mathbf{vec}(R_1^\top X R_2)$ and its binary encoding $B = \mathbf{sgn}(\mathbf{vec}(R_1^\top X R_2))$. We preprocess the video dataset V and image dataset U to be zero-centered and have unit norm, then our goal is to maximize the following objective:

$$\sum_{i=1}^{n} \cos\theta_i^{(U)} + \sum_{j=1}^{k} \cos\theta_j^{(V)} + \mu \sum_{i=1}^{n}\sum_{j=1}^{k} A_{i,j} \cdot \cos\omega_{i,j}, \tag{18}$$

where $\theta_m^{(J)}$ is the angle of the m-th rotated image/video and its binary code, $J \in \{U, V\}$. $\omega_{i,j}$ is the angle between the binary codes of $B_i^{(U)}$ and $B_j^{(V)}$, where $B_i^{(U)} = \mathbf{sgn}(Q_1^\top U_i Q_2)$ and $B_j^{(V)} = \mathbf{sgn}(P_1^\top V_j P_2)$ are the binary codes of the i-th image and j-th video, respectively. Especially, the inner product $A_{i,j} = U_i^\top V_j$ is used as a good surrogate of the Hamming distance to preserve the similarity property of images and videos in the same or different categories, which is 1 (or 0) when it is over (or below) a threshold.

For images, $\cos\theta_i^{(U)}$ is expressed as

$$\cos\theta_i^{(U)} = \frac{\mathbf{vec}(\mathbf{sgn}(Q_1^\top U_i Q_2))^\top}{\sqrt{c}} \frac{\mathbf{vec}(Q_1^\top U_i Q_2)}{\left\|\mathbf{vec}(Q_1^\top U_i Q_2)\right\|_2}. \tag{19}$$

Following [7], we simplify the subsequent optimization by ignoring $\left\|\mathbf{vec}(Q_1^\top U_i Q_2)\right\|_2$, and then we get

$$\begin{aligned} &\sum\nolimits_i \cos\theta_i^{(U)} \\ =&\sum\nolimits_i \Big(\frac{\mathbf{vec}(\mathbf{sgn}(Q_1^\top U_i Q_2))^\top}{\sqrt{c}} \mathbf{vec}(Q_1^\top U_i Q_2)\Big) \\ =&\frac{1}{\sqrt{c}} \sum\nolimits_i (\mathbf{vec}(B_i^{(U)})^\top \mathbf{vec}(Q_1^\top U_i Q_2)) \\ =&\frac{1}{\sqrt{c}} \sum\nolimits_i Tr(B_i^{(U)} Q_2^\top U_i^\top Q_1). \end{aligned} \tag{20}$$

Similarly, we can derive the objectives of video angles and image-to-video angles as follows:

$$\begin{cases} \sum_j \cos\theta_j^{(V)} = \frac{1}{\sqrt{c}} \sum_j Tr(B_j^{(V)} P_2^\top V_j^\top P_1), \\ \sum_{i,j} A_{i,j} \cdot \cos\omega_{i,j} = \frac{1}{c} \sum_{i,j} A_{i,j} \cdot Tr(B_j^{(V)} (B_i^{(U)})^\top). \end{cases} \tag{21}$$

Hence, the objective function is transformed to

$$\begin{aligned} &\sum\nolimits_i Tr(B_i^{(U)} Q_2^\top U_i^\top Q_1) + \sum\nolimits_j Tr(B_j^{(V)} P_2^\top V_j^\top P_1) \\ &+ \frac{\mu}{\sqrt{c}} \sum\nolimits_{i,j} A_{i,j} \cdot Tr(B_j^{(V)} (B_i^{(U)})^\top), \end{aligned} \tag{22}$$

where $B_i^{(U)}, B_j^{(V)} \in \{-1, 1\}^{d\times d}$, $Q_1^\top Q_1 = I, Q_2^\top Q_2 = I, P_1^\top P_1 = I$ and $P_2^\top P_2 = I$.

For optimization, we use block coordinate ascent to alternatingly update $\{B_j^{(V)}\}_{j=1}^k$,$\{B_i^{(U)}\}_{i=1}^n$,Q_1, Q_2,P_1, P_2. The updating processes w.r.t. images and videos are symmetric. Hence we just describe the updates of variables of videos by fixing all the variables of images.

Step 1: Update P_1, with all other variables fixed. We have the following reduced problem:

$$\max_{P_1^\top P_1 = I} Tr(D_1 P_1), \tag{23}$$

where $D_1 = \sum_{j=1}^k (B_j^{(V)} P_2^\top V_j^\top)$. We can solve the above optimization problem using polar decomposition

$$P_1 = Y_1 Z_1^\top, \tag{24}$$

where Z_1 and Y_1 are the left-singular vectors and the top c_1 right-singular vectors of D_1, respectively, by performing SVD.

Step 2: Update P_2, with all the others fixed, we have

$$\max_{P_2^\top P_2 = I} Tr(P_2^\top D_2), \tag{25}$$

where $D_2 = \sum_{j=1}^k (V_j^\top P_1 B_j^{(V)})$. Similar to the previous step, the update for P_2 is $P_2 = Z_2 Y_2^\top$, where Z_2 and Y_2 are the top c_2 left-singular vectors and the right-singular vectors of D_2, respectively, by performing SVD.

Step 3: Update $B_j^{(V)}$, by fixing all the other variables, we have

$$\max_{B_j^{(V)} \in \{-1,1\}^{d\times d}} Tr(B_j^{(V)} D_3), \tag{26}$$

where $D_3 = P_2^\top V_j^\top P_1 + \frac{\mu}{\sqrt{c}} \sum_i A_{i,j} (B_i^{(U)})^\top$. It can be easily seen that the solution to the above problem is as below:

$$B_j^{(V)} = \mathbf{sgn}(D_3^\top). \tag{27}$$

Then, we can similarly update Q_1, Q_2 and $\{B_i^{(U)}\}_{i=1}^n$.

Comparing to the time complexity of full rotation ,i.e, $O(d^2)$, the asymmetric bilinear hashing learning of videos and images significantly reduces the training cost to $O(d_1^2 + d_2^2)$, where $d = d_1 \times d_2$. We summarize the algorithm for optimizing the proposed BBC approach in Algorithm 1.

Algorithm 1 Optimization of Bilinear Binary Coding (BBC).

Input: Subspaces of videos $\{S_j\}|_{j=1}^k$ and images $\{x_i\}|_{i=1}^n$;
Output: Hash function f and g;

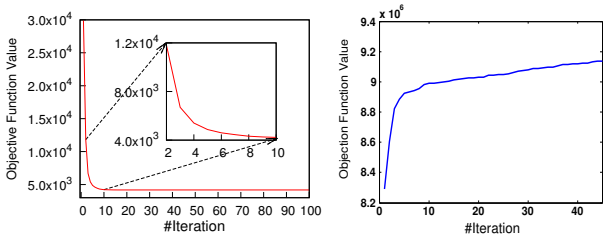
1: Compute $\tilde{S}_j = S_j (S_j^\top S_j)^{-1} S_j^\top$, $j = 1, 2, \ldots, k$;
2: Construct video and image training data as below:

$$\begin{cases} V = \{\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_k\}, \\ U = \{x_1 x_1^\top, x_2 x_2^\top, \ldots, x_n x_n^\top\}; \end{cases}$$

3: Randomly initialize $\{B_j^{(V)}\}_{j=1}^k$,$\{B_i^{(U)}\}_{i=1}^n$,Q_1, Q_2, P_1, P_2;
4: **repeat**
5: Update P_1 by solving the problem (23);
6: Update P_2 by solving the problem (25);
7: Sequentially update $\{B_j^{(V)}\}_{j=1}^k$ by solving the problem (26);
8: Update Q_1 according to the problem (23);
9: Update Q_2 according to the problem (25);
10: Sequentially update $\{B_i^{(U)}\}_{i=1}^n$ according to the problem (26);
11: **until** there is no change to all the variables;
12: **return** $\{B_j^{(V)}\}_{j=1}^k$,$\{B_i^{(U)}\}_{i=1}^n$,Q_1, Q_2, P_1, P_2.

3.5 Algorithm Analysis

In this section, we analyze the convergence of our IBC and BBC approaches. As described above, in each iteration, the updates of all variables make the objective function value of the IBC approach decreased and the BBC approach increased. We conducted empirical study on the convergence property on Vine dataset with 20,000 random training samples and fixed the code length to be 96-bit. As shown in Figure 3, the objective function value becomes relatively stable for IBC to quickly approach the minimum and for BBC to quickly approach the maximum within only 10 iterations. The phenomenon clearly indicates the efficiency of our algorithms.



(a) Convergence curve of IBC (b) Convergence curve of BBC

Figure 3: Convergence study on Vine dataset in 96-bit.

4 EXPERIMENTS

In this part, we evaluate our IBC and BBC approaches for the QBIVR task as compared to state-of-the-art methods.

4.1 Experimental preparation

4.1.1 Datasets. We use four video datasets, a face-image video dataset BBT1 (Big Bang Theory1) [19], an object-based video dataset collected from Vine website[1], an event-based UQE50 (UQ Event dataset with 50 pre-defined events) dataset [38], and a wide range of objects and events dataset Fudan-Columbia Video Dataset (FCVID) [15]. UQE50, BBT1 and FCVID datasets are three publicly-available datasets which have been provided by [38], [19] and (FCVID)[2] respectively. Also, Vine dataset is a valuable micro object-based video dataset which was collected by ourselves. Details on each dataset will be discussed in the next subsections.

4.1.2 Video Modelling. Given the overwhelming accuracy and fast processing speed, we utilize the discriminative CNN method [34] to generate the frame descriptors, except the first dataset BBT1 that has its block Discrete Cosine Transformation (DCT) feature for direct usage. The method has been proved to be effective on TRECVID MEDTest 14 dataset[3]. We follow the FFmpeg[4] to sample the videos at the rate of 5 frames per second as keyframes, and subsequently extract the visual features of keyframes using fc7 layer (4096-d). In view of the potential redundancy, the feature dimensionality in our experiment is further reduced to 1600-d by PCA. Then we obtain the subspace-based video representations with 1600-d keyframe descriptors.

[1]https://vine.co/
[2]http://bigvid.fudan.edu.cn/FCVID/
[3]https://www.nist.gov/itl/iad/mig/med-2014-evaluation
[4]http://ffmpeg.org/

4.1.3 Setup. We compare our IBC and BBC approaches against several state-of-the-art unsupervised hashing methods for large-scale video retrieval, including ALSH [29], SGH [14], ITQ [8], IMH [31] and some effective cross-modal hashing methods such as SCM [39], LSSH [41], CMFH [5], and PDH [26]. For single modality hashing methods, we can transform subspace representations in video datasets to the point representations by projecting or aggregating (e.g., maximum, minimum, average strategies) keyframes. In our methods, each column of the original inner product matrix A is binarized, where the top m largest elements are set to 1 and the rest ones to 0. m is set to 18,000 for FCIVD dataset, 13,000 for the Vine dataset, 1,700 for UQE50 dataset, and 700 for BBT1 dataset based on preliminary tuning. In our IBC approach, the balance parameter λ is empirically set to 100 and the number of the local iteration t is set to 2. In our BBC approach, we firstly initiate the bilinear rotation parameters W_1, W_2, P_1, P_2 randomly and then learn two asymmetric hash functions respectively. The number of the local iteration $iter$ is set to 4 in light of the excellent converging property of our devised BBC approach. The parameters of the rest compared approaches are set as suggested above. In the experiment, the code length is tested in the range of $\{16, 32, 64, 96, 128\}$.

The evaluation metrics are chosen as Hamming ranking including mean of average precision (mAP) and mean precision of the top 500/50 retrieved neighbors (Precision@500/Precision@50).

4.2 BBT1: video retrieval with face images

The Big Bang Theory (BBT) [19] is a sitcom (20 minutes an episode) which includes many full-view shots of multiple characters at a time. It takes place mostly indoors and mainly focuses on $5 \sim 8$ characters. BBT1 consists of 3,341 face videos of the first 6 episodes from season 1 of BBT. We use its provided features, the block 240-d Discrete Cosine Transformation (DCT) features as used in [2], which form a 240×240 covariance video representation.

4.2.1 Compared to other state-of-the-art methods. In this part, we test our approaches on the BBT1 with several popular unsurpervised hashing methods and cross-modal hashing methods, especially an effective heterogeneous spaces hashing method, HER method [19], which has been verified successfully on subspace-based QBIVR task. Following [19], we randomly extract 300 image-video pairs (both elements of the pair come from the same subject) for training and 100 images from the rest as queries for the retrieval task. The results in Table1 show that the cross-modal hashing methods are not suitable for the image-video retrieval even though they are very successful in text-image task. We can also observe that our proposed approaches significantly outperform the effective image-video HER method on both mAP and Precision@50, and moreover, they overcome the limitation of HER method in the high-dimensional feature space. The subsequent experiments on the larger datasets will prove that our approaches can also be applied to high-dimensional feature spaces effectively.

4.3 UQE50: video retrieval with event images

The video dataset UQE50 (UQ Event dataset with 50 pre-defined events) that was downloaded from YouTube[5] [38] aims at the event

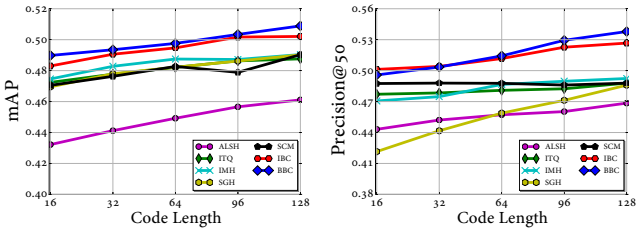
[5]http://www.youtube.com/

Method	mAP			Precision@50		
	32-bit	64-bit	96-bit	32-bit	64-bit	96-bit
ALSH	0.2473	0.2396	0.2467	0.3693	0.3412	0.3604
ITQ	0.3016	0.3026	0.3045	0.3906	0.3926	0.4018
IMH	0.4069	0.4736	0.4808	0.4624	0.4736	0.4944
SGH	0.3135	0.4458	0.4622	0.3965	0.4417	0.4814
LSSH	0.2813	0.2226	0.2737	0.2822	0.3204	0.2996
CMFH	0.3096	0.3296	0.3315	0.3016	0.4126	0.3178
PDH	0.2891	0.2886	0.2884	0.2702	0.2696	0.2702
SCM	0.2535	0.2778	0.2792	0.3265	0.3237	0.3200
HER	0.4977	0.5376	0.5517	0.4572	0.5037	0.5180
IBC	**0.5423**	**0.5670**	**0.5701**	**0.5382**	**0.5428**	**0.5571**
BBC	**0.5501**	**0.5645**	**0.5812**	**0.5537**	**0.5707**	**0.5711**

Table 1: Comparison with the state-of-the-arts (i.e., unsupervised single modalitity hashing methods, cross-modal hashing methods) on BBT1 dataset with different code lengths.

analysis task. The dataset contains 3,462 videos that belong to 50 different event categories, and all the videos are from trending events happened in the last few years whose granularities are comparably larger than the existing video event datasets. To verify the generality of our proposed approaches in different types and sizes of video datasets, we use UQE50 video dataset to compare the performance of our approaches with that of other state-of-the-art methods. Compared with face-image BBT1 dataset, UQE50 is a longer-time event-based video dataset. We randomly use 1,800 videos and 1,800 images chosen from videos as training samples and randomly select the remaining 200 images as test samples.

4.3.1 Compared with other state-of-the-art methods. To examine the practical efficacy of our proposed approaches, in this part, we conduct a similar experiment to BBT1's experiment on UQE50 to evaluate the scalability of our approaches compared with other state-of-the-art hashing methods. Obviously, even for an event-based dataset with longer-time videos, the performance (i.e., mAP, Precision@50) is also satisfactory as shown in Figure 4. As the code length increases, the performance of the two proposed approaches is steadily better than the other methods.



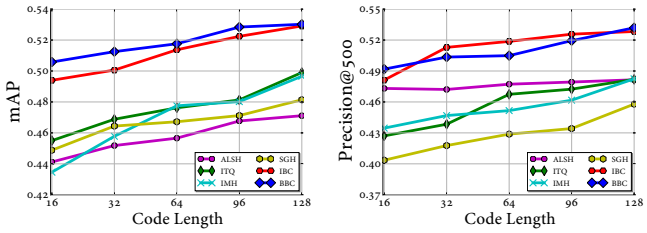
(a) mAP on UQE50 dataset **(b) Precision@50 on UQE50 dataset**

Figure 4: Comparison of IBC, BBC and state-of-the-art hashing methods on UQE50 dataset with different code lengths.

4.4 Vine: video retrieval with object images

Vine is a micro video sharing platform, where users can only share videos which are no more than six seconds by mobile devices. We collect a micro object-based video dataset from Vine comprising 498,000 micro videos in 145 categories, which is a much bigger dataset than BBT1 and UQE50 datasets. We randomly sample 15,000 videos and 15,000 images from videos for training and the rest 1,000 images as test. We report the compared results with hashing methods and the aggregation approach as below.

4.4.1 Compared to other state-of-the-art hashing methods. In comparisons with hashing methods, we treat a query a false case if no point is returned when calculating precision. Ground truths are defined by the category information from the datasets. As Figure 5 shows, the two proposed approaches outperform all the other state-of-the-art methods in terms of both metrics at different code lengths. Note that our approaches remain a much greater ability of expression when the encoding length is as large as 128 bits. In this case, the ascending performance shown in this experiment directly proves the significant potential of our proposed IBC and BBC approaches for large-scale video databases.



(a) mAP on Vine dataset **(b) Precision@500 on Vine dataset**

Figure 5: The performance of mAP and Precision@500 on Vine dataset in 16,32,64,96,128-bit.

4.4.2 Compared to state-of-the-art aggregation approach. Temporal Aggregation Video Retrieval (TAVR) [3], an effective aggregation approach of frame-based video features for QBIVR task, employs the Scalable Compressed Fisher Vectors (SCFV) to reduce retrieval latency and memory requirements for achieving high speed and maintaining good performance. However, the approach sacrifices useful information such as structure similarity when pursuing high efficiency. Moreover, though the binarized fisher features which TAVR uses are more representative and effective than low-level local features, it still fails when competing with deep features, especially the ones after redundancy removing. The performance results of our IBC, BBC approaches and the aggregation approach in 64-bit are clearly illustrated in Table 2.

4.5 FCVID: video retrieval with event/object images

The video dataset FCVID (Fudan-Columbia Video Dataset)[6] is a web video dataset containing 91,223 Web videos annotated manually

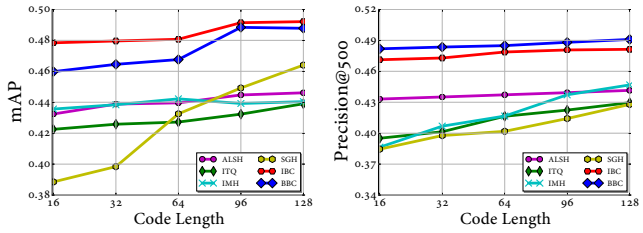
[6]http://bigvid.fudan.edu.cn/FCVID/

Method	mAP	Precision@500
TAVR	0.3785	0.3021
IBC	0.5134	0.5180
BBC	**0.5175**	**0.5049**

Table 2: Comparison of our proposed approaches IBC, BBC and aggregation approach TAVR on Vine dataset in 64-bit.

according to 239 categories. The categories in FCVID cover a wide range of topics like social events (e.g., tailgate party), procedural events (e.g., making cake), objects (e.g., panda), scenes (e.g., beach), etc. These categories are defined very carefully and organized in a hierarchy of 11 high-level groups. In this section, we randomly choose 50,000 as our video dataset and select 23,000 images and videos for training respectively, and then we test another 5,000 images in the video dataset.

4.5.1 Compared with other state-of-the-art methods. In this part, we conduct the same experiment as above to the FCVID dataset for studying the performance of event/object-image retrieval. In view of the wide range of image/video types and higher reliability of the dataset, we can demonstrate the effect of our approaches clearly. As Figure 6 shows, our approaches outperform state-of-the-art hashing methods in terms of both mAP and Precision@500 in different code lengths. Better performance on the four datasets in different types and sizes proves our validity of the proposed IBC and BBC approaches.



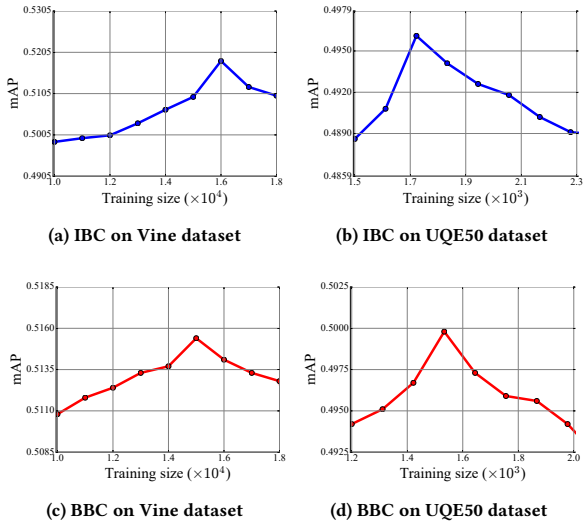
(a) mAP on FCVID dataset **(b) Precision@500 on FCVID dataset**

Figure 6: Comparision of IBC, BBC and other hashing methods on the FCVID dataset with different code lengths.

4.6 Effect of training size on UQE50 and Vine

This experiment mainly evaluates the effect of training size on the searching quality of our IBC and BBC approaches. We perform the experiments on object-based Vine and event-based UQE50 datasets and select the mean of average precision (mAP) as the comprehensive assessment index. We fix the code as 64-bit and vary the training sizes of Vine dataset from 10,000 to 18,000 with a regular interval of 1,000 and UQE50 is tuned from 1,500 to 2,300 with a regular interval of 100.

The results are shown in Figure 7. As we can see, both IBC and BBC approaches have the suitable training size for the best performance respectively. When there are too many training samples, the two approaches perform even worse. The ideal training size of Vine is 16,000 for the IBC approach and 15,000 for the BBC approach. As for the UQE50 dataset, the performance is optimal with the training size of 1,700 for IBC approach and 1,800 for BBC approach. The results give us guidance for choosing the suitable training size. At last, we choose 1,800 for UQE50 dataset and 15,000 for Vine dataset to train in our experiment.



(a) IBC on Vine dataset **(b) IBC on UQE50 dataset**

(c) BBC on Vine dataset **(d) BBC on UQE50 dataset**

Figure 7: The effects of training size of IBC, BBC on mAP performance over Vine and UQE50 with 64 bits fixed.

5 CONCLUSION

In this paper, we developed the Binary Subspace Coding (BSC) framework which includes two different hashing approaches for the subspace-based QBIVR task. Different from traditional video retrieval methods, we focused on subspace-based video representation and discovered a common Hamming space for both images and videos to enable an efficient retrieval. Our proposed geometry-preserving similarity measurement uses a new distance metric to preserve geometric properties between images and videos. Furthermore, we deduced an equivalent MIPS solution to decrease the computational cost significantly. BSC framework is an asymmetric learning model including two hashing approaches that can well handle the domain differences between videos and images efficiently. Especially, the bilinear framework of our BBC approach can minimize the dimensional complexity of video dataset by using compact bilinear projections instead of a single large projection matrix. Experiments on different video datasets demonstrated the advantages of our approaches over the existing methods.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Project 61572108, Project 61502081, Project 61632007 and Project 61602088, and the Fundamental Research Funds for the Central Universities under Project ZYGX2014Z007 and Project ZYGX2015J055.

REFERENCES

[1] Ronen Basri, Tal Hassner, and Lihi Zelnik-Manor. 2011. Approximate Nearest Subspace Search. *TPAMI* 33, 2 (2011), 266–278.

[2] Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. 2013. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *CVPR*. 3602–3609.

[3] Andre F. de Araújo, Jason Chaves, Roland Angst, and Bernd Girod. 2015. Temporal aggregation for large-scale query-by-image video retrieval. In *ICIP*. 1519–1522.

[4] Andre F. de Araújo, Mina Makar, Vijay Chandrasekhar, David M. Chen, Sam S. Tsai, Huizhong Chen, Roland Angst, and Bernd Girod. 2014. Efficient video search using image queries. In *ICIP*. 3082–3086.

[5] Guiguang Ding, Yuchen Guo, and Jile Zhou. 2014. Collective Matrix Factorization Hashing for Multimodal Data. In *CVPR*. 2083–2090.

[6] Yunchao Gong, Sanjiv Kumar, Henry A. Rowley, and Svetlana Lazebnik. 2013. Learning Binary Codes for High-Dimensional Data Using Bilinear Projections. In *CVPR*. 484–491.

[7] Yunchao Gong, Sanjiv Kumar, Vishal Verma, and Svetlana Lazebnik. 2012. Angular Quantization-based Binary Codes for Fast Similarity Search. In *NIPS*. 1205–1213.

[8] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. *TPAMI* 35, 12 (2013), 2916–2929.

[9] Richang Hong, Yang Yang, Meng Wang, and Xian-Sheng Hua. 2015. Learning Visual Semantic Relationships for Efficient Visual Retrieval. *TBD* 1, 4 (2015), 152–161.

[10] Yiqun Hu, Ajmal S. Mian, and Robyn A. Owens. 2011. Sparse approximated nearest points for image set classification. In *CVPR*. 121–128.

[11] Zi Huang, Heng Tao Shen, Jie Shao, Xiaofang Zhou, and Bin Cui. 2009. Bounded coordinate system indexing for real-time video clip search. *TOIS* 27, 3 (2009), 17:1–17:33.

[12] Jianqiu Ji, Jianmin Li, Qi Tian, Shuicheng Yan, and Bo Zhang. 2015. Angular-Similarity-Preserving Binary Signatures for Linear Subspaces. *TIP* 24, 11 (2015), 4372–4380.

[13] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Qi Tian, and Bo Zhang. 2014. Similarity-Preserving Binary Signature for Linear Subspaces. In *AAAI*. 2767–2772.

[14] Qing-Yuan Jiang and Wu-Jun Li. 2015. Scalable Graph Hashing with Feature Transformation. In *IJCAI*. 2248–2254.

[15] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2015. Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks. *CoRR* abs/1502.07209 (2015).

[16] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*. 3270–3278.

[17] Alan J. Laub. 2005. *Matrix analysis - for scientists and engineers*. SIAM.

[18] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. 2016. Feature Learning Based Deep Supervised Hashing with Pairwise Labels. In *IJCAI*. 1711–1717.

[19] Yan Li, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. 2015. Face video retrieval with image query via hashing across Euclidean space and Riemannian manifold. In *CVPR*. 4758–4767.

[20] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. 2015. Semantics-preserving hashing for cross-view retrieval. In *CVPR*. 3864–3872.

[21] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep hashing for compact binary codes learning. In *CVPR*. 2475–2483.

[22] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. 2012. Supervised hashing with kernels. In *CVPR*. 2074–2081.

[23] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with Graphs. In *ICML*. 1–8.

[24] Mohammad Norouzi and David J. Fleet. 2011. Minimal Loss Hashing for Compact Binary Codes. In *ICML*. 353–360.

[25] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. 2010. Improving the Fisher Kernel for Large-Scale Image Classification. In *ECCV*. 143–156.

[26] Mohammad Rastegari, Jonghyun Choi, Shobeir Fakhraei, Hal Daumé III, and Larry S. Davis. 2013. Predictable Dual-View Hashing. In *ICML*. 1328–1336.

[27] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. 2015. Learning Binary Codes for Maximum Inner Product Search. In *ICCV*. 4148–4156.

[28] Fumin Shen, Xiang Zhou, Yang Yang, Jingkuan Song, Heng Tao Shen, and Dacheng Tao. 2016. A Fast Optimization Method for General Binary Code Learning. *TIP* 25, 12 (2016), 5610–5621.

[29] Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS). In *NIPS*. 2321–2329.

[30] Anshumali Shrivastava and Ping Li. 2015. Improved Asymmetric Locality Sensitive Hashing (ALSH) for Maximum Inner Product Search (MIPS). In *UAI*. 812–821.

[31] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*. 785–796.

[32] Raviteja Vemulapalli, Jaishanker K. Pillai, and Rama Chellappa. 2013. Kernel Learning for Extrinsic Classification of Manifold Features. In *CVPR*. 1782–1789.

[33] Ruiping Wang and Xilin Chen. 2009. Manifold Discriminant Analysis. In *CVPR*. 429–436.

[34] Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. 2015. A discriminative CNN video representation for event detection. In *CVPR*. 1798–1807.

[35] Yang Yang, Yadan Luo, Weilun Chen, Fumin Shen, Jie Shao, and Heng Tao Shen. 2016. Zero-Shot Hashing via Transferring Supervised Knowledge. In *ACM MM*. 1286–1295.

[36] Yang Yang, Fumin Shen, Heng Tao Shen, Hanxi Li, and Xuelong Li. 2015. Robust Discrete Spectral Hashing for Large-Scale Image Semantic Indexing. *TBD* 1, 4 (2015), 162–171.

[37] Yang Yang, Zheng-Jun Zha, Yue Gao, Xiaofeng Zhu, and Tat-Seng Chua. 2014. Exploiting Web Images for Semantic Video Indexing Via Robust Sample-Specific Loss. *TMM* 16, 6 (2014), 1677–1689.

[38] Litao Yu, Yang Yang, Zi Huang, Peng Wang, Jingkuan Song, and Heng Tao Shen. 2016. Web Video Event Recognition by Semantic Analysis from Ubiquitous Documents. *TIP* 25, 12, 5689–5701.

[39] Dongqing Zhang and Wu-Jun Li. 2014. Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization. In *AAAI*. 2177–2183.

[40] Yi Zhen and Dit-Yan Yeung. 2012. A probabilistic model for multimodal hash function learning. In *SIGKDD*. 940–948.

[41] Jile Zhou, Guiguang Ding, and Yuchen Guo. 2014. Latent semantic sparse hashing for cross-modal similarity search. In *SIGIR*. 415–424.

[42] Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. 2013. Linear cross-modal hashing for efficient multimedia search. In *ACM MM*. 143–152.