Face Video Retrieval with Image Query via Hashing across Euclidean Space and Riemannian Manifold

Yan Li[1,2], Ruiping Wang[1], Zhiwu Huang[1,2], Shiguang Shan[1], Xilin Chen[1,3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Department of Computer Science and Engineering, University of Oulu, Oulu 90570, Finland
{yan.li, zhiwu.huang}@vipl.ict.ac.cn, {wangruiping, sgshan, xlchen}@ict.ac.cn

Abstract

Retrieving videos of a specific person given his/her face image as query becomes more and more appealing for applications like smart movie fast-forwards and suspect searching. It also forms an interesting but challenging computer vision task, as the visual data to match, i.e., still image and video clip are usually represented quite differently. Typically, face image is represented as point (i.e., vector) in Euclidean space, while video clip is seemingly modeled as a point (e.g., covariance matrix) on some particular Riemannian manifold in the light of its recent promising success. It thus incurs a new hashing-based retrieval problem of matching two heterogeneous representations, respectively in Euclidean space and Riemannian manifold. This work makes the first attempt to embed the two heterogeneous spaces into a common discriminant Hamming space. Specifically, we propose Hashing across Euclidean space and Riemannian manifold (HER) by deriving a unified framework to firstly embed the two spaces into corresponding reproducing kernel Hilbert spaces, and then iteratively optimize the intra- and inter-space Hamming distances in a max-margin framework to learn the hash functions for the two spaces. Extensive experiments demonstrate the impressive superiority of our method over the state-of-the-art competitive hash learning methods.

1. Introduction

Face video retrieval in general is to retrieve video shots containing particular person given one image of him/her [31]. It is an appealing research direction with increasing demands, especially in the era of social networking, when more and more videos are continuously uploaded to the Internet via video blogs, social networking websites, etc. Face video retrieval technology thus can find a wide range of applications in the above context, for instance, 'smart fast-forwards', where the video player can jump to the next shot containing a specific character; retrieving all the shots containing a particular family member from thousands of short videos captured by a digital camera [33]; and rapid locating and tracking suspects from masses of city surveillance videos (e.g., Boston marathon bombings event). For more intuitive understanding, we show a conceptual sample of TV-Series character shots retrieval in Fig. 1. Besides, the inverse retrieval task, i.e., retrieving the face images of one person by using his/her video clip as query, also plays an important role in some scenarios, e.g., determining the identity of an unknown suspect by searching against a huge mug-shot image database with his/her video shot acquired from crime scene CCTV as query; naming a famous person in video based on very large celebrity image database.

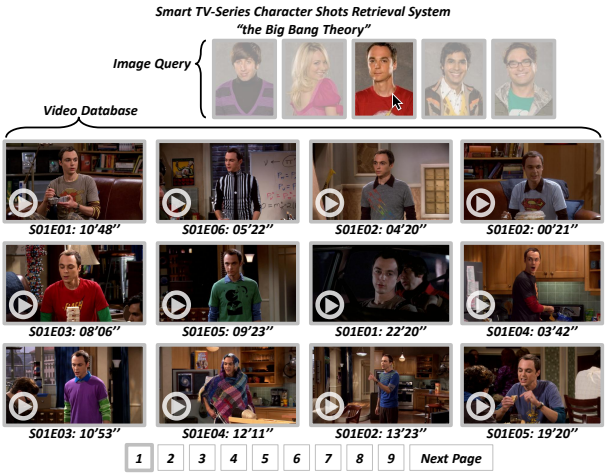


Figure 1: A conceptual illustration of TV-Series (*the Big Bang Theory*) character shots retrieval, where the query is an image of one specific character (*Sheldon Cooper*), and all the shots containing him/her are retrieved and ranked according to their similarities to the query image.

In this paper, we mainly focus on the former retrieval task, i.e., retrieve videos with image query, and set the scenario to character retrieval in TV-Series without loss of generality.

In our current work, query is provided in the form of image, whereas the database contains video clips. Therefore, the core task is to measure the similarity of image and video. Straightforwardly, we can compute the similarity between the query image and each frame of the video, and then integrate these similarities by averaging or taking the maximum. However, this method ignores the correlations among video frames, and also suffers from high computational cost and massive storage demand, especially in case of long video clips with hundreds or even thousands of frames.

Alternatively, a more promising strategy is to model the video frames collectively. Recently, promising methods represent all the frames by single or mixture of linear subspaces [42] [21] [40] [38], affine subspace [7] [17], or covariance matrix [39] [25] [36]. These representations all reside on some specific Riemannian manifolds, namely Grassmann manifold, affine Grassmann manifold and Symmetric Positive Definite (SPD) matrix manifold, respectively. Compared with the former solution treating video as separated frames, these holistic modeling methods lead to more compact representations and superior performance. Among them, covariance matrix, as a second-order statistics, provides a natural compact representation of the set of video frames, thus attracting increasing attention most recently as in [39] [25] [36]. Therefore, in this paper covariance matrix is chosen as the face video representation.

Moreover, retrieval tasks require not only good representation but also low computational complexity in similarity computation for fast search. For this purpose, hash code is one of the best choices, which can achieve fast retrieval with almost constant time complexity and extremely low storage requirement. However, for the task in this paper, hash learning becomes non-trivial, because our query and target are represented in heterogeneous spaces, i.e., one Euclidean space v.s. Riemannian manifold. To our best knowledge, off-the-shelf hash learning methods fail to work in this case. Hashing methods even specifically dealing with multiple modalities cases [6] [22] [43] [29] [44] [26] also can only handle the case where different modalities are all represented in Euclidean spaces (See Fig. 2), but not the case addressed in this paper.

To break the above limitation, this paper proposes a novel framework to embed two entirely heterogeneous spaces, e.g., Euclidean space and Riemannian manifold, into a common discriminant Hamming space. Specifically, we propose a method named Hashing across Euclidean space and Riemannian manifold (HER), which first embeds the two heterogeneous spaces respectively into Reproducing Kernel Hilbert Spaces (RKHS) and then learns the corresponding

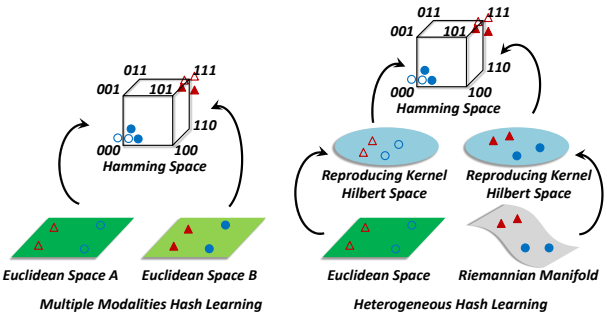


Figure 2: The difference between traditional multiple modalities hash learning methods (the left figure) and our heterogeneous hash learning method (the right figure), where different shapes (i.e., triangles and circles) denote categories.

transformations from either Hilbert space to the final common Hamming space in a max-margin framework. During the learning process, intra- and inter-space discriminability are iteratively optimized for compatibility between the hash codes of the two heterogeneous spaces. To validate our method, we conduct comprehensive experiments on two popular TV-Series, i.e., *the Big Bang Theory* and *Buffy the Vampire Slayer*.

2. Related Work

In this section, we give a brief review of previous literatures that closely relate to our work in either aspects of problem and methodology. In Section 2.1, we overview some existing applications around face video retrieval, and then in Section 2.2 and Section 2.3 we introduce the relevant single modality and multiple modalities hash learning methods, respectively.

2.1. Face Video Retrieval

Recent years have witnessed more and more studies on face video retrieval [33] [2] [3] [10] [31]. Arandjelović and Zisserman [2] [3] built an end-to-end system to retrieve film shots, given one or more query face images. They proposed to obtain an identity preserving and variation insensitive signature image to represent face shot. Anyway, it is a single image based face matching method which does not fully utilize the video information. Instead of matching single faces, Sivic *et al*. [33] developed a video shot retrieval system by matching sets of faces, which are represented as distributions in the form of histogram and measured by Chi-square distance. Nevertheless, these works have large differences from ours. On one hand, they all exploit real-valued video representation rather than binary-valued hash code, which is not quite suited for retrieval task, especially in case of practical large database volume; on the other hand, such works aim to build a complete end-to-end system tailored to face video processing, including shot

boundary detection, face detection and tracking, etc. In contrast, this paper puts its emphasis on the heterogeneous hash learning framework, which is expected to have potential application in more general object retrieval tasks.

2.2. Single Modality Hash Learning

The pioneering hash learning method, i.e., the well-known Locality Sensitive Hashing (LSH) [19], is based on random projections. Although the asymptotic property is theoretically guaranteed, as a data-independent method, LSH still requires long codes to achieve satisfactory precision in practical applications. Realizing the limitation of LSH, recent endeavors aim at data-dependent hashing by exploring either data structure or supervision information to achieve compact hash codes for specific datasets. This new direction is referred to as Hash Function Learning (HFL). Representative unsupervised HFL methods include Spectral Hashing (SH) [41], Anchor Graph Hashing (AGH) [24], Iterative Quantization hashing (ITQ) [12], etc. More recently, semi-supervised and supervised HFL methods are gradually coming into view, such as Semi-Supervised Hashing (SSH) [37], Kernel-based Supervised Hashing (KSH) [23], Discriminative Binary Codes (DBC) [30], and Supervised Iterative Quantization hashing (SITQ) [12], etc. These supervised paradigms move us toward higher performance in practical applications, such as content-based retrieval with massive data.

2.3. Multiple Modalities Hash Learning

The aforementioned single modality HFL methods have been applied to a wide range of real-world tasks with great success. Nevertheless, most of the methods can only deal with data from a single modality. They cannot deal with the cross-modality problem. Nowadays, it is quite common to conduct similarity search involving data from multiple modalities. For instance, given a textual description of certain natural scene, one would like to retrieve some images that depict exactly the described scene. As data from different modalities (e.g., text vs. image) typically reside in different feature spaces, it is reasonable to map the multiple modalities data into a common Hamming space, which will definitely make the cross-modality comparison easier and faster. However, due to its novelty and challenge of this new task, only few methods are proposed for this purpose. Representative methods include the pioneering Cross-Modal Similar Sensitive Hashing (CMSSH) [6], Cross-View Hashing (CVH) [22], Multimodal Latent Binary Embedding (MLBE) [44], Parametric Local Multimodal Hashing (PLMH) [43], Predictable Dual-view Hashing (PDH) [29], and the recent neural network based Multi-Modal NN hashing (MM-NN) [26].

While the above multiple modalities HFL methods have achieved success in applications such as text-image matching, such methods have the limitation that they can only handle the case where the original modalities are all represented in Euclidean spaces. However, in our task, only images lie in Euclidean space, while videos are represented as points lying on SPD Riemannian manifold. Hence, it is infeasible to directly apply the traditional multiple modalities hash learning methods to our task (please see Fig. 2 for more intuitive understanding).

Figure 3: Face samples in two TV-Series, i.e, *the Big Bang Theory* (top row) and *Buffy the Vampire Slayer* (bottom row). Even in a same shot, faces suffer many different types of appearance variations caused by illumination, head pose, expression, occlusion, etc.

3. Video Modeling

Compared with treating video as separated frames and processing it frame by frame, holistic modeling methods, e.g., single or mixture of linear subspaces [42] [21] [40] [38], affine subspaces [7] [17], covariance matrices [39] [25] [36], increasingly exhibit their advantages of not only compact representation but also superior performance. Among these methods, covariance matrix, as the raw second-order statistics of the set of video frames, provides a natural representation for a video with any type of features and any number of frames, and is able to well capture the complicated video structure (see Fig. 3) more faithfully [39]. In fact, as indicated in [39], subspace-based models usually originate from an eigen-decomposition of the covariance matrix without utilizing the information in eigenvalues and non-leading eigenvectors. Taking such into consideration, we resort to covariance matrices for representing videos in this paper.

Let $F = [f_1, f_2, ..., f_n]$ be the data matrix of a video with n frames, where $f_i \in \mathbb{R}^d$ denotes the i^{th} frame with d-dimensional feature description. We represent the video with a $d \times d$ covariance matrix $\mathscr{Y}$ as follows:

$$\mathscr{Y} = \frac{1}{n-1} \sum_{i=1}^{n} (f_i - \bar{f})(f_i - \bar{f})^T, \tag{1}$$

where $\bar{f}$ is the mean of all video frames. The diagonal entries of $\mathscr{Y}$ record the variance of each individual feature, and the off-diagonal entries are their respective correlations. In this way, a video is represented as a nonsingular covariance matrix $\mathscr{Y}$ (in case of singularity, a simple regularization can be imposed to its diagonal [39]), which lies on

Riemannian manifold Sym_d^+ spanned by $d \times d$ Symmetric Positive Definite (SPD) matrices.

Prior to our study here, covariance matrix has been used to characterize local regions within an image, named region covariance [35], and apply to tasks like human detection. However, region covariance is computed within a local region for a single image, whereas our video covariance is the statistic among all frames for a whole video. Moreover, region covariance is intended to depict region texture, whereas ours has the potential to model the appearance variance across frames which is crucial for face video modeling.

4. Heterogeneous Hash Learning

4.1. Problem Description

Assume that we have C categories for training, and data are organized in pairwise structure, i.e., for the c^{th} category we have N_c image-video pairs, where $N = \sum_{c=1}^{C} N_c$ is the total number of training pairs. Both image and individual video frame use the same d-dimensional feature description, as denoted in Section 3. In this paper, we denote a Euclidean point (i.e., an image) by $x_i \in \mathbb{R}^d$, and a Riemannian point (i.e., a video) by $\mathscr{Y}_i \in Sym_d^+$ (here, $\mathscr{Y}_i$ is the frame covariance matrix as defined in Eqn. (1)). Our goal is to learn the hash codes of all the training samples, i.e., $B_e \in \{0,1\}^{K \times N}$ for the N images, $B_r \in \{0,1\}^{K \times N}$ for the N videos, where the subscripts e and r denote Euclidean space and Riemannian manifold, respectively, and K is the hash code length in the final common Hamming space.

4.2. Hash Learning Architecture

As x_i and $\mathscr{Y}_i$ are heterogeneous, it is not trivial to embed them into a common Hamming space directly. To this end, we devise a two-step architecture inspired by [16] [18] to fill the heterogeneous gap and accomplish the common embedding (see Fig. 2). Specifically, in the first step, on the Riemannian manifold side, we propose to map the Riemannian manifold Sym_d^+ into a high dimensional Reproducing Kernel Hilbert Space (RKHS) $\mathscr{H}_r$ via $\eta(\mathscr{Y}_i) : Sym_d^+ \rightarrow \mathscr{H}_r$. There are two advantages by doing this, 1) the mapping transforms the non-linear Riemannian manifold into a (linear) Hilbert space, thus making it possible to utilize algorithms designed for Hilbert space with manifold valued data; 2) as evidenced by the theory of kernel methods in Euclidean space, it yields a much richer representation of the original data distribution [13] [8] [14] [39] [36] [20] [15]. On the Euclidean space side, we can also map the Euclidean space $\mathbb{R}^d$ to another RKHS $\mathscr{H}_e$ via $\varphi(x_i) : \mathbb{R}^d \rightarrow \mathscr{H}_e$ without loss of generality. After the first-step mappings, gap between the two original heterogeneous spaces (i.e., $\mathbb{R}^d$ and Sym_d^+) is dramatically reduced to that between two Hilbert spaces (i.e., $\mathscr{H}_e$ and $\mathscr{H}_r$), and more importantly, this facilitates the subsequent hash functions learning. In the second step, based on the two Hilbert spaces, we learn a common discriminant Hamming space, through iteratively optimizing intra- and inter-space discriminability measured by Hamming distance in a max-margin framework, to guarantee the stability of hash functions of the two heterogeneous spaces.

4.3. Objective Function

To learn a desirable hash functions for the two Hilbert spaces, we believe that three principles need to be taken into consideration, 1) ***discriminability***: the common Hamming space should be first discriminant, where the Hamming distance between samples of the same category should be minimized, meanwhile samples of distinct categories should better have quite different hash codes; 2) ***stability***: let's imagine each hash function (i.e., each bit) as a split in the Hilbert space, we want the splits to be as stable as possible. Intuitively, a split is stable when it has large margins from samples around it [11]. Think about such a disillusionary situation where a split crosses an area with dense samples, many actually neighboring samples will be inevitably assigned different hash values. In a nutshell, similar samples in the feature space should be mapped to similar hash codes within a short Hamming distance; 3) ***compatibility***: due to the heterogeneous representations of two mediums (i.e., images and videos), we should consider not only the intra- but also the inter-space discriminability constraints. Having such principles in mind, we formulate our objective function in Eqn. (2),

$$\begin{aligned}
\min_{W_e,W_r,\xi_e,\xi_r,B_e,B_r} \quad & \lambda_1 E_e + \lambda_2 E_r + \lambda_3 E_{er} \\
& + \gamma_1 \sum_{k\in\{1:K\}} \left\| w_e^k \right\|^2 + C_1 \sum_{\substack{k\in\{1:K\} \\ i\in\{1:N\}}} \xi_e^{ki} \\
& + \gamma_2 \sum_{k\in\{1:K\}} \left\| w_r^k \right\|^2 + C_2 \sum_{\substack{k\in\{1:K\} \\ i\in\{1:N\}}} \xi_r^{ki} \\
s.t. B_e^{ki} &= sgn({w_e^k}^T \varphi(x_i)), \forall k \in \{1:K\}, i \in \{1:N\} \\
B_r^{ki} &= sgn({w_r^k}^T \eta(\mathscr{Y}_i)), \forall k \in \{1:K\}, i \in \{1:N\} \\
B_r^{ki}({w_e^k}^T \varphi(x_i)) &\geq 1 - \xi_e^{ki}, \xi_e^{ki} > 0, \forall k \in \{1:K\}, i \in \{1:N\} \\
B_e^{ki}({w_r^k}^T \eta(\mathscr{Y}_i)) &\geq 1 - \xi_r^{ki}, \xi_r^{ki} > 0, \forall k \in \{1:K\}, i \in \{1:N\},
\end{aligned} \tag{2}$$

where B_*^{ki} is the hash value of the i^{th} sample using the k^{th} split (hash function), w_*^k is the weight vector corresponding to the k^{th} split, ξ_*^{ki} is the slack variable corresponding to the i^{th} sample of the k^{th} split.

In Eqn. (2), the first three terms, i.e., E_e, E_r, and E_{er}, denote the ***discriminability*** energy constraints in Euclidean space, Riemannian manifold, and cross-Euclidean-Riemannian space. The formulations of them can be found in Eqn. (3), Eqn. (4), and Eqn. (5), where $d(\cdot,\cdot)$ can be any distance in the Hamming space, and λ_e, λ_r, and λ_{er}

are the pre-computable trade-off parameters to balance the within-, between-category scales. The original intention to design these energy functions is to minimize the Hamming distance between samples of the same category, and meanwhile maximize the Hamming distance between samples from different categories.

$$E_e = \sum_{c\in\{1:C\}} \sum_{m,n\in c} d(B_e^m, B_e^n) - \lambda_e \sum_{\substack{c_1\in\{1:C\} \\ p\in c_1}} \sum_{\substack{c_2\in\{1:C\} \\ c_1\neq c_2, q\in c_2}} d(B_e^p, B_e^q) \quad (3)$$

$$E_r = \sum_{c\in\{1:C\}} \sum_{m,n\in c} d(B_r^m, B_r^n) - \lambda_r \sum_{\substack{c_1\in\{1:C\} \\ p\in c_1}} \sum_{\substack{c_2\in\{1:C\} \\ c_1\neq c_2, q\in c_2}} d(B_r^p, B_r^q) \quad (4)$$

$$E_{er} = \sum_{c\in\{1:C\}} \sum_{m,n\in c} d(B_e^m, B_r^n) - \lambda_{er} \sum_{\substack{c_1\in\{1:C\} \\ p\in c_1}} \sum_{\substack{c_2\in\{1:C\} \\ c_1\neq c_2, q\in c_2}} d(B_e^p, B_r^q) \quad (5)$$

The fourth and fifth terms in Eqn. (2) correspond to the ***stability*** constraint on hash functions in Euclidean space, which is accomplished by SVM in practice with its inherent max-margin property. Similarly, the sixth and seventh terms in Eqn. (2) correspond to the stability constraint on hash functions in Riemannian manifold. The ***compatibility*** constraint is reflected in two aspects: one is the inter-space discriminability energy term E_{er}, and the other is the cross training scheme incorporated in the last two constraint conditions in Eqn. (2). The motivation behind the cross training scheme lies in the pairwise structure of training data. In particular, we would like to make the two elements (i.e., x_i and $\mathscr{Y}_i$) in a pair share the same code, so cross space training naturally serves as an effective way to achieve it, which definitely strengthens the connection between the two spaces at the same time.

4.4. Iterative Optimization

While it is intractable to find the global minimum of the objective function, here we try to seek a local optima to obtain good hash codes, which are capable of yielding desirable results. In particular, we exploit an iterative block coordinate descent method [34] to go down the objective function. The whole optimization procedure is formulated in Algorithm 1. Here we describe it step by step. First of all (line 1), the aforementioned two mappings, i.e., $\varphi(x_i) : \mathbb{R}^d \to \mathscr{H}_e$, and $\eta(\mathscr{Y}_i) : Sym_d^+ \to \mathscr{H}_r$, are usually implicit in practice. Hence, taking the Euclidean space mapping $\varphi(\cdot)$ as an example, we use the kernel trick [4] by expressing the weight vector w_e^k as a linear combination of all the training samples in the mapped Hilbert space $\mathscr{H}_e$ as

$$w_e^k = \sum_{i=1}^{N} u_e^{ki} \varphi(x_i), \quad (6)$$

where u_e^{ki} is the i^{th} expansion coefficient. Therefore,

$$w_e^{k^T} \varphi(x_j) = \sum_{i=1}^{N} u_e^{ki} \varphi(x_i)^T \varphi(x_j) = u_e^{k^T} K_e^{\cdot j}, \quad (7)$$

Algorithm 1 Optimization

INPUT: Training samples from heterogeneous spaces, i.e., $\{x_i, \mathscr{Y}_i, l_i\}$, where $x_i \in \mathbb{R}^d$, $\mathscr{Y}_i \in Sym_d^+$, $i \in \{1,2,...,N\}$, $l_i \in \{1,2,...,C\}$.
OUTPUT: $B_e, B_r \in \{0,1\}^{K\times N}$.
1. Compute kernel matrices $K_e, K_r \in \mathbb{R}^{N\times N}$ with Eqn. (8) and Eqn. (9)
2. $V_e \in \mathbb{R}^{N\times K}, V_r \in \mathbb{R}^{N\times K} \leftarrow KGMMFA\ or\ KCCA(K_e, K_r)$
3. $B_e \leftarrow sgn(V_e{}^T K_e)$
4. $B_r \leftarrow sgn(V_r{}^T K_r)$
5. **repeat**
6. Optimize B_e with Eqn. (3)
7. Optimize B_r with Eqn. (4)
8. Train K kernel SVMs on K_e to update U_e by using B_r as training labels, and inversely train another K kernel SVMs on K_r to update U_r by using B_e as training labels, where $U_e = [u_e^1, u_e^2, ..., u_e^K] \in \mathbb{R}^{N\times K}$, and $U_r = [u_r^1, u_r^2, ..., u_r^K] \in \mathbb{R}^{N\times K}$
9. $B_e \leftarrow sgn(U_e{}^T K_e)$
10. $B_r \leftarrow sgn(U_r{}^T K_r)$
11. Optimize $B = [B_e, B_r] \in \{0,1\}^{K\times 2N}$ with Eqn. (5) to further update B_e and B_r
12. Train K kernel SVMs on K_e to update U_e by using B_r as training labels, and inversely train another K kernel SVMs on K_r to update U_r by using B_e as training labels
13. **until** Convergence
14. $B_e \leftarrow sgn(U_e{}^T K_e)$
15. $B_r \leftarrow sgn(U_r{}^T K_r)$

where u_e^k is an $N \times 1$ column vector with its i^{th} entry being u_e^{ki}, and $K_e^{\cdot j}$ is the j^{th} column of the kernel matrix $K_e \in \mathbb{R}^{N\times N}$. Here K_e is an $N \times N$ kernel matrix for the Euclidean points, which is computed as follows,

$$K_e^{ij} = \varphi(x_i)^T \varphi(x_j) = exp(-\frac{\left\|x_i - x_j\right\|_2^2}{2{\sigma_e}^2}). \quad (8)$$

Similarly, $K_r \in \mathbb{R}^{N\times N}$ is the kernel matrix for the Riemannian points, which is computed by Eqn. (9).

$$K_r^{ij} = \eta(\mathscr{Y}_i)^T \eta(\mathscr{Y}_j) = exp(-\frac{\left\|\log(\mathscr{Y}_i) - \log(\mathscr{Y}_j)\right\|_F^2}{2{\sigma_r}^2}) \quad (9)$$

Without loss of generality, in this paper we use Gaussian kernel on the Euclidean space side, and Gaussian-logarithm kernel [36] [20] on the Riemannian manifold side. In both kernel functions, σ_e and σ_r can be easily set to the mean distances of training samples. Actually, the kernel mapping mainly severs as a building block to complete the first mapping step and HER welcomes and is compatible with any reasonable explicit or implicit kernel for both spaces.

After the computation of two kernel matrices, i.e., K_e and K_r, we use Kernelized Generalized Multiview Marginal

Fisher Analysis (KGMMFA) [32] or Kernelized Canonical Correlation Analysis (KCCA) [16] to embed the two Hilbert spaces (i.e., $\mathscr{H}_e$ and $\mathscr{H}_r$) into a common Euclidean space for hash codes initialization (line 2∼line 4). After the initialization, we proceed by iterating five steps in sequence (line 5∼line 13). **First**, we optimize Eqn. (3) and Eqn. (4) to update B_e and B_r for promoting the intra-space discriminability. Here we use the binary optimization algorithm proposed in [30][1] with an efficient subgradient descent. **Second**, we use the updated B_* to train K two-class kernel SVMs for each Hilbert space. Specially, we adopt the cross training strategy by using B_e as training labels to train the Riemannian manifold side SVMs with kernel matrix K_r, and vice versa. This strategy was proven to be effective especially for pairwise training samples [29] [27]. **Third**, update the current value of B_* to reflect the hash codes that these SVMs actually yield. **Fourth**, B_e and B_r are mixed together to be optimized with Eqn. (5) for promoting the inter-space discriminability. **Fifth**, the same as the second step, i.e, cross training the SVMs with the updated B_*. The optimization is finished once converged, and usually in practice a couple of iterations can lead to convergence and good hash codes (please refer to our supplementary material).

4.5. Discussion

Application Scope: The Riemannian manifold in this paper is not limited to that spanned by covariance matrices. In fact, video modeling methods like linear subspaces (spanning Grassmann manifold), affine subspaces (spanning affine Grassmann manifold) can also be involved in our framework. Moreover, our framework is not limited to Euclidean space v.s. Riemannian manifold. Actually, it provides down-level compatibility, e.g., Euclidean space v.s. Euclidean space, and Riemannian manifold v.s. Riemannian manifold. Furthermore, as a general heterogeneous hash learning framework, our methodology opens a new way to any potential practical application in which data come from heterogeneous spaces.

Parameters Sensitivity: Although quite a few parameters are observed in Eqn. (2), the proposed method is parameter insensitive as the objective function is optimized in an iterative manner for each component separately, i.e., λ_1, λ_2, λ_3, γ_1, and γ_2 mainly play the role of balancing each component, and were simply set to equally weight those components (λ_1, λ_2, λ_3 are set to 1, and γ_1, γ_2 are set to 0.5). Besides, the only substantial parameters are the soft margin parameters C_1 and C_2, which were simply set to 1 as standard SVM.

[1]This optimization algorithm can guarantee the code diversity in two aspects: a) bit-wise balance: for each training sample, the algorithm guarantees the balance of bit numbers of -1 and 1; b) sample-wise balance: for each bit, the algorithm guarantees the balance of distributions of -1 and 1 on all the training samples.

Initialization Option: The initialization of our method is not limited to KGMMFA and KCCA. Theoretically, any one of the Kernelized Multi-view Learning (KML) [32] methods is competent for this purpose (please find more results in supplementary material).

Kernel Scalability: Inevitably, kernel methods often scale imperfectly with large data size. Fortunately, a series of mathematically principled solutions, e.g., linear random projections [1], low-rank approximation [9], and random features [28], have been well established that are just tailored to the further need of scalability. Moreover, observed from experimental results in Section 5, only a couple of hundred training samples can achieve satisfactory performance.

5. Experiments

In this section, we evaluate our method, i.e., HER[2], on face video retrieval task with two challenging TV-Series. Specifically, we conduct two groups of comparisons, i.e., HER vs. Single Modality Hash (SMH) learning methods in Section 5.2, and HER vs. Multiple Modalities Hash (MMH) learning methods in Section 5.3.

5.1. Databases and Experimental Settings

Databases: The first one consists of 3341 face videos of the first 6 episodes from season 1 of *the Big Bang Theory* (BBT), and the second one consists of 4779 face videos of the first 6 episodes from season 5 of *Buffy the Vampire Slayer* (BVS). These two TV-Series are quite different in their filming styles, and therefore pose different challenges. BBT is a sitcom (about 20 minutes an episode) mostly taking place indoors with a main cast of 5∼8 characters. It includes many full-view shots which contain multiple characters at a time, however the faces are rather small (an average of 75px face size). On the other hand, BVS has about 40 minutes an episode, with a main cast size around 12 sometimes up to 18 in specific episodes. Many shots are set at night and outdoors, resulting in a large range of different illumination. However, it also contains a sizable number of face close-up shots (an average of 116px face size). We use the extracted face videos represented by block Discrete Cosine Transformation (DCT) feature as used in [5]. More specifically, each face frame is represented with a 240-d DCT feature, and thus forms a 240×240 covariance video representation. Faces are aligned and normalized without special preprocessing, and nothing constraint is added to query image. The distribution of face videos per character can be found in supplementary material.

Experimental Settings: The initialization of Algorithm 1 was accomplished by KGMMFA, because KGMMFA

[2]The matlab implementation of HER can be downloaded from `http://vipl.ict.ac.cn/resources/codes`.

Table 1: Comparison with the state-of-the-art single modality and multiple modalities hash learning methods with mAP on two databases. K means the length of hash code.

Type	Method	*the Big Bang Theory*				*Buffy the Vampire Slayer*			
		$K = 16$	$K = 32$	$K = 64$	$K = 128$	$K = 16$	$K = 32$	$K = 64$	$K = 128$
single modality hash learning method	LSH [19]	0.2086	0.2092	0.1963	0.1994	0.1508	0.1517	0.1568	0.1578
	SH [41]	0.2652	0.2665	0.2623	0.2673	0.2046	0.2237	0.2177	0.2222
	ITQ [12]	0.3025	0.2989	0.3029	0.3060	0.1848	0.1972	0.2265	0.2457
	SSH [37]	0.2855	0.2662	0.2584	0.2586	0.2193	0.2202	0.2141	0.2120
	DBC [30]	0.4495	0.4235	0.4005	0.3867	0.3858	0.4460	0.4707	0.4547
	KSH [23]	0.4366	0.4454	0.4567	0.4604	0.3542	0.4149	0.4385	0.4517
	SITQ [12]	0.3909	0.4298	0.4576	0.4799	**0.3869**	0.4580	0.4738	0.4990
multiple modalities hash learning method	CMSSH [6]	0.2047	0.2143	0.2024	0.2478	0.1569	0.1559	0.1593	0.1688
	CVH [22]	0.2110	0.2092	0.2231	0.2407	0.1579	0.1570	0.1644	0.1900
	PLMH [43]	0.2447	0.2461	0.2487	0.2608	0.1859	0.1800	0.1828	0.1853
	PDH [29]	0.2949	0.2903	0.3095	0.2916	0.1769	0.1865	0.1846	0.1980
	MLBE [44]	0.2600	0.2648	0.3917	0.3858	0.1550	0.1720	0.1759	0.1840
	MM-NN [26]	0.3955	0.4664	0.5124	0.4922	0.2207	0.2681	0.3671	0.4045
Ours	**HER**	**0.5049**	**0.5227**	**0.5490**	**0.5539**	0.3770	**0.4852**	**0.5281**	**0.5877**

utilizes more discriminant information compared with KCCA in which only side information is used. The length of hash code ranges from 16 to 128, as no more obvious performance improvement is observed with 256 bits. For the competing methods, source codes of them were kindly provided by the original authors. For fair comparison, important parameters of each method were empirically tuned according to the recommendations in the original references as well as the source codes.

Tasks and Measurements: Our task is conducting face video retrieval with image query. The images were acquired by randomly extracting a frame from each video. For each database, we randomly selected 300 image-video pairs (both elements of the pair come from the same subject) for training (300 is a trade-off between retrieval accuracy and computational cost), and then selected 100 images from the rest as query for the retrieval task. For quantitative evaluation, we use the standard definitions of mean Average Precision (mAP) and precision recall curves calculated among the range of whole database as measurements. For space limitation, we only show the evaluation on image query vs. video database, and actually HER is also qualified to the inverse task, i.e., video query vs. image database (please find corresponding results in supplementary material).

5.2. Comparison with Single Modality Hash Learning Methods

Strictly speaking, SMH learning methods are not qualified to accomplish the cross-modality matching task. Nevertheless, our task has its own characteristic, where video is just composed of frames which actually are images. Therefore, as mentioned in Section 1, we can straightforwardly treat video as a set of separated frames, then compute the similarities between the image and each frame, and finally integrate such similarities by computing the average (in fact, we had evaluated the maximum, minimum, and average strategies. Here only the average version is shown because of its relatively higher performance).

In this group of experiment, we compare HER with seven representative SMH learning methods, including LSH [19], SH [41], ITQ [12], SSH [37], DBC [30], KSH [23], and SITQ [12]. The performance comparison is shown in top rows of Table 1. According to the results on two databases, we have the following two consistent observations: 1) Our method outperforms the SMH learning methods as expected. This is partly attributable to the promising holistic modeling of video by covariance matrix, which can characterize all kinds of complicated variations in face video, including illumination, head pose, facial expression, etc; 2) Supervised SMH methods, i.e., DBC, KSH, SITQ, unsurprisingly outperform the unsupervised and semi-supervised ones. This mainly benefits from the identity label information utilized during the hash functions learning. We can also observe that, in some specific setting, e.g., BVS database with 16 bits hash code, SITQ even surpasses HER. However, the core superiority of our method is the compact representation for videos. Note that, under fixed hash code length for a k-frame video, SMH learning methods will cost k times of bits as much as ours. In case of large volume video, this gap will be unaffordable.

5.3. Comparison with Multiple Modalities Hash Learning Methods

As pointed before, traditional MMH learning methods can only deal with the restricted situation, where the modalities are all represented in Euclidean spaces. Therefore, to conduct the comparison, we have to modify these methods by applying an explicit Riemannian kernel map $\vartheta(\cdot)$ as [39] to map the covariance matrices $\mathscr{Y}$ from Riemannian manifold Sym_d^+ to Euclidean space $\mathbb{R}^{d\times d}$, i.e., $\vartheta(\mathscr{Y}) : \mathscr{Y} \rightarrow log(\mathscr{Y})$. After that, traditional MMH learning methods can be applied to our task.

In this group of experiment, we compare HER with six

representative MMH learning methods, including CMSSH [6], CVH[3] [22], PLMH [43], PDH [29], MLBE [44], and MM-NN [26]. The mAP comparison is shown in bottom rows of Table 1. Moreover, as this class of methods are closely related to our method, we also show the precision recall curves in Fig. 4 (please find more results in supplementary material).

We can see that HER significantly outperforms all the competing methods, due to their inherent limitations as explained below: CMSSH ignores the intra-modality relational information which could be very useful for cross-modality matching; CVH is limited to a relatively narrow class of globally linear multiple modalities hash function learning that often cannot capture well the structure of the data for each modality; PDH only uses the side information which is doomed to limited discriminability; MLBE has limitations on the restrictive global intra-modality weighting matrices involved in the probabilistic model; PLMH models the complex structure of datasets via using different hash functions at different locations, but a lot of sensitive parameters need to be tuned; compared with the above ones, MM-NN performs the best based on a coupled siamese neural network architecture.

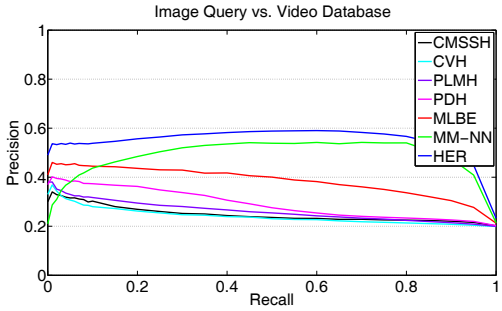
In contrast, the superiority of our method against traditional multiple modalities hash learning methods mainly benefits from three points: 1) the integration of intra- and inter-space discriminability constraints (i.e., E_e, E_r, and E_{er}) via an iterative optimization based on Hamming distance; 2) the two-step architecture, i.e., Euclidean space (Riemannian manifold) to RKHS and then to common Hamming space, involves nonlinear maps from the original spaces into high dimensional Hilbert spaces, which would yield much richer representations of the original data distributions; 3) the max-margin strategy accomplished by SVM further ensures the stability and generalizability of the learned hash functions, which is a crucial element for practical retrieval system.

To further justify the effectiveness of the proposed method, we have also evaluated HER on one more video surveillance database, and also compared HER with several representative key-frame based video classification methods. For space limitation, please find details in supplementary material.

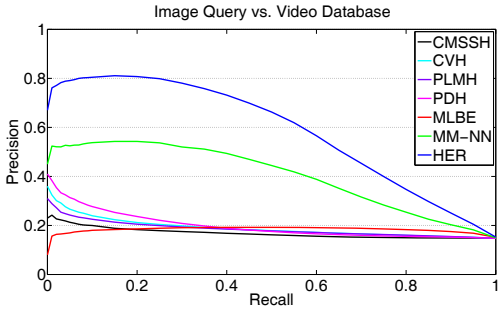
6. Conclusions

In this paper, we have proposed a novel heterogeneous hash learning method named HER, with which two entirely heterogeneous spaces, e.g., Euclidean space and Riemannian manifold, can be embedded into a common discriminant Hamming space. During the learning of hash functions, three principles - discriminability, stability, and compatibility - were explored to iteratively optimize the cross-space hash codes (in a max-margin framework). Extensive experiments on face video retrieval demonstrated the superiority of our method over the state-of-the-art single modality and multiple modalities hash learning methods. For future work, we would like to investigate three possible extensions: 1) integration of temporal information with current video modeling; 2) extension to multiple heterogeneous spaces embedding from the current dual-space version; 3) application to the challenging particular pedestrian retrieval via massive surveillance video.

[3]Because the code is not publicly available, the implementation of CVH is kindly provided by the MLBE authors



(a) *the Big Bang Theory*, $K = 128$



(b) *Buffy the Vampire Slayer*, $K = 128$

Figure 4: Comparison with the state-of-the-art multiple modalities hash learning methods with precision recall curves on two databases. K means the length of hash code. For space limitation, we only showed experimental results with 128 bits, and more results can be found in supplementary material.

Acknowledgements

This work is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61222211, 61379083, and the FiDiPro program of Tekes.

References

[1] D. Achlioptas, F. McSherry, and B. Schölkopf. Sampling techniques for kernel methods. In *NIPS*, volume 1, page 335. MIT Press, 2002.

[2] O. Arandjelović and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *CVPR*, volume 1, pages 860–867. IEEE, 2005.

[3] O. Arandjelović and A. Zisserman. On film character retrieval in feature-length films. In *Interactive Video*, pages 89–105. Springer, 2006.

[4] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[5] M. Bäuml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *CVPR*. IEEE, 2013.

[6] M. Bronstein, A. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601. IEEE, 2010.

[7] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, pages 2567–2573. IEEE, 2010.

[8] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, pages 1932–1939. IEEE, 2009.

[9] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, 6:2153–2175, 2005.

[10] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. 2006.

[11] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *CVPR*, pages 1–8. IEEE, 2007.

[12] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824. IEEE, 2011.

[13] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, pages 376–383. ACM, 2008.

[14] M. Harandi, C. Sanderson, S. Shirazi, and B. Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR*, pages 2705–2712. IEEE, 2011.

[15] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell. Kernel analysis over riemannian manifolds for visual recognition of actions, pedestrians and textures. In *WACV*, pages 433–439. IEEE, 2012.

[16] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[17] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, pages 121–128. IEEE, 2011.

[18] Z. Huang, R. Wang, S. Shan, and X. Chen. Learning euclidean-to-riemannian metric for point-to-set classification. In *CVPR*, pages 1677–1684. IEEE, 2014.

[19] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613. ACM, 1998.

[20] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, pages 73–80. IEEE, 2013.

[21] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *PAMI*, 29(6):1005–1018, 2007.

[22] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365. AAAI Press, 2011.

[23] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE, 2012.

[24] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *ICML*, pages 1–8, 2011.

[25] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[26] J. Masci, M. Bronstein, A. Bronstein, and J. Schmidhuber. Multimodal similarity-preserving hashing. *PAMI*, 2013.

[27] F. Mirrashed and M. Rastegari. Domain adaptive classification. In *ICCV*, pages 2608–2615, 2013.

[28] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[29] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis. Predictable dual-view hashing. In *ICML*, pages 1328–1336, 2013.

[30] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, pages 876–889. Springer, 2012.

[31] C. Shan. Face recognition and retrieval in video. In *Video Search and Mining*, pages 235–260. Springer, 2010.

[32] A. Sharma, A. Kumar, H. Daume, and D. Jacobs. Generalized multiview analysis: A discriminative latent space. In *CVPR*, pages 2160–2167. IEEE, 2012.

[33] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *Image and Video Retrieval*, pages 226–236. Springer, 2005.

[34] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[35] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, pages 1–8. IEEE, 2007.

[36] R. Vemulapalli, J. K. Pillai, and R. Chellappa. Kernel learning for extrinsic classification of manifold features. In *CVPR*, pages 1782–1789. IEEE, 2013.

[37] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431. IEEE, 2010.

[38] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR*, pages 429–436. IEEE, 2009.

[39] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, pages 2496–2503. IEEE, 2012.

[40] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, pages 1–8. IEEE, 2008.

[41] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008.

[42] O. Yamaguchi, K. Fukui, and K.-i. Maeda. Face recognition using temporal image sequence. In *FG*, pages 318–323. IEEE, 1998.

[43] D. Zhai, H. Chang, Y. Zhen, X. Liu, X. Chen, and W. Gao. Parametric local multimodal hashing for cross-view similarity search. In *IJCAI*, pages 2754–2760. AAAI Press, 2013.

[44] Y. Zhen and D.-Y. Yeung. A probabilistic model for multimodal hash function learning. In *KDD*, pages 940–948. ACM, 2012.