Semi-paired hashing for cross-view retrieval

Xiaobo Shen^a, Quan-Sen Sun^{a,*}, Yun-Hao Yuan^b

^a*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*

^b*Department of Computer Science and Technology, Jiangnan University, Wuxi 214122, China*

Abstract

Hashing techniques have been widely applied in the large-scale cross-view retrieval tasks due to the significant advantage of hash codes in computation and storage efficiency. Most existing cross-view hashing methods can only handle fully-paired scenarios, where all samples from different views are paired, however such full pairwise correspondences may not be available in practical applications. In this paper, we propose a novel hashing method, named Semi-paired Hashing (SPH), to deal with a more challenging cross-view retrieval task, where only partial pairwise correspondences are provided in advance. Specifically, SPH aims to preserve within-view similarity and cross-view correlation among multi-view data. Similarity structures within each view are obtained via anchor graph. As limited samples are paired, correlations between unpaired samples are exploited via a simple yet effective approach, which estimates cross-view correlations by partial cross-view pairwise information and within-view similarity structure. Besides, we further incorporate two regression terms between original features and target binary codes to reduce the quantization loss. An efficient iterative algorithm is presented to simultaneously solve hash functions and binary codes. Extensive experiments on two benchmark datasets demonstrate the superiority of SPH over the state-of-the-art methods, especially in the semi-paired scenarios.

Keywords: Hashing, Semi-paired, Cross-view, Retrieval

---

*Corresponding author. Tel: +86 25 84315142; fax: +86 25 84318156.

Email addresses: njust.shenxiaobo@gmail.com (Xiaobo Shen),
sunquansen@njust.edu.cn (Quan-Sen Sun), yyzhbh@163.com (Yun-Hao Yuan)

---

1. Introduction

Recent years have witnessed the explosive growth of the multimedia data, which brings great challenges to information search and retrieval. Hashing-based nearest neighbor search [1] has attracted considerable interests for its great gains of both storage and computation in massive multimedia data. The basic idea of hashing is to learn a set of short binary codes for high-dimensional data while preserving similarity structure in the original space.

One of the most well-known hashing methods is Locality Sensitive Hashing (LSH) [2], which uses locality sensitive functions to embed the data into Hamming space. Due to the limitation of random projection, LSH usually requires long hash code and multiple hash tables to guarantee reasonable retrieval performance. Later machine learning techniques are introduced into hashing, and many learning-based hashing methods [1, 3, 4, 5, 6, 7, 8, 9, 10, 11] are then proposed, including Spectral Hashing (SH) [3], Anchor Graph Hashing (AGH) [4], PCA Hashing (PCAH) [5], Iterative Quantization (ITQ) [12], Inductive Manifold Hashing (IMH) [6, 8], Supervised Discrete Hashing [9], etc. However, these foregoing hashing methods are single-view approaches, which focus on learning binary codes from data with only single view.

Nowadays data with multiple views are often available in many applications, especially in multimedia domains [13, 14, 15]. For example, the wikipedia website provides both text and image descriptions for each entry. It is desirable to leverage multiple views into binary code learning, since such multi-view data [15, 16] usually offers more comprehensive information than single-view data. Existing hashing methods learning from multi-view data can be mainly divided into two categories: *multi-view hashing* (MVH) and *cross-view hashing* (CVH). By leveraging multiple views, MVH [14, 17, 18, 19] aims to learn better codes than single-view hashing, but requires that all views should be available in advance. Unlike the purpose of MVH, CVH is proposed to support cross-view retrieval [13], where a query of one view can search for the relevant results of

Figure 1: Illustration of the proposed Semi-paired Hashing (SPH).

another view. For example, one might need to find pictures on the web that best illustrate given texts, or find texts that best match a given picture. Therefore, CVH is of great practical demand and interest to many applications.

Recently, some useful attempts [20, 21, 22, 23, 24, 25, 26, 27, 28] have been made towards effective cross-view hashing, which exploit correlations and similarity structures across multiple views. While showing promising performances, these methods may encounter the following limitations:

- Most existing cross-view hashing methods, e.g., [21, 24, 25] can only perform in the fully-paired scenarios, which require all samples from different views to have the one-to-one correspondence. However such full pairwise information may not be provided in practical applications.
- Some hashing methods utilize the given pairwise [21, 24, 23, 25] or semantic [20, 22, 26, 27] information to exploit the correlations across views. In the unsupervised semi-paired scenarios, where the above prior information is not or partially available, these methods can not work well.
- Due to the binary constraints (NP hard) in hashing methods, many methods [21, 24, 23, 26] usually adopt the spectral relaxation technique [3] for simplifying the optimization. However, such optimization on continuous variables deviates from the hashing objective, as quantization loss is not

considered in learning process. Consequently, binary codes generated by
50 such way may be suboptimal for retrieval [12, 29].

Based on the above considerations, in this paper, we propose a novel cross-view hashing method, named Semi-paired Hashing (SPH) for retrieval tasks. The process of SPH is illustrated in Figure 1. The main contributions of this paper can be summarized as follows:

- 55 • We propose a hashing method to deal with a *semi-paired cross-view* retrieval task, where only partial pairwise correspondences are known in advance. As limited prior cross-view information is given in the semi-paired scenarios, this task is more challenging.
- 60 • To fully exploit the cross-view correlation, we incorporate an effective approach, which estimates the correlations between unpaired samples from different views by combining the partly given cross-view pairwise and within-view structure information.
- 65 • We add two regression terms between original features and target binary codes to explicitly reduce the quantization loss. Besides, we design an efficient algorithm to iteratively solve the projection matrices and the target binary codes.

The remainder of this paper is organized as follows. We first briefly review related works in Section 2. The details of our proposed method are presented in Section 3. Extensive experimental evaluation are given in 4. Finally, we draw
70 conclusions in Section 5.

2. Related work

In this section, according to the properties of data, we review hashing into two categories, including hashing on single-view data, hashing on multi-view data.

75 2.1. Hashing on single-view data

Most existing hashing methods utilize single-view data to generate hash codes, thus we name this category as single-view hashing methods, which can also be broadly divided into two categories: *data-independent* methods, *data-dependent* methods. The data-independent hashing methods mainly include
80 Locality Sensitive Hashing (LSH) [2] and its extensions [30, 31, 32].

Later considerable attentions have been paid to data-independent hashing methods, which apply some machine learning techniques to generate more compact data-related hash functions. One of the earliest work in this category is Spectral Hashing (SH) [3], which utilizes the distribution of data and turns to
85 be eigen-decomposition problem of a graph Laplacian matrix [33]. Later Anchor Graph Hashing (AGH) [4] is proposed to adopt anchor graph for hashing learning, and can be more efficient than SH. To ease the quantization loss in the binarization process, Gong et al. [12] proposed Iterative Quantization (ITQ), which aims to find an optimal rotation matrix such that the difference between
90 binary codes and original data is minimized. Besides, some other variants in this category can be found in [1, 5, 6, 7, 8, 9].

2.2. Hashing on multi-view data

Recent years more and more multi-view data [13, 17, 16] have been available in real applications, consequently hashing on the multi-view data has received
95 lots of attentions.

One category is multi-view hashing [14, 17, 18, 19, 34], which fuses multiple sources from the same objects to get better binary codes than the single-view methods. Multiple Feature Hashing (MFH) [14] preserves the local structure of each view and globally considers the alignments of all views to learn a group
100 of hash functions. Multiple Feature Kernel Hashing (MFKH) [18] learns hash functions by preserving certain similarities with linearly combined multiple kernels corresponding to different features. Kim et al. [19] proposed Multi-View Spectral Hashing (SU-MVSH), which computes the α -averaged similarity ma-

trix from all views, and adopts the sequential learning approach to obtain the
105 hash function.

Besides, another category is cross-view hashing. Until now many cross-view
hashing methods [20, 21, 22, 23, 24, 25, 26, 27] have been proposed. For exam-
ple, Bronstein et al. [20] proposed the cross-modality search hashing (CMSSH),
which learns hashing functions among different views. Kumar and Udupa [21]
110 extended spectral hashing to multi-view fields. Both Inter-media Hashing (IMH)
[23] and Linear Cross-Modal Hashing (LCMH) [24] introduce intra-view simi-
larity preservation and inter-view consistency to discover a compact Hamming
space, and solve hash functions by eigenvalue decomposition. Collective Matrix
Factorization Hashing (CMFH) [25] learns unified binary codes by collective
115 matrix factorization with latent factor model, which can support cross-view re-
trieval. Zhang and Li [26] proposed a supervised cross-view hashing method,
called Semantic Correlation Maximization (SCM), to seamlessly integrate se-
mantic labels into hash learning.

3. Semi-paired hashing

120 3.1. Problem statement

In some practical applications, samples in the database are often represent-
ed as multiple views. For simplicity, here we only consider two views (e.g.,
text and image), although proposed method can also be extended to handle
the problem with more than two views. Specifically, we assume two different
125 views $X = [x_1, x_2, \dots, x_{n_0}, x_{n_0+1}, \dots, x_{n_x}]$, $Y = [y_1, y_2, \dots, y_{n_0}, y_{n_0+1}, \dots, y_{n_y}]$,
where $x_i \in \mathbb{R}^{d_x}$, $y_i \in \mathbb{R}^{d_y}$, n_x , n_y are the number of samples in view X , Y ,
respectively. Without loss of generality, we assume the first n_0 samples in t-
wo views come from the same n_0 objects, that is, $\{x_i, y_i\}_{i=1}^{n_0}$ are paired, where
 n_0 is the number of pairs, while rest samples lack such one-to-one correspon-
130 dence. Besides, we further assume samples in each view are zero-centered, i.e.,
 $\sum_{i=1}^{n_x} x_i = 0$, $\sum_{i=1}^{n_y} y_i = 0$. As we can see, the above is a *semi-paired multi-view*

scenario, where only partial samples are paired, and thus is more realistic and challenging.

The goal is to learn hashing functions for these two views: $f(x) : \mathbb{R}^{d_x} \rightarrow \{-1, 1\}^c$ and $g(y) : \mathbb{R}^{d_y} \rightarrow \{-1, 1\}^c$, where c is the length of binary codes. These two hash functions map samples of two different views into a common hamming space, where similarity of data with different views should be preserved. Similar to existing works [24, 23], we here consider our hash functions the following forms

$$\begin{aligned} f(x) &= \text{sign} \left(W^{(X)T} x \right) \\ g(y) &= \text{sign} \left(W^{(Y)T} y \right) \end{aligned} \tag{1}$$

where $\text{sign}(\cdot)$ denotes the element-wise sign function, $W^{(X)} \in \mathbb{R}^{d_x \times c}$ and $W^{(Y)} \in \mathbb{R}^{d_y \times c}$ are the projection matrices. Therefore, we aim to jointly learn $W^{(X)}$ and $W^{(Y)}$ to achieve effective cross-view retrieval via hashing.

3.2. Within-view similarity preservation

In this section, we focus on preserving within-view similarity, as some re-searches [3] have shown that similarity structure of data is crucial to hashing. Here within-view similarity preservation is designed to maintain the neighborhood relationships among the samples in each view after being mapped into the Hamming space. One key step for similarity preservation in many graph-based hashing methods is to build a neighborhood graph on the data. The time complexity of graph construction in the traditional procedure is $O(n^2)$, where n is the number of samples, which is intractable for large-scale databases. Here we instead adopt anchor graph [4, 35] for graph construction due to its computation efficiency (time complexity is $O(n)$) and powerful approximation to the true K-NN graph.

Anchor graph uses a small set of samples called anchors to approximate the neighborhood structure. Take view X for example, we first generate m ($m \ll n_x$) anchor points $\{\mu_i\}_{i=1}^m$ by applying scalable K-means clustering [36] on the training data of view X . Then we construct the truncated similarity matrix

$Z^{(X)} \in \mathbb{R}^{n_x \times m}$ between training samples and anchors, where the similarity
 160 between x_i and μ_j is defined as follows:

$$Z_{ij}^{(X)} = \begin{cases} \frac{\exp(\mathcal{D}^2(x_i, \mu_j)/\sigma)}{\sum_{j \in [i]} \exp(\mathcal{D}^2(x_i, \mu_j)/\sigma)}, & \forall j \in [i] \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $[i]$ denotes the indices of s ($s \ll m$) nearest anchors of x_i , $\mathcal{D}^2(x_i, \mu_j) = \|x_i - \mu_j\|^2$, σ is the predefined bandwidth parameter. Thus each row in $Z^{(X)}$ contains only s nonzero entries which sum to 1. With the help of $Z^{(X)}$, the anchor graph provides a powerful approximation to the original similarity matrix

165 as

$$S^{(X)} = Z^{(X)} \Lambda^{-1} (Z^{(X)T}) \quad (3)$$

where $\Lambda = \text{diag}(Z^{(X)T} 1) \in \mathbb{R}^{m \times m}$. Existing research [35] shows that similarity matrix $S^{(X)}$ has some desirable properties, such as non-negative, sparse, low-rank properties.

As aforementioned, a reasonable criterion to achieve similarity preservation
 170 within view is to make similar samples in each view mapped closely in the Hamming space. To this end, we formulate the following optimization problem

$$\min_f \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} S_{ij}^{(X)} \|f(x_i) - f(x_j)\|^2 \quad (4)$$

Similarly to view X , we can obtain $S^{(Y)}$ by applying anchor graph to view Y , then we have

$$\min_g \sum_{i=1}^{n_y} \sum_{j=1}^{n_y} S_{ij}^{(Y)} \|g(y_i) - g(y_j)\|^2 \quad (5)$$

3.3. Cross-view correlation preservation

175 For hash learning from multi-view data, besides preserving within-view similarity, it is critical to analysis the relationships between views, thus in this section, we focus on cross-view correlation preservation. Some researches [13] have shown that if data described in different forms are related to similar topics, they are expected to share a latent space, where latent representations of

180 multi-view feature pairs from the same objects should be close to each other. Specifically in our problem setting, the binary codes of sample pairs, i.e., $\{f(x_i), g(y_i)\}_{i=1}^{n_0}$ should be similar after being transformed by hash functions in (1). Hence, we achieve this goal by solving the following objective function

$$\min_{f,g} \sum_{i=1}^{n_0} \|f(x_i) - g(y_i)\|^2 \quad (6)$$

Note that the above formulation only analyzes the correlations of paired samples, the relationships between rest samples (unpaired samples) are still unknown. Some researches reveal the cross-view correlations via given semantic information [20, 22, 26, 27], which however is difficult to obtain in real applications. Thus how to uncover the cross-view correlations of unpaired samples in the unsupervised setting still remains to solve.

190 Based on some advances of multi-view kernel construction [37], here we incorporate a simple yet effective approach to solve the above problem. **The basic idea is to use the within-view neighborhood relationships to reweight the between-view edges.** Specifically, the correlation between x_i and y_j can be calculated as

$$S_{ij}^{(XY)} = \sum_{k=1}^{n_0} S_{ik}^{(X)} \times S_{kj}^{(Y)} \quad (7)$$

195 Intuitively, $S_{ij}^{(XY)}$ will trend to one when x_i and y_j , meanwhile, are both close to the k -th sample pair, i.e., x_k and y_k , respectively. In other words, if x_i and y_j share more co-occurring paired neighbors, $S_{ij}^{(XY)}$ will be larger. We can rewrite (7) into a compact matrix form

$$S^{(XY)} = S_p^{(X)} \times S_p^{(Y)T} \quad (8)$$

200 where $S_p^{(X)} \in \mathbb{R}^{n_x \times n_0}$, and $S_p^{(Y)} \in \mathbb{R}^{n_y \times n_0}$ consist of the first n_0 columns in $S^{(X)}$, $S^{(Y)}$, respectively, denoting the similarities between pair and unpaired samples in each view, $S^{(XY)} \in \mathbb{R}^{n_x \times n_y}$ measures the cross-view correlation. Based on (8), we optimize the following problem

$$\min_{f,g} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} S_{ij}^{(XY)} \|f(x_i) - g(y_j)\|^2 \quad (9)$$

To this end, summing up (6) and (9), we arrive at our cross-view correlation preservation objective

$$\min_{f,g} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \tilde{S}_{ij}^{(XY)} \|f(x_i) - g(y_j)\|^2 \quad (10)$$

205 where $\tilde{S}^{(XY)}$ can be defined as follows

$$\tilde{S}_{ij}^{(XY)} = \begin{cases} S_{ij}^{(XY)} + 1, & i = j \text{ and } i, j \leq n_0 \\ S_{ij}^{(XY)}, & \text{otherwise} \end{cases} \quad (11)$$

3.4. Formulation

To generate discriminative binary codes, our method fully considers both within-view similarity and cross-view correlation. Overall, by combining (4), (5), and (10), we have our objective as

$$\begin{aligned} \min_{f,g} & \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} S_{ij}^{(X)} \|f(x_i) - f(x_j)\|^2 + \sum_{i=1}^{n_y} \sum_{j=1}^{n_y} S_{ij}^{(Y)} \|g(y_i) - g(y_j)\|^2 \\ & + \lambda \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \tilde{S}_{ij}^{(XY)} \|f(x_i) - g(y_j)\|^2 \end{aligned} \quad (12)$$

where λ is a trade-off parameter to balance the relative importances of within-view term and cross-view term. By some matrix transformations on (12), we obtain the following matrix form based optimization problem

$$\begin{aligned} \min_{B^{(X)}, B^{(Y)}} & \text{tr} \left(B^{(X)T} \left(\lambda \tilde{D}_{col}^{(XY)} + L^{(X)} \right) B^{(X)} - 2\lambda B^{(X)T} \tilde{S}^{(XY)} B^{(Y)} \right. \\ & \left. + B^{(Y)T} \left(\lambda \tilde{D}_{row}^{(XY)} + L^{(Y)} \right) B^{(Y)} \right) \\ \text{s.t.} & \quad B^{(X)} \in \{-1, 1\}^{n_x \times c} \\ & \quad B^{(Y)} \in \{-1, 1\}^{n_y \times c} \end{aligned} \quad (13)$$

where $L^{(X)} = I_{n_x} - S^{(X)}$, $L^{(Y)} = I_{n_y} - S^{(Y)}$ are called the graph Laplacian, $\tilde{D}_{row}^{(XY)} \in \mathbb{R}^{n_y \times n_y}$ ($\tilde{D}_{col}^{(XY)} \in \mathbb{R}^{n_x \times n_x}$) is a diagonal matrix, where each diagonal element equals to the sum of each row (column) in $\tilde{S}^{(XY)}$, and $B^{(X)}$, $B^{(Y)}$ are the binary codes in view X , Y , respectively. The above problem is NP hard

due to the discrete property of binary constraints, and the common scheme to address such problem is to apply spectral relaxation [3]. We put the definition of hash functions into (13), leading to

$$\begin{aligned} \min_W \quad & \text{tr}(W^T H W) \\ \text{s.t.} \quad & W^T W = I \end{aligned} \quad (14)$$

where $W = [W^{(X)T}, W^{(Y)T}]^T \in \mathbb{R}^{(d_x+d_y) \times c}$, and

$$\begin{aligned} H &= \begin{pmatrix} H^{(X)} & -H^{(XY)} \\ -H^{(XY)T} & H^{(Y)} \end{pmatrix} \in \mathbb{R}^{(d_x+d_y) \times (d_x+d_y)}, \\ H^{(X)} &= X \left(\lambda \tilde{D}_{col}^{(XY)} + L^{(X)} \right) X^T \in \mathbb{R}^{d_x \times d_x}, \\ H^{(XY)} &= \lambda X \tilde{S}^{(XY)} Y^T \in \mathbb{R}^{d_x \times d_y}, \\ H^{(Y)} &= Y \left(\lambda \tilde{D}_{row}^{(XY)} + L^{(Y)} \right) Y^T \in \mathbb{R}^{d_y \times d_y}. \end{aligned}$$

The orthogonal constraint for W involved is commonly used for normalization. The above is a typical eigenvalue problem which can be efficiently solved, and the optimal W are selected as the eigenvectors corresponding to the smallest c eigenvalues of H . Note that most existing cross-view hashing methods [20, 21, 23, 24, 26] share similar solving approaches, however incur large quantization loss due to such direct relaxation. Here instead of directly dropping the discrete constraints, we further add two regression terms into (13) to *explicitly* minimize quantization loss, then formulate our final objective

$$\begin{aligned} \min_{B^{(X)}, B^{(Y)}, W^{(X)}, W^{(Y)}} \quad & \text{tr} \left(W^{(X)T} H^{(X)} W^{(X)} - 2 W^{(X)T} H^{(XY)} W^{(Y)T} \right. \\ & \left. + W^{(Y)T} H^{(Y)} W^{(Y)} \right) + \gamma \left(\|B^{(X)} - X^T W^{(X)}\|_F^2 \right. \\ & \left. + \|B^{(Y)} - Y^T W^{(Y)}\|_F^2 \right) \\ \text{s.t.} \quad & B^{(X)} \in \{-1, 1\}^{n_x \times c} \\ & B^{(Y)} \in \{-1, 1\}^{n_y \times c} \end{aligned} \quad (15)$$

where γ is a trade-off parameter balancing the relative importances of similarity preservation terms and quantization loss term. We will detail the joint

optimization for the above problem in the next section.

210 3.5. Solution

As we can see, there is no closed form solution in (15), and this joint optimization function is not convex over all variables simultaneously. Therefore, we present an iterative optimization algorithm as follows:

Update $W^{(X)}$ with fixed $B^{(X)}$, $W^{(Y)}$. We drop some irrelevant terms of $W^{(X)}$, and rewrite (15) as the following form

$$\begin{aligned} \min_{W^{(X)}} \quad & \gamma \|B^{(X)} - X^T W^{(X)}\|_F^2 \\ & + \text{tr} \left(W^{(X)T} H^{(X)} W^{(X)} \right) - 2 \text{tr} \left(W^{(X)T} H^{(XY)} W^{(Y)T} \right) \end{aligned} \quad (16)$$

The above is a typical regularized least square problem. By setting the derivative of $W^{(X)}$ in (16) to 0, we obtain

$$\begin{aligned} -\gamma X B^{(X)} + \gamma X X^T W^{(X)} + H^{(X)} W^{(X)} - H^{(XY)} W^{(Y)} &= 0 \\ \Rightarrow W^{(X)} &= \left(\gamma X X^T + H^{(X)} \right)^{-1} \left(\gamma X B^{(X)} + H^{(XY)} W^{(Y)} \right) \end{aligned} \quad (17)$$

Update $B^{(X)}$ with fixed $W^{(X)}$. By dropping some irrelevant terms of $B^{(X)}$ in (15), we have

$$\begin{aligned} \min_{B^{(X)}} \quad & \|B^{(X)} - X^T W^{(X)}\|_F^2 \\ \text{s.t.} \quad & B^{(X)} \in \{-1, 1\}^{n_x \times c} \end{aligned} \quad (18)$$

(18) can be further reduced into

$$\begin{aligned} \min_{B^{(X)}} \quad & -2 \text{tr} \left(B^{(X)T} X^T W^{(X)} \right) + \text{const} \\ \text{s.t.} \quad & B^{(X)} \in \{-1, 1\}^{n_x \times c} \end{aligned} \quad (19)$$

Clearly, we have the closed solution form of $B^{(X)}$

$$B^{(X)} = \text{sign}(X^T W^{(X)}) \quad (20)$$

215 **Update $W^{(Y)}$ with fixed $B^{(Y)}$, $W^{(X)}$.** Similarly to $W^{(X)}$, we update $W^{(Y)}$ as follows

$$W^{(Y)} = \left(\gamma Y Y^T + H^{(Y)} \right)^{-1} \left(\gamma Y B^{(Y)} + H^{(XY)T} W^{(X)} \right) \quad (21)$$

Update $B^{(Y)}$ with fixed $W^{(Y)}$. Similarly to $B^{(X)}$, we update $B^{(Y)}$ as follows

$$B^{(Y)} = \text{sign}(Y^T W^{(Y)}) \quad (22)$$

Note that since the above optimization is an iterative procedure, we must provide an initialization approach. Here we first solve (14) to derive the projections and binary codes as the initializations of (15). In later experimental section, we can see proposed method quickly converges with such initialization scheme. Finally, we summarize the algorithm of proposed method in Algorithm 1.

Algorithm 1 Semi-paired Hashing

Input: $X \in \mathbb{R}^{d_x \times n_x}$, $Y \in \mathbb{R}^{d_y \times n_y}$, code length c , parameters λ , γ , s .

Output: projection matrices $W^{(X)} \in \mathbb{R}^{d_x \times c}$, $W^{(Y)} \in \mathbb{R}^{d_y \times c}$, binary codes $B^{(X)} \in \mathbb{R}^{n_x \times c}$, $B^{(Y)} \in \mathbb{R}^{n_y \times c}$.

- 1: Initialize $W^{(X)}$, $W^{(Y)}$ as the eigenvectors corresponding to the smallest c eigenvalues in (13);
  - 2: Initialize $B^{(X)} = \text{sign}(X^T W^{(X)})$, $B^{(Y)} = \text{sign}(Y^T W^{(Y)})$;
  - 3: **repeat**
  - 4:   Update $W^{(X)}$ by (17);
  - 5:   Update $W^{(Y)}$ by (21);
  - 6:   Update $B^{(X)}$ by (20);
  - 7:   Update $B^{(Y)}$ by (22);
  - 8: **until** *convergence*
-

4. Experiments

In this section, we evaluate SPH in cross-view retrieval tasks by comparing with several state-of-the-art methods.

4.1. Datasets

Two widely used benchmark datasets are adopted for evaluation:

Table 1: mAP results on Wiki dataset.

Task	Method	Code Length			
		$c = 16$	$c = 32$	$c = 64$	$c = 128$
Image Query v.s. Text Database	CVH	0.1351	0.1347	0.1343	0.1323
	CMSSH	0.1396	0.1651	0.1438	0.1596
	LCMH	0.1212	0.1208	0.1243	0.1270
	IMH	0.1973	0.1648	0.2027	0.2178
	SPH (50%)	0.2149	0.2212	0.2316	0.2331
	SPH (100%)	0.2211	0.2331	0.2425	0.2481
Text Query v.s. Image Database	CVH	0.1091	0.1068	0.1059	0.1072
	CMSSH	0.1176	0.1282	0.1202	0.1181
	LCMH	0.1170	0.1186	0.1242	0.1274
	IMH	0.1589	0.1679	0.1737	0.1728
	SPH (50%)	0.1764	0.1827	0.1904	0.1922
	SPH (100%)	0.2004	0.2135	0.2245	0.2337

- 230 • **Wiki:** The Wiki dataset [13] was collected from Wikipedia consisting of 2,866 multimedia documents. Each image is represented by a 128-dimension SIFT histogram and each text is represented by a 10-dimension topics vector generated by Latent Dirichlet Allocation (LDA) model [38]. Totally 10 categories are considered in his dataset and each document

235 (image-text pair) is labeled by one of them. We use 75% of the pairs as the training set, the remaining 25% as the query set.
- 240 • **Pascal VOC:** The Pascal VOC dataset [39] consists of 9963 image tag pairs, which can be categorized into 20 different classes. Here images with only one object are selected in the experiment, which results in 2808 training and 2841 testing samples [40]. The image features are 512-dimensional Gist features, and the text features are 399-dimensional word frequency features.

Table 2: mAP results on Pascal VOC dataset.

Task	Method	Code Length			
		$c = 16$	$c = 32$	$c = 64$	$c = 128$
Image Query v.s. Text Database	CVH	0.1484	0.1187	0.1651	0.1411
	CMSSH	0.1075	0.1044	0.9900	0.9950
	LCMH	0.1454	0.1097	0.1083	0.1199
	IMH	0.2165	0.2207	0.2051	0.1950
	SPH(50%)	0.2064	0.2116	0.2209	0.2317
	SPH(100%)	0.2177	0.2365	0.2566	0.2691
Text Query v.s. Image Database	CVH	0.9310	0.9450	0.9780	0.9180
	CMSSH	0.9080	0.8990	0.8680	0.8520
	LCMH	0.1162	0.1076	0.1152	0.1335
	IMH	0.1725	0.1706	0.1641	0.1618
	SPH (50%)	0.1851	0.1978	0.2069	0.2194
	SPH (100%)	0.2044	0.2319	0.2628	0.2935

4.2. Experimental setup

In our experiment, we evaluate SPH on two cross-view retrieval tasks: use
an image query in the visual view to search the relevant texts from the text view
(shorted as “*Image Query vs. Text Database*”); use an text query in the textual
view to search the relevant images from the visual view (shorted as “*Text Query*
vs. Image Database”).

We compare SPH with several state-of-the-art related hashing methods, in-
cluding CVH [21], CMSSH [20], LCMH [24], IMH [23]. Note that SPH is unsu-
pervised, thus for a fair comparison, we do not incorporate semantic informa-
tion for all comparing methods. For SPH, we empirically set $s = 5$, λ and γ are
tuned from the range of $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, and best results are
reported.

For evaluation, the *mean Average Precision* (mAP) is used to evaluate the
retrieval performance. For a query q , the average precision (AP) is defined as

$$AP(q) = \frac{1}{L_q} \sum_{r=1}^R P_q(r) \delta_q(r) \quad (23)$$

where L_q is the number of the ground truth neighbors in the retrieved list, $P_q(r)$ is the precision of the top r retrieved results and $\delta_q(r) = 1$ if the r -th result is the true neighbor and 0 otherwise. mAP is the mean of all the queries' APs in the database. In our experiments, we set R as the number of samples in the database.

4.3. Performance comparison

In this section, we evaluate cross retrieval performances of different methods. As SPH can deal with the semi-pair scenario, we evaluate SPH under two different pair settings, i.e., 100% (fully paired), 50% pairwise information, while full pairwise information is provided for other methods.

The mAP results of all methods with different code lengths on two datasets, i.e., Wiki dataset, Pascal VOC dataset are shown in Table 1, 2, respectively. We can see that SPH using 100% pairwise information achieves highest results among all 16 cases (8 cases per dataset). To our surprise, SPH using 50% pairwise information also generally outperforms other methods in most cases. IMH achieves good performances, taking the third place. For the remaining three methods, each obtains relatively similar results, and can outperform each other on some specific cases.

Besides, we also report the Precision-Recall curves of all methods in two different tasks on Wiki dataset, and Pascal VOC dataset, as shown in Fig. 2, 3, 4, 5. From these figures, we can clearly see that SPH using 100% pairwise information achieves the best retrieval performances on nearly all recall levels.

Generally speaking, the above results clearly reveal that proposed SPH can achieve good cross-view retrieval performances even in the limited pairwise information setting.

4.4. Evaluation on semi-paired data

One advantage of SPH is to deal with semi-pair multi-view data, thus in this section, we further evaluate the retrieval performance of SPH on multi-view data with different partial pairwise information. IMH is selected as a

Figure 2: Precision-Recall Curves with different code lengths of Wiki dataset in “Image Query vs. Text Database” task.

Figure 3: Precision-Recall Curves with different code lengths of Wiki dataset in “Text Query vs. Image Database” task.

Figure 4: Precision-Recall Curves of with different code lengths of Pascal VOC dataset in “Image Query vs. Text Database” task.

Figure 5: Precision-Recall Curves of with different code lengths of Pascal VOC dataset in “Text Query vs. Image Database” task.

Figure 6: MAP results with different pairs with fixed code length as 32 on two datasets.

comparison method, since it can also deal with such semi-paired scenarios. We vary the percentage of paired data from 10% to 90%, and report the MAP results of these two methods with fixed code length as 32 in two cross-view retrieval tasks on Fig. 6. From the above figure, we can see that both methods generally improve the retrieval results as the percentage of pairwise information increases. SPH consistently achieves higher performances than IMH on all cases, and more interestingly, SPH using only 10% pairwise information can achieve similar retrieval performances with IMH using 90% pair information. These above results demonstrate that SPH can handle the semi-paired cross-retrieval tasks very well.

4.5. Parameter analysis

In this section, we analysis the effects of two parameters, i.e., λ , γ involved in SPH. We apply SPH on two datasets with fixed code length as 32, and vary the above two parameters from the range of $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. MAP results under all parameter combinations in two cross-view retrieval tasks are reported in Fig. 7. From Fig. 7, we can find the following observations: (1) MAP generally improves as λ increases, which indicates that cross-view correlation is very important to the cross-view retrieval performances. (2) MAP is relatively

(a) Image Query vs. Text Database

(b) Text Query vs. Image Database

(c) Image Query vs. Text Database

(d) Text Query vs. Image Database

Figure 7: MAP results with different combinations of parameters, i.e., λ and γ with fixed code length as 32 on Wiki dataset (a-b) and Pascal VOC dataset (c-d).

Figure 8: Convergence analysis with fixed code length as 32 on two datasets.

low when λ is small and γ is large; it remains high and stable under other
 305 combinations. (3) We can almost obtain the best MAP at around (1, 1) for
 (λ, γ), where similarity preservation and quantization loss make nearly similar
 contributions in SPH.

4.6. Convergence analysis

In this section, we evaluate the convergence of SPH, as it is an iterative
 310 algorithm. We apply SPH on two datasets with fixed code length as 32, and
 compute the objective function value at each iteration. The curves of objective
 function value versus the number of iterations are plotted in Fig. 8, where we
 can clearly see that SPH converges quickly.

5. Conclusion

315 In this paper, we proposed a novel cross-view hashing method, named Semi-
 paired Hashing (SPH). Compared with existing methods, the advantage of SPH
 is to deal with a challenging cross-view retrieval task, where only partial pairwise
 correspondences are provided in advance. Besides considering within-view simi-
 larity structure, SPH could fully utilize partially given pairwise and similarity
 320 information to exploit the cross-view correlation in the semi-paired scenarios.

Furthermore, for the ease of the quantization loss, two regression terms between original features and binary codes were designed. An iterative algorithm without spectral relaxation was presented to jointly learn hash function and binary codes. Experiments on two benchmark datasets validated the effectiveness of proposed method on cross-view retrieval tasks.

6. Acknowledgements

This work is supported by the National Science Foundation of China (Grant No. 61273251, 61402203).

References

- [1] J. Wang, H. T. Shen, J. Song, J. Ji, Hashing for similarity search: A survey, arXiv preprint arXiv:1408.2927 (2014).
- [2] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of International Conference on Very Large Data Bases, 1999, pp. 518–529.
- [3] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proceedings of Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.
- [4] W. Liu, J. Wang, S. Kumar, S.-F. Chang, Hashing with graphs, in: Proceedings of International Conference on Machine Learning, 2011, pp. 1–8.
- [5] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for large-scale search, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 2393–2406.
- [6] F. Shen, C. Shen, Q. Shi, A. Van Den Hengel, Z. Tang, Inductive hashing on manifolds, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1562–1569.

- 345 [7] W. Liu, C. Mu, S. Kumar, S.-F. Chang, Discrete graph hashing, in: Proceedings of Advances in Neural Information Processing Systems, 2014, pp. 3419–3427.
- [8] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, H. T. Shen, Hashing on nonlinear manifolds, IEEE Transactions on Image Processing 24 (2015) 1839–1851.
- 350 [9] F. Shen, C. Shen, W. Liu, H. Tao Shen, Supervised discrete hashing, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 37–45.
- [10] F. Shen, W. Liu, S. Zhang, Y. Yang, H. Tao Shen, Learning binary codes for maximum inner product search, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4148–4156.
- 355 [11] F. Cakir, S. Sclaroff, Adaptive hashing for fast similarity search, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1044–1052.
- [12] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 2916–2929.
- 360 [13] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: Proceedings of ACM International Conference on Multimedia, 2010, pp. 251–260.
- 365 [14] J. Song, Y. Yang, Z. Huang, H. T. Shen, R. Hong, Multiple feature hashing for real-time large scale near-duplicate video retrieval, in: Proceedings of ACM International Conference on Multimedia, 2011, pp. 423–432.
- 370

- [15] N. Chen, J. Zhu, F. Sun, E. P. Xing, Large-margin predictive latent subspace learning for multiview data analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 2365–2378.
- [16] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, *CoRR abs/1304.5634* (2013).
375
- [17] D. Zhang, F. Wang, L. Si, Composite hashing with multiple information sources, in: *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 225–234.
- [18] X. Liu, J. He, D. Liu, B. Lang, Compact kernel hashing with multiple features, in: *Proceedings of ACM International Conference on Multimedia*,
380 2012, pp. 881–884.
- [19] S. Kim, Y. Kang, S. Choi, Sequential spectral learning to hash with multiple representations, in: *Proceedings of European Conference on Computer Vision*, 2012, pp. 538–551.
- [20] M. M. Bronstein, A. M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3594–3601.
385
- [21] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: *Proceedings of International Joint Conference on Artificial Intelligence*, volume 22, 2011, p. 1360.
390
- [22] Y. Zhen, D.-Y. Yeung, Co-regularized hashing for multimodal data, in: *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1376–1384.
- [23] J. Song, Y. Yang, Y. Yang, Z. Huang, H. T. Shen, Inter-media hashing for large-scale retrieval from heterogeneous data sources, in: *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2013, pp. 785–796.
395

- [24] X. Zhu, Z. Huang, H. T. Shen, X. Zhao, Linear cross-modal hashing for efficient multimedia search, in: Proceedings of ACM International Conference on Multimedia, 2013, pp. 143–152.
- [25] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2083–2090.
- [26] D. Zhang, W.-J. Li, Large-scale supervised multimodal hashing with semantic correlation maximization, in: Proceedings of AAAI Conference on Artificial Intelligence, 2014, pp. 2177–2183.
- [27] Y. Hu, Z. Jin, H. Ren, D. Cai, X. He, Iterative multi-view hashing for cross media indexing, in: Proceedings of ACM International Conference on Multimedia, 2014, pp. 527–536.
- [28] J. Masci, M. M. Bronstein, A. Bronstein, J. Schmidhuber, Multimodal similarity-preserving hashing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (2014) 824–830.
- [29] G. Irie, Z. Li, X.-M. Wu, S.-F. Chang, Locally linear hashing for extracting non-linear manifolds, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2123–2130.
- [30] K. Eshghi, S. Rajaram, Locality sensitive hash functions based on concomitant rank order statistics, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 221–229.
- [31] J. Ji, J. Li, S. Yan, B. Zhang, Q. Tian, Super-bit locality-sensitive hashing, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 108–116.
- [32] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 1092–1104.

- 425 [33] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering., in: *Proceedings of Advances in Neural Information Processing Systems*, volume 14, 2001, pp. 585–591.
- [34] X. Liu, J. He, B. Lang, Multiple feature kernel hashing for large-scale visual search, *Pattern Recognition* 47 (2014) 748–757.
- 430 [35] W. Liu, J. He, S.-F. Chang, Large graph construction for scalable semi-supervised learning, in: *Proceedings of International Conference on Machine Learning*, 2010, pp. 679–686.
- [36] X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, in: *Proceedings of AAAI Conference on Artificial Intelligence*, 435 2011.
- [37] V. R. De Sa, P. W. Gallagher, J. M. Lewis, V. L. Malave, Multi-view kernel construction, *Machine learning* 79 (2010) 47–71.
- [38] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, *The Journal of Machine Learning Research* 3 (2003) 993–1022.
- 440 [39] S. J. Hwang, K. Grauman, Reading between the lines: Object localization using implicit cues from image tags, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 1145–1158.
- [40] A. Sharma, A. Kumar, H. Daume III, D. W. Jacobs, Generalized multiview analysis: A discriminative latent space, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2160–2167. 445