Set-to-Set Hashing with Applications in Visual Recognition

I-Hong Jhuo$^1$ and Jun Wang$^2$
Fujitsu Laboratories of America$^1$ and Alibaba Group$^2$
ihjhuo@gmail.com; wongjun@gmail.com

Abstract

Visual data, such as an image or a sequence of video frames, is often naturally represented as a point set. In this paper, we consider the fundamental problem of finding a nearest set from a collection of sets, to a query set. This problem has obvious applications in large-scale visual retrieval and recognition, and also in applied fields beyond computer vision. One challenge stands out in solving the problem—set representation and measure of similarity. Particularly, the query set and the sets in dataset collection can have varying cardinalities. The training collection is large enough such that linear scan is impractical. We propose a simple representation scheme that encodes both statistical and structural information of the sets. The derived representations are integrated in a kernel framework for flexible similarity measurement. For the query set process, we adopt a learning-to-hash pipeline that turns the kernel representations into hash bits based on simple learners, using multiple kernel learning. Experiments on two visual retrieval datasets show unambiguously that our set-to-set hashing framework outperforms prior methods that do not take the set-to-set search setting.

1 Introduction

Searching for similar data samples is a fundamental step in many large-scale applications. As the data size explodes, hashing techniques have emerged as a unique option for approximate nearest neighbor (ANN) search, as it can dramatically reduce both the computational time and the storage space. Successes are seen in areas including computer graphics, computer vision, and multimedia retrieval [17, 43, 45]. Hashing methods perform space partitioning to encode the original high-dimensional data points into binary codes. With the resulting binary hash codes, one can perform extremely rapid ANN search that entails only sublinear search complexity.

Conventional hashing schemes concern point-to-point (P2P) search setting. They either depend on randomization and are data oblivious (represented by the classic Locality Sensitive Hashing – LSH), or are based on advanced machine learning techniques to learn hashing functions that are better tailored to the specific data and/or label distribution. The latter includes unsupervised [10, 51], semi-supervised [43, 44], and supervised hashing [16, 22, 28, 54, 18, 37, 52].

A natural generalization of the point-to-point search is set-to-set (S2S) search. For example, one can pose the facial image recognition problem as one that queries for a nearest subspaces to a given point [50]. Indeed, there are several recent attempts, studying point-to-hyperplane search that is useful for active learning [23], or subspace-to-subspace search [3] that models set-to-set search assuming linear structures in the sets.

In this paper, we consider the set-to-set search problem in its full generality. This general setting finds applications ranging from video-based surveillance to 3D face retrieval from collections of 2D images [2, 40, 38]. Compared to specialized settings discussed above that come with natural notion of distance, a central challenge here is how to measure the distance/similarity between sets.
We propose a similarity measure that captures both the statistical and structural aspects of the sets (Section 3). To learn the hash bits, we adopt dyadic hypercut as a weak learner [27] to derive a boosted algorithm that integrates both the structural and statistical similarities. The whole framework is illustrated in Fig. 1(a). In this paper, we focus on image applications, and hence coin the name Image Set Hashing (ISH). However, the core components of the proposed framework can be extended to generic scenarios.

2 Related Work

Here we discuss representative works in randomization-based hashing and learning-based hashing for P2P setting, and recent work on certain restricted S2S setting. Review of recent development of hashing techniques can be found in [45, 46].

LSH hashing and variants are iconic randomization-based hashing schemes. They are simple in theory and efficient in practice, and flexible enough to handle various distance measures [6, 7, 16]. However, the data oblivious design of the LSH family methods often causes suboptimal recall-precision tradeoff curve. Hence, learning-based hashing schemes have been under active development recently. Generally, both the data and label information is fed into carefully designed learning pipeline to produce more adaptive and efficient hash codes. Depending on whether the label information is in use, these schemes are either unsupervised, e.g., spectral hashing [51], graph hashing, and ITQ (iterative quantization) [10], or (semi-)supervised hashing, including [20, 22, 29, 28, 44]. Particularly, recent efforts have built more powerful hashing schemes on top of deep learning [34, 35, 21, 26]. All these methods only deal with the P2P setting.

Study of the S2S setting started only very recently, and is mostly about image applications. Statistical or geometric assumptions are often made on the sets to facilitate representation. For example, statistical distribution of data points in each set can be assumed, and KL-divergence can be used to measure set similarity [1]. By comparison, point sets can also lie on linear subspaces or more general geometric objects [12, 5, 15, 25, 48]. Among the representation schemes, representation based on covariance matrices has led to superior performances on image sets (video frames) [48, 24, 40]. For instance, in [19], covariance matrix is used in such way, and similarity is then measured via kernel mapping and learning. Promising result has been reported, but the framework is restricted to cases when the query is a single point. For image applications specifically, the Set Compression Tree [32] compresses a set of image descriptors jointly (rather than individual descriptors) and achieve a very small memory footprint (as low as 5 bits). However, all existing representation methods do not account the holistic structural information; this may lead to incorrect similarity measurements on highly nonlinear data distributions. Our representation and hashing scheme is a first attempt to directly address the above problems in the general S2S setting.
3 Structural and Statistical Modeling

In this section, we detail how the structural and statistical information is extracted from the point sets, and how similarity between sets is measured.

3.1 Structure via graph modeling

The idea here is to use graph for discovering structures within data, and then measure the similarity via appropriate kernel mapping on graphs [39, 9, 47], Fig. 1(b) gives an example. To model data points within a set, we derive an affinity matrix $A$ based on quantized pairwise distances [47]: if the distance is larger than a predefined threshold $\mu$, the corresponding affinity value in $A$ is set to 0, and 1 otherwise. We use $x_i$’s to denote the point sets, and $A^i$’s to denote the corresponding affinity matrices thus constructed. With all the $A^i$'s at hand, the point set similarity is defined as:

$$K_g(x_i, x_j) = \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} A_{ip} A_{jq} g(x_{ip}, x_{jq}),$$  

where, $A_{ip} = 1/\sum_{k=1}^{n_i} a_{kp}$, $A_{jq} = 1/\sum_{k=1}^{n_j} a_{jq}$, and $g(x_{ip}, x_{jq}) = \exp(-\gamma g \| x_{ip} - x_{jq} \|^2)$. $\gamma_g$ is a constant and $n_i$ and $n_j$ are the number of data points in $x_i$ and $x_j$, respectively.

To understand the captured structural information, each clique (formed by several 1 elements) in $A^i$ can be regarded as one concept. If $A^i$ is an all-one matrix, all data points in one set belong to one concept and each data point set is considered as one data point. When $A^i$ is an identical matrix, each data point is independent, and no relation can be discovered. When $A^i$ is a clique-based matrix, data points can be clustered into cliques and $K_g$ is a clique-based graph kernel. In this way, we leverage the structural information for our set hashing.

3.2 Statistical information

Covariance matrices have provided effective local region representation for visual recognition and human identification [25, 40]. Intuitively, they describe the local image statistics. In this work, we use covariance matrices to depict the statistical variance of images within each set. Given $N$ image sets, $\mathcal{X} = \{(x_1, l_1), \cdots, (x_i, l_i), \cdots, (x_N, l_N)\}$. $x_i = \{x_{i1}, \cdots, x_{in_i}\}$ is an image set, where $x_{ij} \in \mathbb{R}^d$ represents the $j$th $d$-dimensional feature vector in $x_i$, and the set consists of $n_i$ images. $l_i$’s are the labels of each image set. Each image set is represented with a $d \times d$ covariance matrix:

$$C_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^\top,$$

where $\bar{x}_i$ is the mean feature vector within the set. The diagonal elements of $C_i$ represent the variance of each individual image feature, and the off-diagonal elements are their respective covariance. In this way, the covariance modeling can provide a desirable statistic for semantic variance among all images for an image set. Moreover, we use Gaussian-logrithm kernel [14] to map each covariance matrix of an image set into high dimension space, as follows:

$$K_s(x_i, x_j) = \phi(C_i)^\top \phi(C_j)$$

$$= \exp(-\|\log(C_i) - \log(C_j)\|_F^2 / 2\gamma_s^2),$$

where $\gamma_s$ is a positive constant, which be set to the mean distances of training points. Kernel matrix $K_s$ is for the image sets in Riemannian space and $\| \cdot \|_F$ denotes the matrix Frobenius norm. In the end, each pair of image sets $x_i$ and $x_j$ are mapped by the $K_g$ and $K_s$ kernel functions into a high dimensional space.

4 Image Set Hashing

4.1 Learning framework

Suppose we have an image set dataset $\mathcal{X} = \{x_i, l_i\}$. The goal of hashing is to generate an array of appropriate hash functions $h : \mathbb{R}^d \rightarrow \{0, 1\}$ by a designed function $\Psi$ and each bit is constructed by $h(x) = \text{sign}(\Psi(x))$. However, there is no straightforward way to generate hash codes for each image set. Inspired by [42] [19] [48], we develop a mapping framework to construct hash codes for image sets in a common Hamming space based on multiple kernels. Kernel methods [11] [14] [42] [48] are known to capture and unfold rich information in data distribution. After the mapping process, we can generate hash codes in a Hamming space by simultaneously considering the structural and statistical information and iteratively maximize the discriminant margins based on multiple kernels.
4.2 Weak learners with boosting algorithm for hash functions

Since a multi-class classification problem can always be treated as an array of two-class problems by adopting one-against-one or one-against-all strategies, we design a boosting algorithm to learn binary splits for constructing hash functions. Specifically, we consider dyadic hypercut \([27]\) with multiple kernel functions. A dyadic hypercut \(f\) is generated by a kernel with a pair of different labels in training samples. Specifically, \(f\) is parameterized by positive sample \(x_a\), negative sample \(x_b\), and kernel functions \(\{K_m\}\), i.e., \(m\) indicates statistical kernel \((K_s)\) or structural kernel \((K_g)\), and can be represented as follows:

\[
f(x) = \text{sign}(K_m(x_a, x) - K_m(x_b, x) + \varepsilon),
\]

where \(\varepsilon \in \mathbb{R}\) is a threshold. The size of the totally generated weak learner pool is \(|f| = M \times n_a \times n_b\), where \(M\), \(n_a\) and \(n_b\) are the numbers of kernels, positive training samples and negative training samples, respectively. With an efficient boosting process, we iteratively select a subset of weak learners by considering the learning loss.

Note that the learning process may be susceptible to overfitting when \(|f|\) is large. To alleviate this issue, we adopt a boosting algorithm \([8, 27]\) to combine a number of weak splits (weak learners) into a strong one. Specifically, we iteratively select the discriminant weak learners generated from multiple kernels via maintaining a weighted distribution \(w^t\) over data. Each iteration \(t\) produces a weak hypothesis \(f^t(x) : x \rightarrow \{+1, -1\}\) and a weighted error \(\delta^t\). The learning algorithm is aimed at selecting weak learner \(f^t\) for minimizing \(\delta^t\) followed by updating next distribution \(w^{t+1}\). We adopt exponential loss \([8]\) and minimize the loss function to select the best weak learner \(f^t\) at iteration \(t\) and the best weak learner is computed as:

\[
f^t = \min_f \sum_{i=1}^{N} w_i^t \exp(-l_i f(x_i)),
\]

where \(w_i^t\) indicates the weight of \(x_i\) at iteration \(t\). Once obtaining the best weak learner, we update the data distribution based on weighted errors. The linear combination of weak learners, i.e., a strong split, is computed as follows:

\[
F(x) = \text{sign}(\sum_{t=1}^{T} \lambda_t^t f^t(x)),
\]

where \(\lambda_t^t = \frac{1}{2} \log \frac{1 - \delta^t}{\delta^t}\). At each iteration \(t\), \(F = \sum_{t=1}^{t-1} \lambda^t f^t\) is a linear combination of the \((t-1)\) weak learners.

4.3 Objective Function

With the designed hash functions, the following are desired properties of the hash codes: (1) Each hash value is independent of the binary representation for each sample. (2) When samples are close to each other in feature space (e.g. with similar distributions), the hash codes should induce similar hash values with a small Hamming distance. (3) In the resulting Hamming space, different content should have different hash codes, which push different samples of categories as far as possible, meanwhile gather the samples of the same category close to each other. Based on the criteria, we derive our multiple kernel hashing as follows:

\[
\min_{H_q, H_r} \alpha D_s + \beta D_e + \nu_1 \sum_{r' \in \{1 : R\}} \sum_{i \in \{1 : N\}} \Omega
\]

s.t. \(H_q^{r'} = \text{sign}(\sum_{t=1}^{T} \lambda_t f_q^{r'}(x_i)), \forall i \in \{1 : N\}, \forall r' \in \{1 : R\}\)

\(H_r^{r'} = \text{sign}(\sum_{t=1}^{T} \lambda_t f_r^{r'}(x_i)), \forall i \in \{1 : N\}, \forall r' \in \{1 : R\}\),

where \(H_q^{r'}\) is the hash value of the \(i\)th image set using the \(r'\)th strong split, i.e., hash function, and \(r\) and \(q\) represent the retrieval and query sets in the training process. \(R\) is the number of the splits and \(\Omega = \sum_{t=1}^{T} (\lambda_t^q f_q^{r'}(x_i) + \nu_2 \lambda_t^r f_r^{r'}(x_i))\). \(f_q^{r'}\) is the \(r'\)th strong split generated from the number of \(T\) weighted weak learners (in eq. \([6]\)), \(\lambda_t^q\) is the coefficient of weak learners trained via a boosting algorithm, \(\nu_1\) and \(\nu_2\) are constant parameters.
Algorithm 1 Image Set Hashing

Input: a set of training images, \( X = \{x_i, l_i\} \) is divided into \( q \) and \( r \) two training image sets, where \( x_i = \{x_{ij}\}_{j=1}^{n_i} \in \mathbb{R}^d \), \( i \in \{1, 2, \ldots, N\}, l_i = \{1, 2, \ldots, L\} \).

Initialize: Compute kernel matrices for \( q \) training image sets, i.e., \((K_{m})_{q}\), by using the kernel functions \((K_{g} \text{ and } K_{d})\) according to eq. \((1)\) and eq. \((4)\). Similar to \( r \) training image sets, \((K_{m})_{r}\).

1: \( V_{q} \in \mathbb{R}^{N \times K}, V_{r} \in \mathbb{R}^{N \times K} \leftarrow \) kernel PCA with statistical kernels for \( K_{g} \) and \( K_{d} \), respectively
2: \( H_{q} \leftarrow \text{sign}(V_{q}^{T}(K_{g}))\)
3: \( H_{r} \leftarrow \text{sign}(V_{r}^{T}(K_{d}))\)

Optimization:
4: while not converged do
5: \( \text{Optimize} \quad H_{q}, H_{r} \) with eq. \((5)\)
6: Train \( R \) splits by weak learner selection in eq. \((6)\) on \( q \) kernels \((K_{m})_{q}\) by using \( H_{r} \) as training labels, and inversely train another \( R \) splits on \( r \) kernels \((K_{m})_{r}\) by using \( H_{q} \) as training labels
7: \( H_{q} \leftarrow \text{sign}(\sum_{r'=1}^{R} F_{q}^{r'}(\mathbf{x})) \leftarrow (K_{m})_{q}\)
8: \( H_{r} \leftarrow \text{sign}(\sum_{r'=1}^{R} F_{r}^{r'}(\mathbf{x})) \leftarrow (K_{m})_{r}\)
9: \( \text{Optimize} \quad H = [H_{q}, H_{r}] \in \{0, 1\}^{R \times 2N} \) with eq. \((9)\)
10: Train \( R \) splits by weak learner selection in eq. \((6)\) on \( q \) kernels \((K_{m})_{q}\) by using \( H_{r} \) as training labels, and inversely train another \( R \) splits on \( r \) kernels \((K_{m})_{r}\) by using \( H_{q} \) as training labels
11: Check the convergence condition
12: end while
13: Output: \( \sum_{r'=1}^{R} F_{q}^{r'}(\mathbf{x}) \) and \( \sum_{r'=1}^{R} F_{r}^{r'}(\mathbf{x}) \) for query and database encoding in the testing process, respectively.

The minimization of the first two terms, i.e., \( D_{s} \) and \( D_{c} \), tend to find an optimal discriminative distance between intra- and inter-categories, which capture the discriminative property among all the samples and are defined in eq. \((8)\) and eq. \((9)\), and \( d(\cdot, \cdot) \) indicates distance measure in Hamming space. In addition, \( D_{c} \) can refine the hash codes generated from the training image sets (\( q \) and \( r \) parts) by maximizing the separability of category algorithm in \([31]\). Simultaneous consideration of the two distance functions helps to minimize the within-category distances and meanwhile maximize the between-category dis. By formulating the structural and statistical information with multiple kernels for our objective function, the image set hashing becomes more robust and discriminative.

\[
D_{s} = \sum_{(m,n) \in \mathcal{M}} d(H_{q}^{m}, H_{q}^{n}) - \nu_{3} \sum_{(m,n) \in \mathcal{C}} d(H_{q}^{m}, H_{q}^{n}) \tag{8}
\]

\[
D_{c} = \sum_{(m,n) \in \mathcal{M}} d(H_{q}^{m}, H_{r}^{n}) - \nu_{4} \sum_{(m,n) \in \mathcal{C}} d(H_{q}^{m}, H_{r}^{n}) \tag{9}
\]

where \( \mathcal{M} \) and \( \mathcal{C} \) are represented as intra- and inter-category, \( H_{s} \) can be \( H_{q} \) or \( H_{r} \), and \( \nu_{3} \) and \( \nu_{4} \) are the pre-computable constant parameters to balance the intra-category and inter-category scales.

4.4 Optimization

The objective function optimization problem \([7]\) is a typical nonsmooth, nonconvex multiple variable minimization problem. We derive an iterative block coordinate descent algorithm \([4]\) for the optimization. Algorithm \([1]\) gives the entire algorithm. Here, we highlight several critical steps in our algorithm. We first compute the kernel matrices \( K_{g} \) and \( K_{d} \) with eq. \((1)\) and eq. \((4)\) for \( q \) training and \( r \) training image sets. After the kernel computation, we adopt kernel PCA \([36]\) for the \( q \) and \( r \) two parts to obtain the initial hash codes, i.e., \( H_{q} \) and \( H_{r} \), based on their statistical kernels in Step 1 to Step 3. Next, we update the \( H_{q} \) and \( H_{r} \) codes by optimizing eq. \((8)\) for seeking the discriminability and utilize an efficient subgradient descent algorithm \([41]\) for the binary optimization (the optimization algorithm gives the generated code with two properties: sample-wise balance and bit-wise balance.). In Step 6, we use the updated hash codes, \( H_{q} \) and \( H_{r} \), to train \( R \) two-class strong splits based on multiple kernels. More specifically, we adopt cross-training strategy \([33]\) by using the hash codes, \( H_{r} \), as training labels to train the strong splits with \( q \) training image sets and similar process to \( H_{q} \) hash code with \( r \) training image sets. After that, we update the current hash codes, \( H_{q} \) and \( H_{r} \) by using the learned strong splits based on multiple kernel learning. In order to improve the discriminability, we combine \( H_{q} \) and \( H_{r} \) together to refine the learned hash codes with eq. \((9)\). The process is then repeated. Convergence typically occurs within few outer iterations. Once we obtained the two strong split models (\( \sum_{r'=1}^{R} F_{q}^{r'}(\mathbf{x}) \) and \( \sum_{r'=1}^{R} F_{r}^{r'}(\mathbf{x}) \) in Algorithm 1), we can adopt them to generate query and database hash codes in the testing process, respectively.

**Complexity Analysis:** The time complexity for training ISH is bounded by \( O(\text{RS}^2 + (d + f)\text{RS}^3) \), where \( S \) is the number of image sets, \( f \) is number of dyadic-cuts, and \( d \) indicates the dimension
Table 1: The evaluation results measured by Mean Average Precision on the CIFAR-10 (60K) dataset. The numbers of training samples for P2P and S2S hashing are set to be 1000 points and 195 image sets, respectively. Three methods, LSH, SH, and SSH, are trained with images, while the KLSH, KSH, HER and ISH methods are trained with image sets.

| Compared Methods | Our Method |
|------------------|------------|
| bits             |            |
| 8bits            | LSH [13]   | SH [51]   | SSH [43]   | KLSH [16] | KSH [22] | HER [19] | ISH       |
|                  | 0.1063     | 0.1279    | 0.1165     | 0.1067    | 0.2363   | 0.2284   | 0.2822    |
|                  | 0.1073     | 0.1330    | 0.1416     | 0.1210    | 0.2486   | 0.2672   | 0.3069    |
| 24bits           | 0.1086     | 0.1317    | 0.1512     | 0.1420    | 0.2680   | 0.2772   | 0.3159    |
| 32bits           | 0.1194     | 0.1322    | 0.1574     | 0.1501    | 0.2818   | 0.2937   | 0.3292    |
| 48bits           | 0.1105     | 0.1352    | 0.1629     | 0.1622    | 0.3003   | 0.3154   | 0.3334    |

![Figure 2](image1.png)  
![Figure 3](image2.png)  
![Figure 4](image3.png)

5 Experiments

In this section, we evaluate the effectiveness of the proposed Image Set Hashing (ISH) method on two well-known benchmarks, CIFAR-10 and TV-series, i.e., Big Bang Theory. We also conduct extensive comparison studies with state-of-the-art methods, including Locality Sensitive Hashing (LSH) [13], Spectral Hashing (SH) [51], Kernel LSH (KLSH) [16], Semi-Supervised Hashing (SSH) [44] and supervised methods, Kernel-Based Supervised Hashing (KSH) [22] and Hashing across Euclidean and Riemannian manifold (HER) [19]. For the competing techniques, we adopted the publicly released codes of SH, KLSH, KSH and HER in our experiments. All the experiments set the number of iterations \(T\) to 15 and are performed on the MATLAB platform using a Six-Core Intel Xeon Processor with 2.8 GHz CPU and 48GB RAM.

5.1 Experiment on CIFAR-10

We compare the performance for different hashing techniques on the CIFAR-10 dataset. As a labeled subset of the 80M tiny images, the CIFAR-10 dataset consists of a total of 60K color images, each

\[^{1}\text{http://www.cs.toronto.edu/kriz/cifar.html}\]  
\[^{2}\text{https://cvhci.anthropomatik.kit.edu/ baeuml/datasets.html}\]
We evaluate two test scenarios: Hamming ranking and hash look-up. For the retrieval strategy based on Hamming ranking, the performance measured by mean average precision (MAP) is reported in Table 1 for the hash codes with the length from 8-bit to 48-bit. In addition, Fig. 4 and Fig. 5 show the mean recall and precision curves from different number of returned search samples when using 24-bit hash codes. Clearly, the proposed ISH method produces higher quality of Hamming embedding since it significantly outperforms the competing methods in terms of precisions, recalls, and MAPs. In general, the methods using set information often provide better performance than those based on P2P settings. For instance, the HER method generates the second best MAPs for most of the test cases. The relative performance gains in MAP ranges from 6% to 23.6% compared to the HER method. Such performance gains confirm the value of exploiting statistical and clique-based structural information for hash function design. In Fig. 4 we also evaluate the results using the hash look-up table strategy by showing the precision curve within Hamming radius 2 for hash codes from 8-bit to 48-bit. Again the proposed ISH method achieved the best precisions across all the cases.

5.2 Experiment on TV-series

The Big Bang Theory (BBT) video (image set) benchmark was collected by [4] and contains 3341 face videos from 1 ~ 6 episodes of season one. The dataset includes around 5 ~ 8 main cast characters and has multiple characters at the same full-view scene shot. Even though most of the scenes are taken in indoors, it is still extremely challenging since the resolution of faces regions are quite small with an average size of 75 pixels. In the experiments, we use the provided face features, which are extracted from face videos by block Discrete Cosine Transformation (DCT) feature. In this way, each face is represented by a 512-dimensional GIST feature vector.

In the experiments, we have two different settings. For the first setting, we follow the setting used in [19] and apply still images for the query in training and query in testing process and denote the setting as ISH0. The second setting indicated as ISH, we have 150 image sets for q and r two parts in the training process, respectively. For query in testing, we use 100 image sets and the remaining image sets for database. The setting can completely utilize statistical and structural information. For the comparison, the first group of compared methods consists of seven point-to-point (P2P) hash methods, i.e, LSH, ITQ, SH, Discriminative Binary Codes (DBC) [31], SSH, MM-NN [26], and KSH (point). In addition, we generate kernels for image sets (represented by covariance matrices) and employ them as input for the KLSH, KSH (set), and HER methods. Thus, we have the second group of methods that uses kernels for image sets. Such a modification can be used to further justify the advantage of exploring the structural and statistical information of image sets. The performance evaluated by MAPs for the compared methods is shown in Table 2. We vary the number of hash bits from 8 to 128 bits. As we can see, the method (ISH0) generates the second best MAPs when structural information is ignored in the q training and the query testing process. Moreover, ISH method achieved the best performance for all the tested cases when both of statistical and structural information are considered. This evidence shows that the structural information can help to generate more robust and discriminative hash codes.

In addition, we use 128-bit hash codes and show the performance curves for the seven compared methods. Fig. 5, Fig. 6 and Fig. 7 show the precision, recall, and precision-recall curves, respectively. From these results, the proposed ISH achieves the best performance compared to all the baseline techniques, including both P2P methods and modified S2S methods. The underlying reason lies in that the ISH method can simultaneously capture the statistical (covariance matrix) and structural (graph kernel) information, i.e., combination of weak learners, to generate each hash code. Hence, it captures the most intrinsic characteristics within complicated variations of face images for the same subject. Moreover, if we compare the general performance between the P2P and S2S settings for the KSH method, we can observe that the S2S hashing continuously attains higher search accuracy. It further confirms the advantage of using set information. However, simply employing the covariance
In this paper, we have presented a set-to-set (S2S) ANN search problem and proposed to learn the optimal hash codes for image sets by simultaneously exploiting the statistical and structural information. The key idea is to transform the image sets into a high dimension space where each of image set can be characterized by a graph kernel and statistical measurement. Specifically, the proposed Image Set Hashing (ISH) method captures the relations among each image set based on a clique structure. Meanwhile, it considers the content information using covariance matrix representation. As a result, the proposed S2S hashing achieves a robust and discriminative representation for searching datapoint sets. We have conducted extensive experiments on two well-known benchmarks. The experimental results have demonstrated the effectiveness of the proposed ISH method by showing superior performance over several representative competing approaches. For our future work, we will further extend the proposed method for more general set matching and search problems. In addition, we will consider incorporating sparse representation for set data to further boost the performance.

### Table 2: The evaluation results measured by Mean Average Precision on the on video (image set) benchmark TV-series (BBT). The length of hash codes ranges from 8-bit to 128-bit. Besides the proposed ISH, the first group of compared methods consists of seven P2P hashing methods, i.e., LSH, ITQ, SH, DBC, SSH, MM-NN, and KSH. The second group of compared methods include three modified techniques, i.e., KLSH, KSH, and HER, that use image set information as input.

| Method | 8 bits | 16 bits | 32 bits | 64 bits | 128 bits |
|--------|--------|---------|---------|---------|---------|
| LSH [13] | 0.2109 | 0.2086 | 0.2092 | 0.1963 | 0.1994 |
| ITQ [10] | 0.2935 | 0.3025 | 0.2989 | 0.3029 | 0.3060 |
| SH [51] | 0.2377 | 0.2652 | 0.2665 | 0.2623 | 0.2673 |
| DBC [31] | 0.4489 | 0.4495 | 0.4235 | 0.4905 | 0.3867 |
| SSH [44] | 0.2716 | 0.2855 | 0.2662 | 0.2584 | 0.3003 |
| MM-NN [26] | 0.3752 | 0.3955 | 0.4664 | 0.5124 | 0.4922 |
| KLSH [16] | 0.2450 | 0.2498 | 0.2381 | 0.2256 | 0.2325 |
| KSH [22] (point) | 0.4090 | 0.4366 | 0.4454 | 0.4567 | 0.4604 |
| KSH [22] (set) | 0.4590 | 0.4619 | 0.4534 | 0.4685 | 0.4631 |
| HER [19] | 0.4606 | 0.5049 | 0.5227 | 0.5490 | 0.5539 |
| ISH* | 0.4833 | 0.5279 | 0.5359 | 0.5501 | 0.5712 |
| ISH | 0.5018 | 0.5592 | 0.3864 | 0.6007 | 0.6200 |

**Figure 5:** The evaluation results by mean precision curves for Hamming ranking using 128 bits on the BBT video dataset. The figure is best viewed in color.

**Figure 6:** The evaluation results by mean recall curves for Hamming ranking using 128 bits on the BBT dataset. The figure is best viewed in color.

**Figure 7:** The evaluation results by mean precision-recall curves for Hamming ranking using 128 bits on the BBT dataset. The figure is best viewed in color.

matrix as inputs limits the performance improvements. In summary, with the S2S setting and fully explore statistical and structural information, the proposed ISH yields significant performance gain across all the experiments.

### 6 Conclusion

In this paper, we have presented a set-to-set (S2S) ANN search problem and proposed to learn the optimal hash codes for image sets by simultaneously exploiting the statistical and structural information. The key idea is to transform the image sets into a high dimension space where each of image set can be characterized by a graph kernel and statistical measurement. Specifically, the proposed Image Set Hashing (ISH) method captures the relations among each image set based on a clique structure. Meanwhile, it considers the content information using covariance matrix representation. As a result, the proposed S2S hashing achieves a robust and discriminative representation for searching datapoint sets. We have conducted extensive experiments on two well-known benchmarks. The experimental results have demonstrated the effectiveness of the proposed ISH method by showing superior performance over several representative competing approaches. For our future work, we will further extend the proposed method for more general set matching and search problems. In addition, we will consider incorporating sparse representation for set data to further boost the performance.
