Pattern Recognition 79 (2018) 114–129

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Cross-Modal Discrete Hashing

Venice Erin Liong[a,c], Jiwen Lu[b,*], Yap-Peng Tan[c]

[a] Rapid-Rich Object Search (ROSE) Lab, Interdisciplinary Graduate School, Nanyang Technological University, 639798, Singapore
[b] Department of Automation, Tsinghua University, Beijing 100084, China
[c] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Article history:
Received 11 August 2017
Revised 12 January 2018
Accepted 2 February 2018
Available online 6 February 2018

Keywords:
Multimedia retrieval
Hashing
Cross-modal
Discrete optimization

ABSTRACT

In this paper, we present a new cross-modal discrete hashing (CMDH) approach to learn compact binary codes for cross-modal multimedia search. Unlike most existing cross-modal hashing methods which usually relax the optimization objective function to obtain hash codes, we develop a discrete optimization framework to jointly learn binary codes and a series of hash functions for each modality, so that the performance drop due to the inferior optimization techniques can be avoided. Specifically, we present two cross-modal hashing algorithms called CMDH-linear and CMDH-kernel under the proposed framework, which performs linear and non-linear mappings to learn binary codes, respectively. Different from existing cross-modal hashing methods which maximize the corrections of hash codes from different modalities, our CMDH learns a set of shared binary codes for samples captured from different modalities, so that the modality gap can be effectively removed in cross-modal multimedia retrieval. To further improve the flexibility of our approach for different scenarios, we extend CMDH to unsupervised CMDH (unCMDH) and discrete multi-modal hashing (MMDH), which learns hash codes for training data without label information and with multi-modal labelled data. Experimental results on three benchmark datasets clearly show that our methods achieve competitive results with the state-of-the-arts.

1. Introduction

Due to the massive growth of data in the form of images, text, and videos, efficient information retrieval has become an active research area of the multimedia community. Particularly, how to perform efficient nearest neighbor search in databases has attracted great attention for solving multimedia tasks such as content-based retrieval [1], large-scale object matching [2] and recognition [3–6]. Approximate Nearest Neighbour (ANN) search aims to retrieve the most semantically relevant content from a large database given query data in the most accurate and efficient manner. This problem while interesting, is very challenging due to several factors such as large-scale databases, high-dimensional settings, storage limitations and speed requirement.

While quantization-based [7–10] and tree-based [11–14] methods have been proposed for ANN search, they usually suffer from high-dimensional data and are not suitable for large scale similarity search. Recently, there have been extensive interest in hashing-based methods for ANN search [15–23], which learn a series of hash functions to project each data into compact binary codes such that similar samples are mapped into similar binary codes. By assigning binary codes to high-dimensional feature vectors for retrieval, we are able to significantly improve the similarity search speed and store millions of data in storage [24]. While promising results have been obtained, most existing hashing methods have been limited to single-modal retrieval, where the query instance and gallery are from the same source of multimedia data. In real-world scenarios, it is easy to access multi-modal data which are leveraged to conduct multimedia similarity search. For example, images/videos uploaded in websites are usually tagged with some text descriptions. Hence it is desirable to develop an effective cross-modal hashing approach to exploit the correlation between different modalities.

Cross-modal hashing seeks to learn a series of hash functions for each modality such that binary codes of samples corresponding to the same instance from different modalities are as similar as possible. Existing cross-modal hashing methods can be classified into two categories: unsupervised and supervised. Unsupervised methods [25–28] maximize the inter-modal similarity of training data to learn the correlations. Supervised methods [29–33] use label information of training samples to exploit the semantic relationship of samples from different modalities. However, most existing cross-modal hashing methods perform relaxation on the binary constraint of the optimization objective, which may lead to an

* Corresponding author.
E-mail address: lujiwen@tsinghua.edu.cn (J. Lu).

https://doi.org/10.1016/j.patcog.2018.02.002
0031-3203/

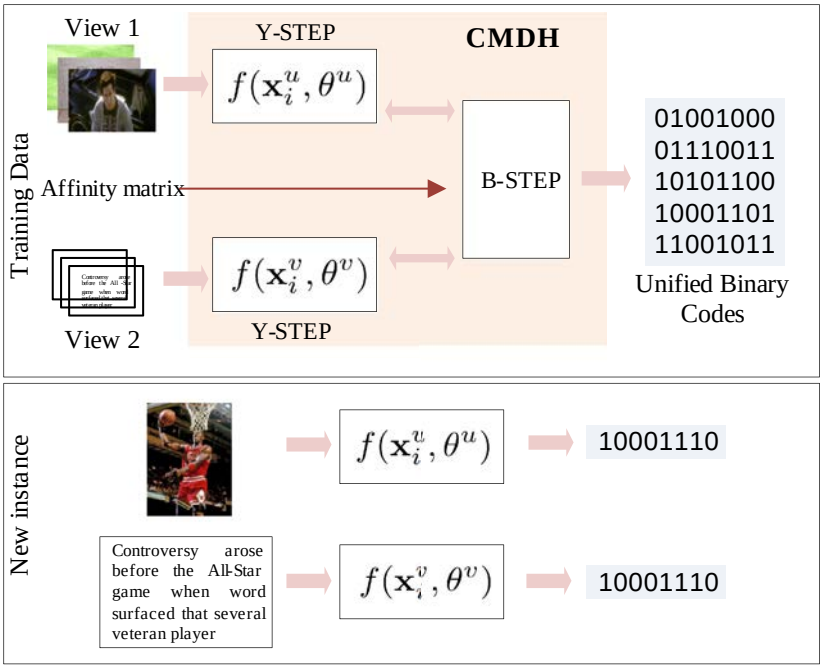


Fig. 1. Basic idea of our proposed Cross-Modal Discrete Hashing approach. In the training stage, our optimization objective consists of two steps which are performed iteratively. The first step is the Y-step which learns the hash functions with fixed binary codes, and the second step is the B-step which learns the unified binary code. In the testing stage, we use the learned hash functions to obtain the binary codes for large-scale cross-modal multimedia retrieval.

accumulated quantization error and possible information loss [34,35]. Moreover, some models [36–38] learn a separate binary code for each modality but of similar semantics, which may lead to inconsistent binary codes.

In this work, we propose a new cross-modal discrete hashing (CDMH) method for cross-modality search. Unlike several existing methods which relax the optimization objective function to address the NP-hard problem, we perform a discrete optimization method that jointly learn binary codes and a series of hash functions. In our approach, we do not relax the binary constraint to avoid approximated solutions, so that our learned hash codes are more representative. Unlike other methods which learn separate hash codes for each modality and maximize the correlations during training, our CDMH method learns a shared set of binary code for samples from different modalities based on their semantic affinities, so that the modality gap is implicitly reduced. Fig. 1 shows the basic idea of our proposed approach which consists of two iterative steps during training to learn a unified binary code and hash functions. The learned hash functions are used to obtain the new binary codes of the query samples during testing stage. Specifically, we present two cross-modal hashing algorithms called CDMH-linear and CDMH-kernel under the proposed CDMH framework, which perform linear and non-linear mappings to learn binary codes, respectively. To improve the flexibility of our CDMH for different scenarios, we further extend our CDMH to unsupervised CDMH (unCDMH) and discrete multi-modal hashing (MMDH) which learn hash codes for training data without label information and multi-modal labelled data, respectively. Experiments on three benchmark datasets showed that our methods achieve competitive results compared to the current state-of-the-art cross-modal hashing methods.

The contributions of this work are summarized as follows:

1. We propose a cross-modal discrete hashing (CDMH) approach to jointly learn a unified binary code and the respective hash functions through a discrete optimization procedure so that no relaxation is done in solving the binary constraints.
2. We develop two algorithms called CDMH-linear and CDMH-kernel based on the proposed CDMH framework which perform linear and non-linear kernel mapping to learn the hash functions, respectively.
3. we extend our CDMH to unsupervised CDMH (unCDMH) and discrete multi-modal hashing (MMDH), which learn hash codes for training data without label information and multi-modal labelled data, respectively.

2. Background

In this section, we briefly review three related topics: 1) single-modal hashing, 2) cross-modal retrieval, and 3) cross-modal hashing.

2.1. Single-modal hashing

Hashing is a popular technique for large scale similarity search because of its efficiency in computation and storage. The basic idea of hashing is to construct multiple hash functions that map each real-valued feature vector to a compact binary code such that semantically similar representations have similar binary codes. A variety of hashing methods have been proposed and can mainly be classified into two categories: *data-independent* [35,39,40] and *data-dependent* [15,41–46]. *Data-independent* hashing methods use randomly map samples into a feature space and then perform binarization to compute binary codes. Representative methods of this category are locality sensitive hashing [39] and its kernelized or discriminative extensions [40,47,48]. *Data-dependent* hashing methods learn efficient hash functions from training data to map samples into compact binary codes. Representative methods in this category include iterative quantization [15], sequential projection learning hashing [16], FastHash using boosted trees [41], supervised discrete hashing [34], and deep hashing [49–51]. While these methods have achieved reasonably good performance in many large scale similarity search systems, most of them are developed for single-modal data, wherein the query and gallery samples

belong to only one type of modality, and are not suitable for cross-modal multimedia retrieval. Most recently, Liu et al. [10] proposed a prototype-based hashing where it uses an adaptive binary quantization to obtain prototypes in the original space which represent a subset of binary codes. [9] implemented a global cluster structure to minimize the quantization loss similar to Liu et al. [10], but ensures that the distance of the inter(intra)-class neighbour pairs are maximized (minimized). While these methods performs an adaptive way of minimizing the quantization loss, our method is a general way to learn a representative binary code in a discrete manner by treating it as a unique variable during the optimization procedure.

2.2. Cross-modal retrieval

Unlike single-modal multimedia retrieval where both query examples and samples stored in the database are from the same multimedia source, cross-modal multimedia retrieval aims to search samples across different modality in which two samples are from different multimedia but may share similar semantics. Generally, there are two main tasks in cross-modal multimedia retrieval: text-image retrieval and image-text retrieval. The first aims to retrieve text documents by using a query image, and the second is to retrieve images by using a query text. Recently, several methods have been proposed for cross-modal retrieval, where the key idea is to learn a common subspace between images and text [52–57] to model their semantic relationship. For example, Rasiwasia et al. [52] used canonical component analysis to map both text documents and images into a latent space. Wang et al. [53] learned a coupled feature space to select the most relevant and discriminative features for cross-modal matching. Yan and Mikolajczyk [55] used deep canonical correlation analysis to maximize the correlation among samples from two modalities. These retrieval methods perform cross-modal matching with high-dimensional features, so are not suitable for large scale cross-modal retrieval. To address the scalability issue, compact binary features are desired.

2.3. Cross-modal hashing

Recently, several cross-modal hashing methods [58–60] have been proposed, and they can be mainly divided into two categories: *unsupervised* and *supervised*. Representative unsupervised methods include collaborative hashing [59], collective matrix factorization hashing (CMFH) [25] and latent semantic sparse hashing (LSSH) [26]. Collaborative hashing technique minimized the quantization loss using orthogonal rotation matrices and minimize the correlation between the two modalities in the Hamming Space. CMFH learned a unified binary code in the training stage by performing matrix factorization with latent factor model; and LSSH learned a unified binary code using sparse coding and matrix factorization iteratively. Typical supervised cross-modal hashing methods are cross modality similarity sensitive hashing (CMSSH) [36], cross-view hashing (CVH) [38], kernel-based supervised hashing for cross-view (KSH-CV) [31], predictable dual-view hashing (PDH) [37], semantic correlation maximization (SCM) [32], Quantized Correlation Hashing (QCH) [30], and semantics preserving hashing (SePH) [29]. CMSSH learned hash functions by preserving the intra-class similarity and performing boosting and eigen-decomposition operations. CVH extended the spectral hashing method from single-view to cross-view by minimizing the cross-modal similarity. KSH-CV learned hash functions to preserve the inter-view similarity in the kernel space by Adaboost. PDH performs dual-view mapping where data points which are near in the original space are also near in the hash space. SCM utilized semantic labels to maximize the semantic correlation in an iterative and sequential manner. QCH ensures that the quantization loss is minimized during cross-modal hash learning, however, it performed relaxation in the binary constraints during optimization. SePH learned unified binary codes for cross-modal data by transforming the hamming distance and affinity matrix in two probability distributions and minimizing their Kullback–Leibler divergence. While these cross-modal hashing methods have achieved encouraging performance, most of them relax the binary constraint in their optimization objective functions, which may lead to a loss of information because of the accumulated quantization error [30,34,35,41].

More recently, [58] implemented a deep cross-modal hashing technique based on pairwise relationships. Specifically, it outputs a hash layer in which the network ensures that the intra-modal loss and inter-modal loss are minimized. While this exploits deep features and learns a unified binary code, it requires more training time due to the requirement of learning the parameters deep networks. In this work, we developed a general cross-modal discrete hashing method to jointly learn binary codes and a series of hash functions for each modality without using any relaxation of constraints.

3. Proposed approach

In this section, we first present the proposed CMDH framework and then propose two algorithms called CMDH-linear and CMDH-kernel which perform linear and non-linear kernel mapping to learn the hash functions, respectively. Lastly, we present two extensions called unsupervised CMDH (unCMDH) and discrete multi-modal hashing (MMDH) which learns hash codes for training data without label information and multi-modal labelled data, respectively.

3.1. CMDH

Let $\mathbf{X}^u = [\mathbf{x}_1^u, \cdots, \mathbf{x}_N^u] \in \mathbb{R}^{d^u \times N}$ and $\mathbf{X}^v = [\mathbf{x}_1^v, \cdots, \mathbf{x}_N^v] \in \mathbb{R}^{d^v \times N}$ be the training set of two different modalities (U and V), respectively, d^u and d^v are its corresponding feature dimension (where d^u need not to be equal to d^v), and N is the number of training samples. Our CMDH aims to learn a shared binary feature vector, $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_N] \in \{-1, 1\}^{K \times N}$ of length K and a series of hash functions for each modality simultaneously. To achieve this, we formulate the following optimization objective:

$$\begin{aligned} \min_{\mathbf{b},\mathbf{y}^u,\mathbf{y}^v} J &= J_1 + \eta J_2 \\ &= \sum_{i,j=1,i\neq j}^{N} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 A_{ij} \\ &\quad + \eta \sum_{i=1}^{N} (\|\mathbf{b}_i - \mathbf{y}_i^u\|_2^2 + \|\mathbf{b}_i - \mathbf{y}_i^v\|_2^2) \qquad (1) \\ \text{subject to} \quad & \mathbf{b}_i \in \{-1, 1\}^K \end{aligned}$$

where $\mathbf{y}_i^u, \mathbf{y}_i^v \in \mathbb{R}^K$ is a continuous variable based on embedding function f with parameter θ^u and θ^v as follows:

$$\mathbf{y}_i^u = f(\mathbf{x}_i^u, \theta^u) \qquad (2)$$

$$\mathbf{y}_i^v = f(\mathbf{x}_i^v, \theta^v) \qquad (3)$$

and $\mathbf{b}_i$, $\mathbf{b}_j$ is the common binary vector to be learned for the ith and jth cross-modal data. The embedding function $f(\cdot)$ can be any linear or non-linear mapping function parameterized by θ, that transforms the input feature modality, $\mathbf{x}^{u/v}$, to a continuous real-value code $\mathbf{y}^{u/v}$ such that can then be used to obtain the shared

binary feature vector, **b**. In our work, we will present a linear and kernel embedding function as presented in the succeeding sections.

In (1), J_1 ensures the hamming distance of the learned binary codes which are semantically similar are small as much as possible, J_2 ensures the quantization loss between binary codes and the corresponding learned real-valued codes in (2) and (3) are minimized, and η is a constant variable that balances J_1 and J_2. A_{ij} is an affinity matrix which represents the semantic relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$, which can be obtained from the cosine similarity of label information of samples as follows:

$$A_{ij} = \frac{l_i \cdot l_j}{\|l_i\| \|l_j\|} \tag{4}$$

where l_i, $l_j \in [0, 1]^{1 \times L}$ is the label information for the ith and jth data having L classes, respectively.

Given $\mathbf{Y}^* = [\mathbf{y}_1^*, \cdots, \mathbf{y}_N^*] \in \mathbb{R}^{K \times N}$[1] and normalizing the affinity matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, we re-write (1) into a matrix form as follows:

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{Y}^u, \mathbf{Y}^v} J &= -\mathrm{tr}(\mathbf{B}^\top \mathbf{A} \mathbf{B}) \\ &\quad + \eta(\|\mathbf{B} - \mathbf{Y}^u\|_F^2 + \|\mathbf{B} - \mathbf{Y}^v\|_F^2) \\ \text{subject to} \quad & \mathbf{B} \in \{-1, 1\}^{N \times K} \end{aligned} \tag{5}$$

The optimization problem in (5) is non-convex due to the binary constraints, which makes it difficult to solve. However, it can be addressed using an iterative approach where we keep other variables fixed and solve one iteratively. The following presents the detailed procedure as follows:

B-step: We solve for **B** by fixing $\mathbf{Y}^u$ and $\mathbf{Y}^v$. Having fixed $\mathbf{Y}^u$ and $\mathbf{Y}^v$, we obtain the following discrete optimization problem:

$$\begin{aligned} \min_{\mathbf{B}} J(\mathbf{B}) &= -\mathrm{tr}(\mathbf{B}^\top \mathbf{A} \mathbf{B}) \\ &\quad + \eta(\|\mathbf{B} - \mathbf{Y}^u\|_F^2 + \|\mathbf{B} - \mathbf{Y}^v\|_F^2) \\ \text{subject to} \quad & \mathbf{B} \in \{-1, 1\}^{N \times K} \end{aligned} \tag{6}$$

Having removed all these constant terms, we re-write (6) as follows:

$$\begin{aligned} \min_{\mathbf{B}} J(\mathbf{B}) &= -\mathrm{tr}(\mathbf{B}^\top \mathbf{A} \mathbf{B}) - \eta \mathbf{B}(\mathbf{Y}^u + \mathbf{Y}^v) \\ \text{subject to} \quad & \mathbf{B} \in \{-1, 1\}^{N \times K} \end{aligned} \tag{7}$$

This binary quadratic problem can be solved through a linearization technique according to [35]. We obtain a closed-form solution as follows:

$$\mathbf{B}^{(r+1)} = \mathrm{sgn}(2\mathbf{A}\mathbf{B}^{(r)} + \eta(\mathbf{Y}^u + \mathbf{Y}^v)) \tag{8}$$

where $\mathbf{B}^{(r)}$ is the current binary matrix at iteration r, and $\mathbf{B}^{(r+1)}$ is the new binary matrix that will be used for the Y-step as $\mathbf{B} = \mathbf{B}^{(r+1)}$.

Y*-step: We solve for $\mathbf{Y}^*$ by fixing **B** and $\mathbf{Y}^{\tilde{*}}$. The discrete optimization problem can be re-written as:

$$\min_{\mathbf{Y}^*} J(\mathbf{Y}^*) = \|\mathbf{B} - \mathbf{Y}^*\|_F^2 \tag{9}$$

which can be further rewritten as follows in terms of f and θ^*.

$$\min_{\theta^*} J(\theta^*) = \|\mathbf{B} - f(\mathbf{X}^*, \theta^*)\|_F^2 \tag{10}$$

The solution to (10) varies from $f(\cdot)$ which can be any representative linear or non-linear embedding function. Algorithm 1 shows the general optimization procedure of our CMDH approach.

Out-of-Sample Extension: In the training stage of our CMDH, we learn shared binary codes for cross-modal data, but we also need to obtain modal-specific hash functions for out-of-sample data points. To obtain the binary codes for each query sample, we ensure that the quantization loss between binary codes and the corresponding learned real-valued codes in each modality is minimized as follows:

$$\begin{aligned} \min_{\mathbf{b}^*_{query}} J_2(\mathbf{b}^*_{query}) &= \|\mathbf{b}^*_{query} - \mathbf{y}^*_{query}\|_2^2 \\ \text{subject to} \quad & \mathbf{b}^*_{query} \in \{-1, 1\}^K \end{aligned} \tag{11}$$

Similar to the B-step optimization problem, we obtain a closed-form solution for each query sample as follows:

$$\mathbf{b}^u_{query} = \mathrm{sgn}(\mathbf{y}^u_{query}) \tag{12}$$

$$\mathbf{b}^v_{query} = \mathrm{sgn}(\mathbf{y}^v_{query}) \tag{13}$$

Algorithm 1: CMDH.

Input: Training set $\mathbf{X}^u$ and $\mathbf{X}^v$, affinity matrix **A**.
Output: Parameters θ^u, θ^v.
Step 1 (Initialization):
1.1. Initialize $\mathbf{B}^{(1)}$ randomly
1.2. Initialize θ^u and θ^v according to $f(\cdot)$
Step 2 (Optimization):
for $r = 1, 2, \cdots,$ *maxiter* **do**
2.1. Solve $\mathbf{B}^{(r+1)}$ using (8);
2.2. Solve θ^u based on (10);
2.3. Solve θ^v based on (10).
If $r > 1$ and $|J_r - J_{r-1}| < \varepsilon$, go to **Return**.
end
Return: θ^u, θ^v.

3.2. CMDH-linear

We develop a CMDH-linear method by using a linear mapping $\mathbf{W}^* \in \mathbb{R}^{K \times d^*}$ to project each training sample **X** in different modalities as follows:

$$f_{lin}(\mathbf{x}^*, \mathbf{W}^*) = \mathbf{W}^{*\top} \mathbf{x}^* \tag{14}$$

from (10), the optimization is re-written as:

$$\min_{\mathbf{W}^*} J(\mathbf{W}^*) = \|\mathbf{B} - \mathbf{W}^{*\top} \mathbf{X}^*\|^2 \tag{15}$$

Since (15) is a regularized least squares problem, we can easily obtain a closed-form solution to **W** as follows:

$$\mathbf{W}^u = (\mathbf{X}^u \mathbf{X}^{u\top} + \lambda \mathbf{I})^{-1} \mathbf{X}^u \mathbf{B}^\top \tag{16}$$

$$\mathbf{W}^v = (\mathbf{X}^v \mathbf{X}^{v\top} + \lambda \mathbf{I})^{-1} \mathbf{X}^v \mathbf{B}^\top \tag{17}$$

Algorithm 2 details the implementation of our proposed CMDH-linear method.

3.3. CMDH-kernel

To better exploit the non-linear relationship of samples, we also develop a CMDH-kernel method by using a non-linear mapping in the kernel space. Specifically, we first map samples from each modality to a set of g points, also called as *anchors* defined as $[\mathbf{a}_1, \cdots, \mathbf{a}_g]$. We perform random selection on the training data for the anchor points. Based on empirical results, obtaining the anchors using k-means or randomly does not make any difference. Then, we learn a linear projection matrix, $\mathbf{P}^* \in \mathbb{R}^{K \times g}$, which maps samples from this kernel space into a low-dimensional space.

$$f_{kernel}(\mathbf{x}^*, \mathbf{P}^*) = \mathbf{P}^{*\top} \phi(\mathbf{x}^*) \tag{18}$$

[1] For ease of presentation, we represent $* = \{u, v\}$ and $\tilde{*} = \{v, u\}$.

Algorithm 2: CMDH - *linear*.

Input: Training set $\mathbf{X}^u$ and $\mathbf{X}^v$, affinity matrix $\mathbf{A}$.
Output: Parameters $\mathbf{W}^u, \mathbf{W}^v$.
Step 1 (Initialization):
1.1 Perform centering in $\mathbf{X}^u$ and $\mathbf{X}^v$
1.2 Initialize $\mathbf{B}^{(1)}$ randomly
1.3 Initialize $\mathbf{W}^u$ and $\mathbf{W}^v$ randomly
Step 2 (Optimization):
for $r = 1, 2, \cdots,$ *maxiter* **do**
  2.1 Solve $\mathbf{B}^{(r+1)}$ using (8);
  2.2 Solve $\mathbf{W}^u$ using (16);
  2.3 Solve $\mathbf{W}^v$ using (17).
  If $r > 1$ and $|J_r - J_{r-1}| < \varepsilon$, go to **Return**.
end
Return: $\mathbf{W}_u, \mathbf{W}_v$.

where $\phi(\mathbf{x}) = [\exp(\mathcal{D}(\mathbf{x}, \mathbf{a}_1)/\sigma), \cdots, \exp(\mathcal{D}(\mathbf{x}, \mathbf{a}_g)/\sigma)]$ is the RBF kernel mapping function with σ as the kernel bandwidth parameter, and $\mathcal{D}(\cdot)$ is the ℓ_2 distance function.

Similarly, from (10), the optimization is re-written as:

$$\min_{\mathbf{P}^*} J(\mathbf{P}^*) = \|\mathbf{B} - \mathbf{P}^{*\top}\phi^*(\mathbf{X}^*)\|_F^2 \tag{19}$$

It can be seen that (19) is also a regularized least squares problem, and $\mathbf{P}$ can be easily solved by the following closed-form solution:

$$\mathbf{P}^u = (\phi(\mathbf{X}^u)\phi(\mathbf{X}^u)^\top)^{-1}\phi(\mathbf{X}^u)\mathbf{B}^\top \tag{20}$$

$$\mathbf{P}^v = (\phi(\mathbf{X}^v)\phi(\mathbf{X}^v)^\top)^{-1}\phi(\mathbf{X}^v)\mathbf{B}^\top \tag{21}$$

Algorithm 3 details the implementation of our proposed CMDH-kernel method.

Algorithm 3: CMDH - *kernel*.

Input: Training set $\mathbf{X}^u$ and $\mathbf{X}^v$, affinity matrix $\mathbf{A}$, number of anchor points g.
Output: Parameters $\mathbf{P}^u, \mathbf{P}^v$.
Step 1 (Initialization):
1.1 Perform centering in $\mathbf{X}^u$ and $\mathbf{X}^v$
1.2 Initialize $\mathbf{B}^{(1)}$ randomly
1.3 Select g anchor points, $\{a_i^u\}_{i=1}^g$ and $\{a_i^v\}_{i=1}^g$ randomly from $\mathbf{X}^u$ and $\mathbf{X}^v$, respectively
Step 2 (Optimization):
for $r = 1, 2, \cdots,$ *maxiter* **do**
  2.1 Solve $\mathbf{B}^{(r+1)}$ using (8);
  2.2 Solve $\mathbf{P}^u$ using (20);
  2.3 Solve $\mathbf{P}^v$ using (21).
  If $r > 1$ and $|J_r - J_{r-1}| < \varepsilon$, go to **Return**.
end
Return: $\mathbf{P}_u, \mathbf{P}_v$.

3.4. CMDH extensions

Unsupervised CMDH (unCMDH): While our CMDH is developed for supervised hashing, it can be easily extended to unsupervised in which only the pairwise information is available. An estimated affinity matrix, $\tilde{\mathbf{A}}$, can be obtained based on the neighborhood structure of each modalities using anchor graphs [61]. We assume that instances that are very much near (distance-wise) to each other in both modalities are semantically related to some extent. We are able to obtain data-to-anchor affinity matrix, $\mathbf{Z}^* \in \mathbb{R}^{N\times G}$, where G is the number of anchors used. This can lead to a data-to-data affinity matrix $\mathbf{A}^* = \mathbf{Z}^*\Lambda^{*-1}\mathbf{Z}^{*\top} \in \mathbb{R}^{N\times N}$ where $\Lambda^* = \text{diag}(\mathbf{Z}^{*\top}\mathbf{1}) \in \mathbb{R}^{G\times G}$. The final affinity matrix is then the addition of the affinity matrices of respective modalities:

$$\tilde{\mathbf{A}} = \mathbf{A}^u + \mathbf{A}^v \tag{22}$$

This affinity matrix is then normalized so that each column would have a sum of 1. We then follow the steps in Algorithm 1 but replacing the original $\mathbf{A}$ with $\tilde{\mathbf{A}}$.

Discrete Multi-Modal Hashing (MMDH): Our method can also be easily extended for multi-modal experiments. We also propose a Multi-Modal Discrete Hashing (MMDH) method to hash data which have multiple modalities.

Given a group of M different modalities $U = \{u^{(1)}, \cdots, u^{(M)}\}$ and based on the initial formulation in (1), we obtain new objective formulation suitable for multiple modalities as follows:

$$\begin{aligned}\min_{\mathbf{b}, \{\mathbf{y}^{u^{(m)}}\}_{m=1}^M} J = &\sum_{i,j=1, i\neq j}^{N} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 A_{ij} \\ &+ \eta \sum_{m=1}^{M}\sum_{i=1}^{N} \|\mathbf{b}_i - \mathbf{y}_i^{u^{(m)}}\|_2^2 \\ \text{subject to} \quad &\mathbf{b}_i \in \{-1, 1\}^K\end{aligned} \tag{23}$$

Similarly, following the simplification and derivation of CMDH, we obtain a new B-step optimization procedure:

$$\mathbf{B}^{(r+1)} = \text{sgn}\left(2\mathbf{A}\mathbf{B}^{(r)} + \eta \sum_{m=1}^{M} \mathbf{Y}^{u^{(m)}}\right) \tag{24}$$

The Y-step optimization procedure remains the same but with updates for M modality types. This is then repeated until convergence. The new formulation is presented in Algorithm 4. It is then straightforward to implement the linear and kernel form of MMDH based on Algorithms 2 and 3, respectively.

Algorithm 4: MMDH.

Input: Training set $\mathbf{X}^{u^{(1)}}, \cdots \mathbf{X}^{u^{(M)}}$, affinity matrix $\mathbf{A}$.
Output: Parameters $\{\theta^{u^{(m)}}\}_{m=1}^M$.
Step 1 (Initialization):
  1.1. Initialize $\mathbf{B}^{(1)}$ randomly
  1.2. Initialize $\{\theta^{u^{(m)}}\}_{m=1}^M$ according to $f(\cdot)$
Step 2 (Optimization):
for $r = 1, 2, \cdots,$ *maxiter* **do**
  2.1. Solve $\mathbf{B}^{(r+1)}$ using (24);
  2.2. Solve $\{\theta^{u^{(m)}}\}_{m=1}^M$:
  for $m = 1, 2, \cdots, M$ **do**
    Solve $\theta^{u^{(m)}}$ based on (10).
  end
  If $r > 1$ and $|J_r - J_{r-1}| < \varepsilon$, go to **Return**.
end
Return: $\{\theta^{u^{(m)}}\}_{m=1}^M$.

3.5. Relation to Discrete Graph Hashing (DGH) [35]

The DGH's formulation is written as follows:

$$\begin{aligned}\max \mathcal{Q}(\mathbf{B}, \mathbf{Y}) = &\text{tr}(\mathbf{B}^\top\mathbf{A}\mathbf{B}) + \rho\text{tr}(\mathbf{B}^\top\mathbf{Y}) \\ \text{s.t.} \quad &\mathbf{B} \in \{1, -1\}^{N\times K}, \mathbf{Y} \in \mathbf{R}^{N\times K}, \\ &\mathbf{1}^\top\mathbf{Y} = 0, \mathbf{Y}^\top\mathbf{Y} = N\mathbf{I}_K\end{aligned} \tag{25}$$

(a) Wiki (b) MIRFlickr (c) NUS-WIDE

Fig. 2. Sample images - text/tag pair for the Wiki, MIRFlickr, and NUS-WIDE dataset, respectively.

where $\mathbf{A}$ is the affinity matrix based on the anchor graph Laplacian, $\mathbf{L} = \mathbf{I}_N - \mathbf{A}$. $\mathbf{Y}$ is a continuous variable to soften the constraints $\mathbf{1}^\top \mathbf{B} = 0$ and $\mathbf{B}^\top \mathbf{B} = N\mathbf{I}_K$. In this formulation, the distance between $\mathbf{B}$ and $\mathbf{Y}$ is minimized and controlled by a factor ρ.

While our method may have some similarities in the mathematical formulation with the Discrete Graph Hashing (DGH) work, there are 3 key differences of our CMDH method compared to DGH. First, our CMDH addresses the problem of cross-modal/multimodal hashing instead of single-modal hashing by learning a unified binary code for two modalities. Hence, we have investigated a cross-modal hashing implementation based on discrete optimization. Second, unlike DGH which only extract the affinity matrix from anchors based on distances of the samples, our method can perform supervised and unsupervised learning in which we can exploit the label information of the database in order to obtain a similarity matrix. By doing so, we obtain representative shared binary codes. Lastly, in DGH, $\mathbf{Y}$ is defined as a continuous variable solved through singular value decomposition (SVD), in which a suboptimal solution is derived. Differently, our method defines $\mathbf{Y}$ as output of an embedding function (linear or kernel) of the feature input. Hence, we perform iterative and alternative optimization procedure to solve the function parameters accordingly. By doing so, we represent $\mathbf{Y}$ in a general way suitable for unseen data, based on the assumption that $\mathbf{Y}$ is modelled by a defined parametric function. Our general formulation can then be extended to other embedding functions such as boosting, mixture of Gaussians or neural network models.

4. Experiments

We conducted experiments on three widely used datasets including Wiki,[2] MIRFlicker,[3] and NUS-WIDE[4] for cross-modality retrieval to evaluate the performance of our approach. Sample image-text/tag pairs from these three datasets are shown in Fig. 2. The following describes the details of experimental settings and results.

4.1. Datasets and experimental settings

The Wiki dataset introduced by Rasiwasia et al. [52], contains 2866 image-text pairs collected from featured Wikipedia articles. For each pair, the image is represented by a 128-dimensional SIFT feature vector, and the text article is represented by a 10-dimensional feature vector which is computed by the Latent Dirichlet Allocation (LDA) model. Each image-text pair is annotated by one of 10 categories. Similar to previous works [25,29], we use 25% samples from the dataset for query and employed the rest as training and gallery instances.

The MIRFlickr dataset introduced by Huiskes and Lew [62], contains 25000 image-text pairs collected from Flickr, where images are annotated with textual tags. Each image is represented with a 150-dimensional edge histogram, and each text is represented by a 500-dimensional feature vector which is also computed by the LDA model. Each image-text pair is annotated with one or more of the pre-defined 24 labels. Similar to [29], we use the textual tags that appeared at least 20 times. We randomly select only 5% samples of the dataset for query and use the rest as gallery samples. We also randomly select 5000 pairs from the gallery set to train the model.

The NUS-WIDE dataset introduced by Chua et al. [63], contains 269648 image-text pairs collected from the web where images are annotated with 81 label tags. Following the same settings in previous works [25,26,29], we select the top 10 most frequent labels and obtain a new subset which consists of 186577 image-text pairs. Each image is represented by a 500-dimensional SIFT feature vector, and each text is represented by a 1000-dimensional feature vector which is computed by the bag-of-words (BoW) model. We use 1% samples of the dataset as query instances, and use the rest as gallery samples. We also randomly select 5000 pairs in the gallery set to train the model.

To evaluate the performance of different cross-modal hashing methods, we employed the *mean average precision* (mAP) which is defined as the mean of the average precision (AP) of the top R retrieved instances as follows:

$$AP = \frac{1}{M}\sum_{r=1}^{R} p(r)\cdot\delta(r) \tag{26}$$

where M is the number of relevant instances in the database for the query instance $p(r)$ denotes the precision of the top r retrieved set, and $\delta(r)$ is an indicator of relevance of a given rank (which is set to 1 if relevant and 0 otherwise). For the MIRFlickr and NUS-WIDE datasets which have multiple labels, the relevant instance for a query is defined as instances sharing at least one label. In our experiments, we set $R = 100$ for all datasets. Besides mAP, we also used the precision-recall curve and top-N precision curve to evaluate the performance of different methods.

4.2. Experimental results

Comparisons with State-of-the-art Cross-Modal Hashing Methods: We compared our CMDH with several state-of-the-art cross-modal hashing methods including CMFH [25], LSSH [26], PDH [37], CCA-ITQ [15], CVH [38], SCM-Orth [32], SCM-Seq [32], SePH-rnd [29] and SePH-km [29]. Specifically, CMFH, LSSH, PDH, and CCA-ITQ exploit the correspondence of cross-modal similarity to learn hash codes, and CVH, SCM-Orth, SCM-Seq, SePH-rnd

[2] http://www.svcl.ucsd.edu/projects/crossmodal/.
[3] http://press.liacs.nl/mirflickr/.
[4] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm/.

Table 1
The mAPs of different cross-modal hashing methods on the Wiki and MIRFlickr25k dataset, where images were used as query samples and texts/tags were employed as gallery samples, respectively.

	Wiki				MIRFlickr25k			
Method	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [38]	0.1988	0.1772	0.1613	0.16	0.6209	0.6112	0.6057	0.5933
CCA-ITQ [15]	0.2342	0.2406	0.2306	0.2241	0.6448	0.6420	0.6333	0.6350
PDH [37]	0.1841	0.1799	0.1858	0.1887	0.6202	0.6291	0.6321	0.6366
LSSH [26]	0.2239	0.226	0.2199	0.2149	0.6361	0.6297	0.6247	0.6292
CMFH [25]	0.2260	0.2466	0.2484	0.2547	0.6432	0.6403	0.6304	0.6204
unCMDH - *ker*	0.2354	0.2373	0.2612	0.2582	0.6937	0.6947	0.7053	0.7083
unCMDH - *lin*	0.2157	0.2296	0.2331	0.2437	0.6715	0.6810	0.6849	0.6929
SCM - *orth* [32]	0.1985	0.1794	0.1644	0.1615	0.6863	0.6913	0.6983	0.6996
SCM - *seq* [32]	0.1388	0.1398	0.1389	0.1377	0.6498	0.6348	0.6160	0.6120
SePH - *rnd* [29]	0.2681	0.2862	0.2937	0.2915	0.6986	**0.7057**	0.7122	0.7129
SePH - *km* [29]	0.2691	0.2775	**0.2940**	0.2928	**0.7020**	0.7052	0.7117	0.7174
CMDH - *ker*	**0.2705**	**0.2866**	0.2894	**0.2982**	0.6915	0.7046	**0.7126**	**0.72204**
CMDH - *lin*	0.2484	0.2576	0.2700	0.2731	0.6861	0.6815	0.6867	0.6950

Table 2
The mAPs of different cross-modal hashing methods on the NUS-WIDE dataset, where images were used as query samples and texts/tags were employed as gallery samples, respectively.

Method	16 bits	32 bits	64 bits	128 bits
CVH [38]	0.4211	0.4197	0.4119	0.4125
CCA-ITQ [15]	0.4265	0.4267	0.4200	0.4152
PDH [37]	0.5021	0.5282	0.5432	0.5434
LSSH [26]	0.4780	0.4836	0.4815	0.4940
CMFH [25]	0.4456	0.4570	0.4752	0.4740
unCMDH - *ker*	0.5184	0.5411	0.5683	0.5784
unCMDH - *lin*	0.4300	0.4242	0.4224	0.4215
SCM - *orth* [32]	0.5306	0.5373	0.5481	0.5433
SCM - *seq* [32]	0.4874	0.4722	0.4452	0.4133
SePH - *rnd* [29]	0.5558	0.5596	0.5665	0.5807
SePH - *km* [29]	0.5503	0.5603	**0.5724**	**0.5854**
CMDH - *ker*	**0.5622**	**0.5691**	0.5720	0.5849
CMDH - *lin*	0.5370	0.5450	0.5623	0.5705

and SePH-km exploit the label information of samples to learn discriminative hash codes. The standard implementations from the original authors using the default parameters were used for all methods except CVH. We carefully implemented the CVH method since its code is not publicly available. We conducted experiments with the same randomly selected training, gallery and query sets, and repeated the experiments 10 times and obtained the average as the final performance. For our CMDH-kernel and CMDH-linear, we set the balancing parameter η to be 0.5 for both methods. To be consistent with the kernel mapping of SePH, we use 500 anchors and σ of 0.6 for the CMDH-kernel. For other compared methods, we set the default parameters according to the original papers.

For consistency and fair comparison of the different methods, we first evaluated these methods using the out-of-sample extension of the gallery set. Tables 1–4 show the mAP results of different methods on different datasets where the binary code length was set at 16, 32, 64 and 128, respectively. Figs. 3–5 show the precision vs. recall curves and precision curves vs the retrieval number N on the Wiki, MirFlickr, and NUS-WIDE datasets, respectively. We clearly see from these results that our CMDH-kernel obtains the best performance along with SePH, and outperforms all other compared cross-modal hashing methods. While our CMDH-kernel is competitive with the SePH in terms of retrieval accuracy, our method have less computational costs due to simpler optimization procedure. It can also be seen that there are instances where some methods show decrease in performance when the bit length reaches 128 dimensions, particularly the CMFH and SCM. Our method does not suffer from this limitation.

Overall, our implementation showed competitive performance in all three datasets which may be attributed to two key concepts. First, different from CVH, CCA-ITQ, and PDH, our method learns a unified binary code which implicitly eliminate the modality gap between two modalities. This in turn, learns a more representative hashing function for each modality. Second, different from LSSH, CMFH, and SePH, our method learns the unified binary code as a separate variable in a discrete manner. This avoids the possible approximation loss caused by any relaxation during optimization. We can also see from the results that representing the hashing function as a non-linear representation, such as a kernel function, can significantly improve the performance. While, the SePH-km also performs well as it uses a kernel representation, it however tries to minimize the quantization loss of real-value output and binary code explicitly. In addition, a gradient descent optimization was used to obtain local optimum, however, based on our experiments, it would require more training time.

Comparisons with State-of-the-art Cross-Modal Hashing Methods using Shared Binary Code as Gallery: To better show the advantage of learning shared binary codes, we also conducted experiments on the Wiki and MIRFlickr datasets where our CMDH was trained on the whole gallery set and the final shared binary codes were used for search during testing. We compared our CMDH with three other cross-modal hashing methods which also learn shared binary codes for retrieval: LSSH, CMFH, and SePH. To further improve the performance of different methods, we extracted convolutional neural networks (CNN) feature for each image instead of the conventional SIFT feature in the Wiki dataset. We used the pre-defined deep model in [64], particularly the CNN-F structure which consists of 5 conventional and 3 fully-connected layers, to represent each image to with a 4096-dimensional CNN feature and employed PCA to reduce it into 512 dimensions. Table 5 shows the mAP results for this experiment where the binary code length was set at 16, 32, 64 and 128, respectively. We see that the performance of all methods on the MIRFlickr dataset significantly improved because the shared binary codes were used for retrieval. Moreover, the performance is further improved on the Wiki dataset due to the use of CNN features. This also shows that our method is flexible to any type of feature used.

Comparisons with State-of-the-art Unsupervised Cross-Modal Hashing Methods: In Tables 1–5, we also include the performance of our unCMDH. As can be seen, it is competitive with other

Table 3
The mAPs of different cross-modal hashing methods on the Wiki and MIRFlickr25k dataset, where texts/tags were used as query samples and images were employed as gallery samples, respectively.

Method	Wiki				MIRFlickr25k			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [38]	0.2692	0.2297	0.2047	0.1821	0.6176	0.6121	0.6055	0.5985
CCA-ITQ [15]	0.3735	0.3627	0.3514	0.3328	0.6353	0.6303	0.6277	0.6256
PDH [37]	0.1937	0.1958	0.2020	0.2026	0.6152	0.6220	0.6301	0.6360
LSSH [26]	0.5318	0.5606	0.5662	0.5660	0.6462	0.6530	0.6560	0.6690
CMFH [25]	0.3463	0.3701	0.3815	0.3980	0.6336	0.6367	0.6362	0.6358
unCMDH - *ker*	0.3606	0.4132	0.4746	0.5046	0.7015	0.7082	0.7217	0.7216
unCMDH - *lin*	0.3519	0.3784	0.3926	0.4041	0.6742	0.6754	0.6745	0.6839
SCM - *orth* [32]	0.2593	0.2170	0.1898	0.1769	0.6747	0.6867	0.7024	0.7080
SCM - *seq* [32]	0.1454	0.1489	0.1465	0.1459	0.6587	0.6342	0.6158	0.6106
SePH - *rnd* [29]	0.6100	0.6411	0.6512	0.6619	0.7037	0.7220	0.7358	0.7451
SePH - *km* [29]	0.6053	0.6383	0.6483	0.6520	0.7052	0.7201	0.7358	0.7454
CMDH - *ker*	**0.6125**	**0.6438**	**0.6598**	**0.6642**	**0.7157**	**0.7273**	**0.7432**	**0.7551**
CMDH - *lin*	0.4133	0.4388	0.4576	0.4613	0.6953	0.6917	0.7001	0.7103

Table 4
The mAPs of different cross-modal hashing methods on the NUS-WIDE dataset where texts/tags were used as query samples and images were employed as gallery samples, respectively.

Method	16 bits	32 bits	64 bits	128 bits
CVH [38]	0.4161	0.4114	0.4079	0.4102
CCA-ITQ [15]	0.4201	0.4171	0.4164	0.4192
PDH [37]	0.4881	0.5120	0.5350	0.5451
LSSH [26]	0.4789	0.5078	0.5167	0.5156
CMFH [25]	0.4553	0.4683	0.4744	0.4769
unCMDH - *ker*	0.5166	0.5307	0.5764	0.5918
unCMDH - *lin*	0.4371	0.4271	0.4257	0.4309
SCM - *orth* [32]	0.5162	0.5444	0.5527	0.5621
SCM - *seq* [32]	0.4878	0.4702	0.4529	0.4429
SePH - *rnd* [29]	0.5777	0.5820	0.5931	0.6075
SePH - *km* [29]	0.5833	0.5812	**0.6191**	**0.6245**
CMDH - *ker*	**0.5885**	**0.5936**	0.6005	0.6142
CMDH - *lin*	0.5547	0.5588	0.5697	0.5805

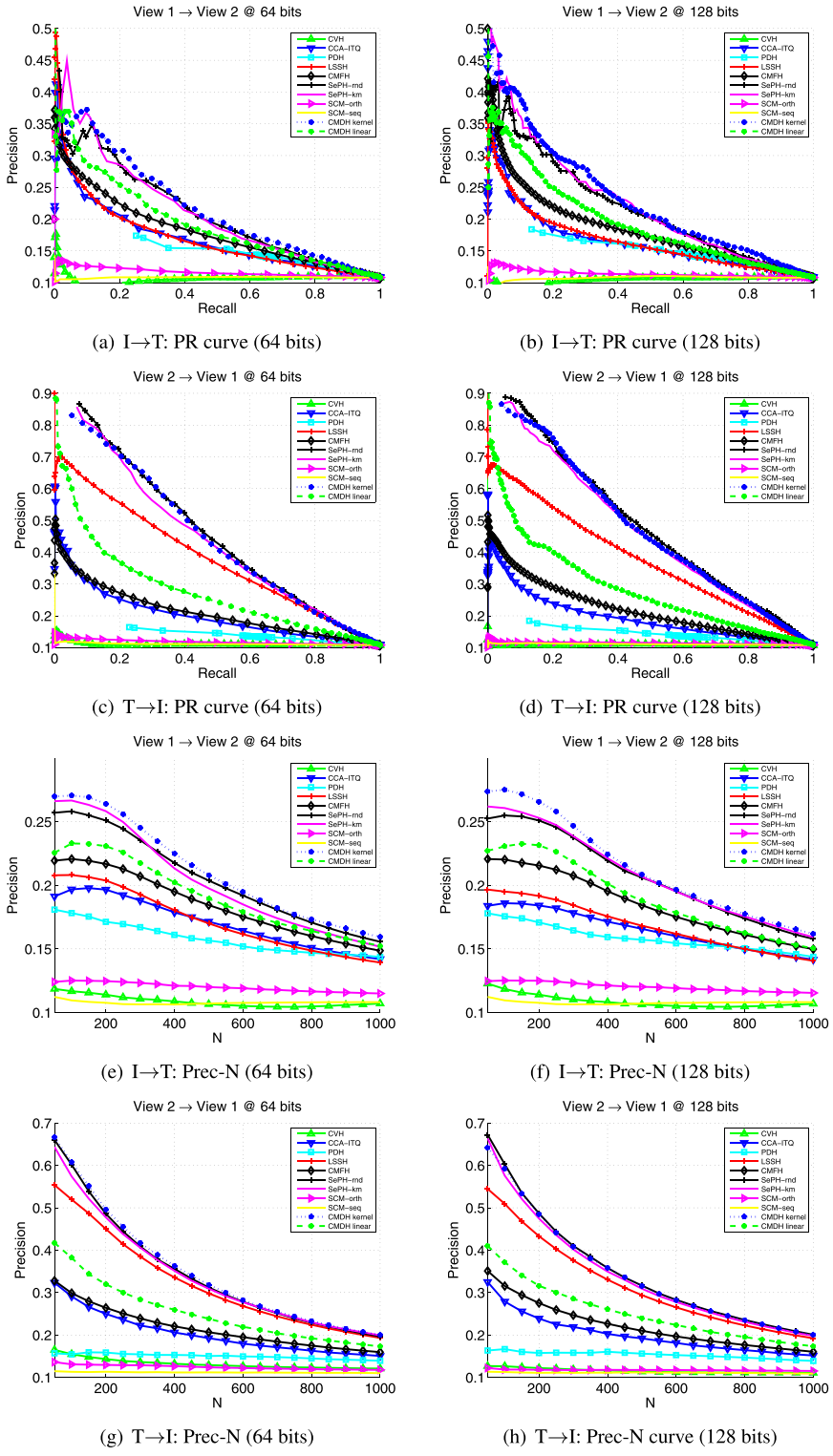
unsupervised methods particularly with the popular LSSH and CMFH method. This shows that our method is flexible with or without label information.

Comparisons with State-of-the-art Multi-Modal Hashing Methods: To show the advantage of our MMDH method, we conduct another experiment that uses a multi-modal set-up. We implement the Wiki experiment using three modalities. The different modalities used are the SIFT image features, CNN image features and Text features. Among the methods compared previously, only CMFH present a multi-modal extension which we call CMFH-MM. For our implementation we use the learned joint binary code to obtain similar gallery codes. Table 6 shows the mAP of our MMDH method compared to CMFH-MM. It can be seen that the performance is lower than that of Table 5. This is because it is more challenging to learn a unified binary code that would be consistent for the three modalities provided. Nevertheless, our MMDH method still achieves competitive or even better performance than CMFH for multi-modal experiments.

Similarly, we perform our multi-modal hashing method in the NUS-WIDE experiment. Differently, we use the out-of-sample extension approach to obtain the binary codes for the gallery set. Here, we use the block-wise color histogram (LAB) image feature as the additional modality. Table 7 shows the mAPs for the NUS-WIDE multi-modal experiments. While the LAB features to Text retrieval experiment gave lower retrieval performance in general, it probably may be due to the weak feature representation. Nevertheless, It can be seen that our method still outperforms CMFH-MM in all retrieval scenarios even in out-of-sample-extension setting. This may be because of our discrete binary code learning step and kernel representation, on top of our unified binary code learning. Interestingly, our unsupervised method also per-

Table 5
The mAPs of different cross-modal hashing methods on different datasets where shared binary codes were used as gallery samples.

	Method	Wiki				MIRFlickr25k			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Image to Text/Tag	LSSH [26]	0.3892	0.3866	0.3861	0.3806	0.6407	0.6450	0.6460	0.6547
	CMFH [25]	0.2742	0.2876	0.2960	0.3045	0.6408	0.6327	0.6306	0.6244
	unCMDH - *ker*	0.3725	0.3928	0.4062	0.4138	0.6981	0.6959	0.7029	0.7146
	unCMDH - *lin*	0.3238	0.3355	0.3517	0.3635	0.6820	0.6798	0.6842	0.6875
	SePH - *rnd* [29]	0.4336	0.4455	0.4620	0.4862	0.7189	0.7229	0.7291	0.7352
	SePH - *km* [29]	0.4482	0.4601	**0.4735**	0.4818	**0.7279**	**0.7283**	**0.7333**	0.7325
	CMDH - *ker*	**0.4610**	**0.4651**	0.4729	**0.4869**	0.7238	0.7203	0.7312	**0.7316**
	CMDH - *lin*	0.4364	0.4479	0.4401	0.4573	0.6856	0.6844	0.6833	0.6818
Text/Tag to Image	LSSH [26]	0.6119	0.6427	0.6560	0.6840	0.7033	0.7268	0.7352	0.7405
	CMFH [25]	0.4004	0.4212	0.4582	0.4833	0.7002	0.7191	0.7387	0.7597
	unCMDH - *ker*	0.5776	0.6341	0.6376	0.6500	0.7563	0.7611	0.7854	0.7848
	unCMDH - *lin*	0.6060	0.6160	0.6219	0.6401	0.7389	0.7582	0.7653	0.7825
	SePH - *rnd* [29]	0.7166	0.7192	0.7252	0.7281	0.7945	0.8047	0.8194	0.8275
	SePH - *km* [29]	0.7081	0.7225	0.7269	0.7296	0.8074	0.8086	0.8226	0.8344
	CMDH - *ker*	**0.7427**	**0.7416**	**0.7494**	**0.7502**	**0.8102**	**0.8185**	**0.8364**	**0.8390**
	CMDH - *lin*	0.6982	0.7001	0.7040	0.7102	0.7761	0.7903	0.8010	0.8032



(a) I→T: PR curve (64 bits)

(b) I→T: PR curve (128 bits)

(c) T→I: PR curve (64 bits)

(d) T→I: PR curve (128 bits)

(e) I→T: Prec-N (64 bits)

(f) I→T: Prec-N (128 bits)

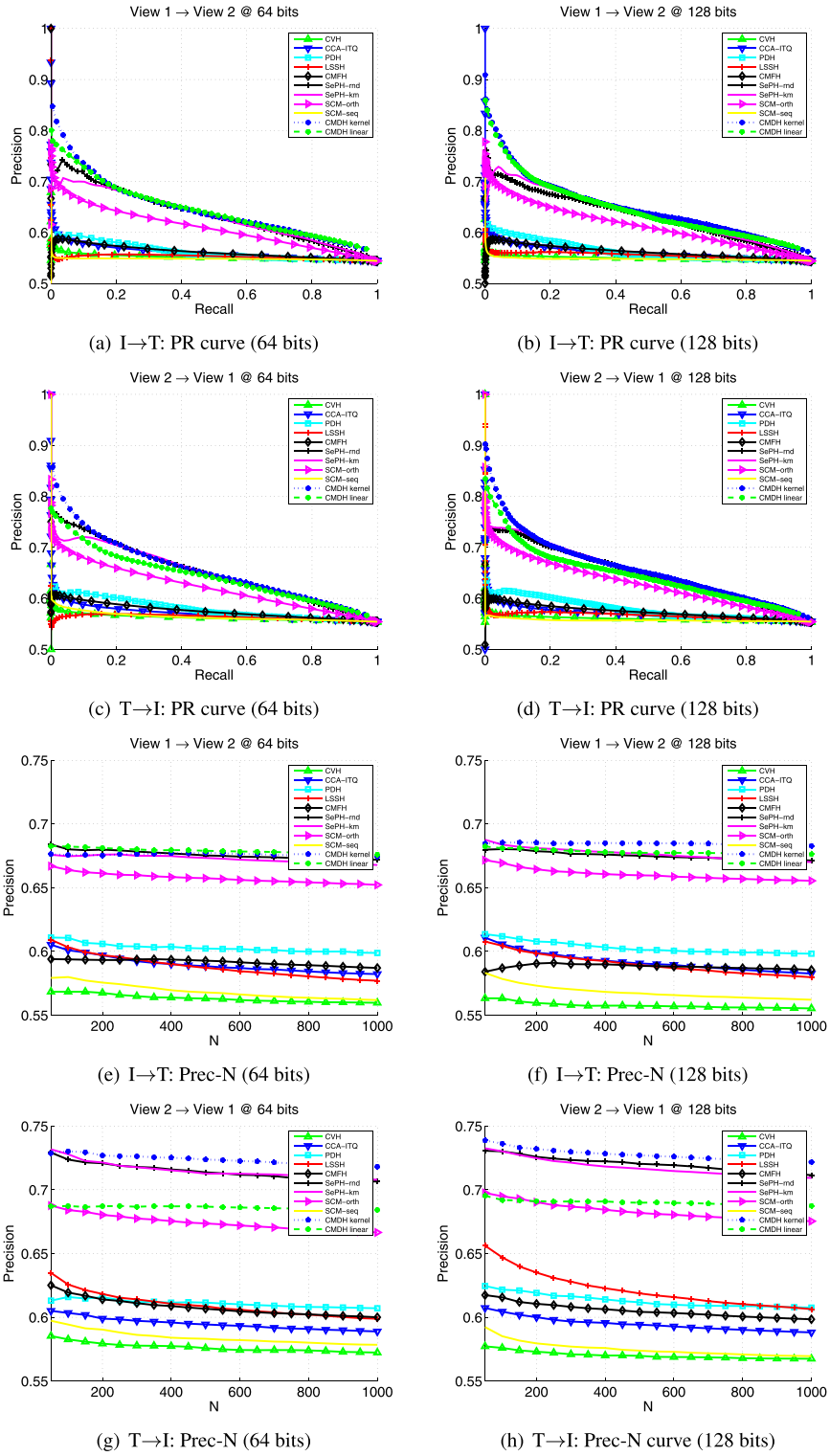
(g) T→I: Prec-N (64 bits)

(h) T→I: Prec-N curve (128 bits)

Fig. 3. Precision-Recall (PR) curves and Precision-N curves on the Wiki dataset of varying code lengths for Image-to-Text and Text-to-Image tasks.

forms better than CMFH-MM which shows that our affinity matrix estimation based on neighbourhood structure and anchor graphs is effective.

Comparisons of Different Learning Strategies: To further show the advantage of the discrete optimization and joint binary codes learning strategy of our CMDH, we constructed two other baselines: relaxed cross modal hashing (RCMH) and individual cross modal discrete hashing (ICMDH), which learn binary codes with a relaxation and for each modality individually, respectively. For RCMH, we solve (1) using a relaxed optimization similar to [15,16,29] where we removed the sign function and used the signed magnitude such that $J = -\text{tr}(\mathbf{Y}^{u\top}\mathbf{A}\mathbf{Y}^{u} + \mathbf{Y}^{v\top}\mathbf{A}\mathbf{Y}^{v}) + \|\mathbf{B} - \mathbf{Y}^{u}\|_F^2 + \|\mathbf{B} - \mathbf{Y}^{v}\|_F^2$ which can be solved using a gradient descent method. For ICMDH, we learned individual binary codes for each



(a) I→T: PR curve (64 bits)

(b) I→T: PR curve (128 bits)

(c) T→I: PR curve (64 bits)

(d) T→I: PR curve (128 bits)

(e) I→T: Prec-N (64 bits)

(f) I→T: Prec-N (128 bits)

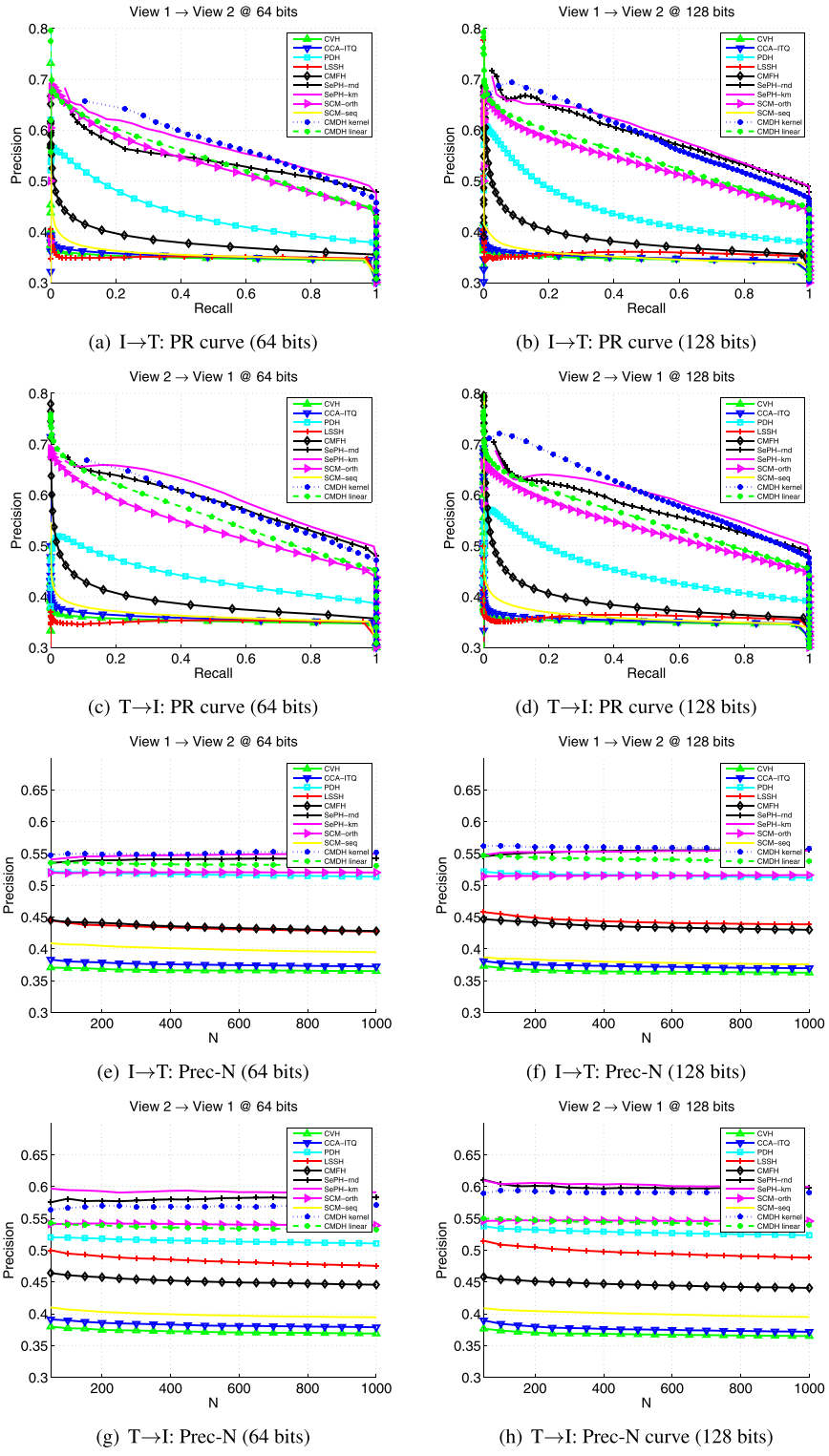
(g) T→I: Prec-N (64 bits)

(h) T→I: Prec-N curve (128 bits)

Fig. 4. Precision-Recall (PR) curves and Precision-N curves on the MIRFlickr dataset of varying code lengths for Image-to-Text and Text-to-Image tasks.

modality such that the formulation becomes $J = \sum_{*} -\text{tr}(\mathbf{B}^{*\top}\mathbf{A}\mathbf{B}^{*}) + \|\mathbf{B}^{*} - \mathbf{Y}^{*}\|_F^2$, which can be solved similar to our iterative discrete optimization. We repeat the first experiment on the Wiki and MIR-Flickr datasets for RCMH and ICMDH. Table 8 shows the average mAP (from mAPs obtained in image-text and text-image retrieval experiments) of our CMDH and the other two baseline methods. It can be seen that our CMDH outperforms both RCMH and ICMDH, which indicates that both the discrete optimization technique and the unified binary codes learning strategy are useful to improve the retrieval performance in our CMDH method.

Parameter Sensitivity: We investigated the sensitivity of our optimization method with respect to the balancing parameter η. We performed experiments at varying values of $\eta = [0.2, 0.4, 0.5, 0.6, 0.8, 1]$ when the code length was set as 16, 32, 64



(a) I→T: PR curve (64 bits)
(b) I→T: PR curve (128 bits)
(c) T→I: PR curve (64 bits)
(d) T→I: PR curve (128 bits)
(e) I→T: Prec-N (64 bits)
(f) I→T: Prec-N (128 bits)
(g) T→I: Prec-N (64 bits)
(h) T→I: Prec-N curve (128 bits)

Fig. 5. Precision-Recall (PR) curves and Precision-N curves on the NUS-WIDE dataset of varying code lengths for Image-to-Text and Text-to-Image tasks.

and 128, respectively. Fig. 6(a)–(c) show the mAP of our method versus different length of the binary codes. As can be seen, the performance is stable when η is set as in the range of [0.4, 0.6].

Convergence: We evaluated the convergence of our optimization procedure for both CMDH-kernel and CMDH-linear. Fig. 8(a)–(f) show the value of the objective function during training on the NUS-WIDE dataset for various bit sizes. We see that CMDH-kernel can converge with 20–30 iterations, and CMDH-linear can converge around 100–150 iterations. We find that a linear model takes a longer time in obtaining an optimum solution than the kernel mapping.

Computational Complexity: We compared the train and test time for different cross-modal hashing methods. The test time represents the time it took to hash in an out-of-sample extension

Table 6
The mAPs of different multi-modal hashing methods on the Wiki dataset where shared binary codes were used as gallery samples.

	Method	16 bits	32 bits	64 bits	128 bits
Image-SIFT feature	CMFH-MM [25]	0.1992	0.2030	0.2090	0.2123
	unMMDH-ker	0.2258	0.2510	0.2577	0.2624
	unMMDH-lin	0.2326	0.2377	0.2528	0.2499
	MMDH-ker	**0.2437**	**0.2529**	**0.2668**	**0.2651**
	MMDH-lin	0.2154	0.2407	0.2422	0.2486
Text Feature	CMFH-MM [25]	0.3802	0.4034	0.4376	0.4671
	unMMDH-ker	0.4239	0.5229	0.5626	0.5870
	unMMDH-lin	0.4678	0.5283	0.5735	0.5785
	MMDH-ker	**0.6634**	**0.6755**	**0.6894**	**0.7026**
	MMDH-lin	0.6544	0.6677	0.6746	0.6797
Image-CNN feature	CMFH-MM [25]	0.2573	0.2697	0.2761	0.2828
	unMMDH-ker	0.3035	0.3516	0.3733	0.3898
	unMMDH-lin	0.2745	0.3228	0.3548	0.3661
	MMDH-ker	**0.3704**	**0.3867**	**0.4005**	**0.4058**
	MMDH-lin	0.3050	0.3297	0.3286	0.3330

Table 7
The mAPs of different multi-modal hashing methods on the NUS-WIDE dataset where out-of-sample binary codes were used as gallery samples.

	Method	16 bits	32 bits	64 bits	128 bits
Image SIFT-BoW feature → Text Feature	CMFH-MM [25]	0.4327	0.4582	0.4728	0.4730
	unMMDH-ker	0.4897	0.5048	0.5252	0.5428
	unMMDH-lin	0.3853	0.4078	0.42336	0.4308
	MMDH-ker	**0.5435**	**0.5542**	**0.5690**	**0.5783**
	MMDH-lin	0.4868	0.5172	0.5148	0.5443
Text Feature → Image SIFT-BoW feature	CMFH-MM [25]	0.4495	0.4668	0.4690	0.4687
	unMMDH-ker	0.4824	0.4995	0.5233	0.5483
	unMMDH-lin	0.3818	0.3931	0.4070	0.4229
	MMDH-ker	**0.5594**	**0.5821**	**0.6037**	**0.6081**
	MMDH-lin	0.4680	0.5025	0.5033	0.5369
Image block-wise LAB feature → Text Feature	CMFH-MM [25]	0.3642	0.3710	0.3759	0.3821
	unMMDH-ker	0.3685	0.3795	0.3751	0.3820
	unMMDH-lin	0.3673	0.3725	0.3787	0.3713
	MMDH-ker	0.3676	0.3656	0.3780	0.3728
	MMDH-lin	**0.3746**	**0.3806**	**0.3839**	**0.3876**

Table 8
The average mAPs of our CMDH on Wiki and MIRFlickr datasets when different learning strategies were employed.

Method	Wiki		MIRFlickr25k	
	64 bits	128 bits	64 bits	128 bits
CMDH-*ker*	**0.4768**	**0.4812**	**0.7279**	**0.7387**
RCMH-*ker*	0.2976	0.3148	0.5857	0.6086
ICMDH-*ker*	0.3575	0.3664	0.6384	0.6446
CMDH-*lin*	**0.3638**	**0.3672**	**0.6934**	**0.7026**
RCMH-*lin*	0.3245	0.3223	0.6283	0.6268
ICMDH-*lin*	0.3490	0.3567	0.6625	0.6591

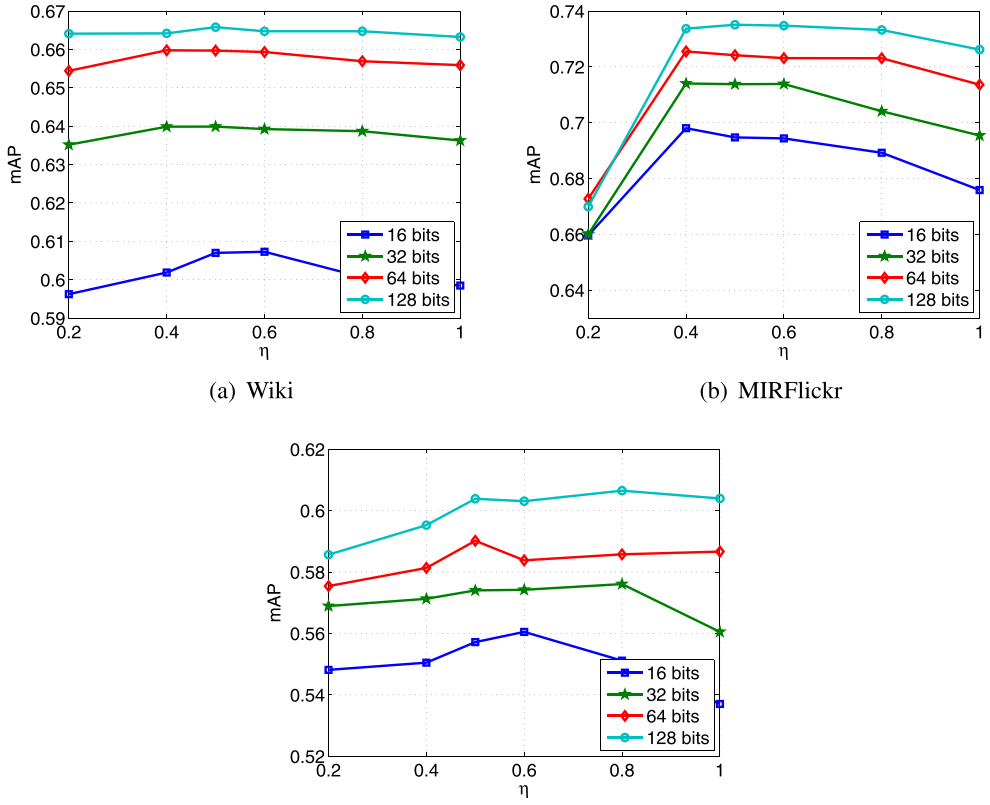
Table 9
Computational time (s) of different cross-modal hashing methods on the Wiki and NUS-WIDE datasets for code length of 16 bits.

Method	Wiki		NUS-WIDE	
	Train	Search	Train	Search
CVH	0.129	0.0018	1.103	0.639
CCA-ITQ	0.198	0.0049	1.345	2.218
PDH	0.746	0.0044	16.77	1.644
LSSH	44.36	3.356	106.2	5.8e3
CMFH	0.327	0.0122	344.2	1.651
SCM-orth	0.071	0.0037	15.42	1.424
SCM-seq	0.035	0.0037	0.953	1.231
SePH-rnd	211.8	0.242	1.2e3	3.214
SePH-km	224.2	0.285	1.2e3	3.541
CMDH-ker	2.124	0.042	34.95	2.881
CMDH-lin	1.557	0.004	109.67	2.494

implementation for all query and gallery samples. Our hardware consists of a PC configuration of 3.20GHz i5-3470 CPU and 32.0GB RAM. Table 9 shows the time complexity for training and testing on the Wiki dataset which has 2150 training instances and on the NUS-WIDE dataset which contains 5000 training instances. As shown, SePH requires the largest time for training despite its strong performance. This is mainly due to the complex optimization of minimizing the Kullback–Leibler divergence. On the other hand, our CMDH presents an acceptable speed with better performance than existing cross-modal hashing method. Since our optimization is iterative and straightforward, we can easily add more training samples without sacrificing computational time as shown in Fig. 7.

4.3. Discussion

The above experimental results suggest the following three key observations:



(a) Wiki (b) MIRFlickr

(c) NUS-WIDE

Fig. 6. The mAP of our CMDH versus different values of η on the Wiki, MIRFlickr, and NUS-WIDE datasets.

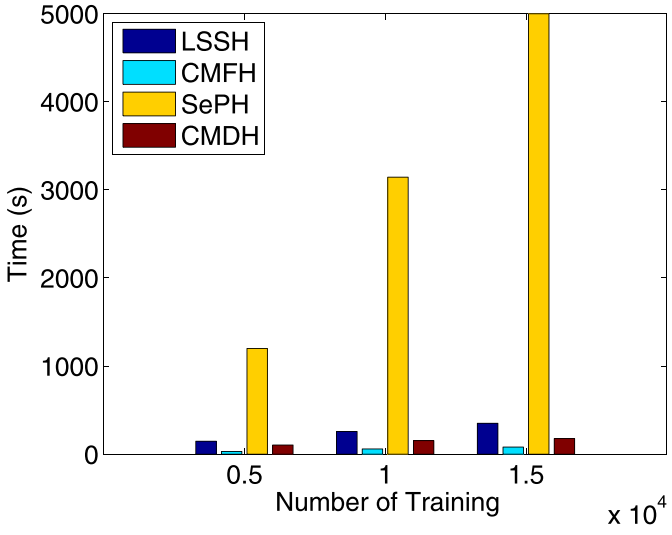


Fig. 7. Training time of different cross-modal hashing methods versus different sizes of the training set.

1. Our discrete cross-modal hashing approach achieves very competitive or even better performance with the state-of-the-art cross-modal hashing methods on different benchmark datasets. This is because we performed an optimization step that avoids relaxation in the binary constraints which lessens the information loss during hash function learning. Moreover, we learned a joint binary code to implicitly reduced the modality gap between the cross-modal data.
2. Our unCMDH and MMDH methods also show competitive performance, which shows the flexibility of our approach in addressing different scenarios in cross-modal hashing.
3. Our discrete cross-modal hashing approach is not only competitive in retrieval performance but also yields an acceptable training speed compared to the state-of-the-art SePH and LSSH methods, in whiche these obtain good mAP performance but require significant time for training.

5. Conclusion

In this paper, we have proposed a cross-modal discrete hashing (CMDH) method to learn compact binary codes for cross-modality retrieval. Specifically, we have presented two forms of hashing algorithms called CMDH-linear and CMDH-kernel under the proposed framework, which perform linear and non-linear mappings to learn binary codes, respectively. Furthermore, we have extended our CMDH method to unsupervised CMDH (unCMDH) and discrete multi-modal hashing (MMDH) to make it suitable for unsupervised hashing and multi-modal hashing. Experimental results on three widely used cross-modal datasets have clearly showed the effectiveness of our proposed approach. How to develop more non-linear mapping models such as boosting or a deep neural network seems an interesting future work Figs. 3–5.

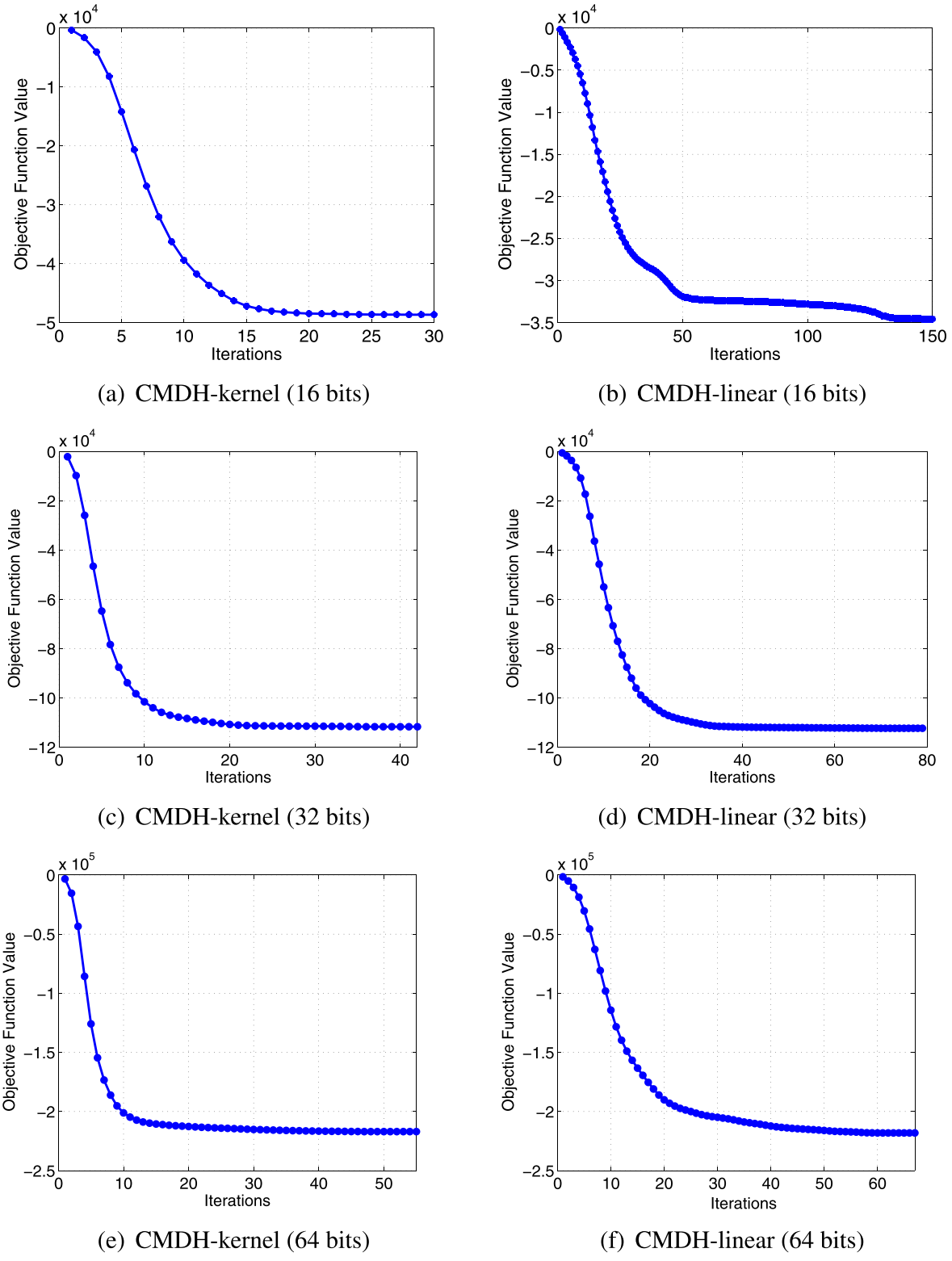


Fig. 8. Objective function values of our CMDH-kernel and CMDH-linear on the NUS-WIDE dataset when the code length is set as 16, 32 and 64.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61672306. This work was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University Singapore. The ROSE Lab is supported by the Infocomm Media Development Authority, Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

References

[1] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: NIPS, 2009, pp. 1042–1050.

[2] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, in: TPAMI, 34, 2012, pp. 1704–1716.
[3] A. Torralba, R. Fergus, Y. Weiss, Small codes and large image databases for recognition, in: CVPR, 2008, pp. 1–8.
[4] A. Torralba, R. Fergus, W.T. Freeman, 80 Million tiny images: a large data set for nonparametric object and scene recognition, in: TPAMI, 30, 2008, pp. 1958–1970.
[5] T. Yao, Z. Wang, Z. Xie, J. Gao, D.D. Feng, Learning universal multiview dictionary for human action recognition, Pattern Recognit. 64 (2017) 236–244.
[6] X. Li, M. Fang, J.-J. Zhang, J. Wu, Learning coupled classifiers with rgb images for rgb-d object recognition, Pattern Recognit. 61 (2017) 433–446.
[7] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, in: TPAMI, 33, 2011, pp. 117–128.
[8] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: ACM on Computational Geometry, 2004, pp. 253–262.
[9] X. Liu, Z. Li, C. Deng, D. Tao, Distributed adaptive binary quantization for fast nearest neighbor search, TIP, 2017.
[10] X. Liu, B. Du, C. Deng, M. Liu, B. Lang, Structure sensitive hashing with adaptive product quantization, in: TSCVT, 46, 2016, pp. 2252–2264.
[11] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: ACM Symposium on Theory of Computing, 1998, pp. 604–613.
[12] C. Silpa-Anan, R. Hartley, Optimised kd-trees for fast image descriptor matching, in: CVPR, 2008, pp. 1–8.
[13] M. Muja, D.G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, in: TPAMI, 36, 2014, pp. 2227–2240.
[14] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration., VISAPP, 2, 2009.
[15] Y. Gong, S. Lazebnik, Iterative quantization: A procrustean approach to learning binary codes, in: CVPR, 2011, pp. 817–824.
[16] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for large-scale search, in: TPAMI, 34, 2012, pp. 2393–2406.
[17] V.E. Liong, J. Lu, Y.-P. Tan, J. Zhou, Deep video hashing, in: TMM, 19, 2017, pp. 1209–1219.
[18] T. Song, J. Cai, T. Zhang, C. Gao, F. Meng, Q. Wu, Semi-supervised manifold-embedded hashing with joint feature representation and classifier learning, Pattern Recognit. 68 (2017) 99–110.
[19] X. Bai, C. Yan, H. Yang, L. Bai, J. Zhou, E.R. Hancock, Adaptive hash retrieval with kernel based similarity, 2017.
[20] J. Tang, Z. Li, X. Zhu, Supervised deep hashing for scalable face image retrieval, 2017.
[21] J. Song, L. Gao, L. Liu, X. Zhu, N. Sebe, Quantization-based hashing: a general framework for scalable image and video retrieval, 2017.
[22] Z. Chen, J. Zhou, Collaborative multiview hashing, 2017.
[23] D. Zhai, X. Liu, H. Chang, Y. Zhen, X. Chen, M. Guo, W. Gao, Parametric local multiview hamming distance metric learning, 2017.
[24] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: NIPS, 2008, pp. 1753–1760.
[25] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: CVPR, 2014, pp. 2083–2090.
[26] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, Q. Tian, Towards codebook-free: scalable cascaded hashing for mobile image search, in: TMM, 16, 2014, pp. 601–611.
[27] X. Zhu, Z. Huang, H.T. Shen, X. Zhao, Linear cross-modal hashing for efficient multimedia search, in: ACM MM, 2013, pp. 143–152.
[28] M. Kafai, K. Eshghi, B. Bhanu, Discrete cosine transform locality-sensitive hashes for face retrieval, in: TMM, 16, 2014, pp. 1090–1103.
[29] Z. Lin, G. Ding, M. Hu, J. Wang, Semantics-preserving hashing for cross-view retrieval, in: CVPR, 2015, pp. 3864–3872.
[30] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, J. Wang, Quantized correlation hashing for fast cross-modal search, in: IJCAI, 2015, pp. 25–31.
[31] J. Zhou, G. Ding, Y. Guo, Q. Liu, X. Dong, Kernel-based supervised hashing for cross-view similarity search, in: ICME, 2014, pp. 1–6.
[32] D. Zhang, W.-J. Li, Large-scale supervised multimodal hashing with semantic correlation maximization, in: AAAI, 2014, pp. 2177–2183.
[33] J. Masci, M.M. Bronstein, A.M. Bronstein, J. Schmidhuber, Multimodal similarity-preserving hashing, in: TPAMI, 36, 2014, pp. 824–830.
[34] F. Shen, C. Shen, W. Liu, H. Shen, Supervised discrete hashing, in: CVPR, 2015, pp. 37–45.
[35] W. Liu, C. Mu, S. Kumar, S.-F. Chang, Discrete graph hashing, in: NIPS, 2014, pp. 3419–3427.
[36] M.M. Bronstein, A.M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in: CVPR, 2010, pp. 3594–3601.
[37] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, L. Davis, Predictable dual-view hashing, in: ICML, 2013, pp. 1328–1336.
[38] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: IJCAI, 22, 2011, pp. 1360–1365.
[39] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: FOCS, 2006, pp. 459–468.
[40] P. Jain, B. Kulis, K. Grauman, Fast image search for learned metrics, in: CVPR, 2008, pp. 1–8.
[41] G. Lin, C. Shen, A. van den Hengel, Supervised hashing using graph cuts and boosted decision trees, in: TPAMI, 37, 2015, pp. 2317–2331.
[42] M.A. Carreira-Perpinán, R. Raziperchikolaei, Hashing with binary autoencoders, in: CVPR, 2015, pp. 557–566.
[43] X.-J. Mao, Y.-B. Yang, N. Li, Hashing with pairwise correlation learning and reconstruction, in: TMM, 19, 2017, pp. 382–392.
[44] C. Leng, J. Wu, J. Cheng, X. Bai, H. Lu, Online sketching hashing, in: CVPR, 2015, pp. 2503–2511.
[45] X. Bai, H. Yang, J. Zhou, P. Ren, J. Cheng, Data-dependent hashing based on p-stable distribution, IEEE Trans. Image Process. 23 (12) (2014) 5033–5046.
[46] Y. Lv, W.W. Ng, Z. Zeng, D.S. Yeung, P.P. Chan, Asymmetric cyclical hashing for large scale image retrieval, in: TMM, 17, 2015, pp. 1225–1235.
[47] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: ICCV, 2009, pp. 2130–2137.
[48] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, in: NIPS, 2009, pp. 1509–1517.
[49] V.E. Liong, J. Lu, G. Wang, P. Moulin, J. Zhou, Deep hashing for compact binary codes, in: CVPR, 2015, pp. 2475–2483.
[50] F. Zhao, Y. Huang, L. Wang, T. Tan, Deep semantic ranking based hashing for multi-label image retrieval, in: CVPR, 2015, pp. 1556–1564.
[51] R. Xia, Y. Pan, H. Lai, C. Liu, S. Yan, Supervised hashing for image retrieval via image representation learning, in: AAAI, 2014, pp. 2156–2162.
[52] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: ACMM, 2010, pp. 251–260.
[53] K. Wang, R. He, W. Wang, L. Wang, T. Tan, Learning coupled feature spaces for cross-modal matching, in: ICCV, 2013, pp. 2088–2095.
[54] L. Ma, Z. Lu, L. Shang, H. Li, Multimodal convolutional neural networks for matching image and sentence, 2015, pp. 2623–2631.
[55] F. Yan, K. Mikolajczyk, Deep correlation for matching images and text, in: CVPR, 2015, pp. 3441–3450.
[56] F. Feng, X. Wang, R. Li, Cross-modal retrieval with correspondence autoencoder, in: ACM MM, 2014, pp. 7–16.
[57] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G.R. Lanckriet, R. Levy, N. Vasconcelos, On the role of correlation and abstraction in cross-modal multimedia retrieval, in: TPAMI, 36, 2014, pp. 521–535.
[58] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, X. Gao, Pairwise relationship guided deep hashing for cross-modal retrieval., in: AAAI, 2017, pp. 1618–1625.
[59] X. Liu, J. He, C. Deng, B. Lang, Collaborative hashing, in: CVPR, 2014, pp. 2139–2146.
[60] K. Ding, B. Fan, C. Huo, S. Xiang, C. Pan, Cross-modal hashing via rank-order preserving, in: TMM, 19, 2017, pp. 571–585.
[61] W. Liu, J. Wang, S. Kumar, S.-F. Chang, Hashing with graphs, in: ICML, 2011, pp. 1–8.
[62] M.J. Huiskes, M.S. Lew, The mir flickr retrieval evaluation, in: ACM MIR, 2008, pp. 39–43.
[63] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, Nus-wide: A real-world web image database from national university of singapore, in: ACM CIVR, 2008, pp. 48–57.
[64] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, BMVC, 2014.

Venice Erin Liong received the B.S. degree from the University of the Philippines Diliman, Quezon City, Philippines in 2010, and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon City, South Korea, in 2013. She is currently pursuing a Ph.D. degree in the Interdisciplinary Graduate School, Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore. Her research interests include computer vision, biometrics and pattern recognition.

Jiwen Lu received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, and the Ph.D. degree in electrical engineering from the Nanyang Technological University, Singapore, in 2003, 2006, and 2012, respectively. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. From March 2011 to November 2015, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. His current research interests include computer vision, pattern recognition, and machine learning. He has authored/co-authored over 180 scientific papers in these areas, where more than 80 papers are published in the IEEE Transactions journals and top-tier computer vision conferences such as CVPR, ICCV, ECCV and NIPS. He is an elected member of the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, an elected member of the Multimedia Systems and Applications Technical Committee of the Circuits and Systems Society. He serves as an Associate Editor of the IEEE Trans. on Circuits and Systems for Video Technology, Pattern Recognition, and Journal of Visual Communication and Image Representation, and served as an Associate Editor for Pattern Recognition, Neurocomputing, and the IEEE Access. He is/was a Workshop Chair/Special Session Chair/Area Chair for more than 10 international conferences. He was a recipient of the National 1000 Young Talents Plan Program of China in 2015. He is a senior member of the IEEE.

Yap-Peng Tan received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1993, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1995 and 1997, respectively, all in electrical engineering. From 1997 to 1999, he was with Intel Corporation, Chandler, AZ, and Sharp Laboratories of America, Camas, WA. In November 1999, he joined the Nanyang Technological University of Singapore, where he is currently Professor and Associate Chair (Academic) of the School of Electrical and Electronic Engineering. He is the principal inventor or co-inventor on 15 U.S. patents in the areas of image and video processing. His current research interests include image and video processing, content-based multimedia analysis, computer vision, pattern recognition, and data analytics. He served as the Chair of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society from 2012 to 2014, a member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society from 2009 to 2013, and a voting member of the IEEE International Conference on Multimedia & Expo (ICME) Steering Committee from 2011 to 2012, and Chairman of the IEEE Signal Processing Singapore Chapter from 2009 to 2010. He has also served as Associate Editor of the IEEE Signal Processing Letters (since 2016), IEEE Transactions on Multimedia (since 2014) and the IEEE Access (since 2013), an Editorial Board Member of the EURASIP Journal on Advances in Signal Processing and EURASIP Journal on Image and Video Processing, Guest Editor for special issues of several journals including the IEEE Transactions on Multimedia, and a member of the Multimedia Systems and Applications Technical Committee (MSA TC) and Visual Signal Processing and Communications Technical Committee (VSPC TC) of the IEEE Circuits and Systems Society. He is the Tutorial Co-Chair of the 2016 IEEE International Conference on Multimedia and Expo (ICME 2016) and Technical Program Co-Chair of the 2019 IEEE International Conference on Image Processing (ICIP 2019), and was the Finance Chair of the 2004 IEEE International Conference on Image Processing (ICIP 2004), General Co-Chair of the 2010 IEEE International Conference on Multimedia and Expo (ICME 2010), Technical Program Co-Chair of the 2015 IEEE International Conference on Multimedia and Expo (ICME 2015), and General Co-Chair of the 2015 IEEE International Conference on Visual Communications and Image Processing (VCIP 2015).