Frustratingly Easy Cross-Modal Hashing

Dekui Ma[†], Jian Liang[§,¶,*], Xiangwei Kong[†], Ran He[§,‡,¶]
[†] School of Information and Communication Engineering, Dalian University of Technology
[§] Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR), CASIA
[‡] CAS Center for Excellence in Brain Science and Intelligence Technology (CEBSIT)
[¶] University of Chinese Academy of Sciences (UCAS)
madk@mail.dlut.edu.cn, {jian.liang, rhe}@nlpr.ia.ac.cn, kongxw@dlut.edu.cn

ABSTRACT

Cross-modal hashing has attracted considerable attention due to its low storage cost and fast retrieval speed. Recently, more and more sophisticated researches related to this topic are proposed. However, they seem to be inefficient computationally for several reasons. On one hand, learning coupled hash projections makes the iterative optimization problem challenging. On the other hand, individual collective binary codes for each content are also learned with a high computation complexity. In this paper we describe a *simple yet effective* cross-modal hashing approach that can be implemented in just three lines of code. This approach first obtains the binary codes for one modality via unimodal hashing methods (e.g., iterative quantization (ITQ)), then applies simple linear regression to project the other modalities into the obtained binary subspace. Obviously, it is non-iterative and parameter-free, which makes it more attractive for many real-world applications. We further compare our approach with other state-of-the-art methods on four benchmark datasets (i.e., the Wiki, VOC, LabelMe and NUS-WIDE datasets). Despite its extraordinary simplicity, our approach performs remarkably and generally well for these datasets under different experimental settings (i.e., large-scale, high-dimensional and multi-label datasets).

CCS Concepts

•**Information systems** → **Top-k retrieval in databases;** *Novelty in information retrieval; Similarity measures;* Web indexing;

Keywords

cross-modal hashing; cross-media retrieval; image and text; double alignment

1. INTRODUCTION

With the fast development of the Internet, multi-modal data posted on the websites (e.g., Twitter and Facebook) are emerging, which makes retrieving heterogeneous content become more and more significant. Over the last decade, numerous cross-modal retrieval approaches [2, 31, 9] have been proposed, several unsupervised methods [10, 11] are also proposed. [15] first proposed a cross-modal retrieval method based on classification models. [14] further proposed a cross-modal learning method based pairwise classification, and obtained better retrieval performance. Recently owing to the benefits of lower storage costs and higher query speeds, hashing methods [25, 21, 27, 5] have gained even more popularity over traditional cross-modal retrieval methods. Since the text (i.e., articles, textual descriptions and tags) and image modality are even common in real-world web applications, we mainly focus on the text-image cross-modal retrieval in this work.

Cross-modal hashing (CMH) methods aim to map heterogeneous data into the common low-dimensional hamming space, where similarities among both intra- and inter- modalities are preserved. For semantic-preserving hashing methods, the heterogeneous data sharing the identical labels are required to be close to each other in the hamming space. The key issue for CMH is how to exploit the relationship of heterogeneous data efficiently for obtaining the hash projections.

Depending on whether the semantic labels for the observations are utilized or not, existing CMH methods can be roughly divided into two categories: supervised [32, 26, 13] and unsupervised [29, 3, 19] methods. Among the unsupervised ones, [8] extended [24] to the multimodal setting through minimizing the weighted distance, while [3] utilized collective matrix factorization from different modalities of one instance to obtain the hash functions with latent factor model. Besides, [29] captured the salient structures of images and learns latent concepts from texts through using sparse coding and matrix factorization respectively.

Supervised CMH methods make full use of provided semantic labels to learn discriminative hash functions via some other criterion like label-similarity preserving [6]. [28] tried to maximize the semantic correlation and learn the hash functions greedily. [13] made full use of the semantic similarity matrix to obtain the optimal binary codes for each observation. [1] was proposed to embed data from different feature space into a common metric space.

Furthermore, CMH methods can be optimized via two solutions, i.e., iterative and two-step optimization methods. When the former ones optimize the overall objective function iteratively, the two-step solutions first obtain an optimal binary code for each pair, then minimize the objective w.r.t. merely hash functions. Regarding of the time complexity, the iterative solution needs to optimize the binary codes and hash functions simultaneously, resulting in a higher time complexity. The two-step framework seems simple, however, it takes too much time in finding optimal binary codes for multiple modalities.

[*] The corresponding author; the first two authors contributed equally to this paper.

MM '16, October 15–19, 2016, Amsterdam, The Netherlands.
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2964284.2967218

Intuitively, unimodal hashing consists of two procedures, 1) finding the corresponding holes (binary codes) in hamming space, 2) solving the parameters of assumed hash functions which map origin features into known holes. The key issue for two-step CMH methods is how to obtain the code for each multimodal content efficiently. Here we propose a novel strategy double alignment based hashing (DASH) for the trivial two-step framework. The code generated by one single modality is reused for the other modality. In that way, we avoid the huge time-complexity and pass the semantic information from generated codes instead of explicit semantic labels, which proves effective for cross-modal learning. Extensive experimental results under large-scale, high-dimensional, multi-label settings are shown in the experiment section to validate the effectiveness and efficiency of our DASH method.

2. CORRELATION ALIGNMENT CROSS-MODAL HASHING

In this section, we first introduce some preliminary knowledge such as notations, terms and definition of hash function. Then we provide some motivations behind our proposed DASH and implementation details. Without loss of generality, we consider bimodal case (i.e., image-tag or image-text) for each instance, which is easy to present and understand.

2.1 Notations and Problem Definition

Supposing that we have n observations described in two modalities $X^{(m)} \in \mathbf{R}^{d_m \times n}$, $x_i^{(m)}$ denotes the i-th observation in the m-th modality, and d_m is the dimensionality of the m-th modality. Moreover, we also have *semantic labels* $y_i \in \{0, 1\}^k$ for each observation $x_i = [x_i^{(1)}, x_i^{(2)}]$, $i \in [1, n]$, where k is the amount of semantic categories, and $y_{i,j} = 1$ denotes that the i-th observation belongs to the j-th category. Note that each observation is not limited to one semantic category (e.g., on multi-label datasets).

For each matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, its i-th row, j-th column are denoted by $\mathbf{m}^i$, $\mathbf{m}_j$ respectively, and $M_{i,j}$ lies in the i-th row and j-th column. The Frobenius norm of any matrix $\mathbf{M}$ is defined as $||\mathbf{M}||_F = \sqrt{\sum_{i=1}^n ||\mathbf{m}^i||_2^2}$, and the trace of the square matrix M is defined as $Tr(\mathbf{M}) = \sum_i \mathbf{M}_{i,i}$. Moreover, M^T denotes the transpose of a vector or matrix M.

The goal of CMH is to learn two hash functions $\{f_m(\cdot)\}_{m=1}^2$ for each modality. The functions $f_m(\cdot)$ are further defined as below: $f_m(x_i^{(m)}) = sgn(W_m^T x_i^{(m)})$, $sgn(\cdot)$ denotes the element-wise sign function, and $W_m \in \mathbf{R}^{d_m \times c}$ is the learned projection matrix, where c is the length of binary codes. Without loss of generality, we assume that the data points are all zero-centered, i.e., $\sum_{i=1}^n x_i^{(1)} = \mathbf{0}, \; \sum_{i=1}^n x_i^{(2)} = \mathbf{0}$.

2.2 Motivation and Framework

Recently, SePH [13] exploits Kullback-Leibler divergence with the semantic label similarities matrix to seek optimal binary codes for multiple observations, then applies logistic regression to obtain corresponding hash projections. This method resembles the framework of two-step unimodal hashing in [12], which also first learns the optimal binary code via semantic or locality preserving criterion, followed by learning hash functions via boosting tree or other various classifiers. However, both of them suffer from large computational complexity during the first stage, which are not flexible for large-scale data hashing.

ITQ [4] proposes a classic iterative quantization method to generate binary codes, which enjoys a linear time complexity $\mathcal{O}(n)$ in training data size n and achieves promising retrieval results at the same time. Inspired by the success achieved by the co-training strategy [30, 23], we attempt to directly utilize the binary codes generated by such fast unimodal hashing methods, and apply them to the other embedded modality to discover the optimal hash function. In this way, we can avoid the huge time complexity brought by seeking optimal binary codes and hash projections simultaneously. Besides, benefiting from the semantic embedding such as Canonical Correlation Analysis (CCA), we still preserve the semantic similarities in the hamming space. The overview of our proposed DASH is shown in Figure 1, and we summarize the algorithm in Algorithm 1.

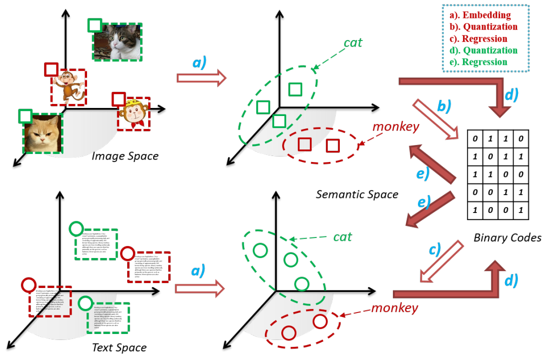

Figure 1: Overview of the proposed DASH, circles denote the text modality while squares denote the image modality. Red and green indicates two semantic categories, 'monkey' and 'cat', respectively. Our DASH follows unidirectional path *'a-b-c'* while traditional two-step methods go along with the path *'a-d-e'* and iterative methods follows the cycled path *'a-(d-e)$_n$'*. (Best viewed in colors).

2.3 Relationship to Iterative View-specific CMH Methods

Most previous view-specific works on CMH try to obtain the binary codes $B_m \in \{+1, -1\}^{c \times n}$ and hash function $W_m \in \mathbb{R}^{d_m \times c}$, $m = 1, 2$ in a unified framework as is shown below:

$$\begin{aligned} \min_{W_1, W_2, B_1, B_2} \mathcal{L} = \|B_1 - W_1^T X^{(1)}\|_F^2 + \|B_2 - W_2^T X^{(2)}\|_F^2 \\ + \alpha \Omega(Y, B_1, B_2) + \beta \Phi(W_1, W_2), \end{aligned} \tag{1}$$

where $\Omega(Y, B_1, B_2) = \|B_1 - B_2\|_F^2$ is an ordinary alignment setting to keep codes close to each other. Note that $\|B_1 - B_2\|_F^2$ equals to $-2tr(B_1^T B_2) + const$ due to the hard discrete constraints.

Given B_1 and W_1 from last iteration, we minimize the following term versus B_2 and W_2 (here $\Phi(W_1, W_2) = \gamma \sum_i \|W_i\|_F^2$),

$$\min_{B_2, W_2} \|B_2 - W_2^T X^{(2)}\|_F^2 - 2\alpha tr(B_1^T B_2) + \gamma \|W_2\|_F^2. \tag{2}$$

Then the optimal W_2 is explicitly given by following term,

$$\arg\min_{W_2} \|B_2 - W_2^T X^{(2)}\|_F^2 = (X^{(2)} X^{(2)T} + \gamma I)^{-1} X^{(2)} B_2^T. \tag{3}$$

The term of miminizing B_2 is given below,

$$\begin{aligned} &\min_{B_2} \|B_2 - W_2^T X^{(2)}\|_F^2 - 2\alpha tr(B_1^T B_2) \\ &= -2tr((W_2^T X^{(2)} + \alpha B_1)^T B_2) + const. \end{aligned} \tag{4}$$

When the trade-off parameter α becomes larger, the optimal code B_2 apparently equals to B_1. This result also corresponds with our

DASH, because the feedback from B_2 to B_1 is ignored due to the equivalence. Hence, we propose a rather simple approach to deal with cross-modal hashing problem. Note that γ is fixed to $1e^{-3}$, resulting in our DASH being a parameters-free method.

Algorithm 1 Double **A**lignment ba**S**ed **H**ashing (DASH)

```
Input: Data matrices $X^{(m)} \in \mathbf{R}^{d_m \times n}$, $m = 1, 2$, semantic label matrix $Y \in \mathbf{R}^{k \times n}$ and hash code length $c$.
Output: Hash projection matrices $W_m \in \mathbf{R}^{d_m \times c}$ , $m = 1, 2$.
Procedure:
    1. Obtain $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ via CCA with $Y$;
    2. Solve $B$ and $W_1$ with $\hat{X}^{(1)}$ via ITQ;
    3. Compute $W_2$ with Eqn. 3.
```

2.4 Computation Complexity Analysis

During the training procedure, CCA semantic embedding of each modality occupy $\mathcal{O}(d_1^3 + d_2^3 + ncd_1 + ncd_2)$ due to the generalized eigenvalue decomposition problem, and ITQ occupies $\mathcal{O}(c^3 + ncd_1)$, where c is the code length. Besides, least-square linear regression occupies $\mathcal{O}(nd_2^2c)$. As a result, the overall time complexity is $\mathcal{O}(nd^2 + d^3)$, where d is the dimension of longer original features. It is linear with the training data size, guaranteeing that our DASH is suitable for large-scale datasets.

3. EXPERIMENTS

We compared our DASH with other start-of-the-art cross-modal hashing algorithms on four commonly used datasets: Wiki, LabelMe, VOC and NUS-WIDE. To compare the general retrieval performance on various settings, we extend these four basic datasets to seven different detailed datasets shown in Table 1.

3.1 Experimental Setting

3.1.1 Datasets

The **Wiki** dataset consists of 2,866 items which were collected from 'Wikipedia' and classified into 10 semantic categories. We adopt the same setting as [2]. The **Wiki++** extends the Wiki dataset with deep image features and high-dimensional text feature as [22].

The **LabelMe** dataset [17] consists of 2,686 fully annotated outdoor images from 8 scene categories. For the text modality, we generate the object account vector via the LabelMe toolbox the same as [11]. It is randomly split into training/testing set as 3:1.

The **VOC_full** dataset consists of 9,963 image-tag pairs classified as 20 different classes [7]. We choose images associated with only one object as [18] and obtain the **VOC** dataset. The CNN image features are extracted via Caffe[1] of the VOC_full dataset, denoted as **VOC_full+**.

The **NUS-WIDE** dataset [13, 3] is composed of 186,577 annotated web images associated with corresponding tags. Here we choose 1% of image-text pairs coming from the largest 10 classes randomly as our testing data, and the rest as training data.

3.1.2 Baseline Methods

Unsupervised methods: CVH [8], CMFH [3], PDH [16] and LSSH [29]; Supervised ones, SePH [13], IMH [20], CMSSH [1] and SCM [28]. For fair comparisons, all training instances are utilized for IMH and linear regression hash functions are adopted for SePH. All the source codes are kindly provided by the authors.

The methods of obtaining B with image and text are abbreviated as DASH_i and DASH_t, respectively. The definitions of SCM_Orth and SCM_Seq are the same as [28]. We regard two items as true neighbors if they share one same class at least. In order to eliminate the effects of random initialization, all the results are averaged over 5 runs.

Table 1: Statistics of several benchmark datasets (For label*, s denotes single-label, and m denotes multi-label).

Dataset	# training / testing	# image / text	# categories	labels*
Wiki	2,173 / 693	128 / 10	10	s
Wiki++	2,173 / 693	4,096 / 5,000	10	s
LabelMe	2,014 / 672	512 / 470	8	s
VOC	2,808 / 2,841	512 / 399	20	s
VOC_full	5,011 / 4,952	512 / 399	20	m
VOC_full+	5,011 / 4,952	4,096 / 399	20	m
NUS-WIDE	184,671 / 1,906	500 / 1,000	10	m

3.1.3 Evaluation Scheme

Performance of all the methods are measured with the mean average precision (MAP) that is widely used for retrieval methods and normalized discounted cumulative gain (NDCG), which is widely used for single-label and multi-label retrieval methods, respectively.

3.2 Experimental Results

3.2.1 Results for the Single-label Datasets

Table 2: MAP@100 result on three single-label datasets for different tasks. The best values are shown in boldface.

Image query	**Wiki**			**LabelMe**			**VOC**		
# of bits	16	24	32	16	24	32	16	24	32
CMSSH	23.1	18.9	17.3	53.1	55.4	57.9	22.5	20.3	24.1
CVH	21.3	20.3	19.6	33.2	32.8	31.5	19.8	19.0	18.9
IMH	20.4	19.2	18.5	41.5	38.4	31.5	27.4	24.6	22.9
PDH	21.6	20.5	22.6	50.6	51.3	54.7	20.0	20.3	20.0
CMFH	25.9	27.6	28.1	36.8	42.3	24.0	29.5	31.0	30.3
SCM_Orth	19.8	19.4	17.3	47.3	48.0	46.9	23.4	20.9	20.3
SCM_Seq	27.7	27.4	27.6	64.2	65.0	66.7	34.7	38.2	37.9
SePH	28.2	**30.9**	**31.1**	65.6	65.3	**70.1**	38.5	43.6	46.3
DASH_i	24.4	24.1	24.0	49.4	41.9	40.3	34.0	28.2	24.4
DASH_t	**28.9**	30.5	**31.1**	**67.7**	**68.6**	69.2	**45.0**	**49.2**	**52.1**

Text query	**Wiki**			**LabelMe**			**VOC**		
# of bits	16	24	32	16	24	32	16	24	32
CMSSH	22.3	22.5	19.2	53.7	57.6	57.7	26.9	27.2	27.6
CVH	19.7	19.4	18.3	34.7	34.2	33.3	20.3	19.5	19.3
IMH	21.2	19.5	18.4	43.1	39.5	36.6	27.9	24.9	23.0
PDH	19.4	18.2	19.4	50.7	51.7	53.0	19.1	18.9	19.2
CMFH	26.5	28.6	29.5	36.7	43.3	24.6	28.2	30.8	28.8
SCM_Orth	19.7	19.4	16.4	36.0	36.7	29.9	21.9	19.6	18.0
SCM_Seq	27.6	27.6	28.4	69.8	69.9	72.0	31.0	34.0	34.2
SePH	27.4	28.6	**31.1**	73.2	74.2	**77.8**	34.3	39.1	39.0
DASH_i	24.9	25.8	25.4	57.4	50.4	51.6	37.7	30.4	29.9
DASH_t	**27.8**	**29.6**	29.5	**74.7**	**74.3**	74.5	**38.1**	**39.2**	**39.8**

We compare DASH with other methods on three single-label datasets – Wiki, LabelMe and VOC. From Table 2, we can find that DASH_t and SePH achieve the best performances on all three datasets. Especially for the VOC dataset, our DASH outperforms the second best SePH at every bit.

3.2.2 Results for the High-dimensional Datasets

We evaluate DASH and other methods on two high dimensional datasets, Wiki++ and VOC_full+. From Table 3, we can summarize that DASH achieves the best MAP in all cases, especially on VOC_full+, its MAP value achieves nearly 90%. Compared with the second best method on Wiki++, the maximum gains of DASH_t reaches 14.2% for image query and 13.1% for text query. Note that DASH_i and DASH_t have similar performance. That is because the image is represented by 4,096 CNN features, which have rich semantic information.

[1] http://caffe.berkeleyvision.org/.

Table 3: MAP@100 result on two high-dimensional datasets for different tasks. The best values are shown in boldface.

Image query	**Wiki++**			**VOC_full+**		
# of bits	16	24	32	16	24	32
CMSSH	31.1	29.3	29.7	70.4	73.7	73.8
CVH	14.3	14.3	14.3	74.2	72.4	66.7
IMH	28.6	28.4	27.7	68.5	65.2	63.7
CMFH	25.6	26.9	27.0	47.3	48.6	47.8
SCM_Orth	25.8	24.2	21.0	67.2	65.4	63.8
SCM_Seq	37.2	40.1	40.1	77.3	80.2	81.8
SePH	33.6	36.5	37.0	82.8	84.8	87.3
DASH_i	39.7	38.5	38.2	**84.9**	88.1	88.1
DASH_t	**42.5**	**41.7**	**41.6**	**84.9**	**88.9**	**89.2**

Text query	**Wiki++**			**VOC_full+**		
# of bits	16	24	32	16	24	32
CMSSH	37.0	31.2	32.4	70.4	77.1	79.2
CVH	15.0	14.7	14.3	80.5	75.9	65.6
IMH	28.9	29.0	28.0	71.1	66.2	61.3
CMFH	25.6	26.8	26.9	46.0	49.2	49.7
SCM_Orth	26.8	24.8	21.9	71.4	64.1	57.3
SCM_Seq	40.5	43.3	42.7	76.7	80.0	81.1
SePH	42.7	44.4	45.9	83.6	86.5	88.5
DASH_i	46.2	44.8	45.4	**91.8**	93.1	92.8
DASH_t	**48.3**	**47.8**	**47.5**	88.8	**93.6**	**94.2**

3.2.3 Results for the Large-scale Datasets

Table 4: Result on large-scale dataset (NUS-WIDE) for different tasks. The best values are shown in boldface.

Image query	**MAP@100**			**NDCG@10**		
# of bits	16	24	32	16	24	32
CMSSH	52.6	51.2	50.6	34.2	33.2	32.0
CVH	49.3	48.1	46.7	32.2	31.5	30.1
IMH	45.8	45.4	43.9	28.8	28.4	27.2
PDH	52.7	53.5	54.5	31.6	32.3	33.1
CMFH	41.2	40.7	42.7	23.9	23.8	25.6
SCM_Orth	49.8	47.7	47.9	32.6	31.1	31.4
SCM_Seq	**60.5**	**61.1**	**62.6**	40.3	40.1	**42.0**
SePH	55.0	56.7	56.0	35.1	35.7	36.1
DASH_i	55.7	55.3	55.9	34.4	33.7	34.5
DASH_t	60.3	57.9	59.1	**41.1**	**40.8**	41.6

Text query	**MAP@100**			**NDCG@10**		
# of bits	16	24	32	16	24	32
CMSSH	50.1	49.1	48.8	35.2	28.9	30.1
CVH	49.5	48.2	46.8	33.6	32.4	31.0
IMH	45.5	45.3	44.0	29.1	28.9	28.1
PDH	52.7	51.3	51.5	31.6	32.1	30.4
CMFH	41.2	41.8	42.7	24.3	26.2	27.1
SCM_Orth	49.1	45.4	47.3	30.9	28.8	27.9
SCM_Seq	58.2	60.6	61.4	36.8	38.7	40.8
SePH	57.5	58.6	57.2	38.5	36.8	37.5
DASH_i	**61.9**	**63.8**	**63.4**	**44.0**	**47.4**	**45.8**
DASH_t	58.4	57.3	56.6	38.9	37.6	37.2

From Table 4, we can easily find that our DASH performs best for text query. For image query, the MAP values of SCM are higher than that of DASH, while the NDCG values of SCM are smaller, which illustrates that DASH_t achieve better results with slight retrieval instances.

3.2.4 Results for the Multi-label Datasets

Cross-modal methods are evaluated on two multi-label datasets – NUS_WIDE and VOC_full. It is shown in Table 4 and Table 5 that DASH, SCM_Seq and SePH always perform better than other methods. In more than half of the cases, DASH achieves the best performance, especially in NUS_WIDE for text query.

Table 5: Result on multi-label dataset (VOC_full) for different tasks. The best values are shown in boldface.

Image query	**MAP@100**			**NDCG@10**		
# of bits	16	24	32	16	24	32
CMSSH	55.0	55.0	54.7	23.0	26.5	30.0
CVH	51.1	50.1	49.8	25.0	24.0	22.9
IMH	53.2	48.5	48.8	26.5	23.6	22.7
PDH	40.8	36.8	39.0	19.6	16.9	17.4
CMFH	50.5	52.6	52.6	25.1	25.9	26.0
SCM_Orth	51.2	49.3	49.4	24.5	23.4	23.4
SCM_Seq	56.1	60.1	61.3	27.8	**30.4**	**30.6**
SePH	**57.2**	58.9	**66.0**	25.8	26.7	27.3
DASH_i	55.8	54.1	54.4	**28.4**	27.4	26.7
DASH_t	**57.2**	**65.1**	65.3	28.3	30.1	29.7

Text query	**MAP@100**			**NDCG@10**		
# of bits	16	24	32	16	24	32
CMSSH	36.4	40.0	40.4	19.1	25.0	24.7
CVH	43.5	41.9	37.4	31.6	30.8	23.3
IMH	45.7	39.9	40.3	30.4	24.7	26.6
PDH	34.0	33.8	34.1	17.0	17.5	17.1
CMFH	41.6	44.3	43.3	26.7	31.2	28.5
SCM_Orth	42.5	38.6	35.1	29.9	25.1	21.8
SCM_Seq	47.7	50.5	52.2	32.5	32.9	36.5
SePH	50.2	52.8	**53.2**	35.7	38.0	37.6
DASH_i	**52.1**	48.1	46.3	**41.4**	35.8	33.0
DASH_t	51.4	**53.1**	51.8	40.4	**42.2**	**40.2**

3.2.5 Training Time

Table 6 shows the training time of the supervised hashing methods on three challenging datasets. Generally, CMSSH, SCM_Orth and DASH cost relatively less time, and DASH always performs better than the other two methods. Generally, the retrieval performance of SePH always follows SCM_seq and DASH in above tables. However, SePH is not applicable for the large-scale datasets. Even SePH randomly chooses 5,000 data as training set, the training time of SePH is still huge. By contrast, SCM has a strong ability to adapt to large-scale data, but it needs large training time cost in terms of processing high-dimensional data. For the VOC_full+ dataset, the training time of SePH and SCM is 100 times more than DASH. Moreover, the training time of SCM_seq is linear with the length of hash bits. Generally, our DASH is applicable for high-dimensional, large-scale datasets, and achieves the best or competitive cross-modal retrieval performance.

Table 6: Training time (in seconds) of supervised hashing methods on three datasets.

datasets	**NUS-WIDE**		**VOC_full+**		**Wiki++**	
# of bits	16	32	16	32	16	32
CMSSH	8	8	17	15	18	13
IMH	64	64	121	113	17	11
SCM_Orth	2	2	90	105	12	13
SCM_Seq	10	14	1,489	2,909	318	863
SePH	2,378	2,647	1,858	1,900	759	861
DASH_i	12	14	16	16	6	6
DASH_t	12	13	16	16	6	6

4. CONCLUSION

In this paper, we propose a *simple yet effective* approach named DASH for cross-modal hashing. This non-iterative and parameter-free DASH method is frustratingly easy to implement in three code lines. Extensive experimental results illustrate the advantages of our DASH over other existing state-of-the-art methods, which further confirms that DASH is flexible to various settings, including high-dimensional, large-scale and multi-label datasets.

5. ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No. 61502073, 61473289), the Open Projects Program of National Laboratory of Pattern Recognition (No. 2014-07349) and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB02070000).

6. REFERENCES

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

[2] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. R. Lanckriet, R. Levy, and N. Vasconcelos. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):521–535, 2014.

[3] G. Ding, Y. Guo, and J. Zhou. Collective matrix factorization hashing for multimodal data. In *Proc. CVPR*, pages 2083–2090, 2014.

[4] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.

[5] R. He, Y. Cai, T. Tan, and L. S. Davis. Learning predictable binary codes for face indexing. *Pattern Recognition*, 48(10):3160–3168, 2015.

[6] X. He and P. Niyogi. Locality preserving projections. In *Proc. NIPS*, pages 153–160, 2003.

[7] S. J. Hwang and K. Grauman. Reading between the lines: Object localization using implicit cues from image tags. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1145–1158, 2012.

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

[9] J. Liang, D. Cao, R. He, Z. Sun, and T. Tan. Principal affinity based cross-modal retrieval. In *Proc. ACPR*, pages 126–130, 2015.

[10] J. Liang, R. He, Z. Sun, and T. Tan. Group-invariant cross-modal subspace learning. In *Proc. IJCAI*, pages 1739–1745, 2016.

[11] J. Liang, Z. Li, D. Cao, R. He, and J. Wang. Self-paced cross-modal subspace matching. In *Proc. SIGIR*, pages 569–578, 2016.

[12] G. Lin, C. Shen, and A. van den Hengel. Supervised hashing using graph cuts and boosted decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11):2317–2331, 2015.

[13] Z. Lin, G. Ding, M. Hu, and J. Wang. Semantics-preserving hashing for cross-view retrieval. In *Proc. CVPR*, pages 3864–3872, 2015.

[14] A. K. Menon and D. Surian. Cross-modal retrieval : a pairwise classification approach. In *Proc. SDM*, pages 199–207, 2015.

[15] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proc. MM*, pages 251–260, 2010.

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

[17] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

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

[19] X. Shen, F. Shen, Q.-S. Sun, and Y.-H. Yuan. Multi-view latent hashing for efficient multimedia search. In *Proc. MM*, pages 831–834, 2015.

[20] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *Proc. SIGMOD*, pages 785–796, 2013.

[21] D. Wang, X. Gao, X. Wang, and L. He. Semantic topic multimodal hashing for cross-media retrieval. In *Proc. IJCAI*, pages 3890–3896, 2015.

[22] K. Wang, R. He, L. Wang, W. Wang, and T. Tan. Joint feature selection and subspace learning for cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. doi:10.1109/TPAMI.2015.2505311.

[23] W. Wang and Z.-H. Zhou. Analyzing co-training style algorithms. In *Proc. ECML*, pages 454–465, 2007.

[24] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. NIPS*, pages 1753–1760, 2009.

[25] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang. Quantized correlation hashing for fast cross-modal search. In *Proc. IJCAI*, pages 3946–3952, 2015.

[26] Z. Yu, F. Wu, Y. Yang, Q. Tian, J. Luo, and Y. Zhuang. Discriminative coupled dictionary hashing for fast cross-media retrieval. In *Proc. SIGIR*, pages 395–404, 2014.

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

[28] D. Zhang and W.-J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *Proc. AAAI*, pages 2177–2183, 2014.

[29] J. Zhou, G. Ding, and Y. Guo. Latent semantic sparse hashing for cross-modal similarity search. In *Proc. SIGIR*, pages 415–424, 2014.

[30] Z.-H. Zhou and M. Li. Semi-supervised regression with co-training. In *Proc. IJCAI*, pages 908–913, 2005.

[31] F. Zhu, L. Shao, and M. Yu. Cross-modality submodular dictionary learning for information retrieval. In *Proc. CIKM*, pages 1479–1488, 2014.

[32] Y. Zhuang, Z. Yu, W. Wang, F. Wu, S. Tang, and J. Shao. Cross-media hashing with neural networks. In *Proc. MM*, pages 901–904, 2014.