

OPTIMAL

Center for OPTical IMagery Analysis and Learning, Xi'an, China

光學影像分析與學習中心 · 中國西安 ·



Learning Hash Functions Using Sparse Reconstruction

Yong Yuan, Xiaoqiang Lu, and Xuelong Li

Center for OPTical IMagery Analysis and Learning (OPTIMAL)



Roadmap

■ *Overview*



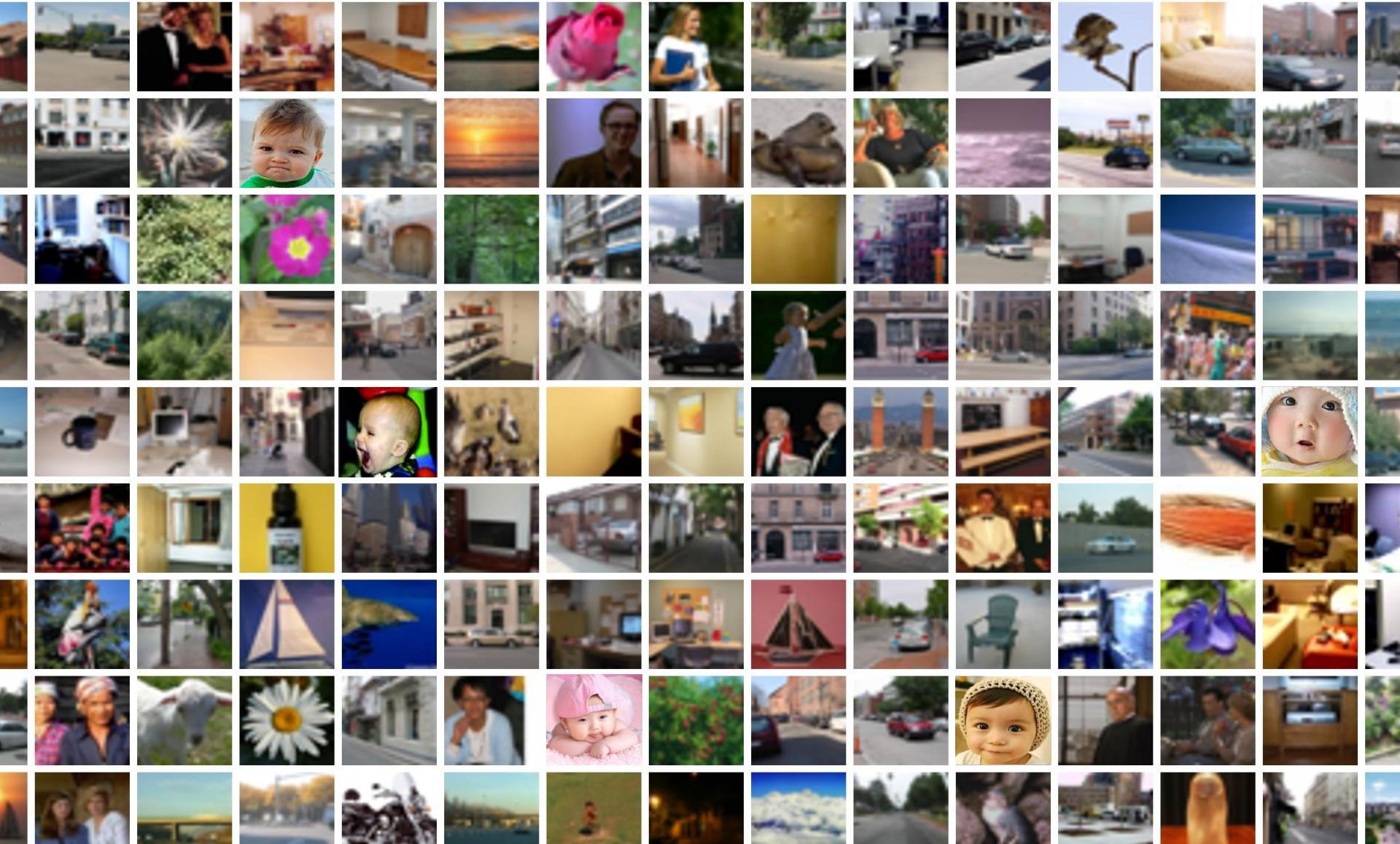
■ *Method*

■ *Experiments*

■ *Summary*

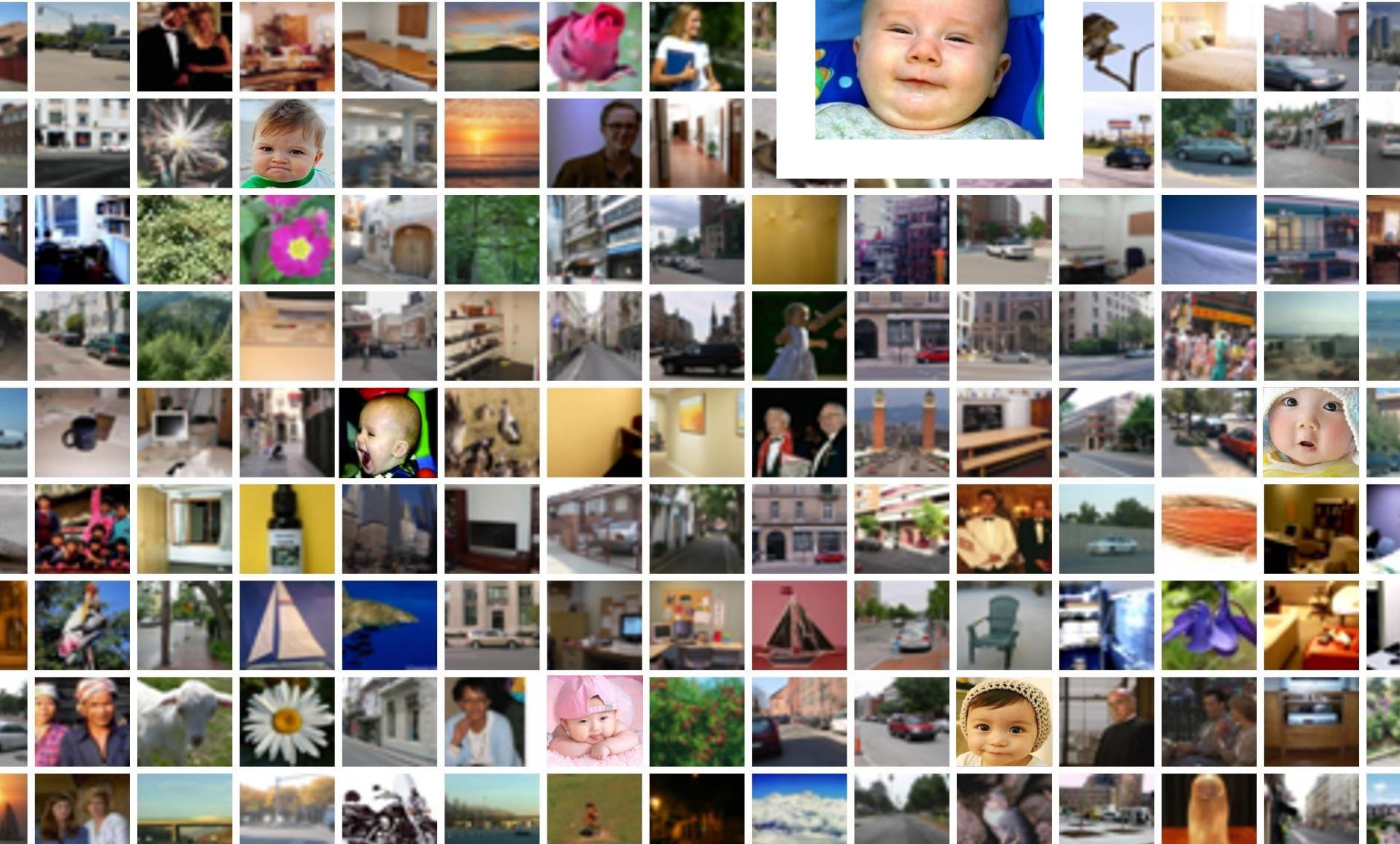


Near Neighbor Search

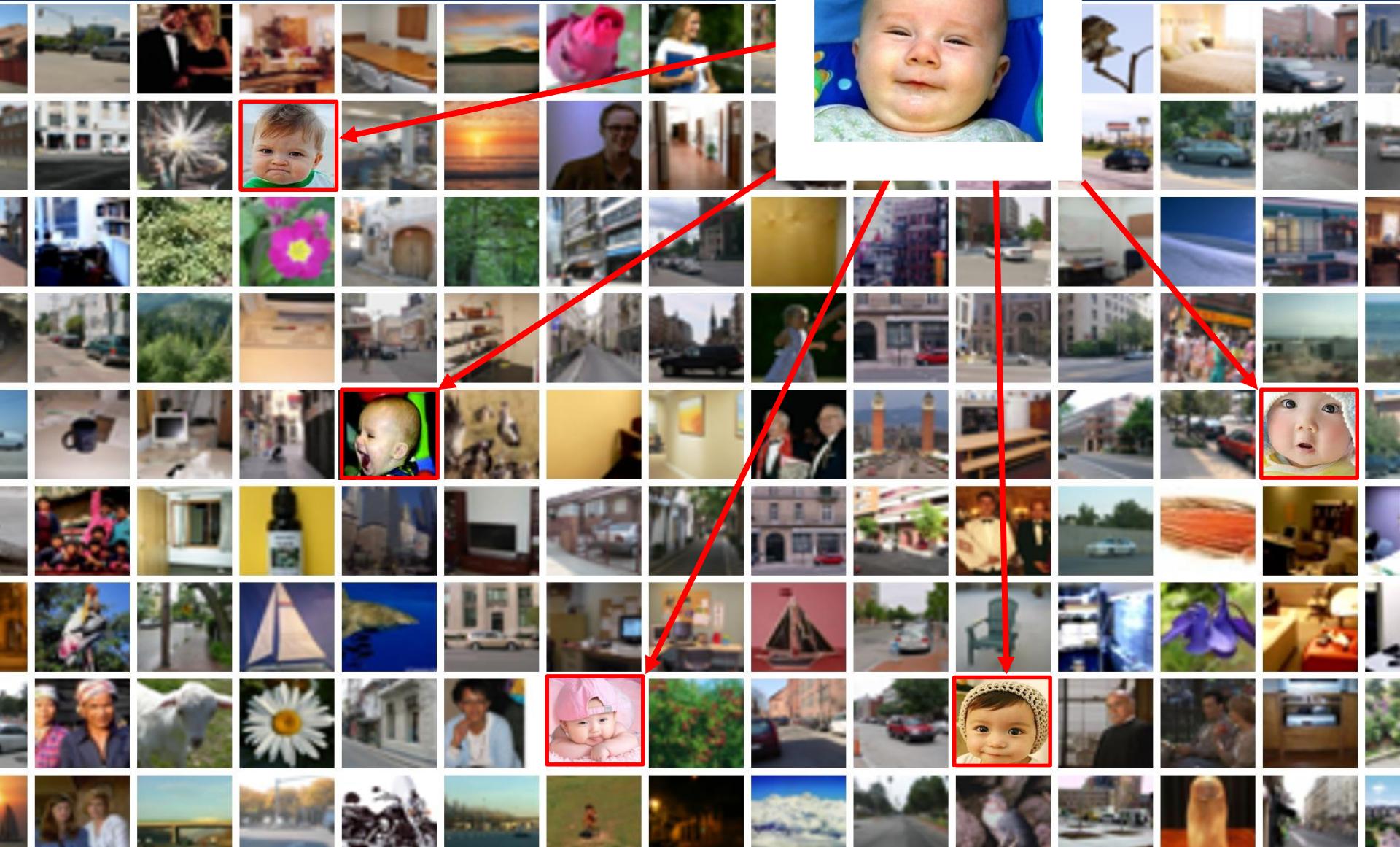




Near Neighbor Search

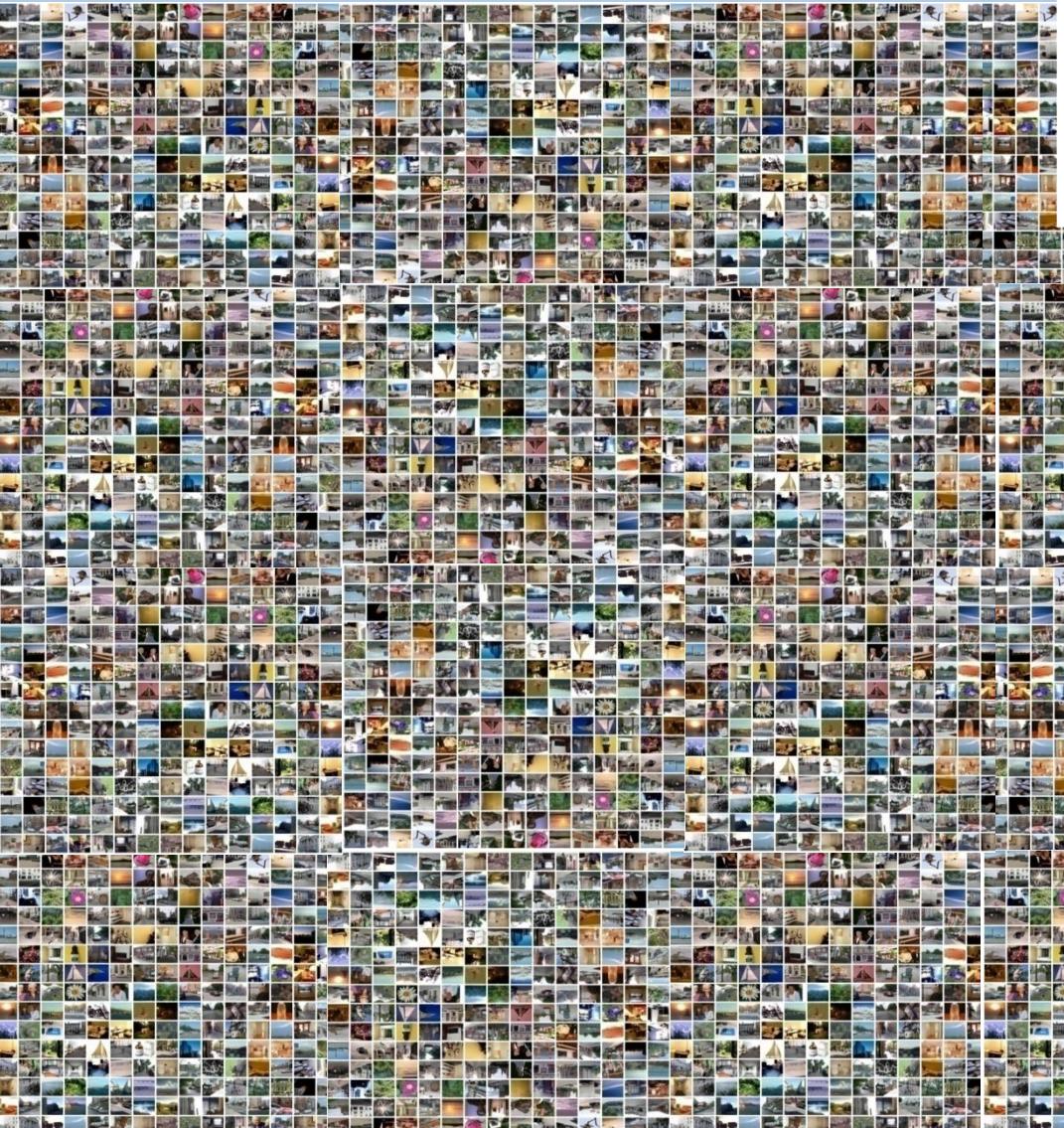


Near Neighbor Search





Near Neighbor Search





Near Neighbor Search



Big data



Near Neighbor Search



How do a content based image retrieval engine search a query efficiently?



Similarity Preserving Binary Hashing

...

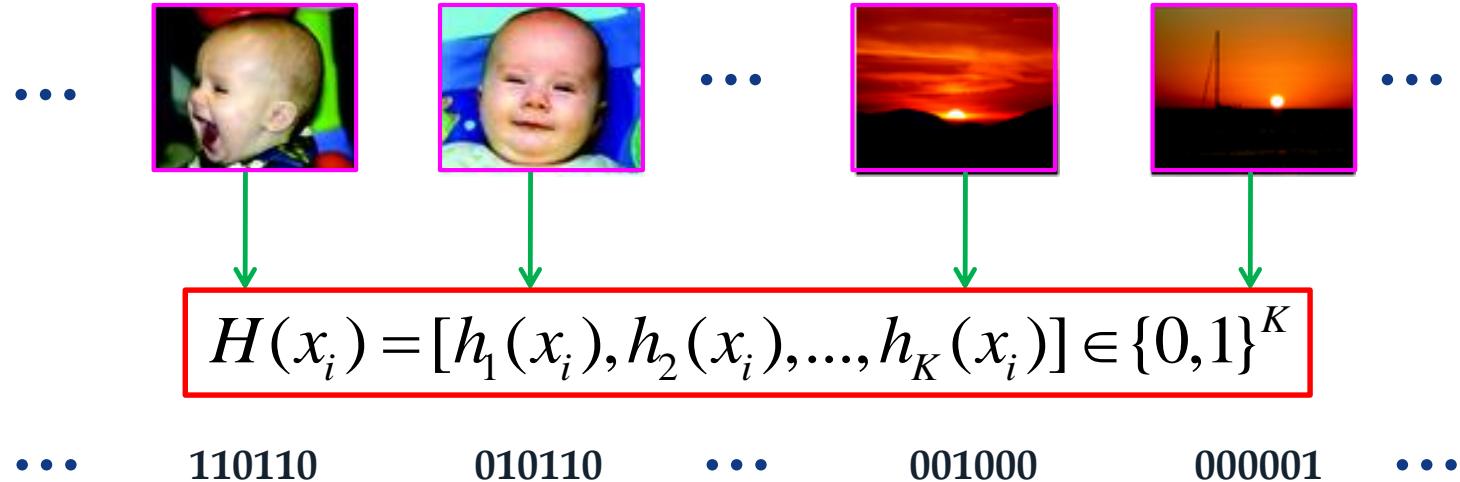


...

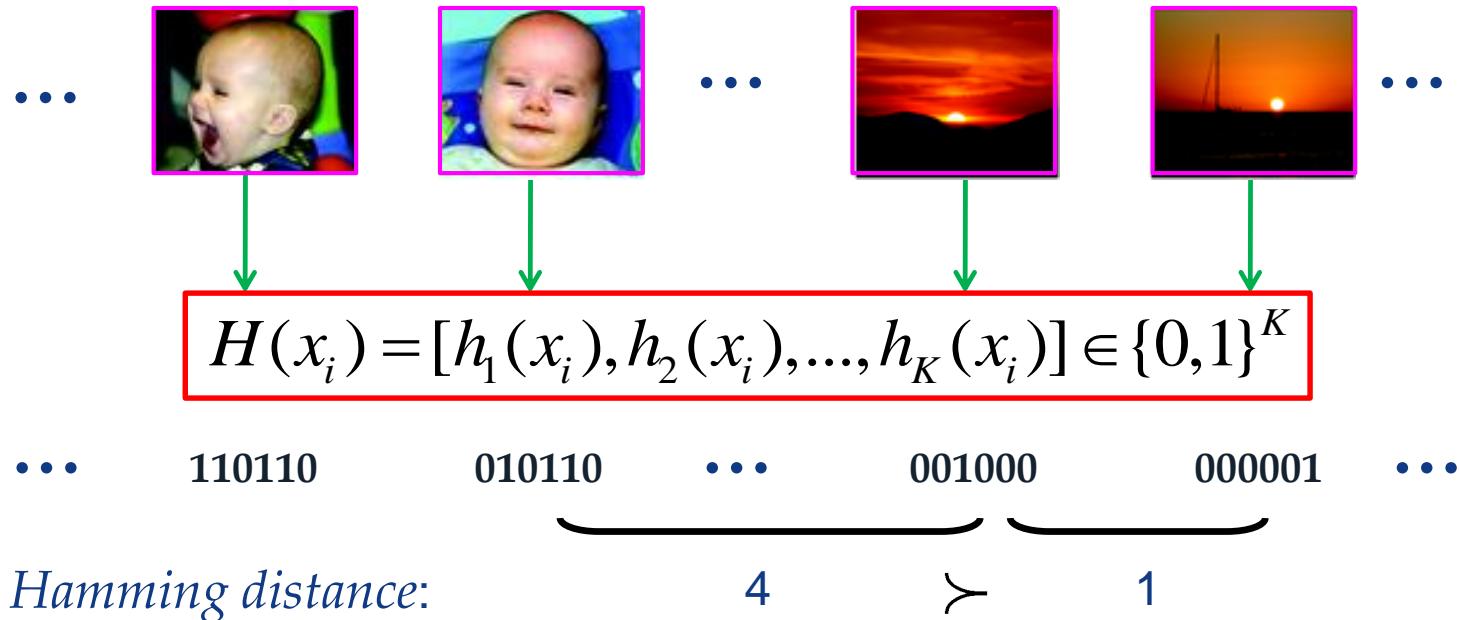


...

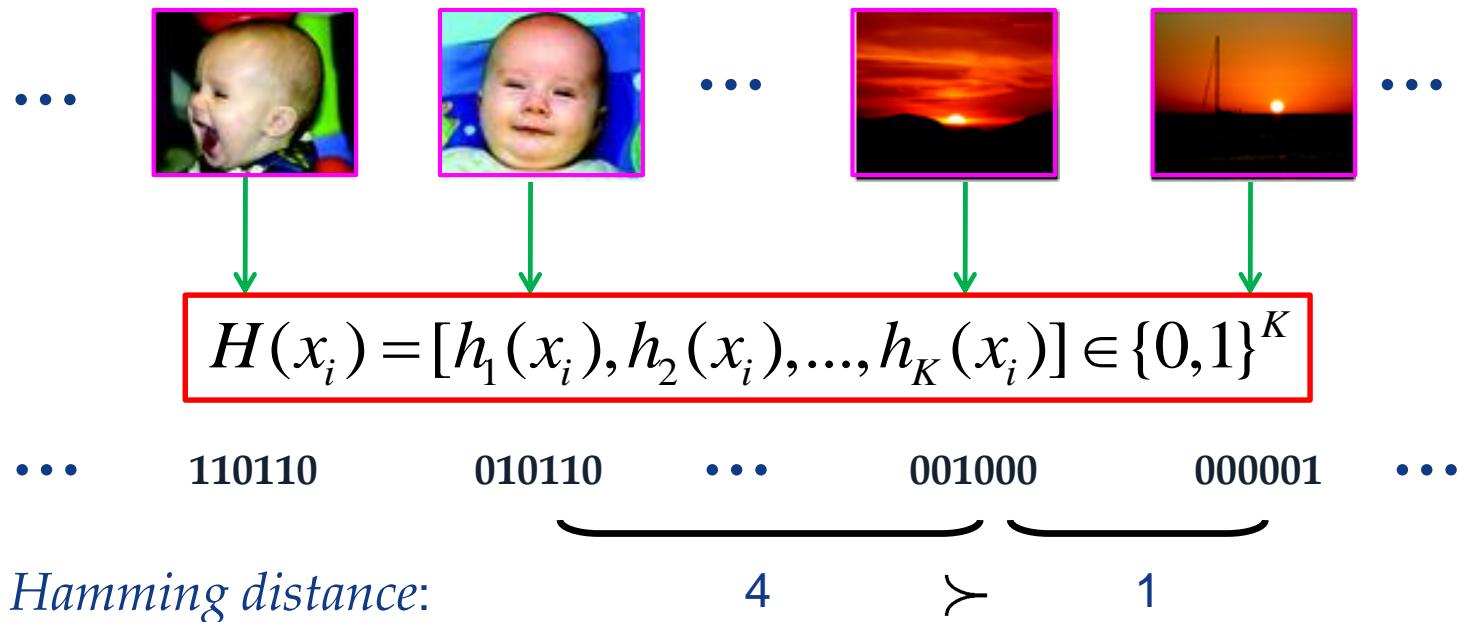
Similarity Preserving Binary Hashing



Similarity Preserving Binary Hashing



Similarity Preserving Binary Hashing



- Why binary codes?
 - Query time is constant or sub-linear.
 - Binary codes are storage-efficient.

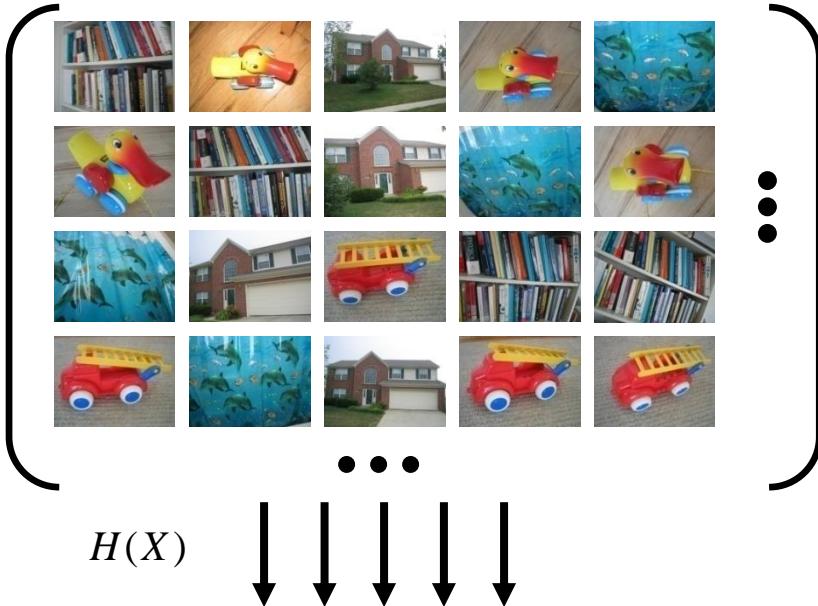
Hashing Using Compact Codes

Dataset



Hashing Using Compact Codes

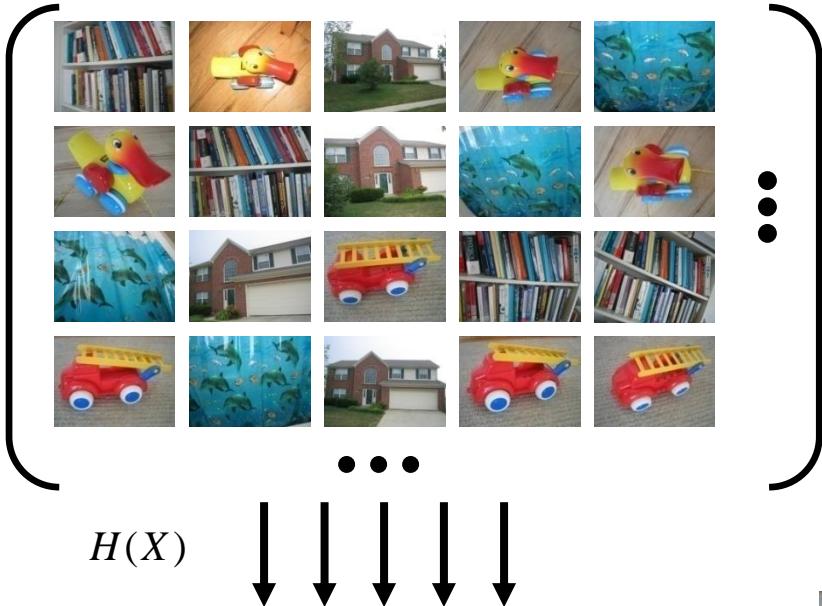
Dataset



Simple hash table

Hashing Using Compact Codes

Dataset



$H(X)$

$H(x_q)$
→ 110111



Q

XOR

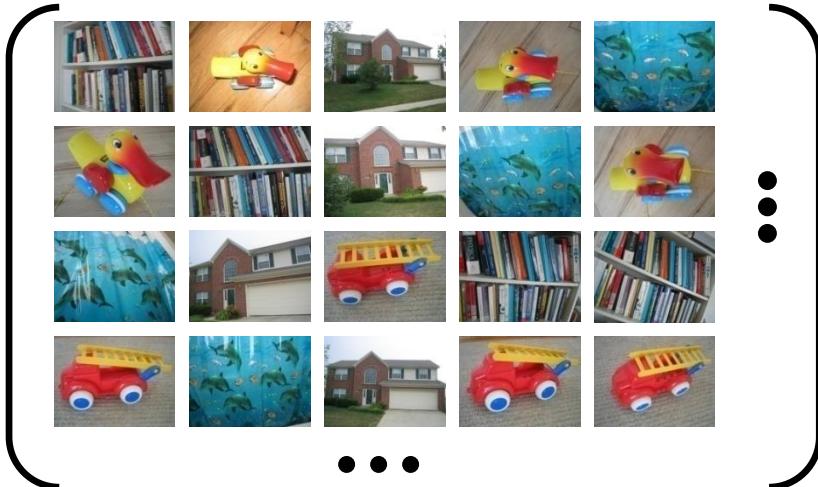
000101					⋮
110111					⋮
100001					⋮



Simple hash table

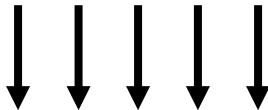
Hashing Using Compact Codes

Dataset

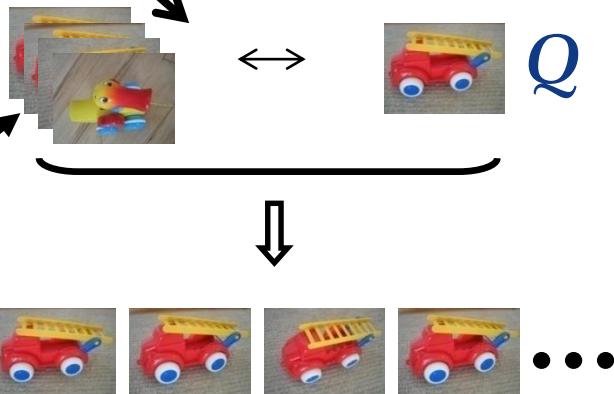


*Search the hash table
for a small set of images.*

$H(X)$



$\ll N$



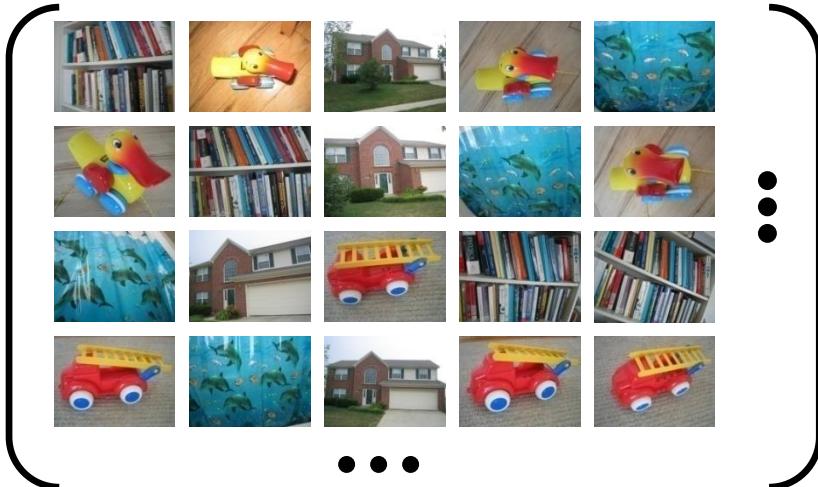
000101					...
110111					...
100001					...

Simple hash table

⋮

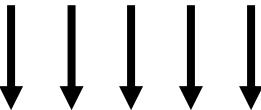
Hashing Using Compact Codes

Dataset

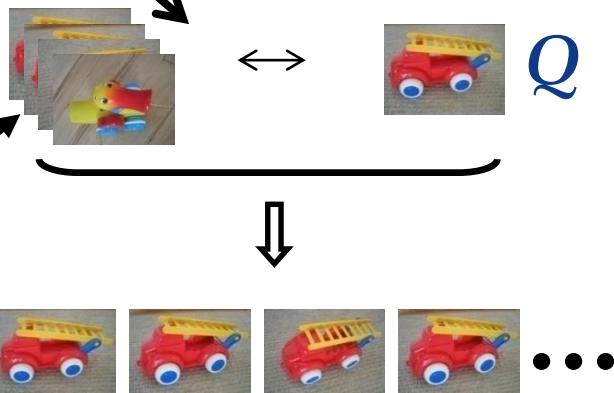


*Search the hash table
for a small set of images.*

Key point: $H(X)$



$\ll N$



$H(x_q)$
 $\rightarrow 110111$



Q

XOR

000101						...
110111						...
100001						...

Simple hash table



Two of the Representative



IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 12, DECEMBER 2012

2283

Semi-Supervised Hashing for Large-Scale Search

Jun Wang, Member, IEEE, Sanjiv Kumar, Member, IEEE, and Shih-Fu Chang, Fellow, IEEE

Abstract—Hashing-based approximate nearest neighbor (ANN) search in huge databases has become popular due to its computational and memory efficiency. The popular hashing methods, e.g., Locality Sensitive Hashing and Spectral hash function based on random or principal projections, the resulting hashes are either not very accurate or are not linearly separable. These methods are designed for a given metric similarity. On the contrary, semantic similarity is usually given in terms of samples. There exist supervised hashing methods that can handle such semantic similarity, but they are pre-labeled data are small or noisy. In this work, we propose a semi-supervised hashing (SSH) framework that minimizes over the labeled set and an information theoretic regularizer over both labeled and unlabeled sets. Based on the present three different semi-supervised hashing methods, including orthogonal hashing, nonnegative hashing, hashing. Particularly, the sequential hashing method generates subset codes in which each hash function is determined by the previous ones. We further show that the sequential learning paradigm can be extended to unlabelled data pairs as well. Extensive experiments on four large datasets (up to 60 million samples) demonstrate the performance of the proposed SSH methods over state-of-the-art supervised and unsupervised hashing techniques.

Index Terms—Hashing, nearest neighbor search, binary codes, semi-supervised hashing, pairwise labels, seq-

1 INTRODUCTION

With data including documents, images, and videos are growing rapidly. For example, the photo sharing website Flickr has over 8 billion images. The video sharing website YouTube receives more than 24 hours of uploaded

visual descriptions usually have hundreds of dimensions. Therefore, both exhaustive search, storage of the original data is a critical bottleneck.



The director of DMVV, Shih-Fu Chang: They have constant query time and also need substantially reduced storage as they usually store only compact codes. TPAMI, 2012.

data [1]. In addition, the semantic similarity search in these applications also suffer from the curse of dimensionality since

detailed survey of the tree-based ANN search algorithms can be found in [10]. However, the performance of tree-based methods is drastically degraded for high-dimensional data, being mostly restricted to the worst case of a linear search [11].

In addition, tree-based methods also suffer from memory constraints. In many cases, the size of the data structure is bigger than the original data itself. Hence, hashing-based ANN techniques have attracted more attention recently. They have constant query time and also need substantially reduced storage as they usually store only compact codes.

Recently, we focus on binary codes derived from linear vectors $\mathbf{X} \in \mathbb{R}^{d \times n}$, the goal in hashing is to learn suitable K-bit binary codes $\mathbf{Y} \in \mathbb{R}^{K \times n}$. To generate \mathbf{Y} , K binary hash functions are used. Linear projection-based hash functions have been widely used in the literature since they are very simple and efficient. Also, they have achieved

* J. Wang is with the Business Analytics and Mathematical Sciences Department, IBM T.J. Watson Research Center, RM 21-28, 1101 Kitchawan Rd, Box 124, Yorktown Heights, NY 10598.
E-mail: wangjun@us.ibm.com.

+ S. Kumar is with Google Research, 76 Ninth Avenue, 1st Floor, New York, NY 10011. E-mail: skumar@google.com.

++ S.-F. Chang is with the Department of Electrical and Computer Engineering, Columbia University, 440 W. 118th St., New York, NY 10027. E-mail: shifuchang@columbia.edu.

Manuscript received 12 Dec., 2010; revised 1 Aug. 2011; accepted 28 Jan. 2012; published online 6 Feb. 2012.

Recommended for acceptance by C. Stachniss.
For information on obtaining reprints of this article, please send email to: tpami@cs.columbia.edu, and reference IEEE Log Number TPAMI-2011-04-0059.

Digital Object Identifier no. 10.1109/TPAMI.2012.44

© 2012 IEEE. Manuscript received 12 Dec., 2010; revised 1 Aug. 2011; accepted 28 Jan. 2012; published online 6 Feb. 2012.

Published by the IEEE Computer Society.

The image consists of two main parts. The left side shows a white rectangular card with a black border, containing a research abstract. The right side shows a color photograph of a smiling woman with long brown hair, identified as Kristen Grauman.

OPTIMAL. Xi'an. China

光學影像分析與學習中心 · 中國西安 ·



Roadmap

■ *Overview*

■ *Method*



■ *Experiments*

■ *Summary*

Motivations

Projection	Learning paradigm	Method	Year
Data-independent	Unsupervised	<i>LSH</i>	1999
		<i>SIKH</i>	2009(<i>NIPS</i>)
		<i>SH</i>	2008(<i>NIPS</i>)
		<i>CH</i>	2011(<i>ICCV</i>)
		<i>ITQ</i>	2012(<i>TPAMI</i>)
		<i>KMH</i>	2013(<i>CVPR</i>)
		<i>KRH</i>	2014(<i>CVPR</i>)
		<i>S3PLH</i>	2010(<i>ICML</i>)
		<i>SSH</i>	2012(<i>TPAMI</i>)
		<i>BSPLH</i>	2012(<i>TKDE</i>)
		<i>BSSC</i>	2006
		<i>BRE</i>	2009(<i>NIPS</i>)
		<i>KS</i>	2012(<i>CVPR</i>)

A limitation of most hash based methods to explore the sparse reconstructive relationship of the data constructing hash functions. (Motivation A)

Motivations

If a sparse reconstruction is adopted, to maximize the information provided by each bit, how to incorporate this requirement with reconstruction simultaneously? (Motivation B)

Figure 10.2 shows a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

For a code to be efficient, we require that each bit has a 50% chance of being one or zero, and that different bits are independent of each other.

Does a balance between them improve the performance?

For a code to be efficient, we require that each bit has a 50% chance of being one or zero, and that different bits are independent of each other.

NIPS, 2012
How to build the balance?

Fig. 10.2 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

Fig. 10.3 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

Using maximum entropy principle, a binary bit that gives balanced partitioning of X provides maximum information. TPAMI, 2012.

Fig. 10.4 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

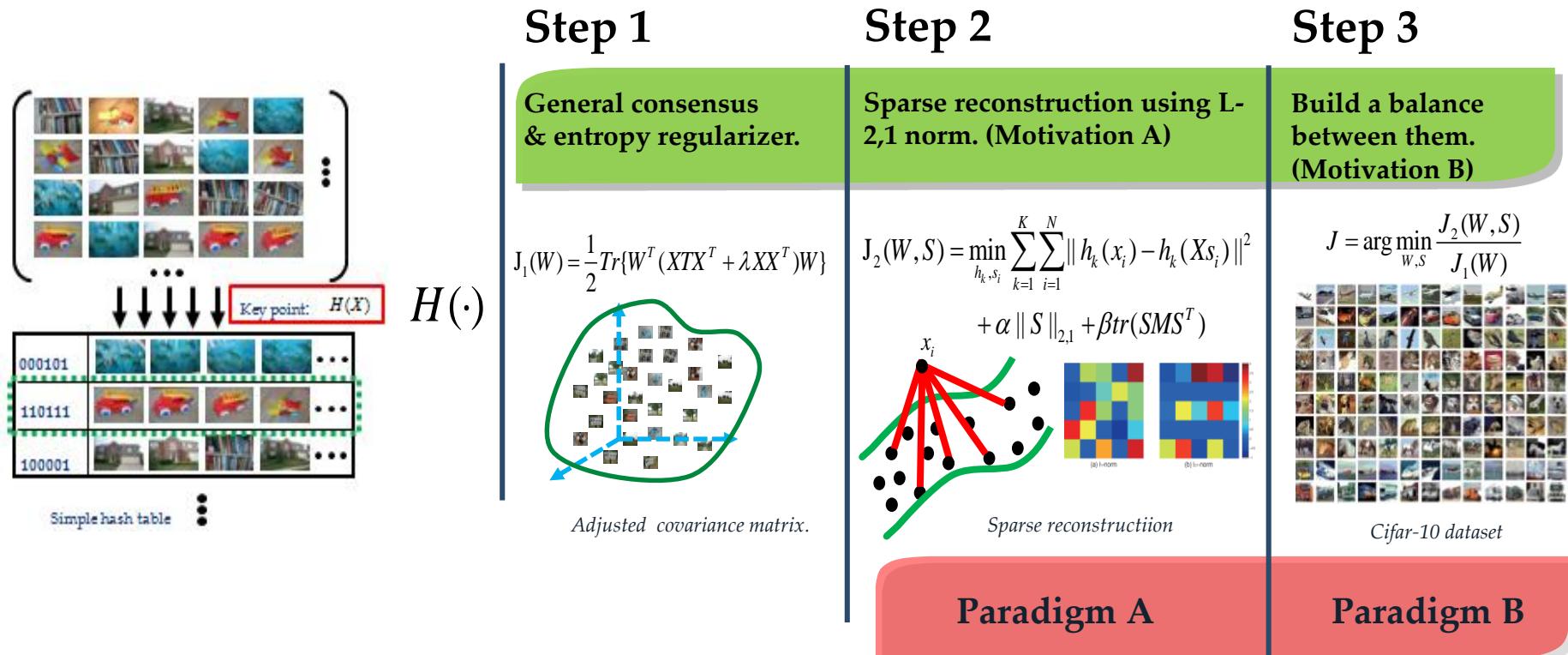
Fig. 10.5 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

Fig. 10.6 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

Fig. 10.7 A 2D mask with a sparse reconstruction. The mask has the following properties: it is a 2D binary mask with a sparse reconstruction.

Framework of Our Method

Flow chart of the proposed method to learn hash functions.





Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k)$$



Step 1:

■ Hash Function:

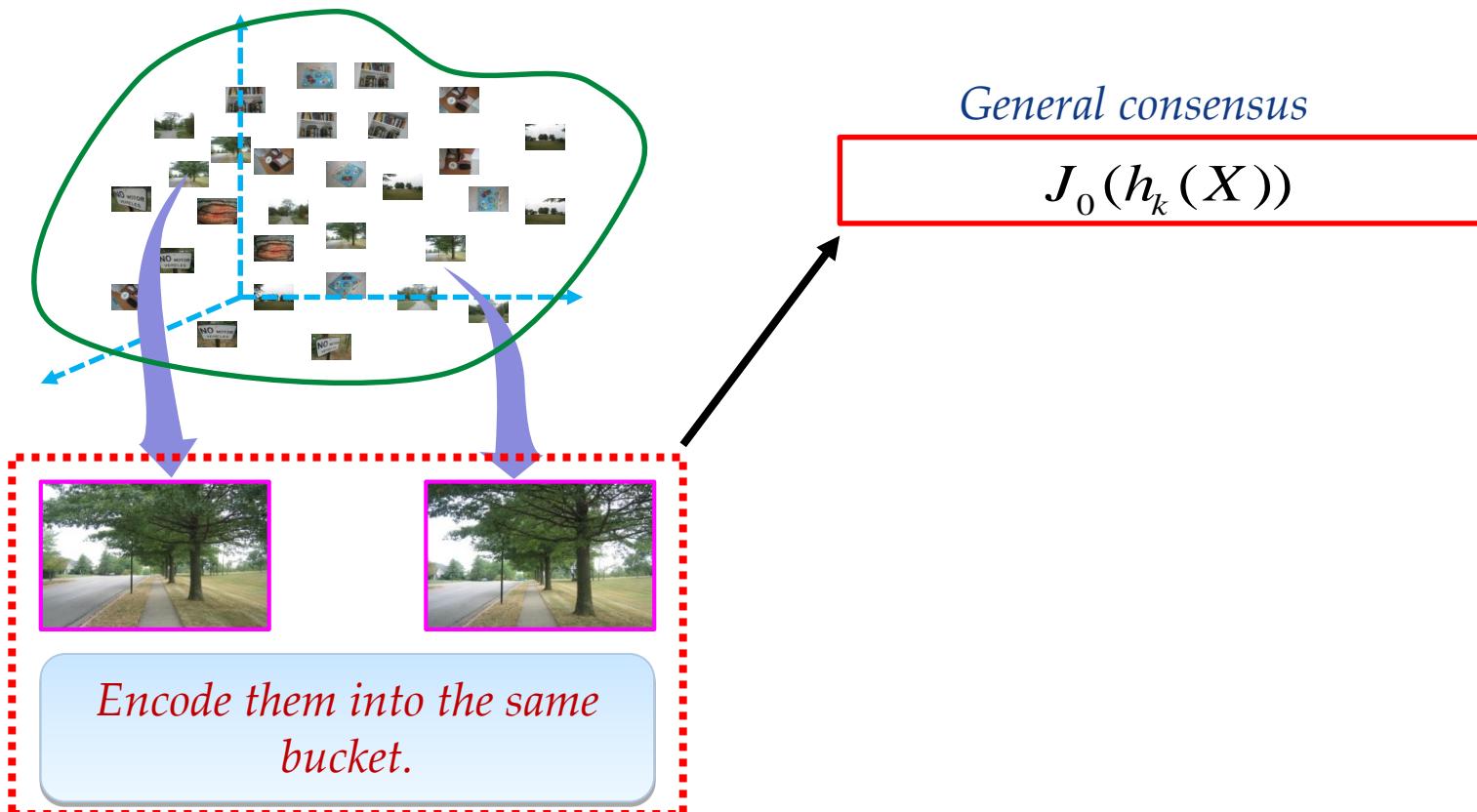
$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

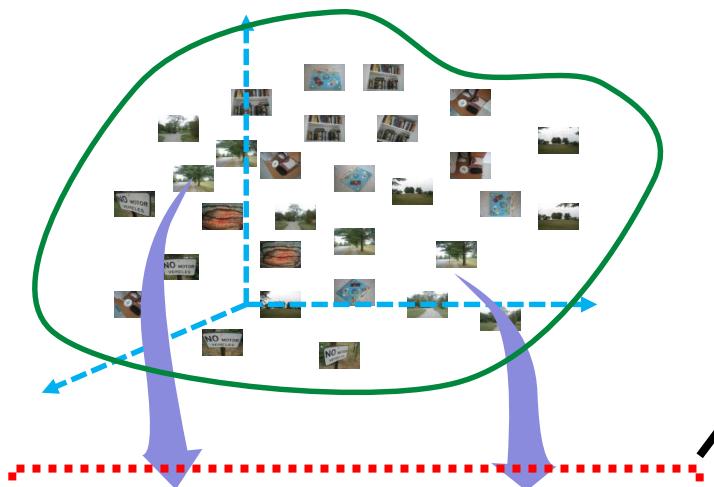


Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer



General consensus

$$J_0(h_k(X))$$

$$J_0(h_k(X)) = \sum_{i,j=1}^N t_{i,j} \langle H(x_i), H(x_j) \rangle$$

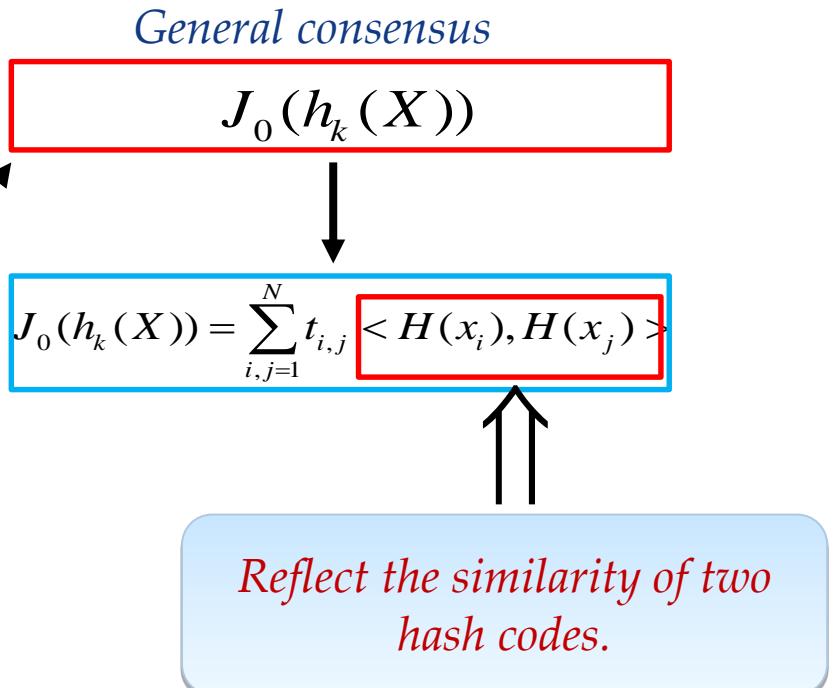
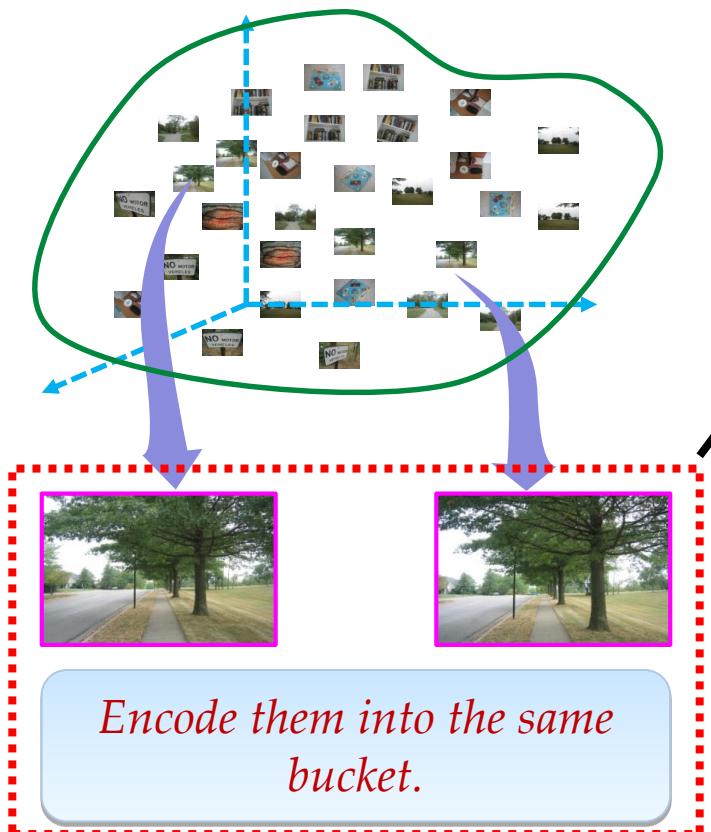
Encode them into the same
bucket.

Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

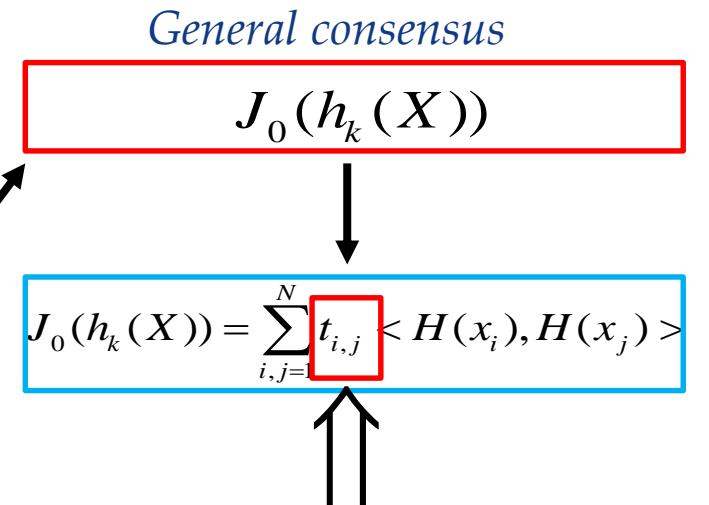
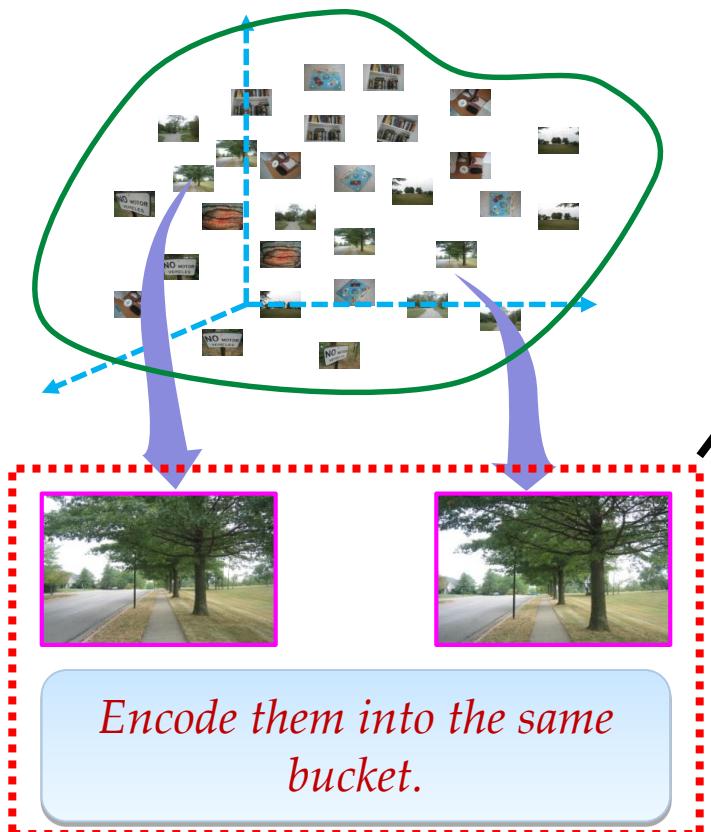


Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer



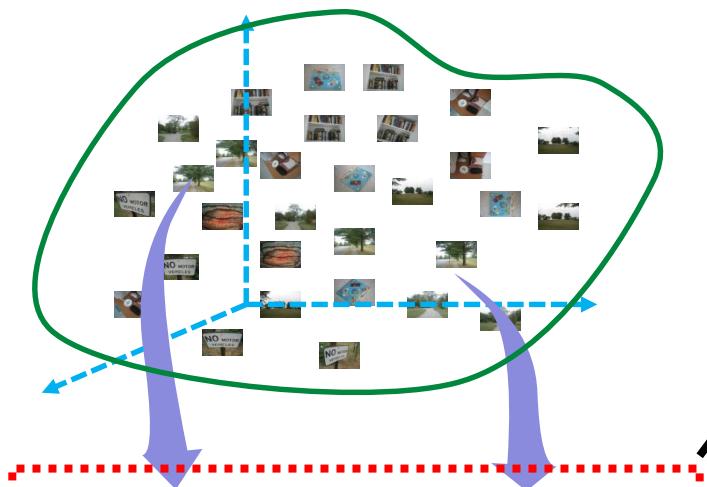
Weight element denoted the same in complementary hashing.

Step 1:

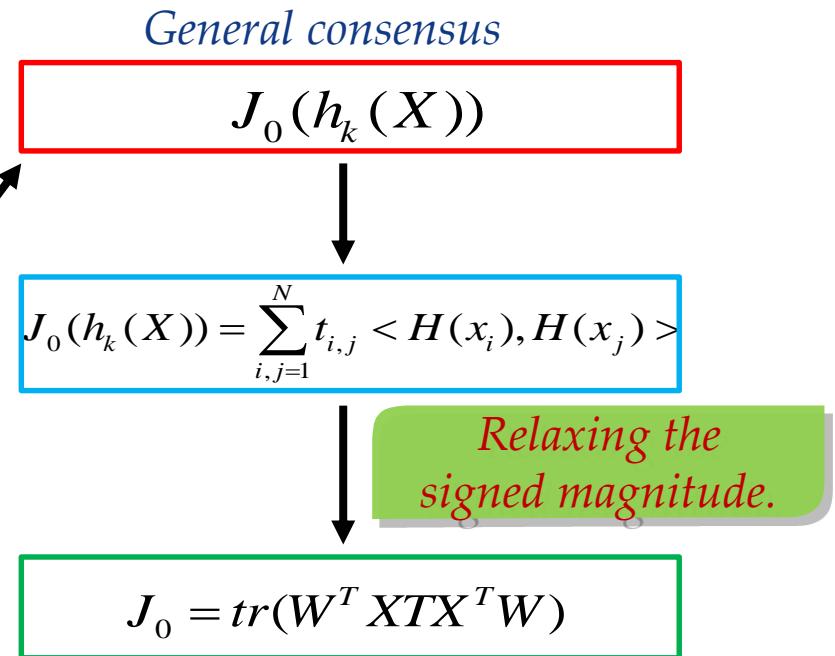
■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer



Encode them into the same bucket.





Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

Each bit requires a 50% chance of being one or zero.



Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

Each bit requires a 50% chance of being one or zero.

$$\sum_{i=1}^N H(x_i) = 0$$



Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

Each bit requires a 50% chance of being one or zero.

$$\sum_{i=1}^N H(x_i) = 0$$

Maximize the variance of hash functions.

$$\max_W \text{tr}(W^T X X^T W)$$

Entropy regularizer [1].

[1]. J. Wang, O. Kumar, and S.F. Chang, "Semi-supervised hashing for scalable image retrieval," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3424-3431, 2010.

Step 1:

■ Hash Function:

$$h_k(x) = \text{sgn}(w_k^T x_i + b_k), \quad b_k = E[w_k^T x_i] \xrightarrow{0} \Rightarrow h_k(x) = \text{sgn}(w_k^T x_i)$$

■ General consensus & entropy regularizer

Each bit requires a 50% chance of being one or zero.

$$\sum_{i=1}^N H(x_i) = 0$$

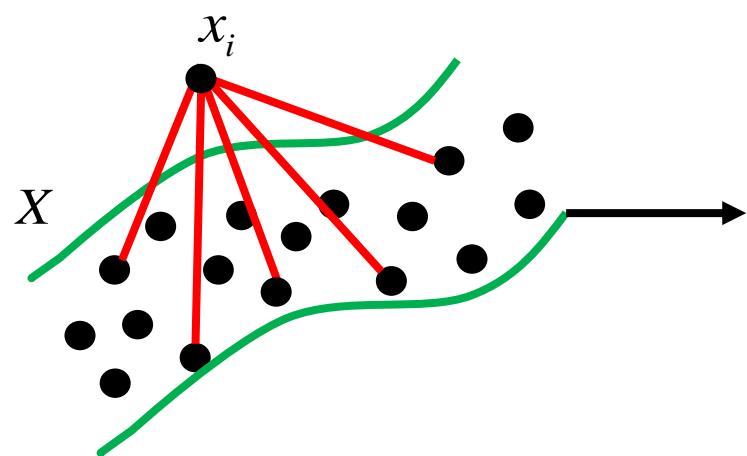
Maximize the variance of hash functions.

$$\max_W \text{tr}(W^T X X^T W)$$

Combining the general consensus and the entropy regularizer.

$$J_1 = \max_W \text{tr}(W^T X X^T W + \lambda W^T X X^T W)$$

Step 2:



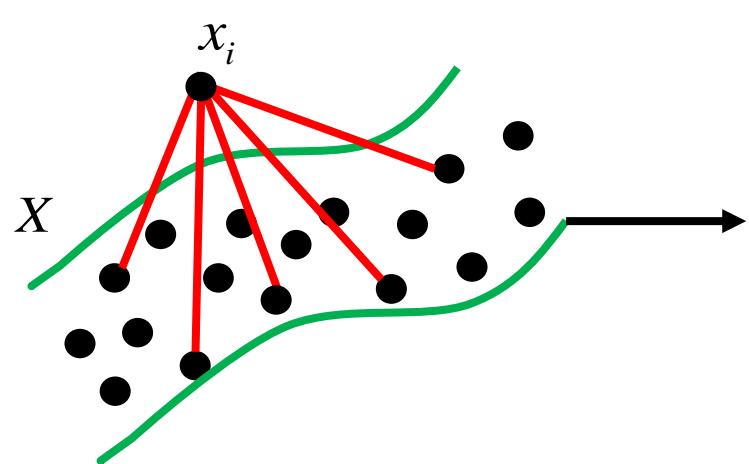
Sparsity preserving projections (SPP)

$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^N \sum_{j=1}^N (s_i - s_j)^2 A_{ij}$$

$$s.t. \quad x_i = Xs_i, i = 1, 2, \dots, N$$

$$1 = 1^T s_i$$

Step 2:



Sparsity preserving projections (SPP)

$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^N \sum_{j=1}^N (s_i - s_j)^2 A_{ij}$$

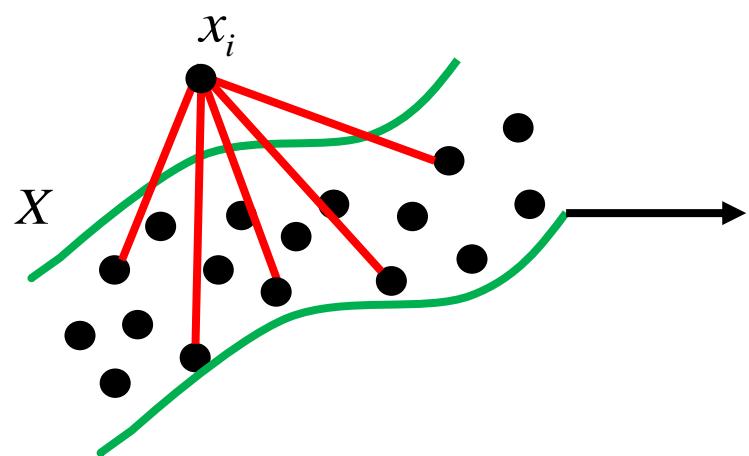
$$s.t. \quad x_i = Xs_i, i = 1, 2, \dots, N$$

$$1 = 1^T s_i$$

$$\min_{h_k, S} \sum_{k=1}^K \sum_{i=1}^N \| h_k(x_i) - h_k(Xs_i) \|^2 + \alpha \| S \|_{2,1} + \beta \text{tr}(SMS^T)$$



Step 2:



Sparsity preserving projections (SPP)

$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^N \sum_{j=1}^N (s_i - s_j)^2 A_{ij}$$

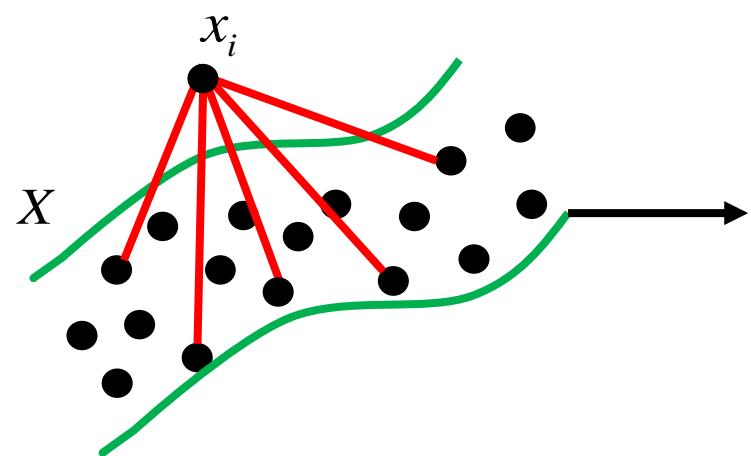
$$s.t. \quad x_i = Xs_i, i = 1, 2, \dots, N$$

$$1 = 1^T s_i$$

$$\min_{h_k, S} \sum_{k=1}^K \sum_{i=1}^N \| h_k(x_i) - h_k(Xs_i) \|^2 + \alpha \| S \|_{2,1} + \beta \text{tr}(S M S^T)$$

M is the graph Laplacian matrix.

Step 2:



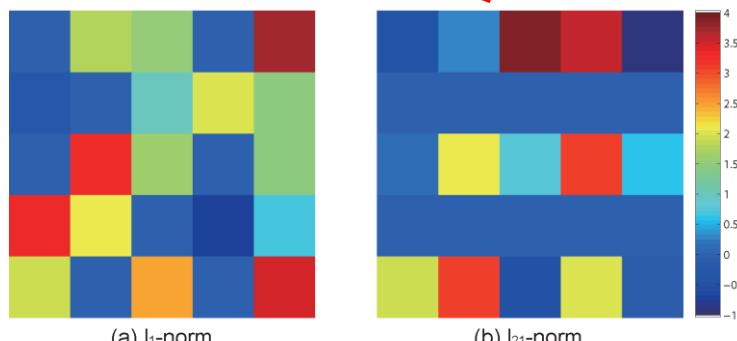
Sparsity preserving projections (SPP)

$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^N \sum_{j=1}^N (s_i - s_j)^2 A_{ij}$$

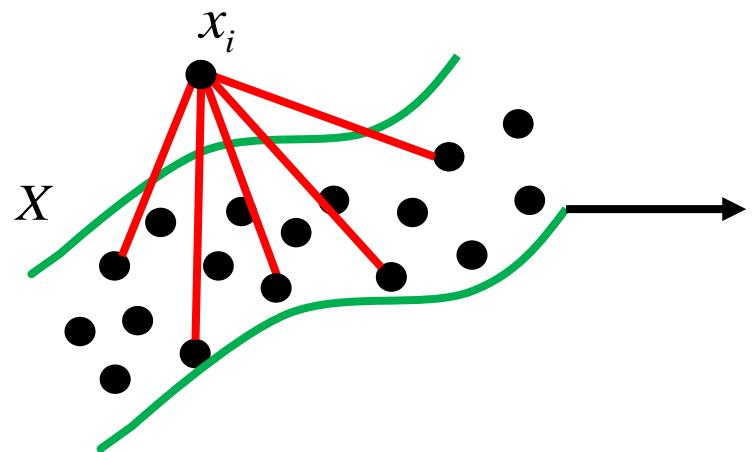
$$s.t. \quad x_i = Xs_i, i = 1, 2, \dots, N$$

$$1 = 1^T s_i$$

$$\min_{h_k, S} \sum_{k=1}^K \sum_{i=1}^N \| h_k(x_i) - h_k(Xs_i) \|^2 + \alpha \| S \|_{2,1} + \beta \text{tr}(SMS^T)$$



Step 2:



Sparsity preserving projections (SPP)

$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^N \sum_{j=1}^N (s_i - s_j)^2 A_{ij}$$

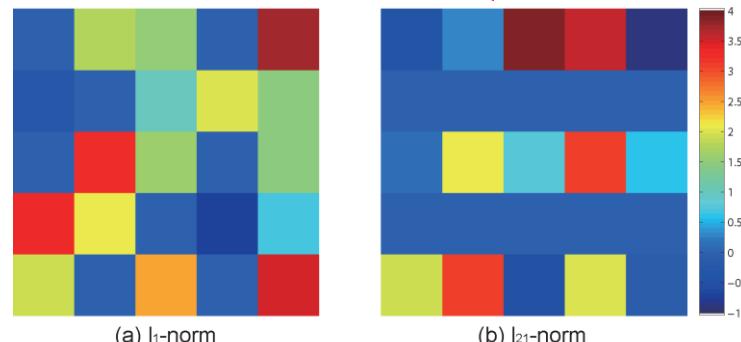
$$s.t. \quad x_i = Xs_i, i = 1, 2, \dots, N$$

$$1 = 1^T s_i$$

$$\min_{h_k, S} \sum_{k=1}^K \sum_{i=1}^N \| h_k(x_i) - h_k(Xs_i) \|^2 + \alpha \| S \|_{2,1} + \beta \text{tr}(SMS^T)$$

$$J_2 = \min_{W, S} \text{tr}(W^T X V X^T W) + \alpha \text{tr}(S U S^T) + \beta \text{tr}(S M S^T)$$

$$V = (I - S - S^T + S^T S), \quad U_{ii} = \frac{1}{2 \| s_i \|}$$



Step 3:

■ Final objective function:

$$J(W, S) = \frac{J_2}{J_1} = \min_{W, S} \frac{\text{tr}(W^T X V X^T W) + \alpha \text{tr}(S U S^T) + \beta \text{tr}(S M S^T)}{\text{tr}\{W^T (X T X^T + \lambda X X^T) W\}}$$



Incorporates to build a **balance** between the maximum of empirical accuracy combined with the information theory, manifold structure and the minimum of sparse reconstruction of data points.



Optimization

- Step 1: Fix S and update W :

$$J(W) = \min_W \frac{\text{tr}(W^T X V X^T W)}{\text{tr}(W^T X (T + \lambda I) X^T W)}$$

Generalized eigenvalue decomposition problem.

- Step 2: Fix W and update S :

$$J(S) = \min_S \text{tr}(W^T X V X^T W) + \alpha \text{tr}(S U S^T) + \beta \text{tr}(S M S^T)$$

Take derivation with respect of S .

Algorithm

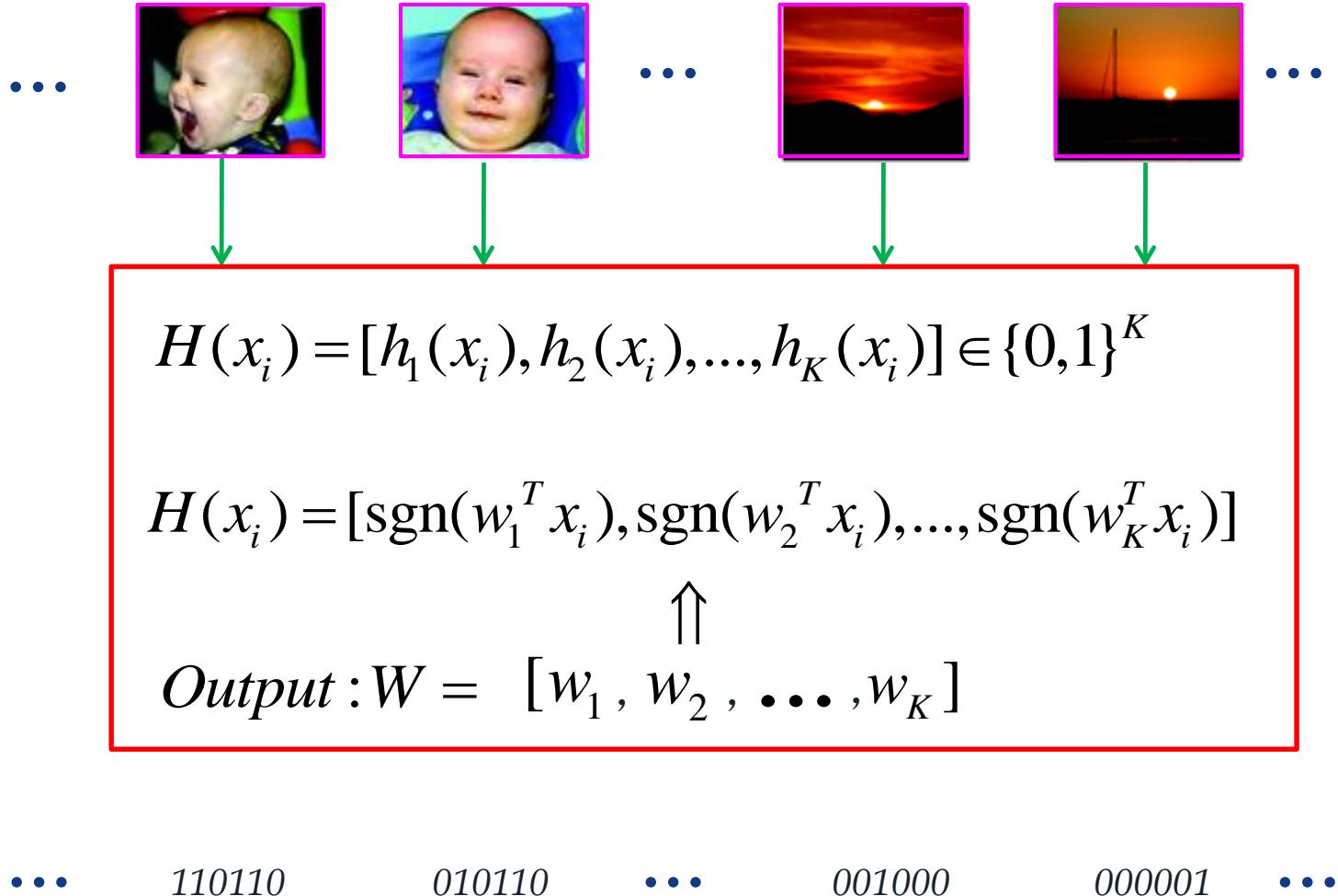
Algorithm 1 Sparse Reconstruction Hashing

Input: A set of training data X (zero-centered), length of hash codes K , the parameters α, β, η .

Output: the sparse weight matrix S , the map matrix W .

- 1: Initialize $S_0 = I_{N \times N}$, and compute the diagonal matrix U , the i th element of $U_{ii} = \frac{1}{\|s_i\|_2 + \zeta}$.
- 2: **repeat**
- 3: Computer the map matrix W by using generalized eigenvalue decomposition problem in Eq.13.
- 4: Computer the sparse matrix S by Eq.15.
- 5: Update the diagonal matrix U .
- 6: **until** convergence

Encode Binary Codes





Roadmap

■ *Overview*

■ *Method*

■ *Experiments*



■ *Summary*

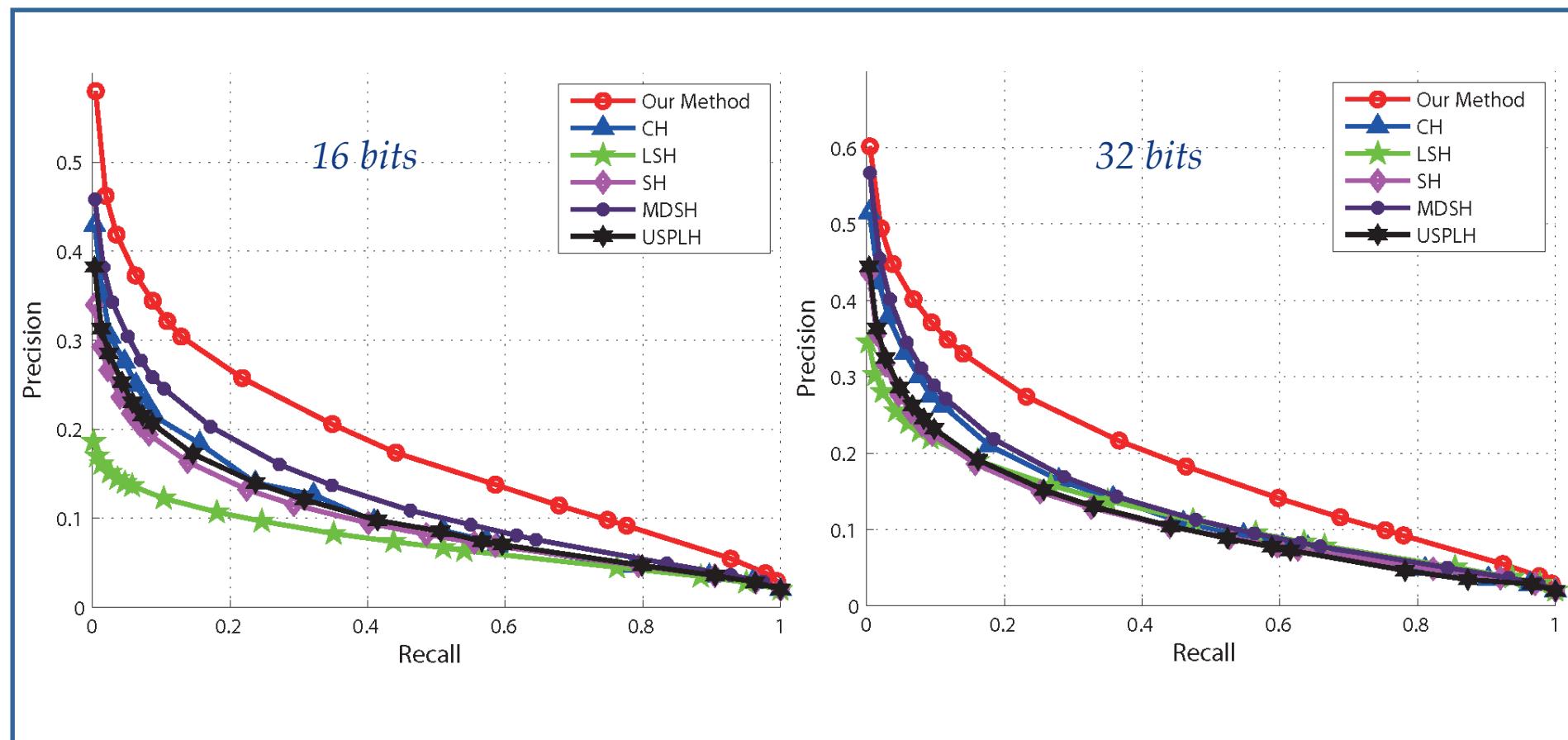


Experiments

- **Datasets:**
 - *Cifar-10 Dataset : 60k*320d Gist.*
 - *Tiny100k Dataset : 100K*384d Gist.*
- **Evaluation protocol:**
 - *For each query, find all points within a certain hamming radius or find M nearest neighbors by exhaustive hamming ranking.*
- **Settings:**
 - *Cifar-10 data*
Unsupervised: 59k for training (database), 1K for query.
 - *Tiny100k images data*
Unsupervised: 96k for training (database), 4K for query.
 - *Parameter selection*
Alpha: 0.5, beta: 0.1.

Experiments

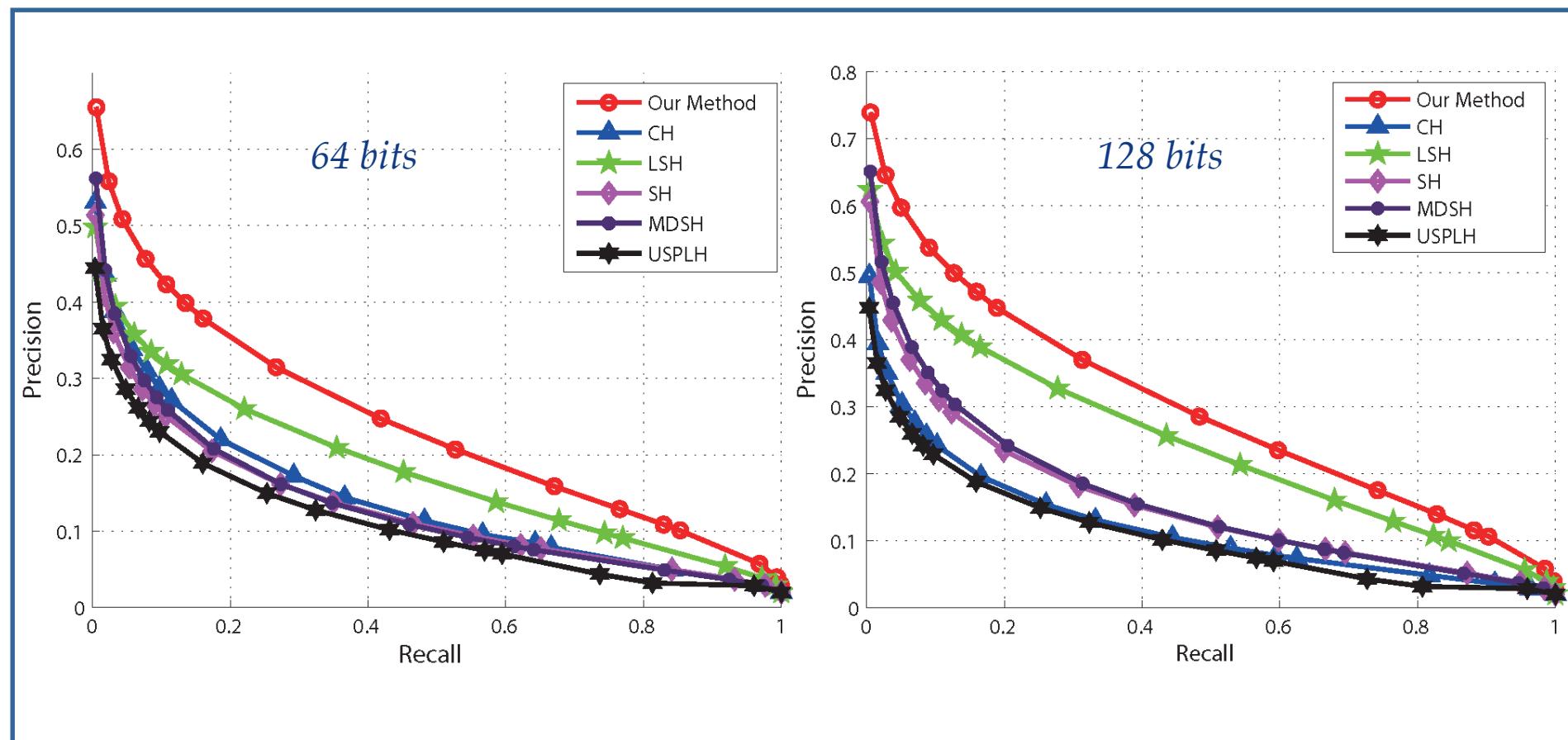
Cifar-10 precision recall curve





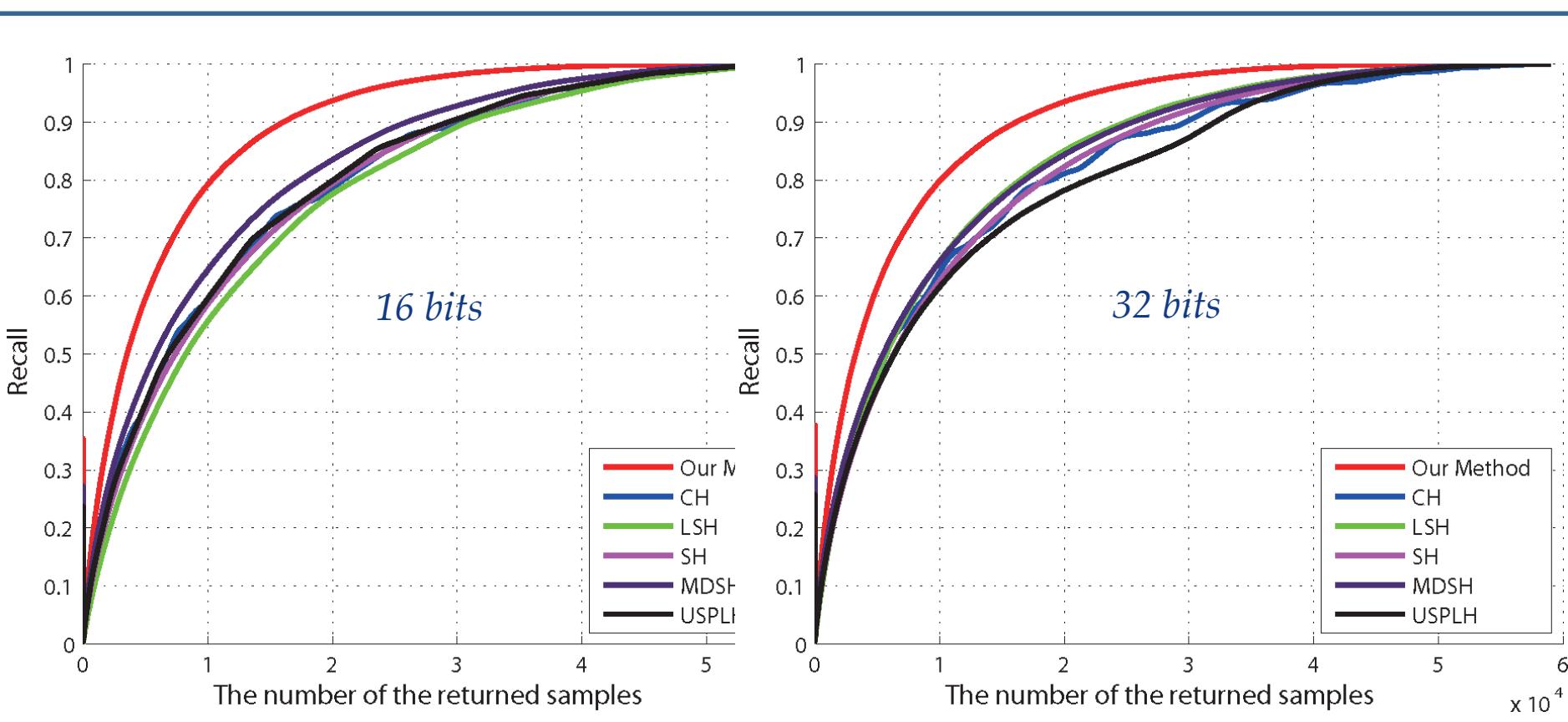
Experiments

Cifar-10 precision recall curve



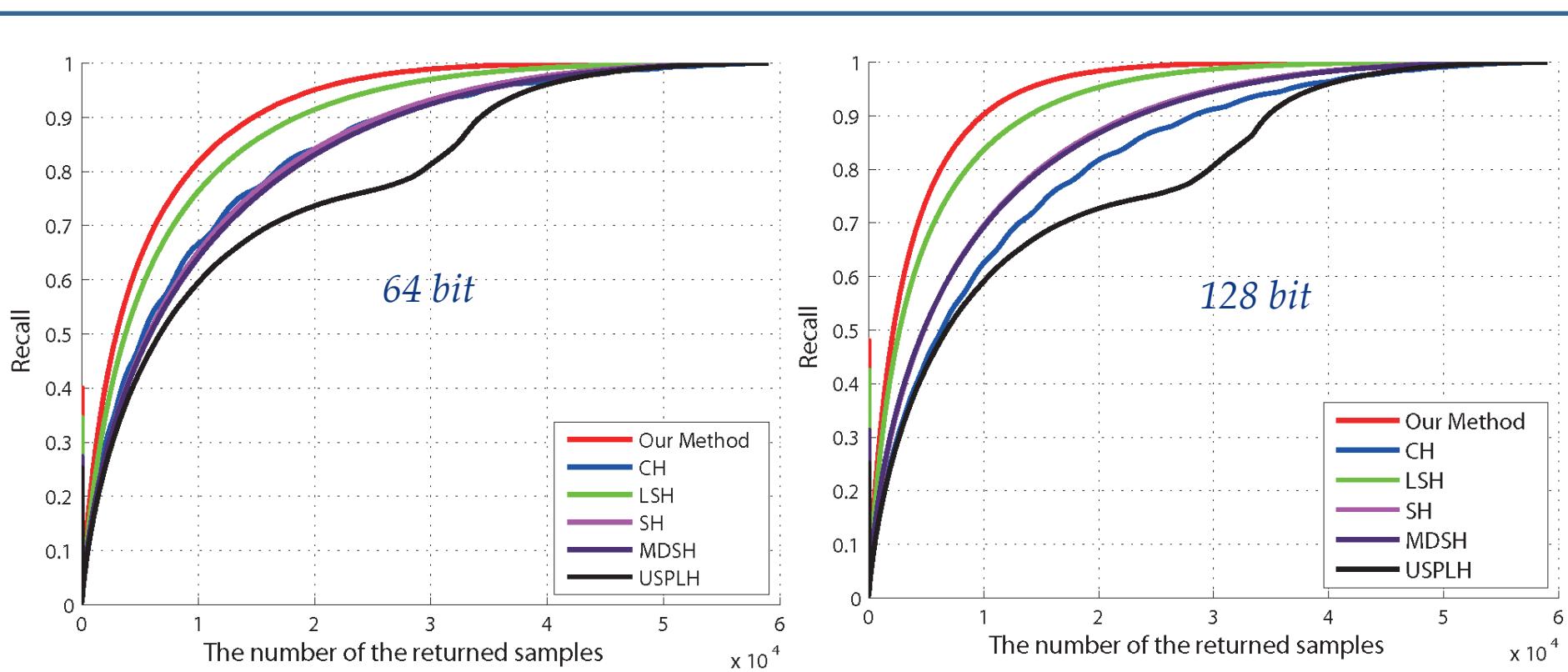
Experiments

Cifar-10 recall-the number of return samples curve



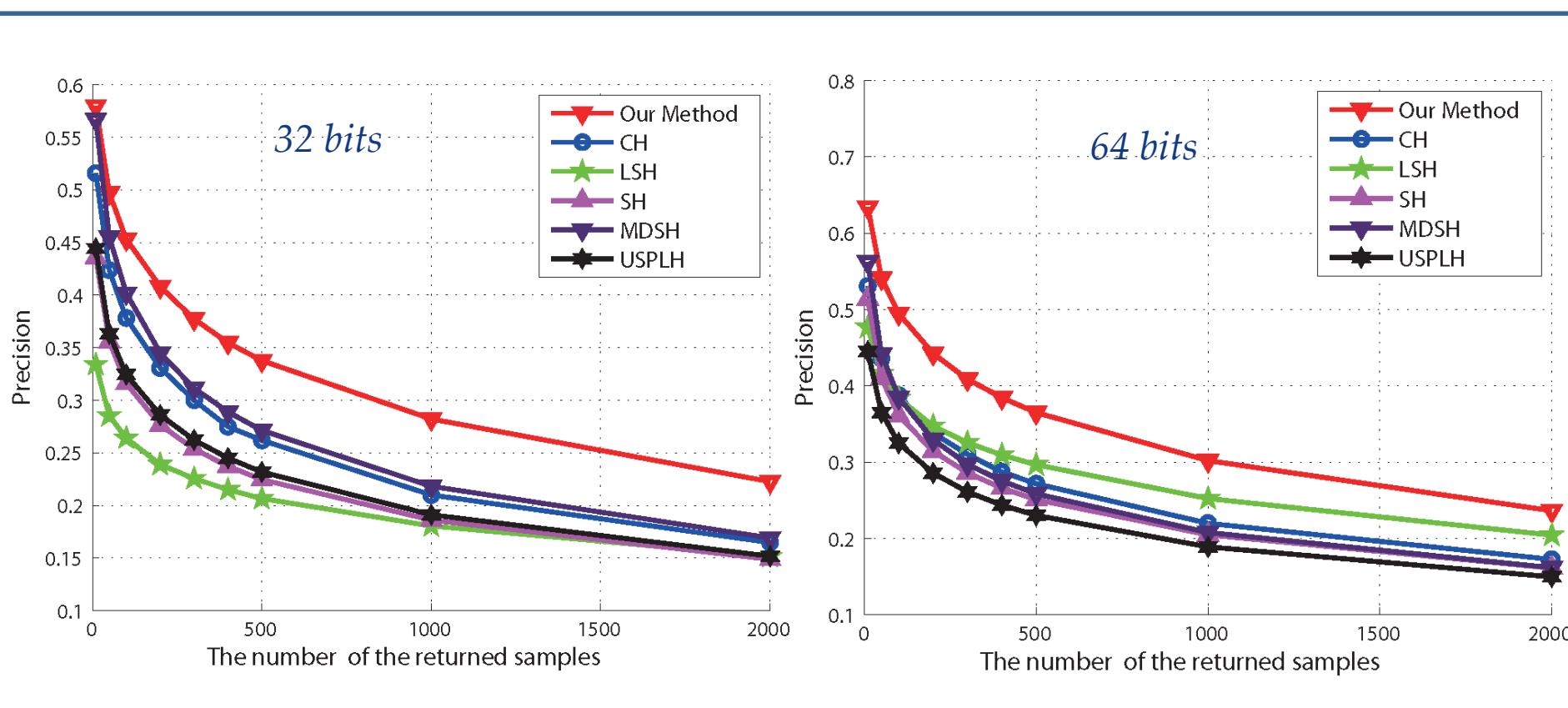
Experiments

Cifar-10 recall-the number of return samples curve



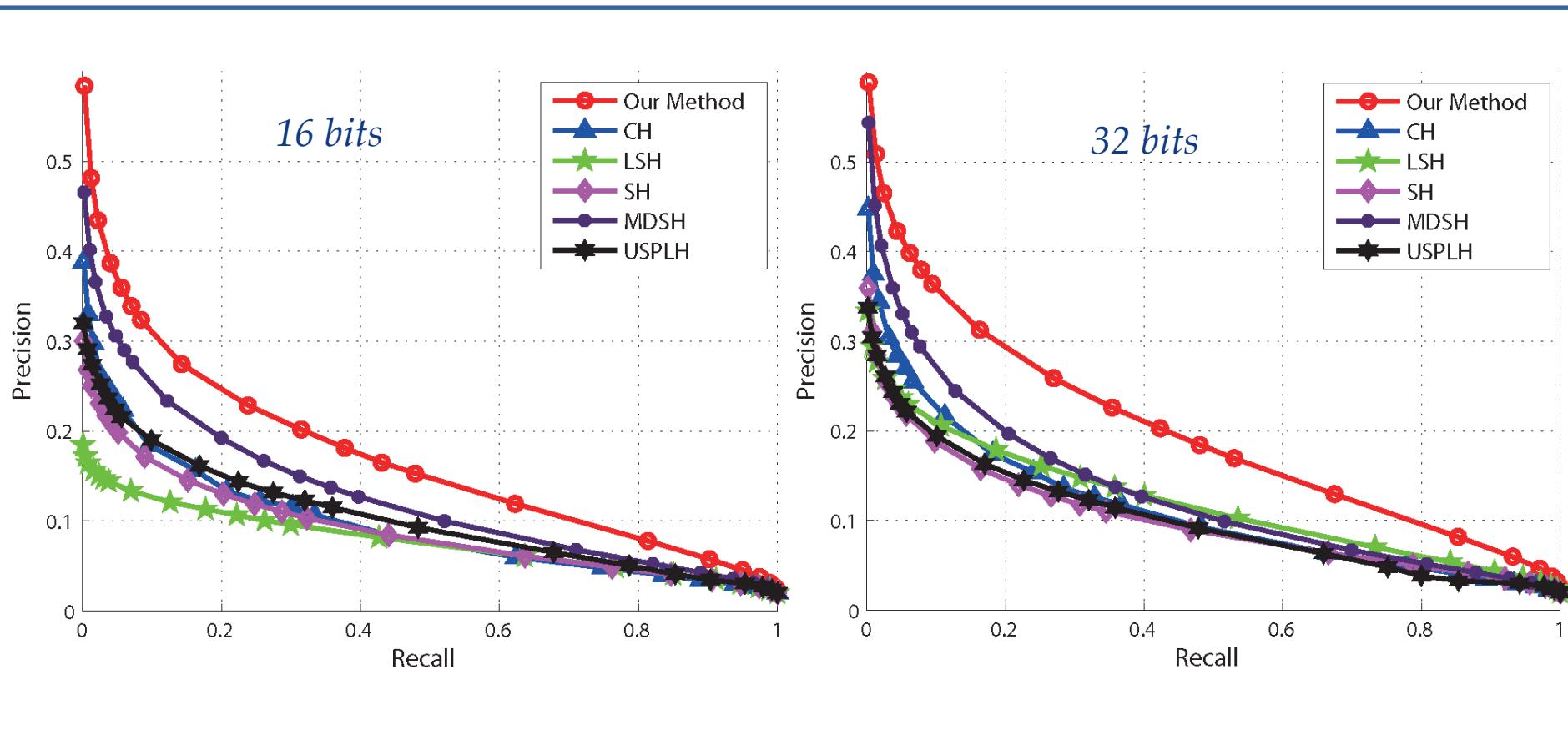
Experiments

Cifar-10 precision-the number of return samples curve



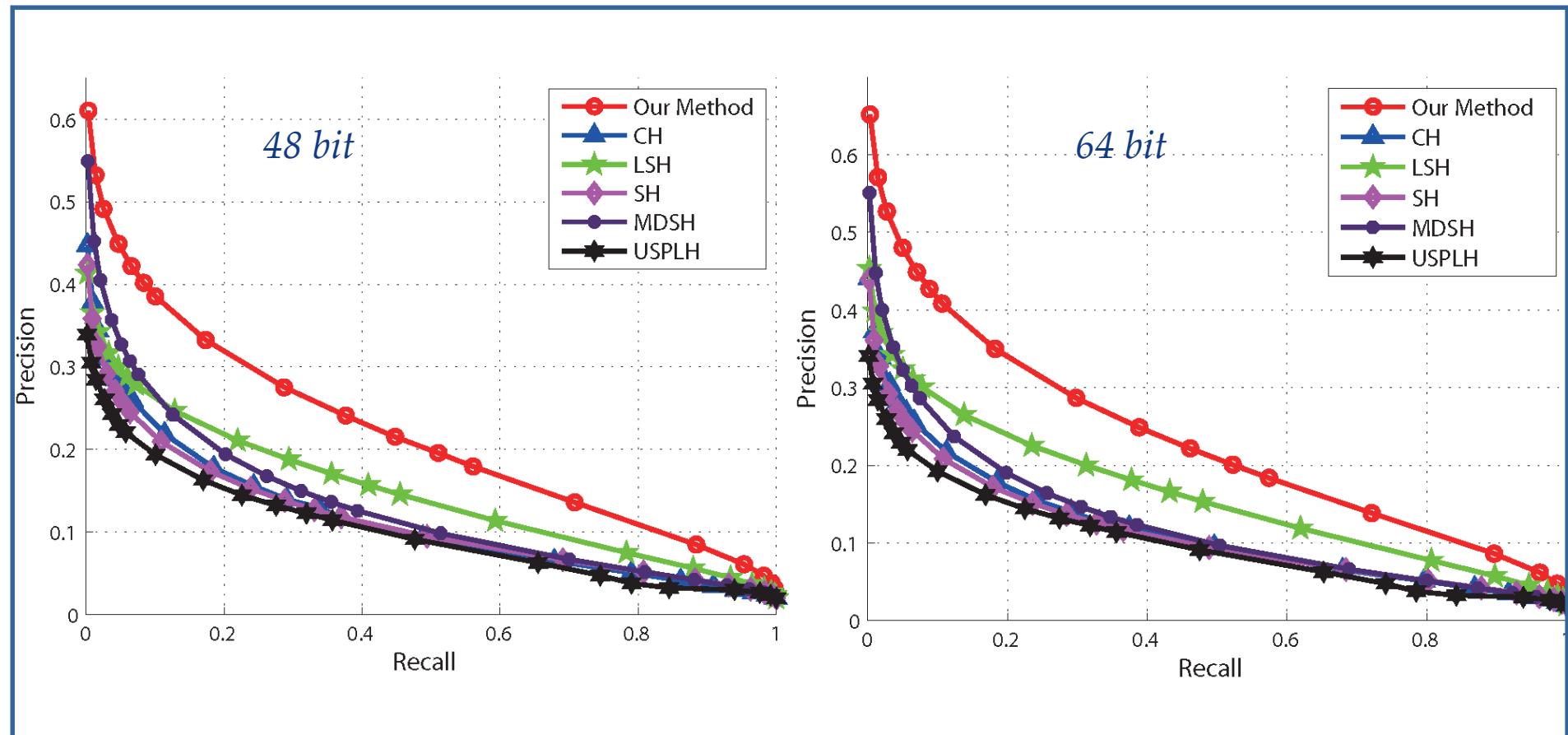
Experiments

Tiny100k precision recall curve



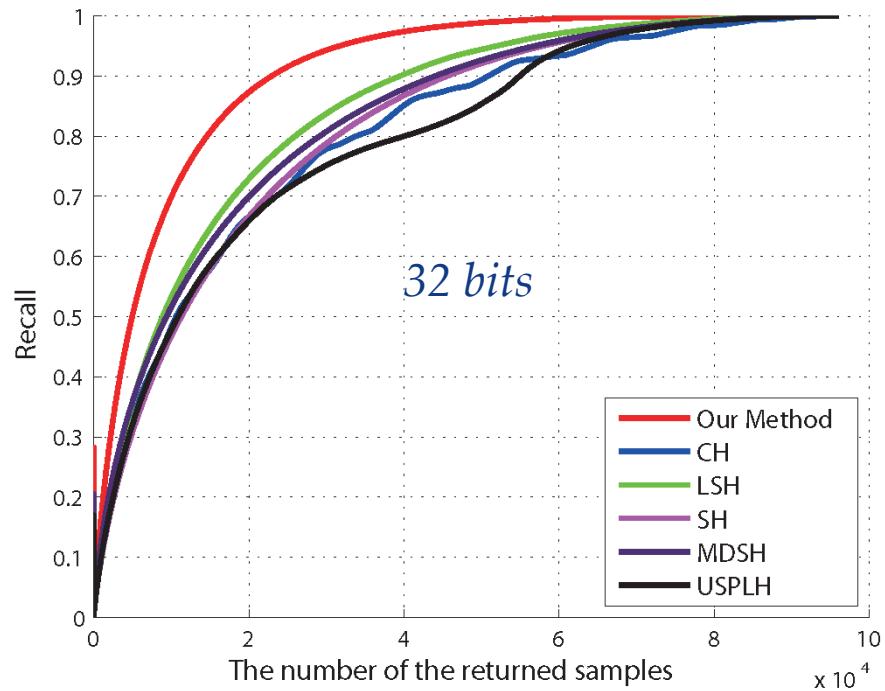
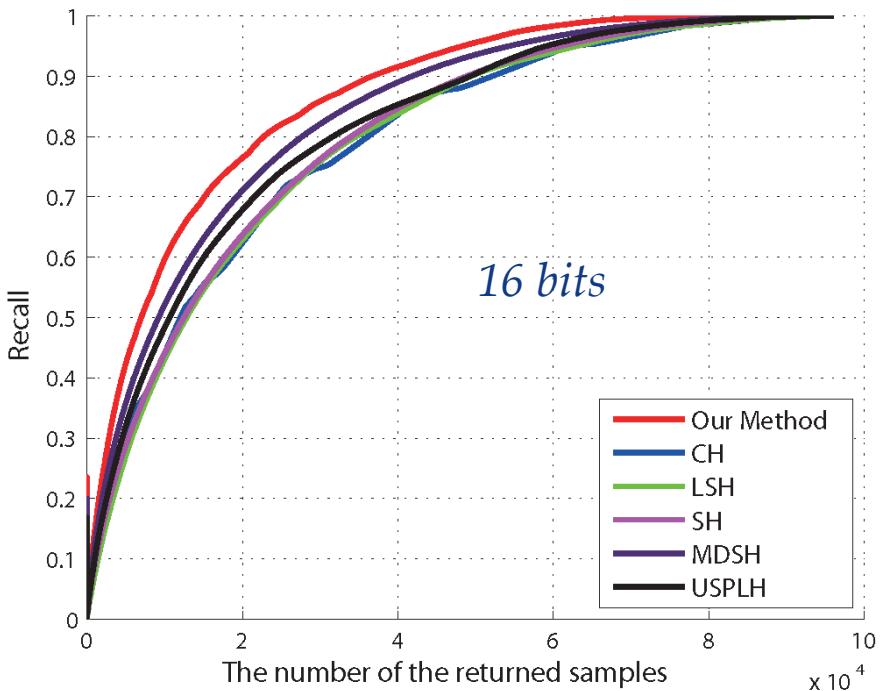
Experiments

Tiny100k precision recall curve



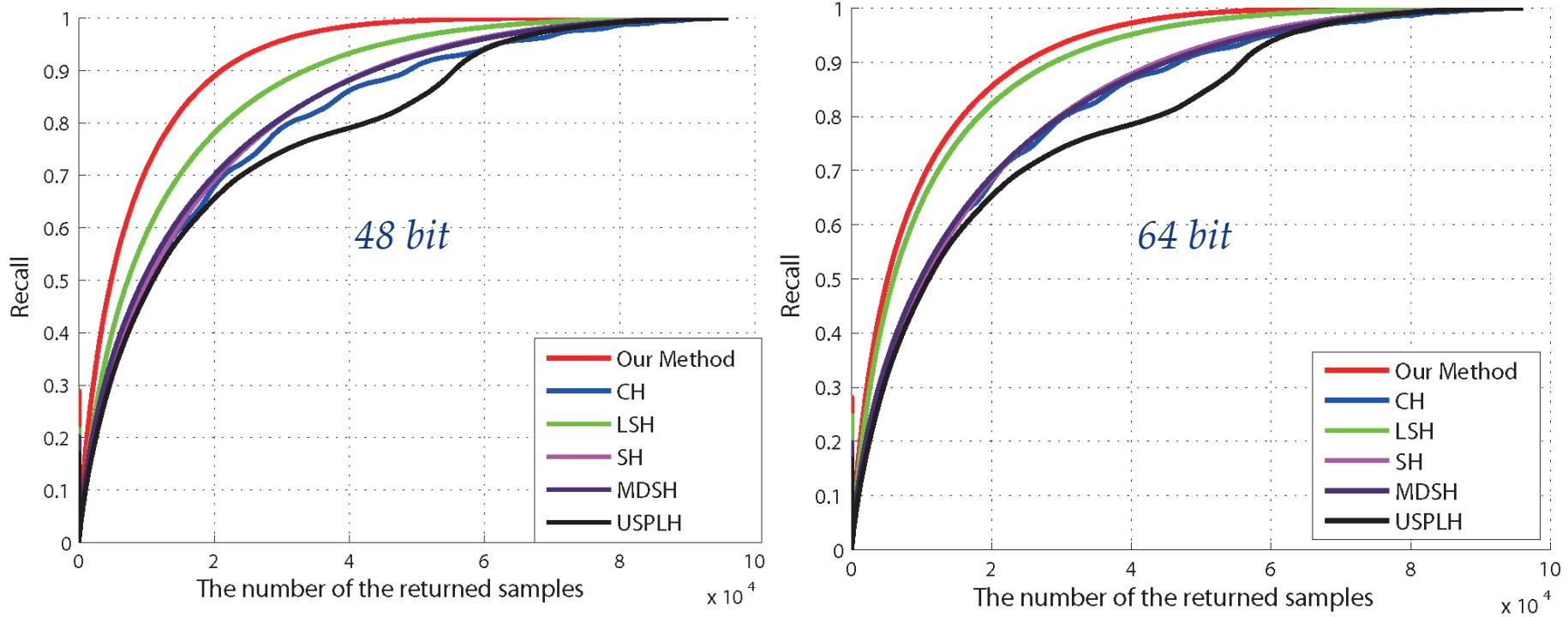
Experiments

Tiny100k recall-the number of return samples curve



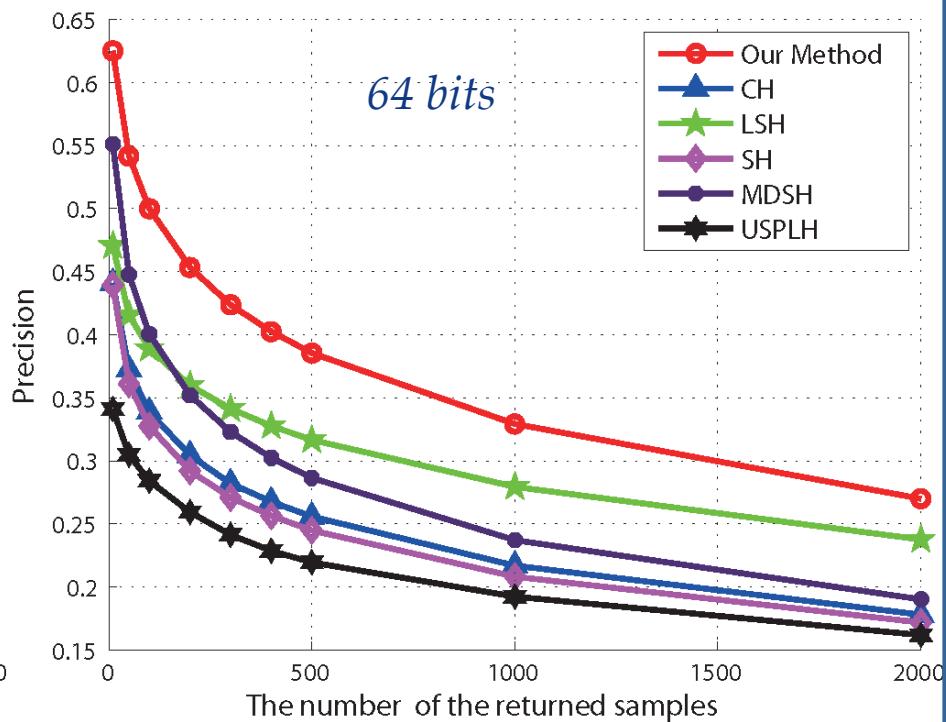
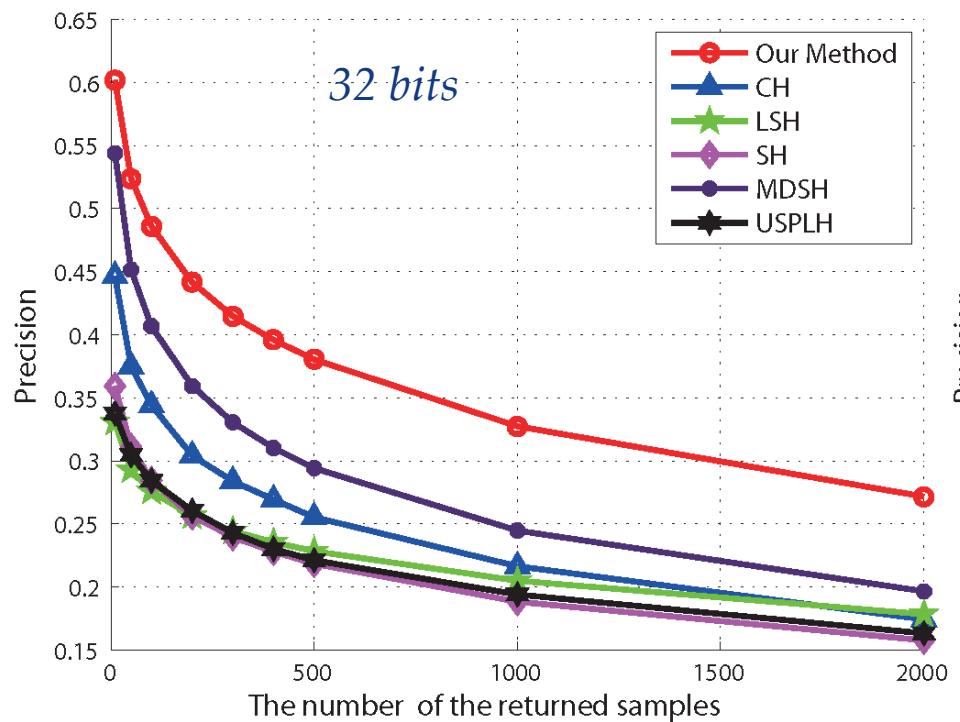
Experiments

Tiny100k recall-the number of return samples curve



Experiments

Tiny100k precision-the number of return samples curve



Experiments

Our method



SH



MDSH



Query image



LSH



CH

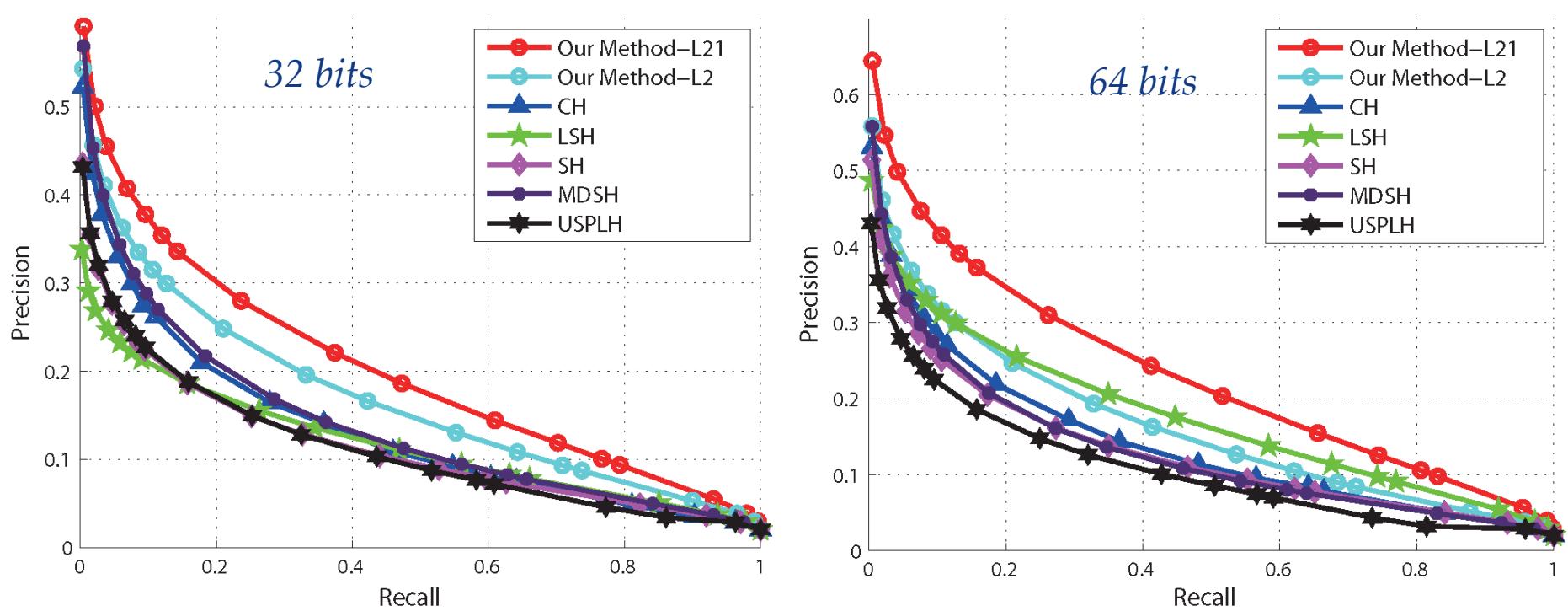


USPLH



L21 Norm VS L2

Cifar-10 precision recall curve





Roadmap

- *Overview*
- *Method*
- *Experiments*
- *Summary*



Summary

➤ Contributions

A **novel sparse reconstruction** is proposed by introducing the $l_{2,1}$ -norm into SSP framework to preserve **potential discrimination information** of hash functions.

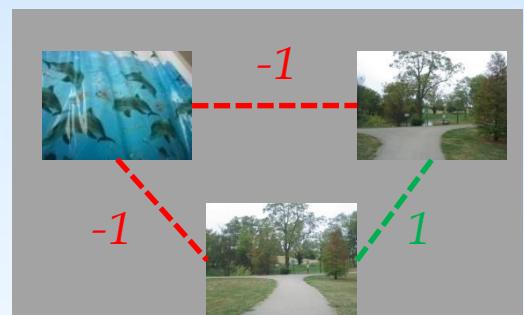
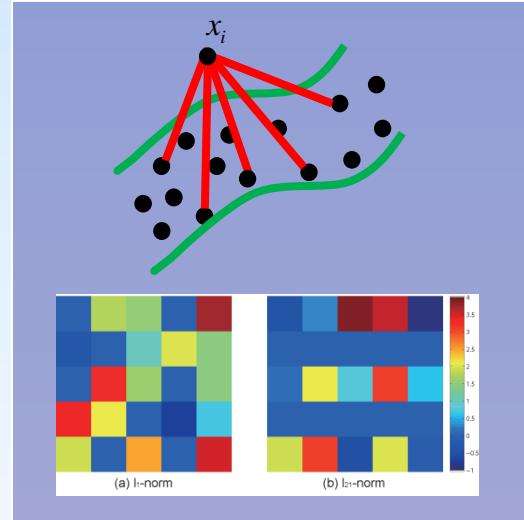


Build a **balance** between the maximum of empirical accuracy combined with the **information theory**, **manifold structure** and the minimum of **sparse reconstruction** of data points.

➤ Future work



More efficient structure models and **labels** **information** will be explored to solve ANN search problem.



OPTIMAL

Center for OPTical IMagery Analysis and Learning, Xi'an, China

光學影像分析與學習中心 · 中國西安 ·



Thank You

Q & A

中國科學院西安光學精密機械研究所
瞬態光學與光子技術國家重點實驗室