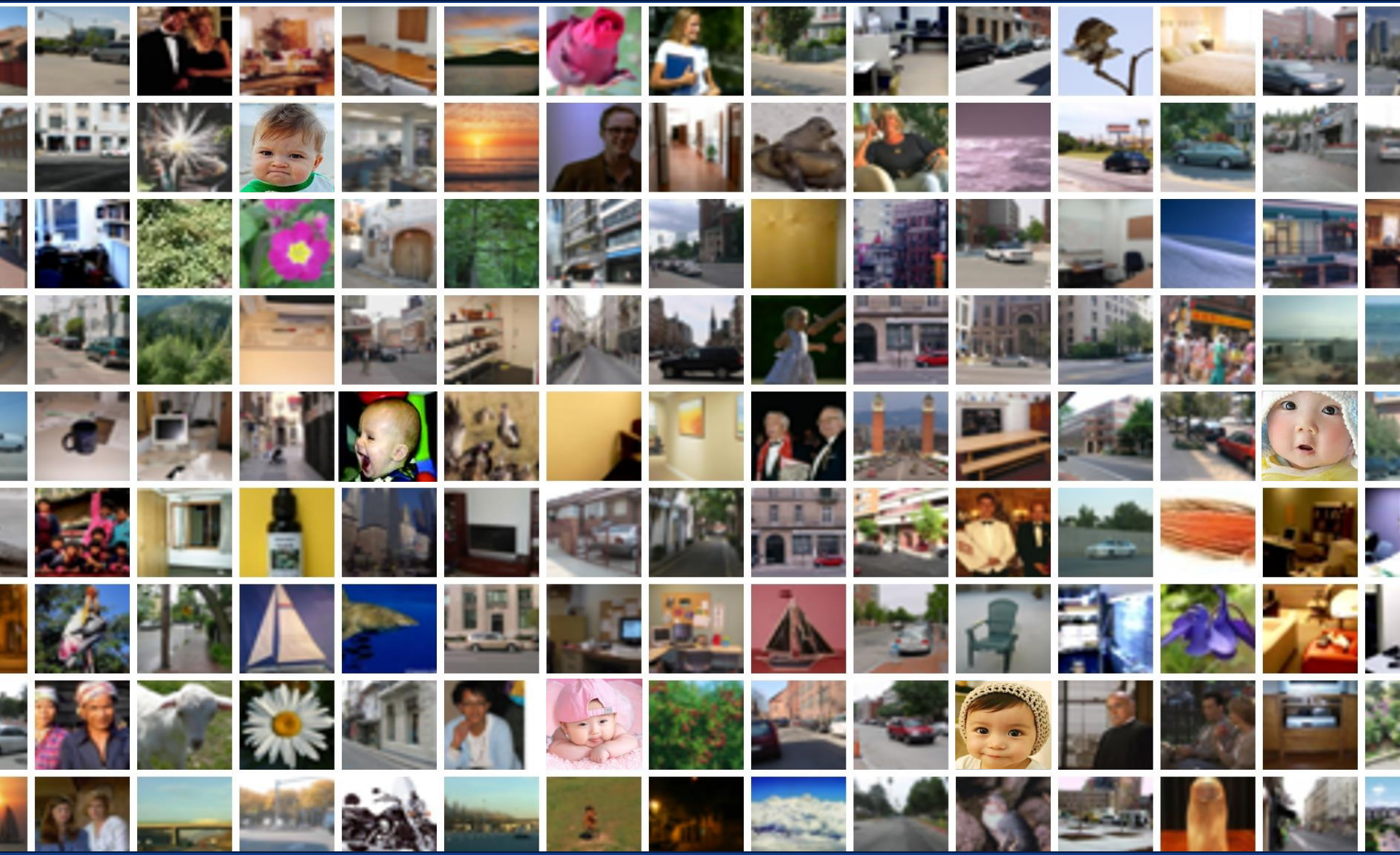
Learning Hash Functions Using Sparse Reconstruction

Yong Yuan, Xiaoqiang Lu, and Xuelong Li
Center for OPTical IMagery Analysis and Learning (OPTIMAL)

Roadmap

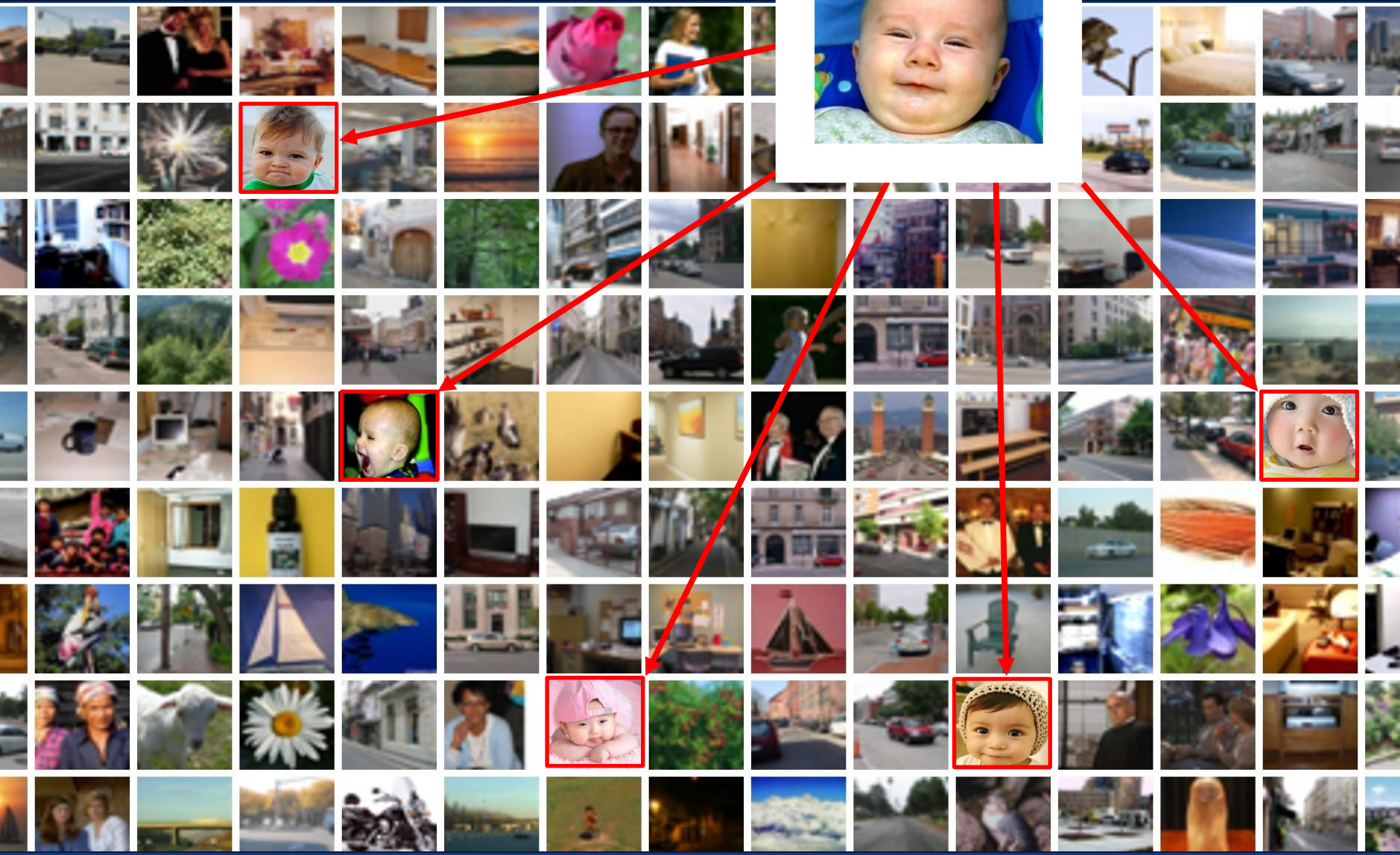
- *Overview* 
- *Method*
- *Experiments*
- *Summary*

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

…   …   …

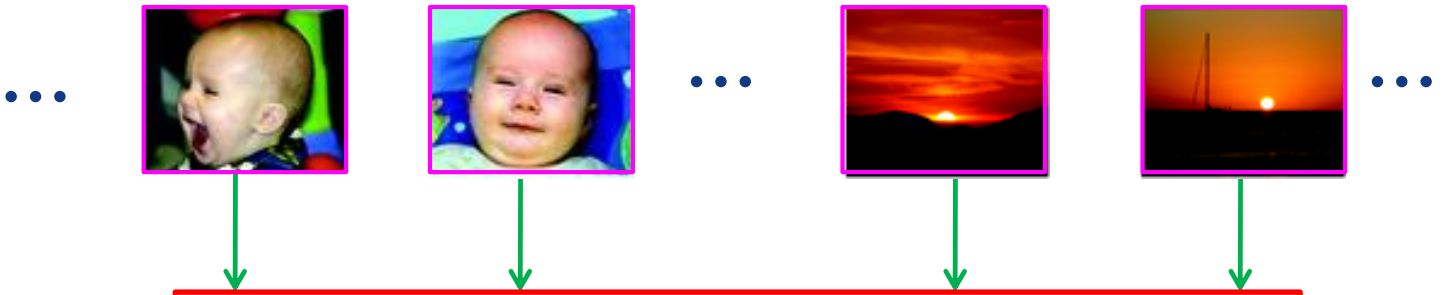
Similarity Preserving Binary Hashing




$$H(x_i) = [h_1(x_i), h_2(x_i), ..., h_K(x_i)] \in \{0,1\}^K$$

110110 010110 001000 000001

Similarity Preserving Binary Hashing

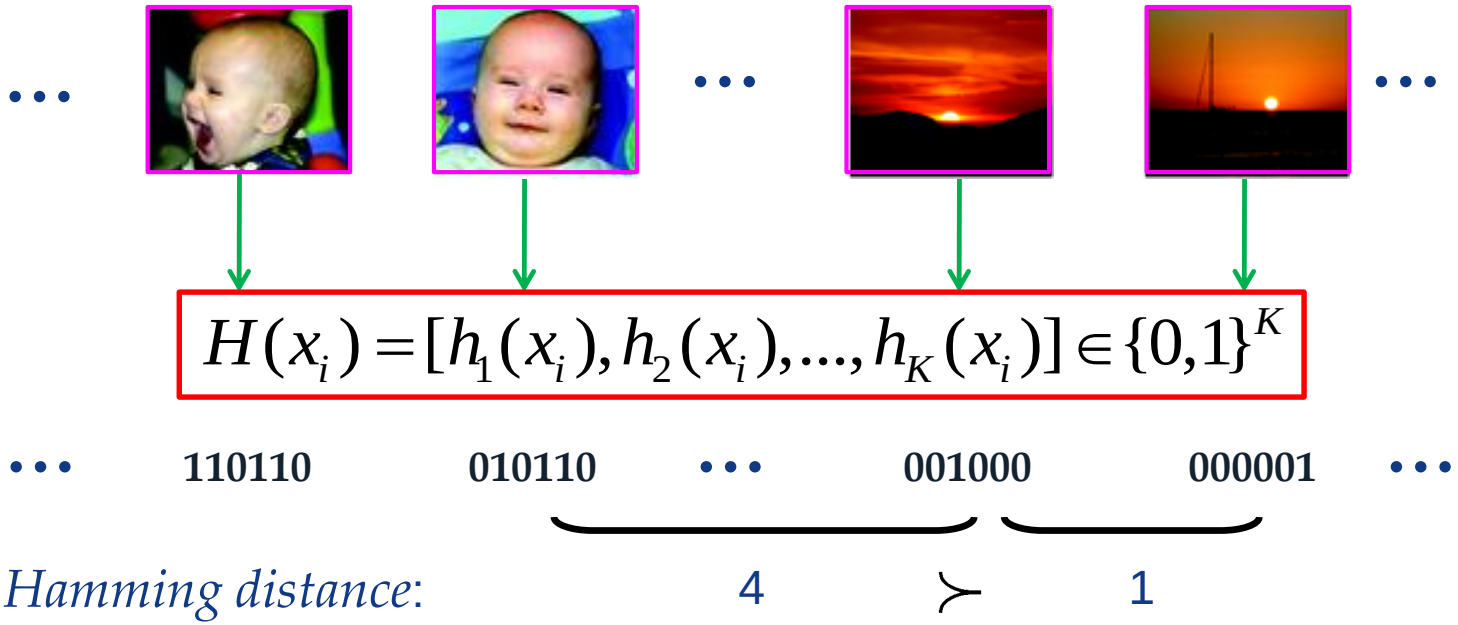


$$H(x_i) = [h_1(x_i), h_2(x_i), ..., h_K(x_i)] \in \{0,1\}^K$$

… 110110 010110 … 001000 000001 …

Hamming distance: 4 $\succ$ 1

Similarity Preserving Binary Hashing



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

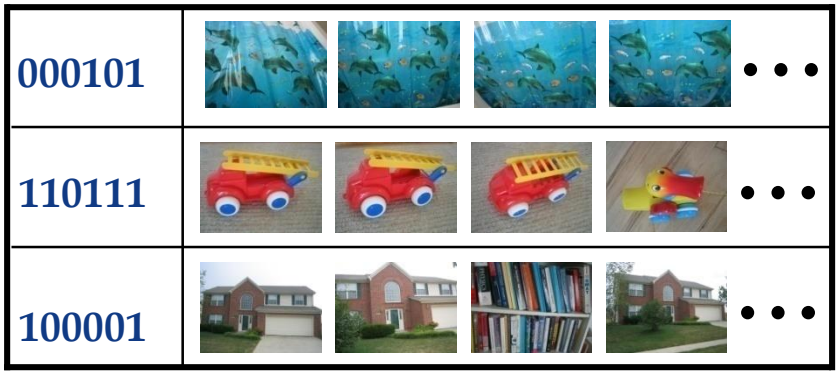
Hashing Using Compact Codes

Dataset

Hashing Using Compact Codes

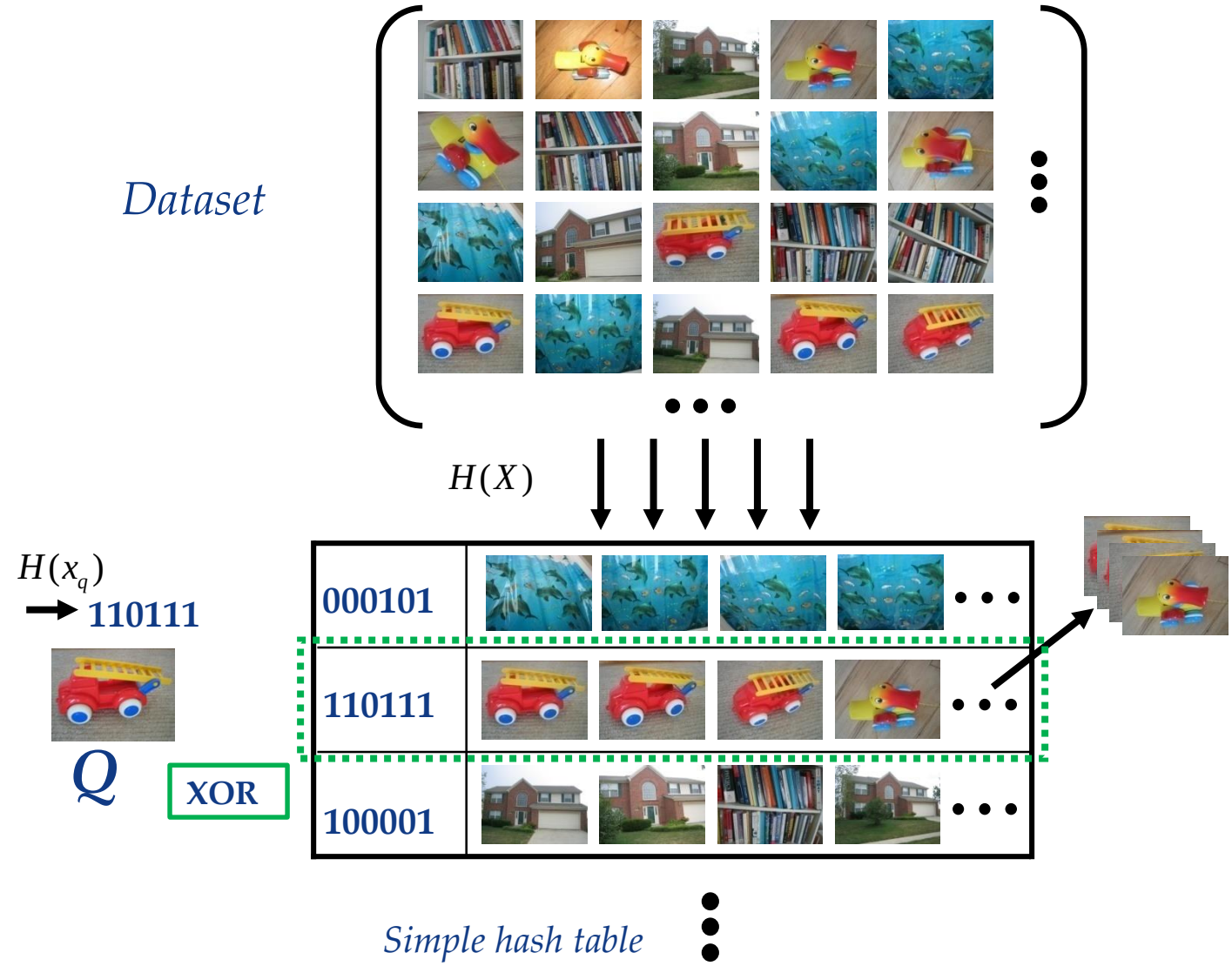
Dataset

$H(X)$

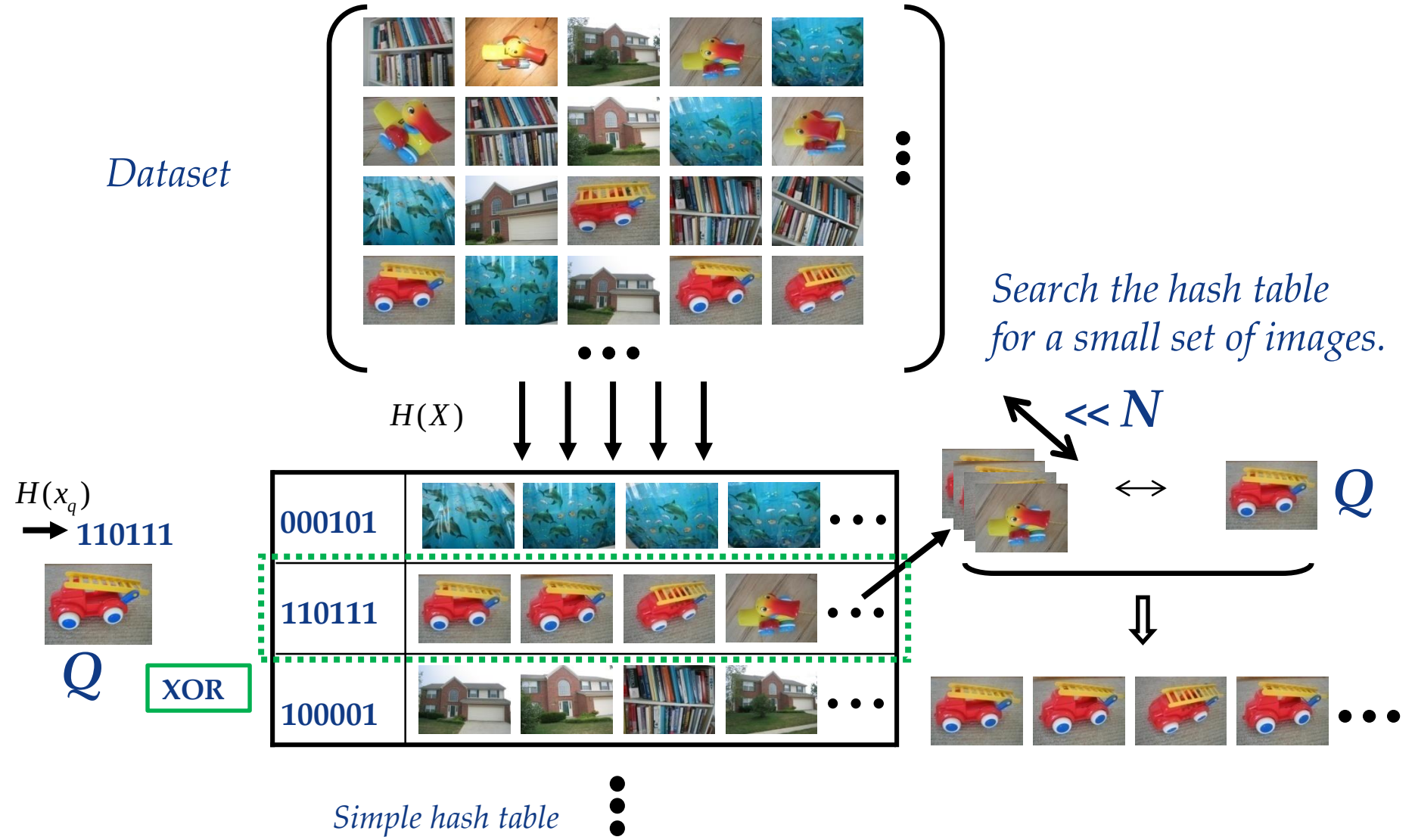


Simple hash table

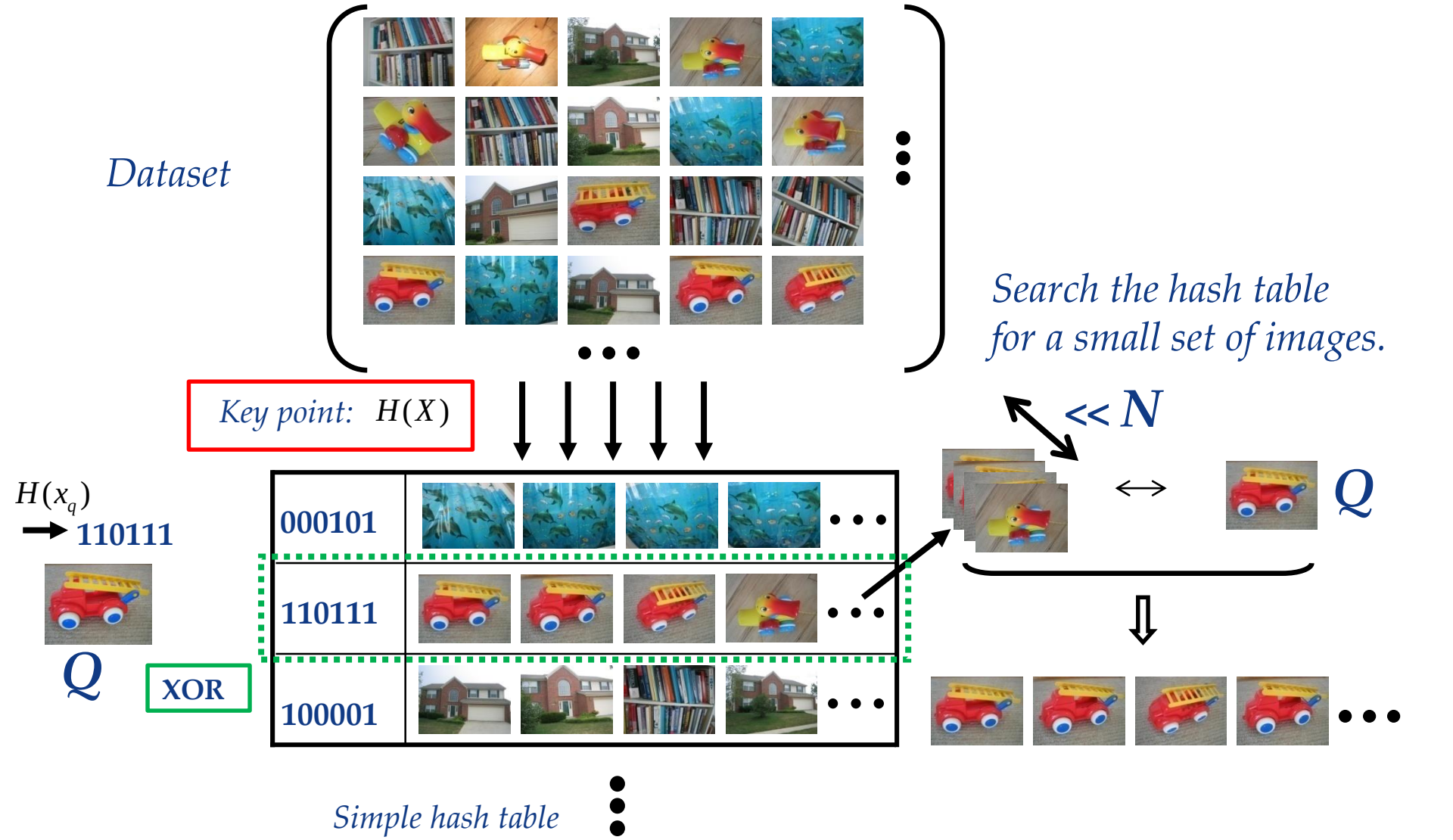
Hashing Using Compact Codes



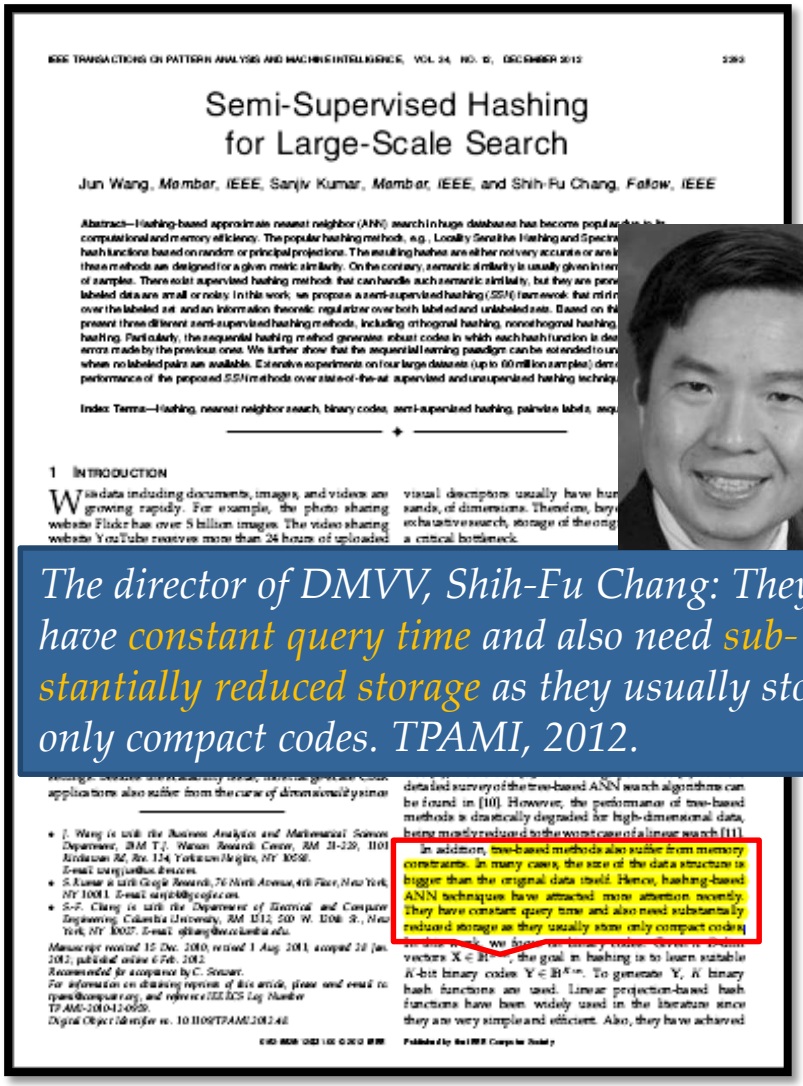
Hashing Using Compact Codes



Hashing Using Compact Codes



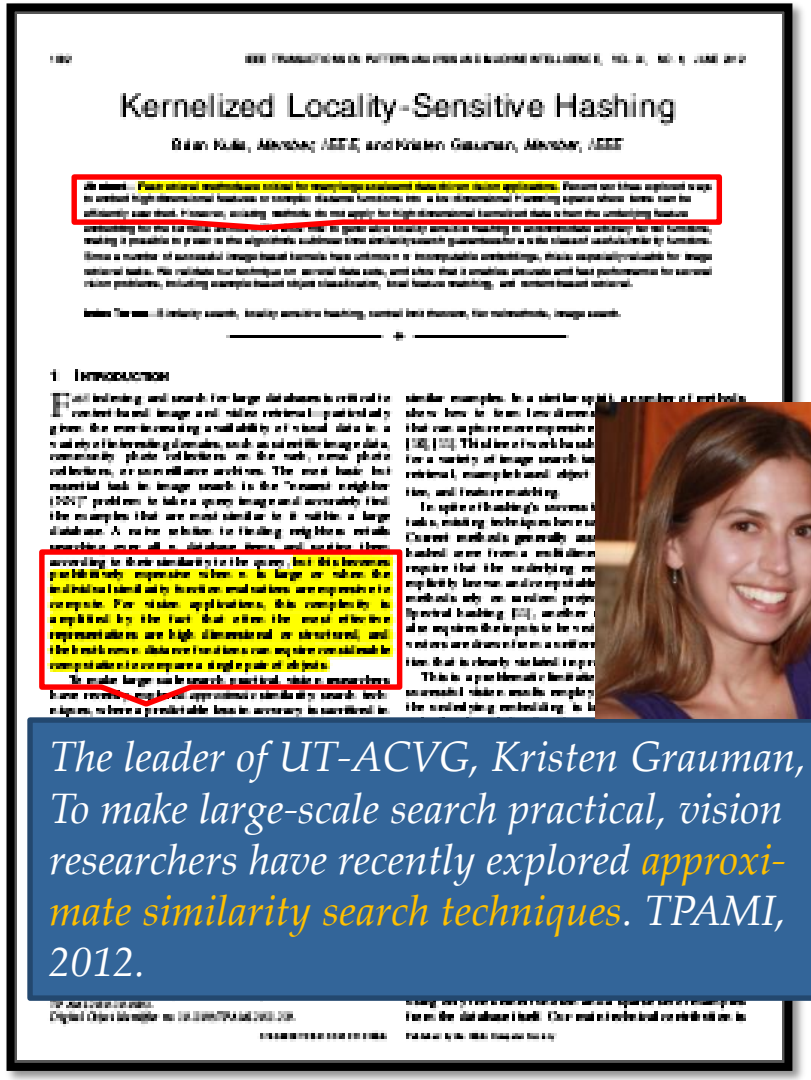
Two of the Representative



Semi-Supervised Hashing for Large-Scale Search

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

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



Kernelized Locality-Sensitive Hashing

Brian Kulis, Member, IEEE, and Kristen Grauman, Member, IEEE

The leader of UT-ACVG, Kristen Grauman, To make large-scale search practical, vision researchers have recently explored approxi-mate similarity search techniques. TPAMI, 2012.

Roadmap

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

Motivations

Projection	Learning paradigm	Method	Year
Data-independent	Unsupervised	*LSH*	*1999*
		SIKH	*2009(NIPS)*
		SH	*2008(NIPS)*
		CH	*2011(ICCV)*
		ITQ	*2012(TPAMI)*
		KMH	*2013(CVPR)*
		KRH	*2014(CVPR)*
	Semi-supervised	*S3PLH*	*2010(ICML)*
		SSH	*2012(TPAMI)*
		BSPLH	*2012(TKDE)*
	Supervised	*BSSC*	*2006*
		BRE	*2009(NIPS)*
		KSH	*2012(CVPR)*

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

Motivations

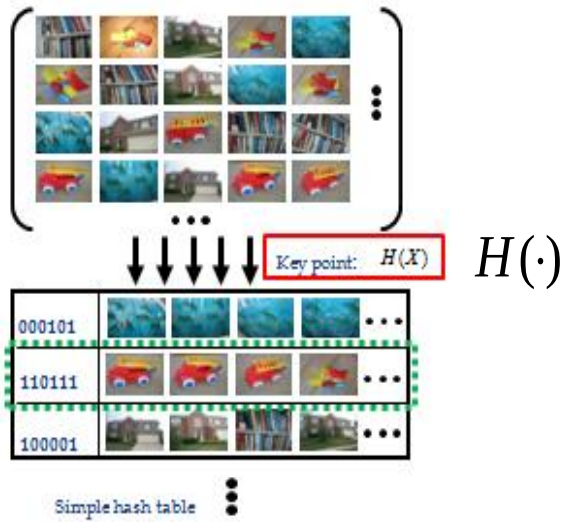
If a sparse reconstruction is adopted, to maximize the information provided by each bit, how to incorporate this requirement with reconstruction simultaneously? (Motivation B)

Does a balance between them improve the performance?

How to build the balance?

Framework of Our Method

Flow chart of the proposed method to learn hash functions.



$H(\cdot)$

Step 1

General consensus & entropy regularizer.

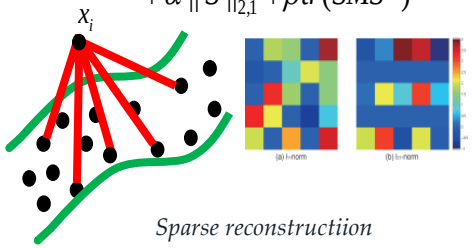
$$J_1(W) = \frac{1}{2} Tr\{W^T(XTX^T + \lambda XX^T)W\}$$

Adjusted covariance matrix.

Step 2

Sparse reconstruction using L-2,1 norm. (Motivation A)

$$J_2(W,S) = \min_{h_k, s_i} \sum_{k=1}^{K} \sum_{i=1}^{N} \| h_k(x_i) - h_k(Xs_i) \|^2 + \alpha \| S \|_{2,1} + \beta tr(SMS^T)$$



Sparse reconstructiion

Step 3

Build a balance between them. (Motivation B)

$$J = \arg\min_{W,S} \frac{J_2(W,S)}{J_1(W)}$$

Cifar-10 dataset

Paradigm A **Paradigm B**

Step 1:

- *Hash Function:*

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

Step 1:

- *Hash Function:*

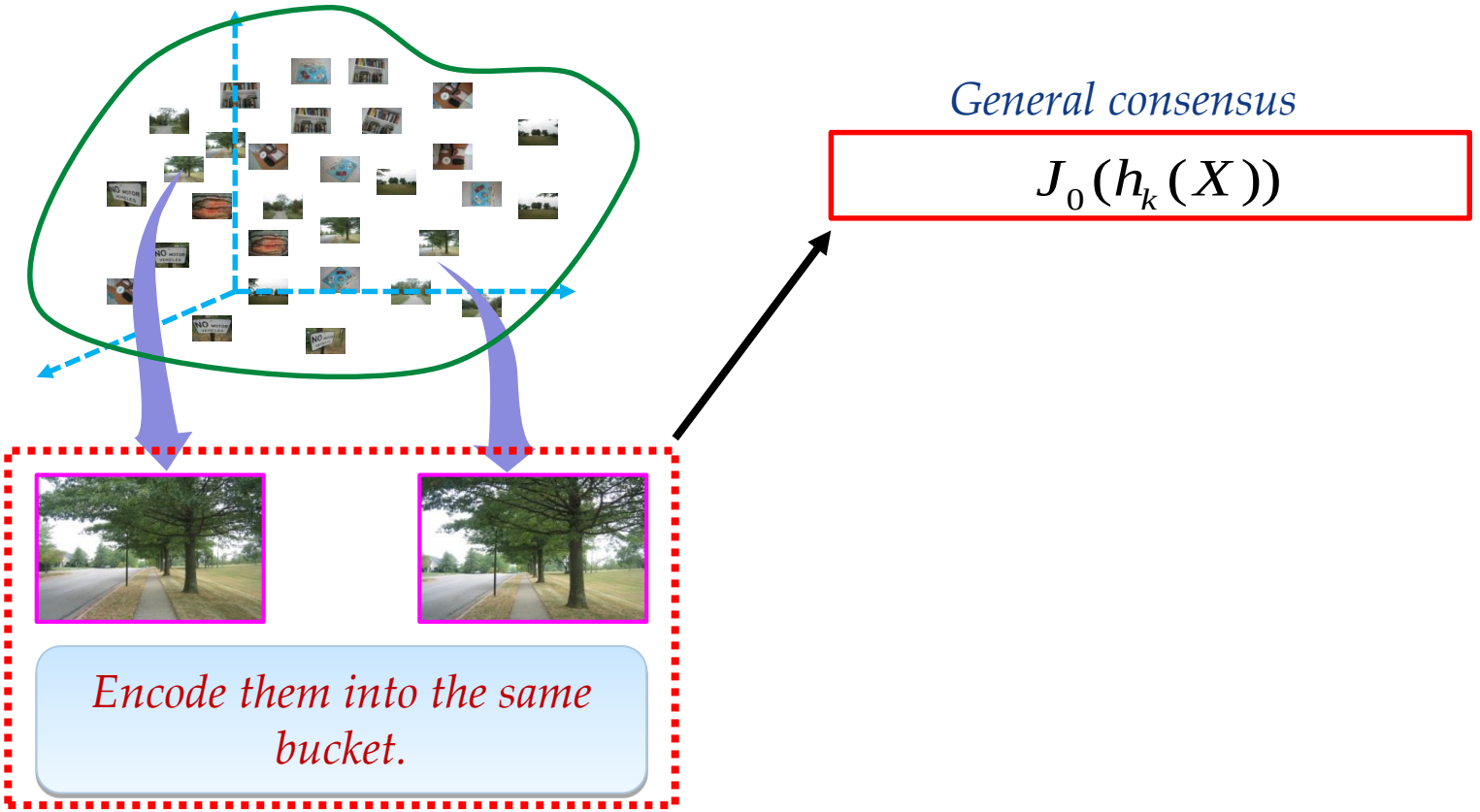
$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad , \quad b_k = E[w_k^T x_i] \xrightarrow{\;0\;} \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

Step 1:

- *Hash Function:*

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

- *General consensus & entropy regularizer*

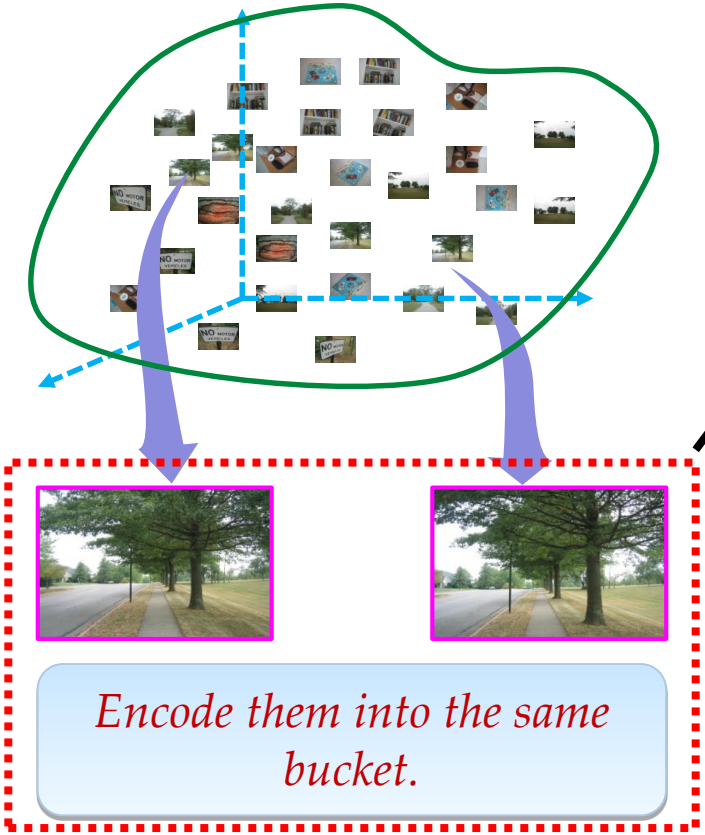


Step 1:

- *Hash Function:*

$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad , \quad b_k = E[w_k^T x_i] \to 0 \quad \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

- *General consensus & entropy regularizer*



General consensus

$$J_0(h_k(X))$$

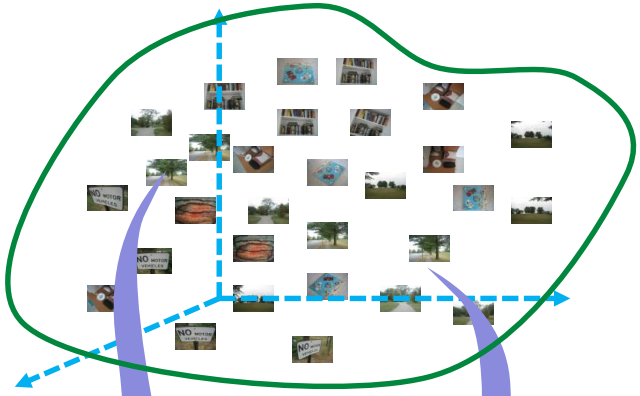
$$J_0(h_k(X)) = \sum_{i,j=1}^{N} t_{i,j} < H(x_i), H(x_j) >$$

Step 1:

- *Hash Function:*

$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad , \quad b_k = E[w_k^T x_i] \to 0 \quad \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

- *General consensus & entropy regularizer*



Encode them into the same bucket.

General consensus

$$J_0(h_k(X))$$

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

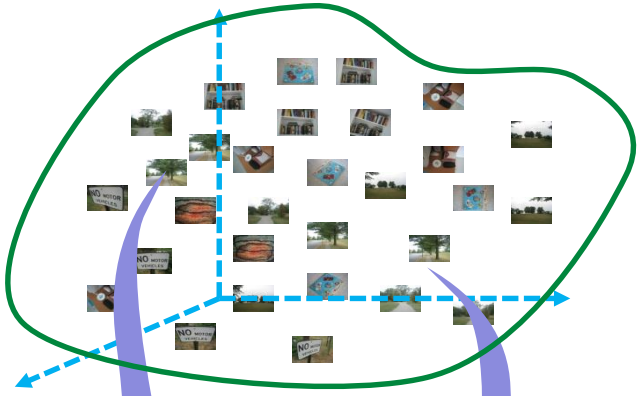
Reflect the similarity of two hash codes.

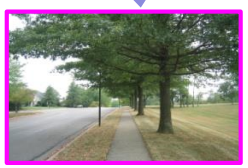
Step 1:

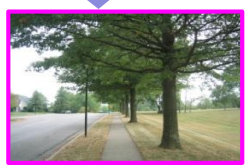
- *Hash Function:*

$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad , \quad b_k = E[w_k^T x_i] \to 0 \quad \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

- *General consensus & entropy regularizer*







Encode them into the same bucket.

General consensus

$$J_0(h_k(X))$$

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

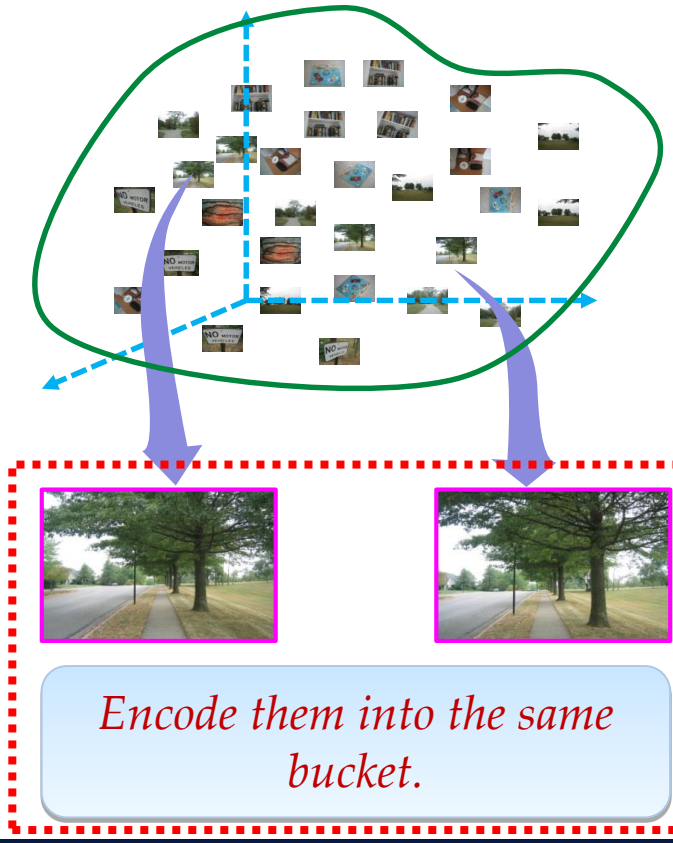
Weight element denoted the same in complementary hashing.

Step 1:

- *Hash Function:*

$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad , \quad b_k = E[w_k^T x_i] \to 0 \quad \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

- *General consensus & entropy regularizer*



Encode them into the same bucket.

General consensus

$$J_0(h_k(X))$$

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

Relaxing the signed magnitude.

$$J_0 = tr(W^T XTX^T W)$$

Step 1:

- *Hash Function:*

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

- *General consensus & entropy regularizer*

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

Step 1:

- *Hash Function:*

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

- *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:*

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

- *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.

$$\max_W tr(W^T X X^T W)$$

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:*

$$h_k(x) = \mathrm{sgn}(w_k^T x_i + b_k) \quad b_k = E[w_k^T x_i] \to 0 \Longrightarrow \quad h_k(x) = \mathrm{sgn}(w_k^T x_i)$$

- *General consensus & entropy regularizer*

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

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

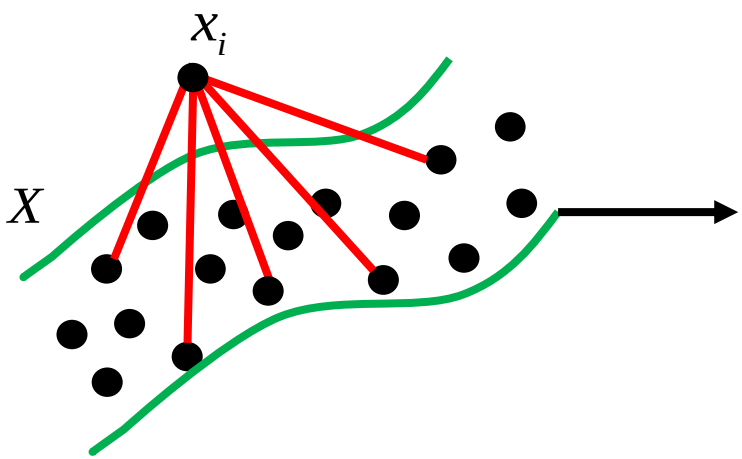
Maximize the variance of hash functions.

$$\max_W tr(W^T X X^T W)$$

Combining the general consensus and the entropy regularizer.

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

Step 2:



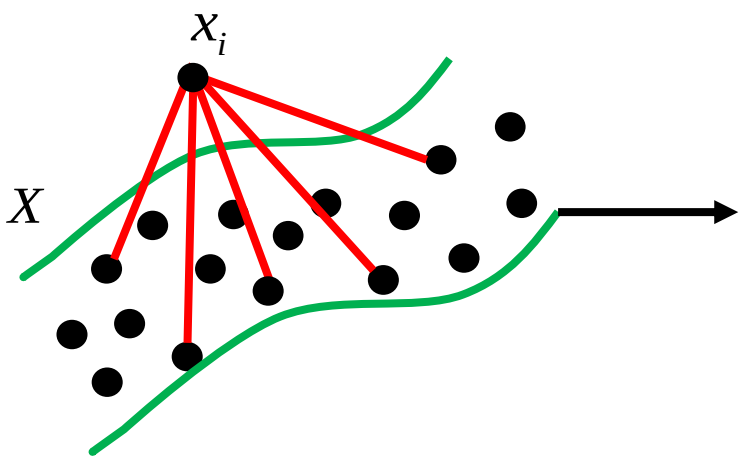
Sparsity preserving projections (SPP)

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

$$s.t. \quad x_i = X s_i, i = 1, 2, ..., N$$

$$1 = 1^T s_i$$

Step 2:



Sparsity preserving projections (SPP)

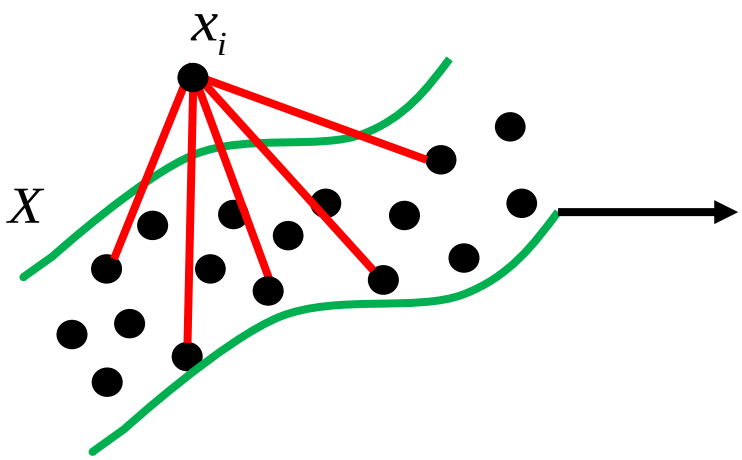
$$\min_{s_i} \| s_i \|_1 + \beta \sum_{i=1}^{N} \sum_{j=1}^{N} (s_i - s_j)^2 A_{ij}$$

$$s.t. \quad x_i = X s_i, i = 1, 2, ..., N$$

$$1 = 1^T s_i$$

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

Step 2:



Sparsity preserving projections (SPP)

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

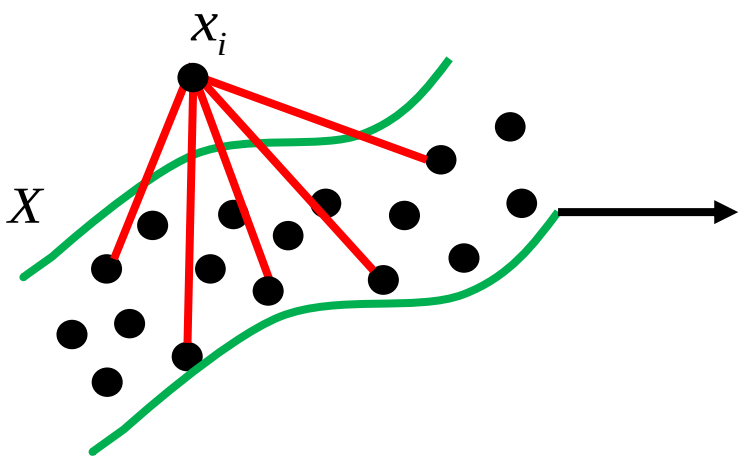
$$s.t. \quad x_i = X s_i, i = 1, 2, ..., N$$

$$1 = 1^T s_i$$

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

M is the graph Laplacian matrix.

Step 2:



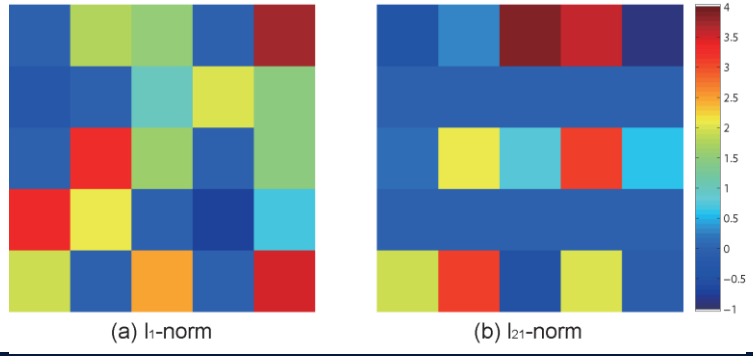
Sparsity preserving projections (SPP)

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

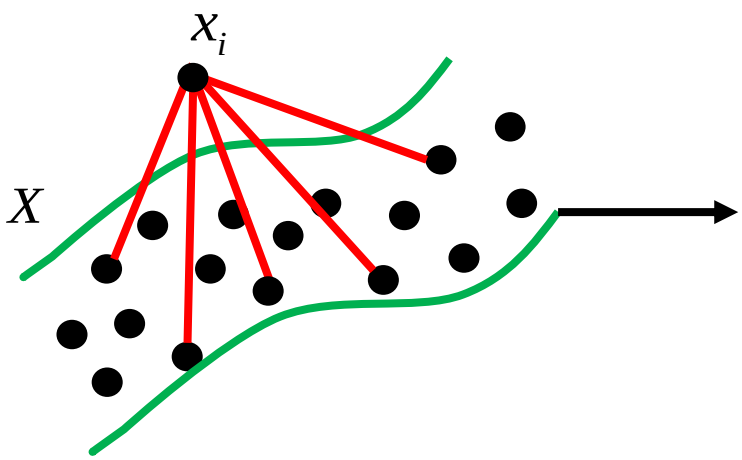
$$s.t. \quad x_i = X s_i, i = 1, 2, ..., N$$

$$1 = 1^T s_i$$

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



Step 2:



Sparsity preserving projections (SPP)

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

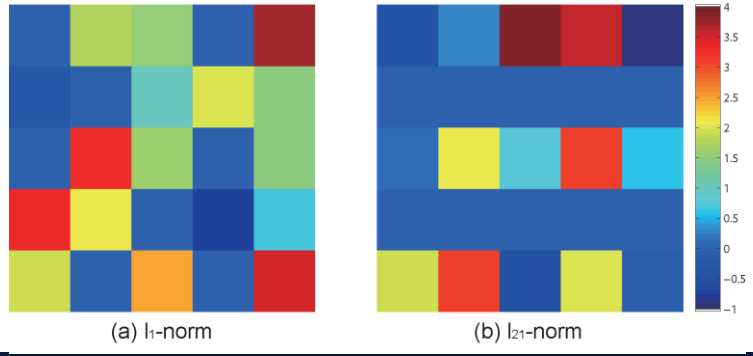
$$s.t. \quad x_i = X s_i, i = 1, 2, ..., N$$

$$1 = 1^T s_i$$

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

$$J_2 = \min_{W,S} tr(W^T XVX^T W) + \alpha tr(SUS^T) + \beta tr(SMS^T)$$

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



Step 3:

- *Final objective function:*

$$J(W,S)=\frac{J_2}{J_1}=\min_{W,S}\frac{tr(W^T XVX^T W)+\alpha tr(SUS^T)+\beta tr(SMS^T)}{tr\{W^T(XTX^T+\lambda XX^T)W\}}$$

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*:*

$$J(W) = \min_{W} \frac{tr(W^T XVX^T W)}{tr(W^T X(T+\lambda I)X^T W)} \Longrightarrow$$

Generalized eigenvalue decomposition problem.

- *Step 2: Fix* W *and update* S *:*

$$J(S) = \min_{S} tr(W^T XVX^T W) + \alpha tr(SUS^T) + \beta tr(SMS^T) \Longrightarrow$$

Take derivation with respect of S *.*

Algorithm

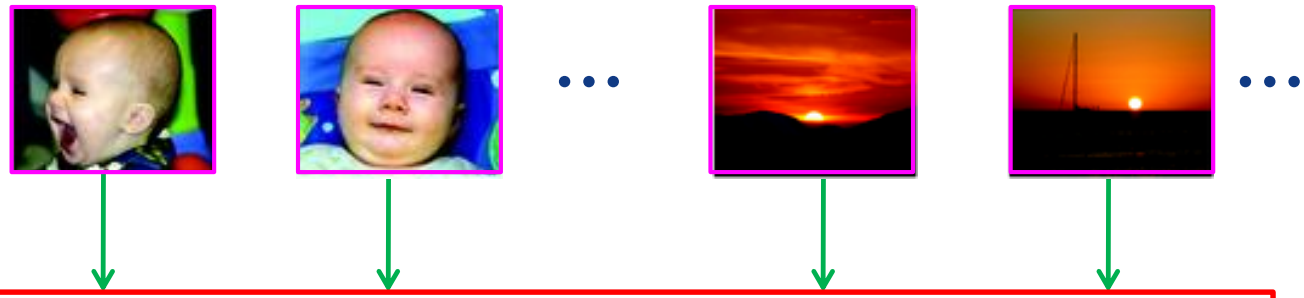
Algorithm 1 Sparse Reconstruction Hashing

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 ith element of $U_{ii} = \frac{1}{\|s_{i\cdot}\|_2+\zeta}$.
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



$$H(x_i) = [h_1(x_i), h_2(x_i), ..., h_K(x_i)] \in \{0,1\}^K$$

$$H(x_i) = [\text{sgn}(w_1^T x_i), \text{sgn}(w_2^T x_i), ..., \text{sgn}(w_K^T x_i)]$$

$$\Uparrow$$

$$Output: W = [w_1, w_2, \bullet\bullet\bullet, w_K]$$

... 110110 010110 ... 001000 000001 ...

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*

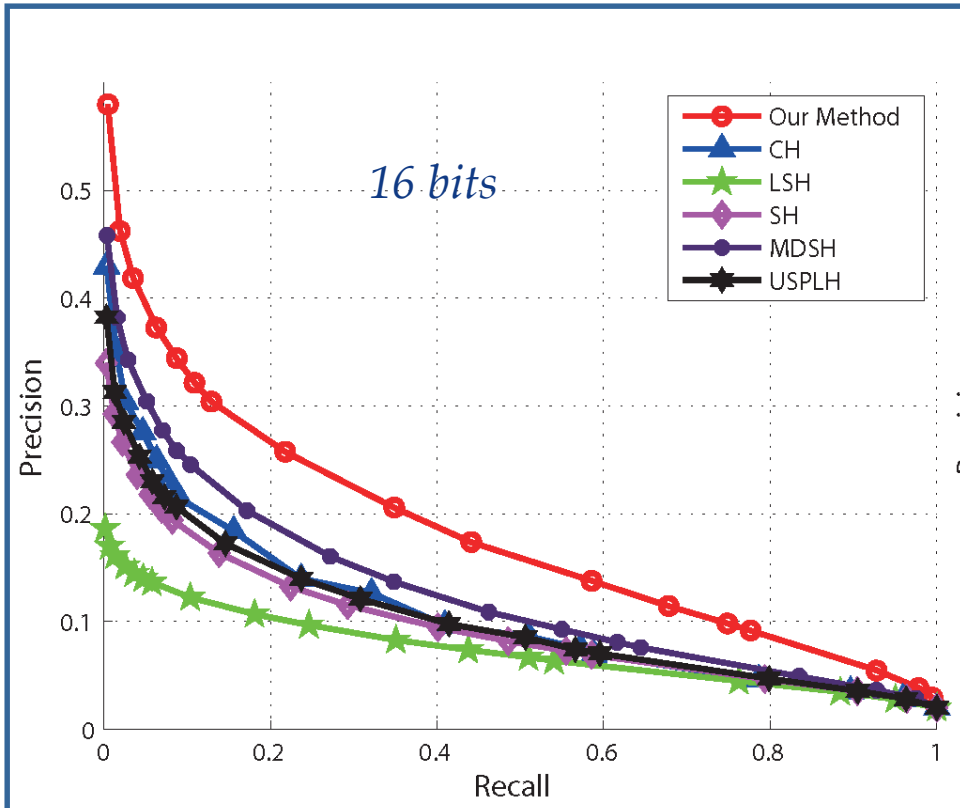
    Unsupervised: 59k for training (database), 1K for query.
  - *Tiny100k images data*

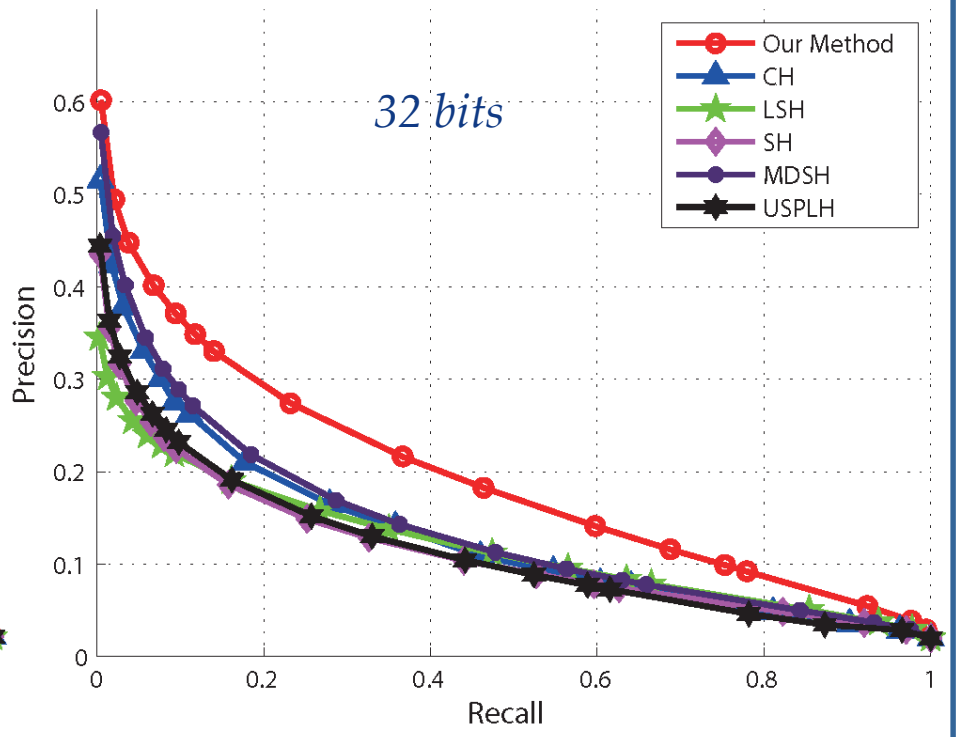
    Unsupervised: 96k for training (database), 4K for query.
  - *Parameter selection*

    Alpha: 0.5, beta: 0.1.

Experiments

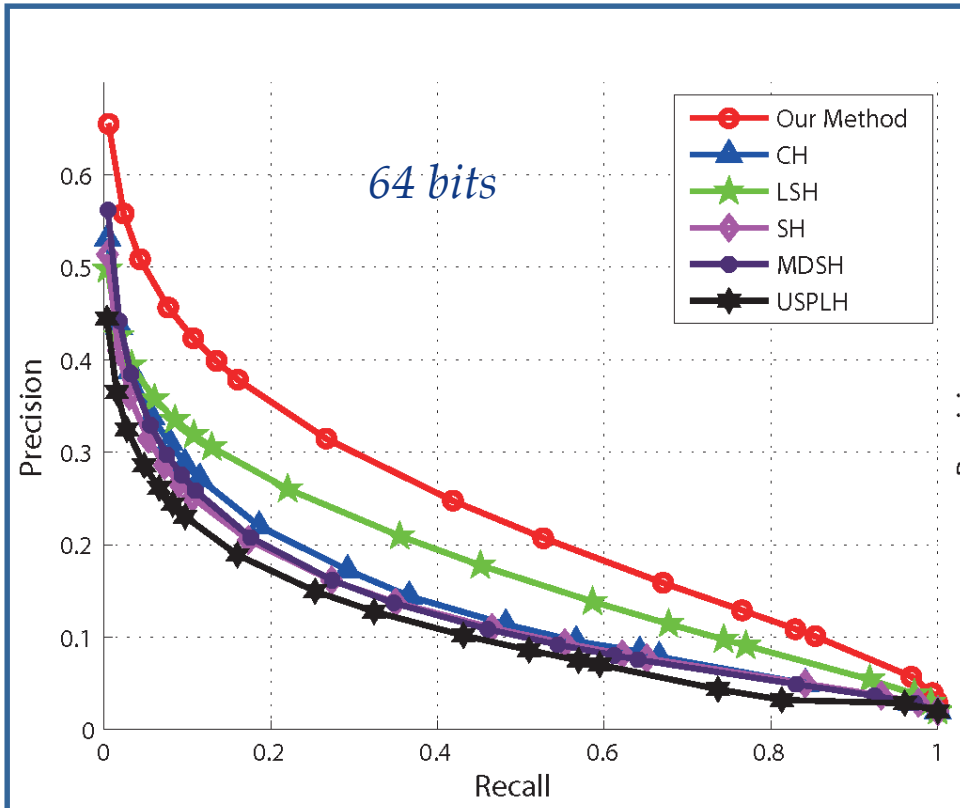
Cifar-10 precision recall curve

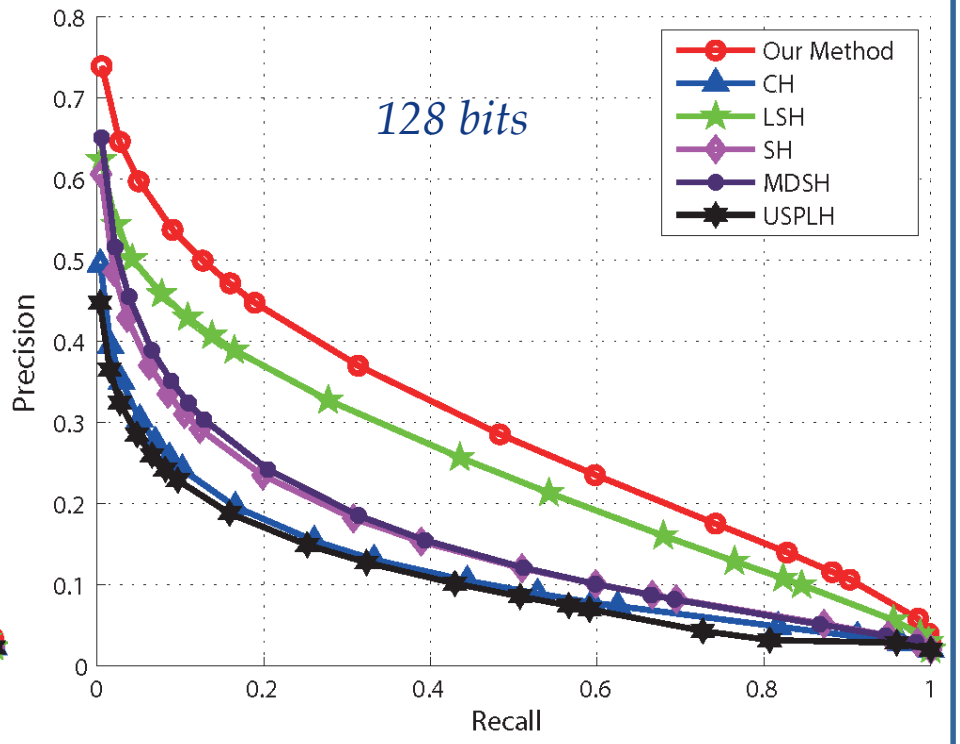




Experiments

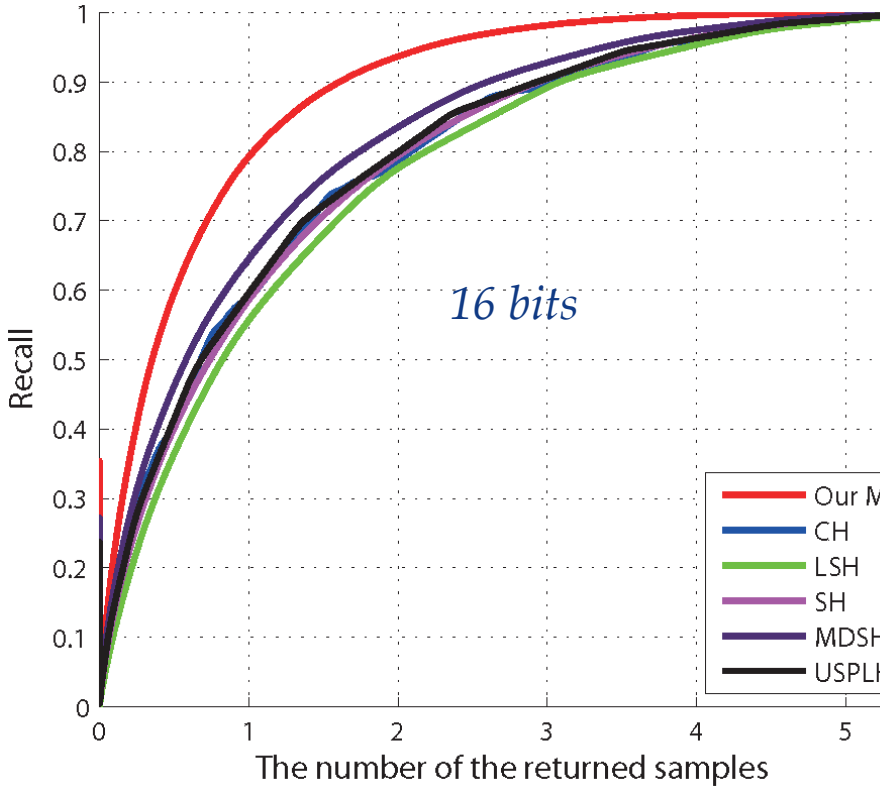
Cifar-10 precision recall curve

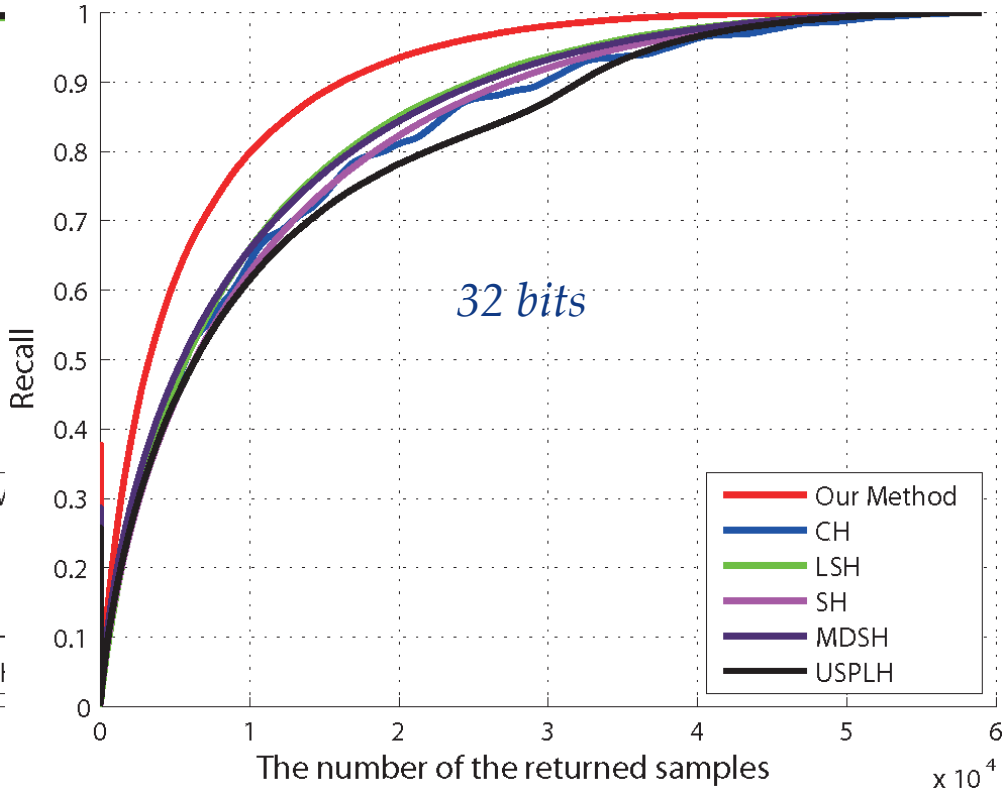




Experiments

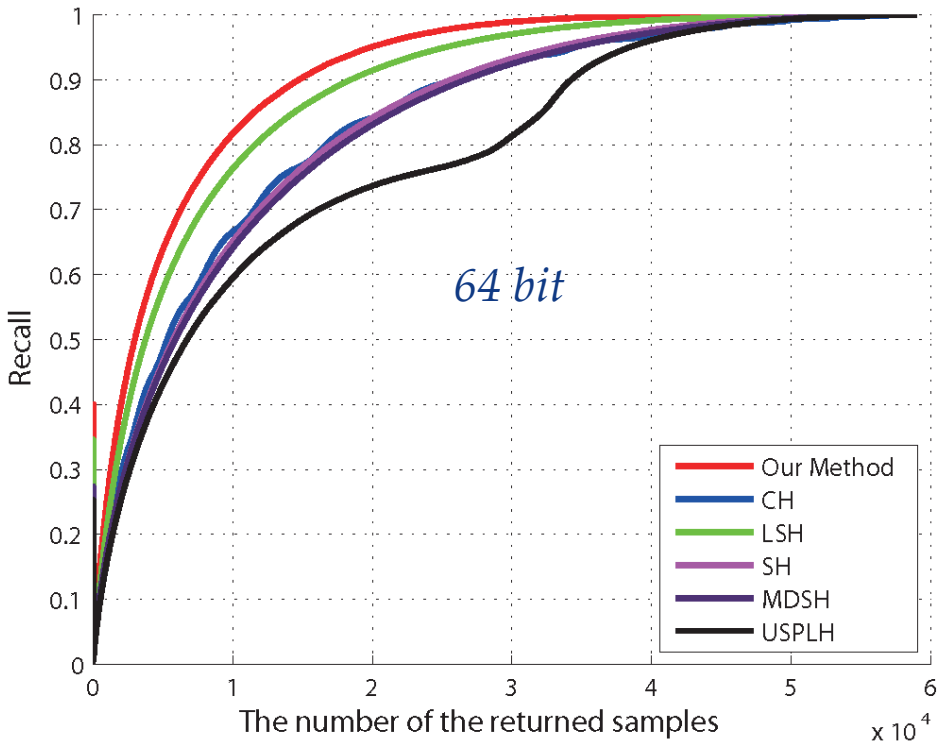
Cifar-10 recall-the number of return samples curve

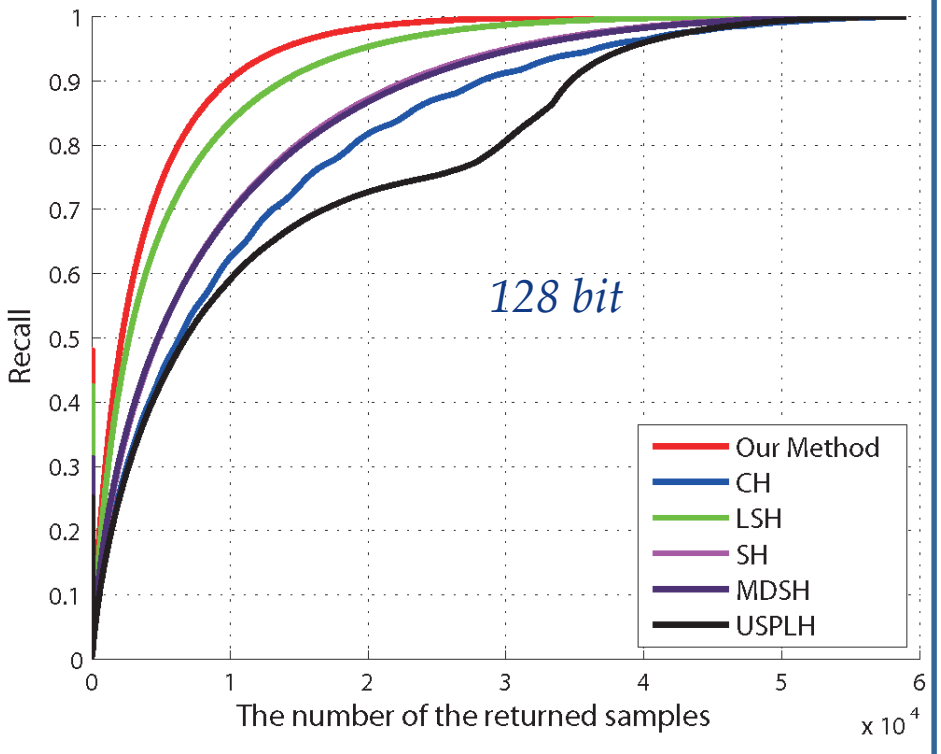




Experiments

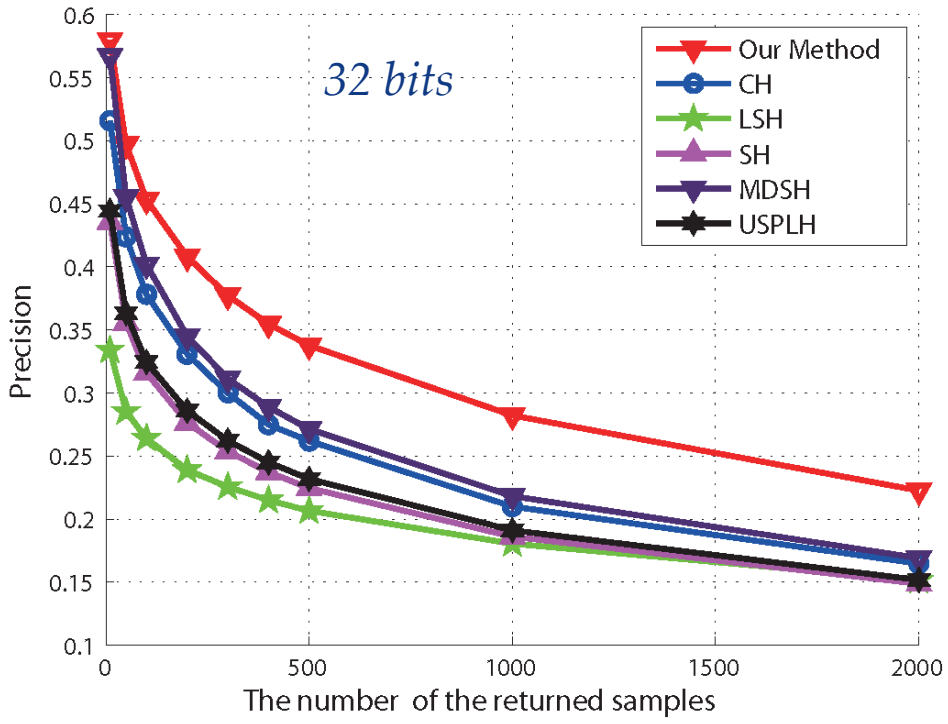
Cifar-10 recall-the number of return samples curve

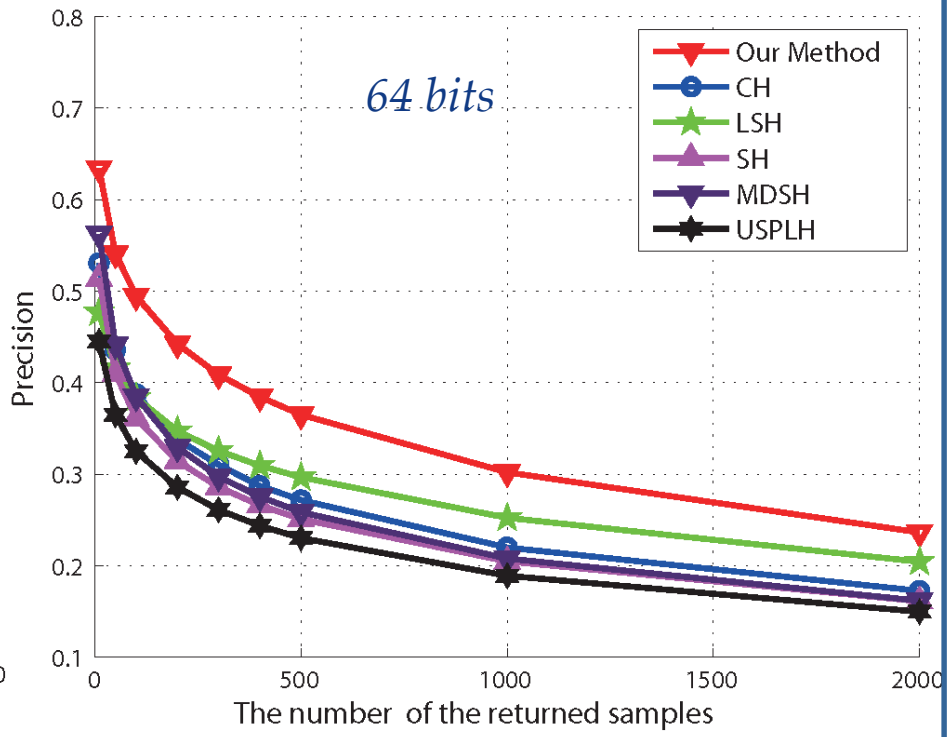




Experiments

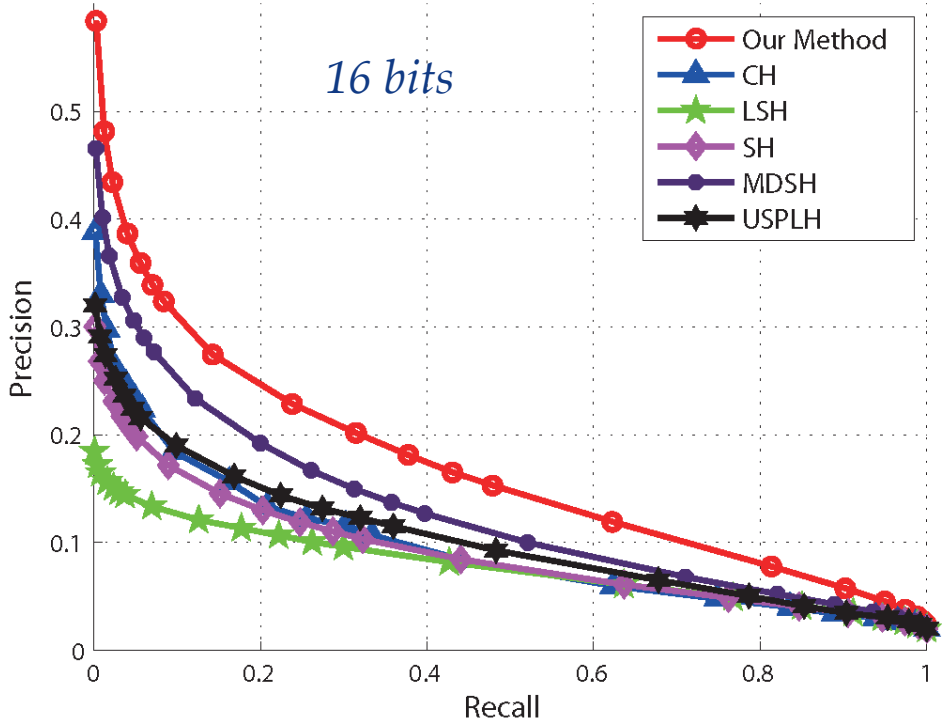
Cifar-10 precision-the number of return samples curve

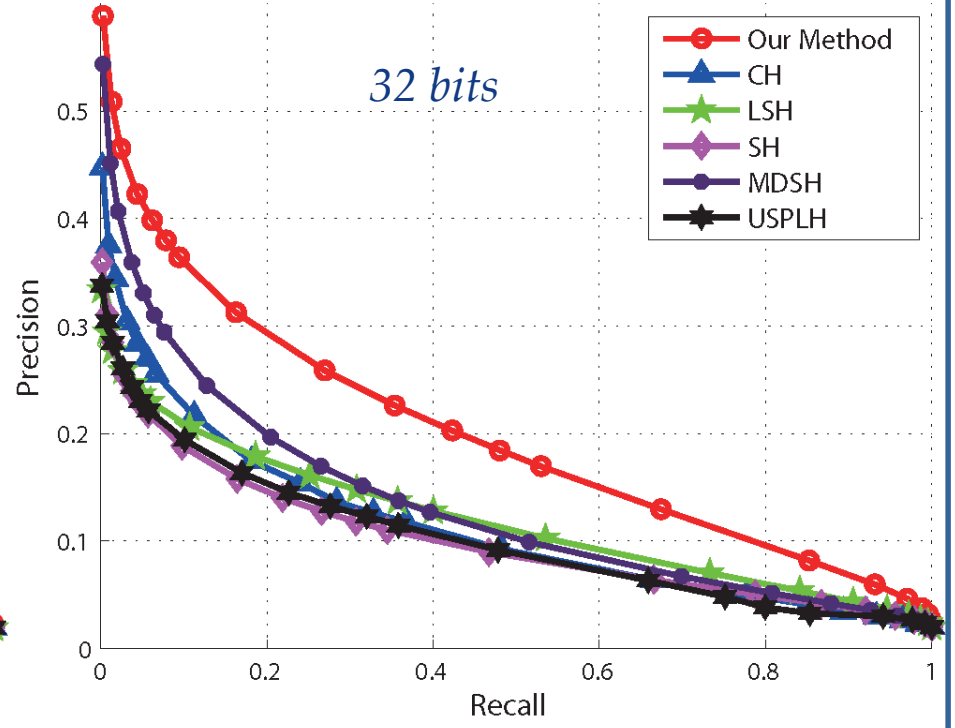




Experiments

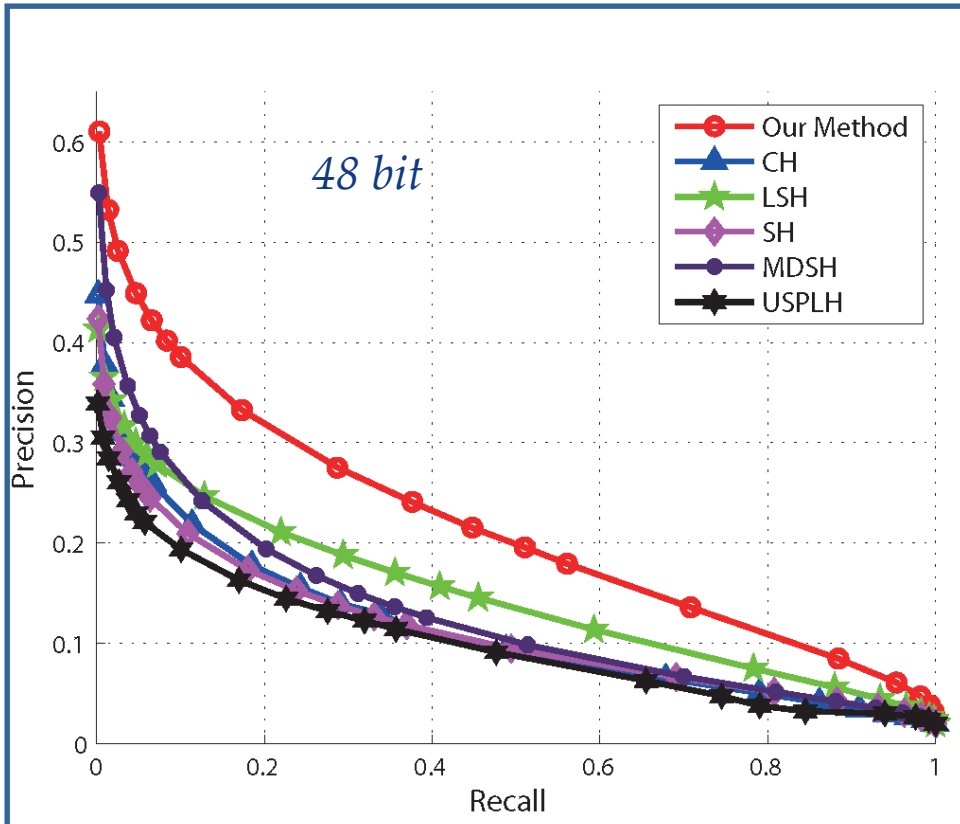
Tiny100k precision recall curve

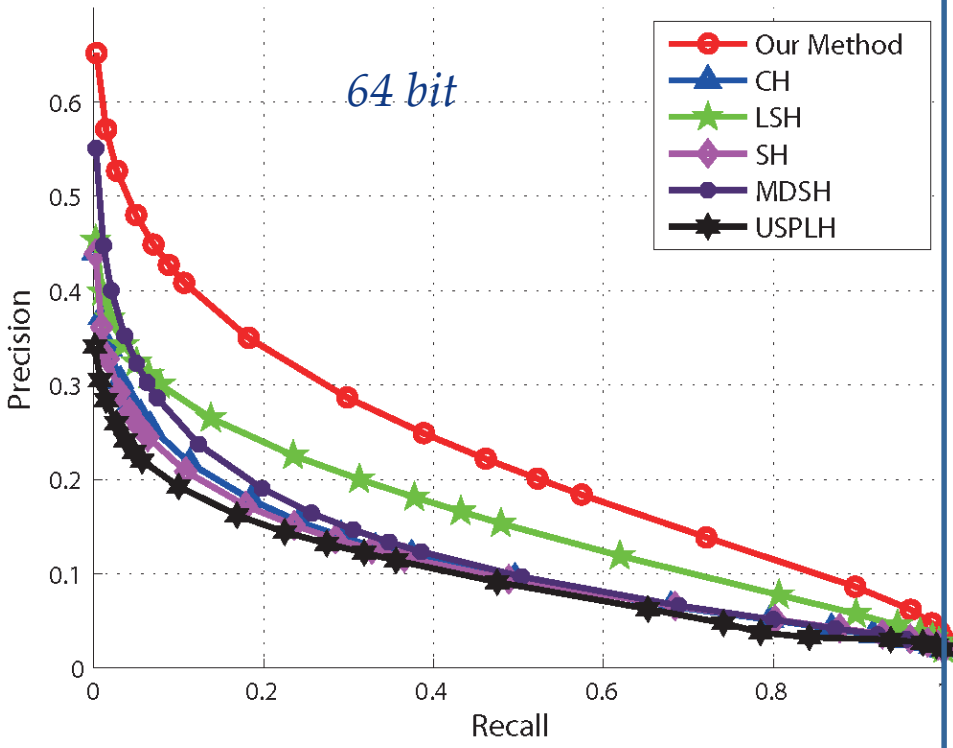




Experiments

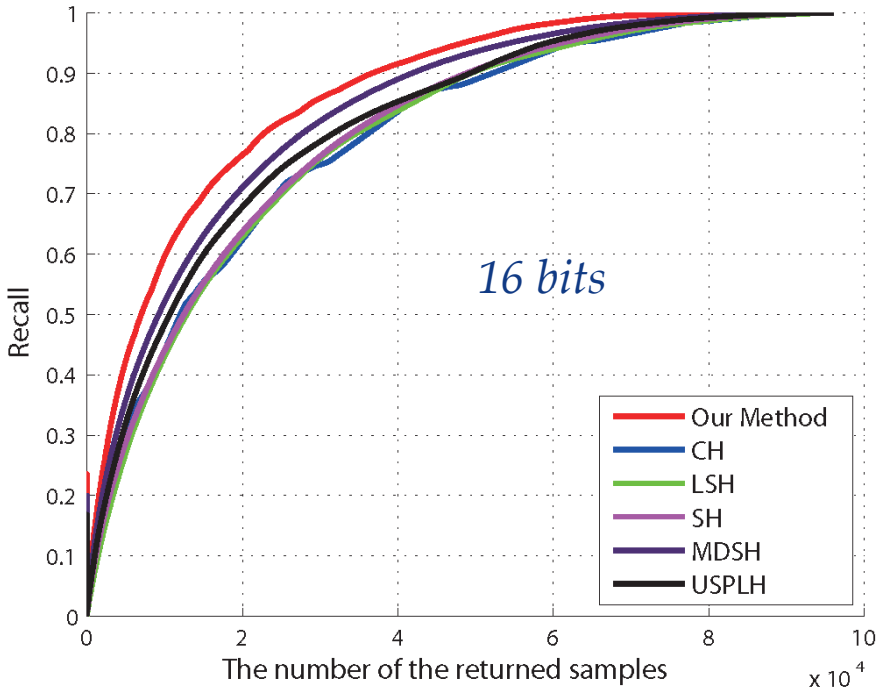
Tiny100k precision recall curve

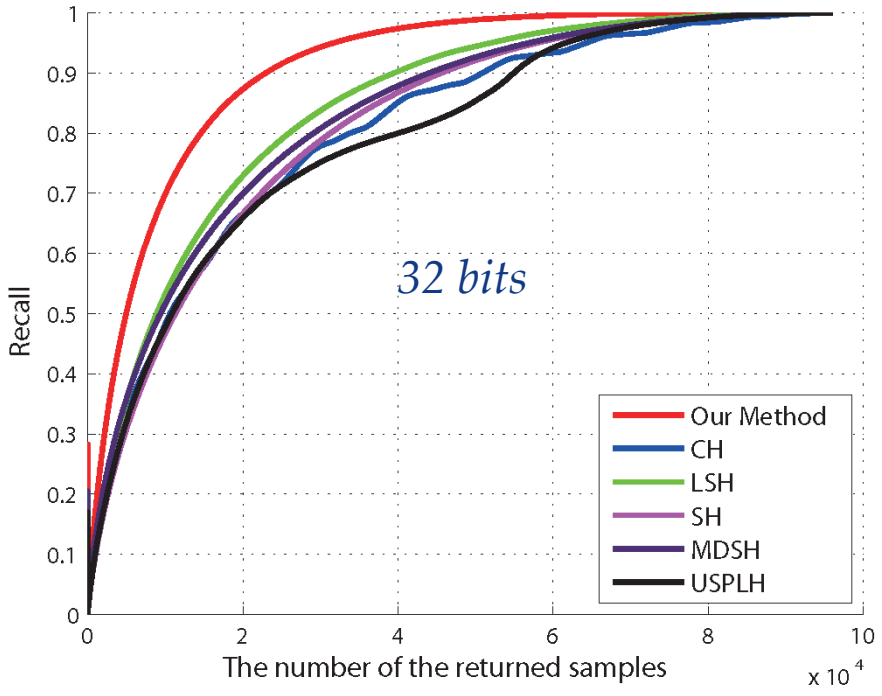




Experiments

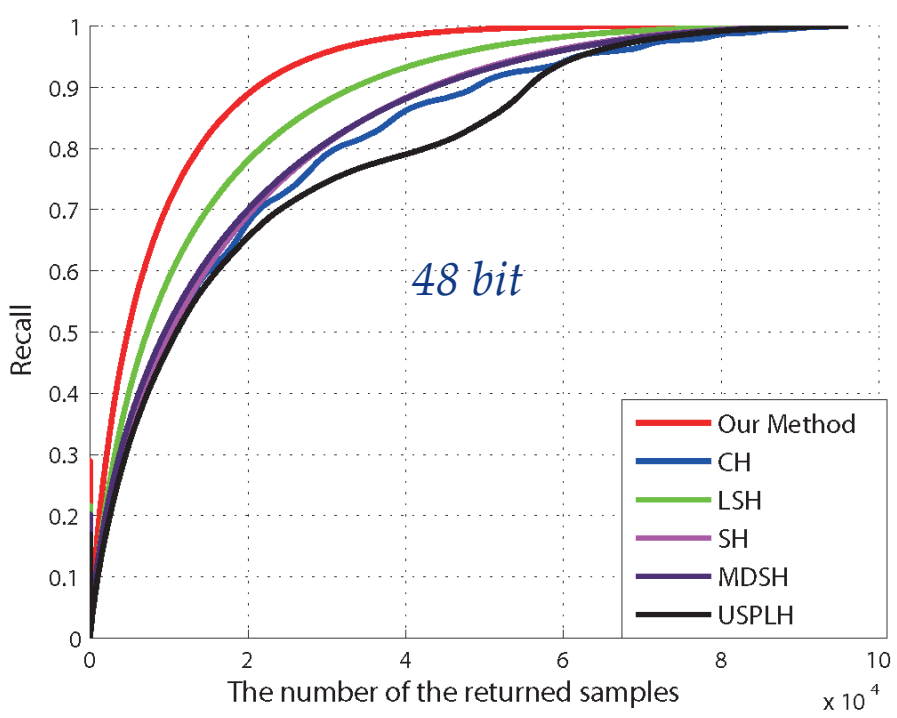
Tiny100k recall-the number of return samples curve

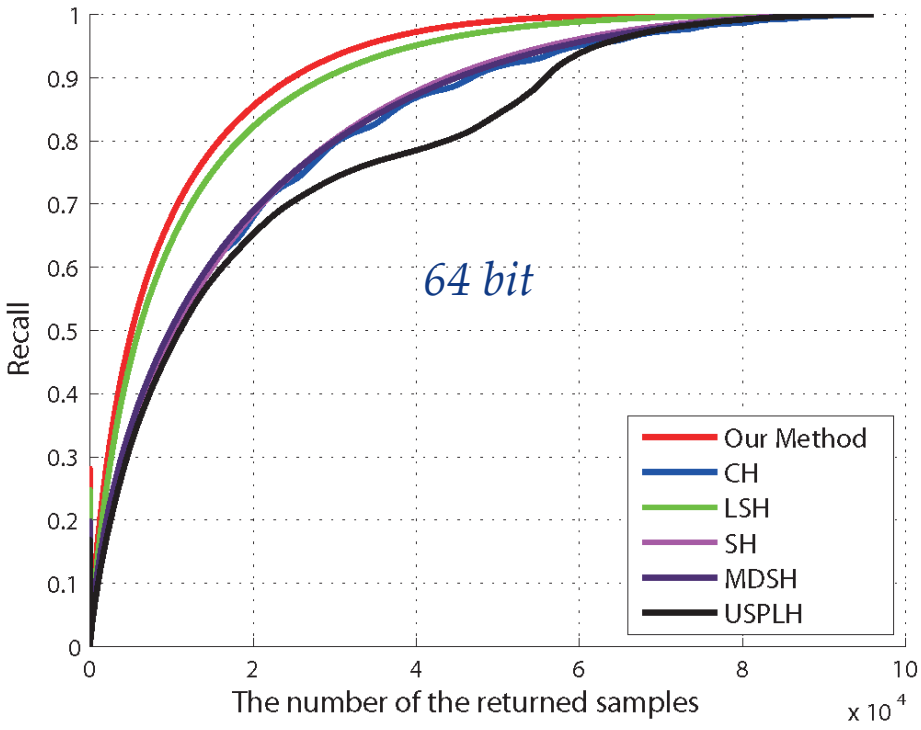




Experiments

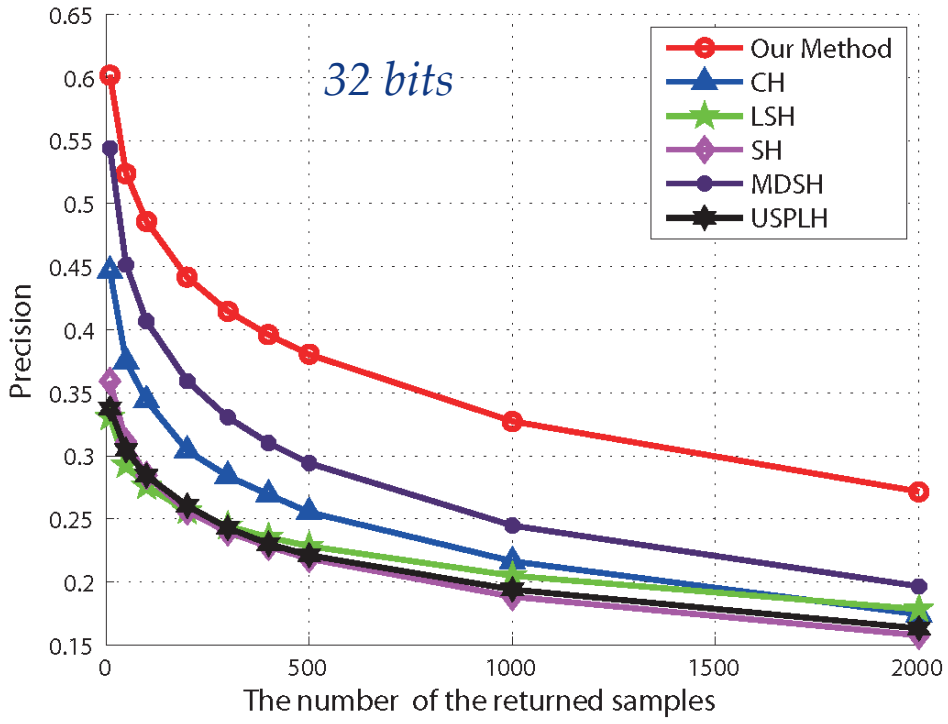
Tiny100k recall-the number of return samples curve

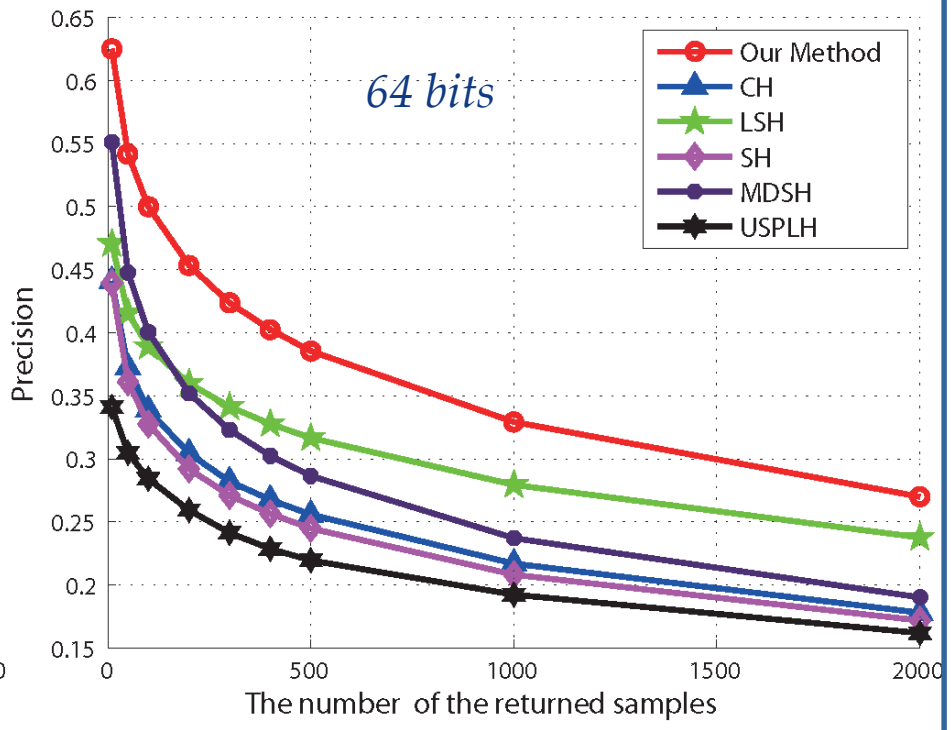




Experiments

Tiny100k precision-the number of return samples curve





Experiments

Our method

SH

MDSH

Query image

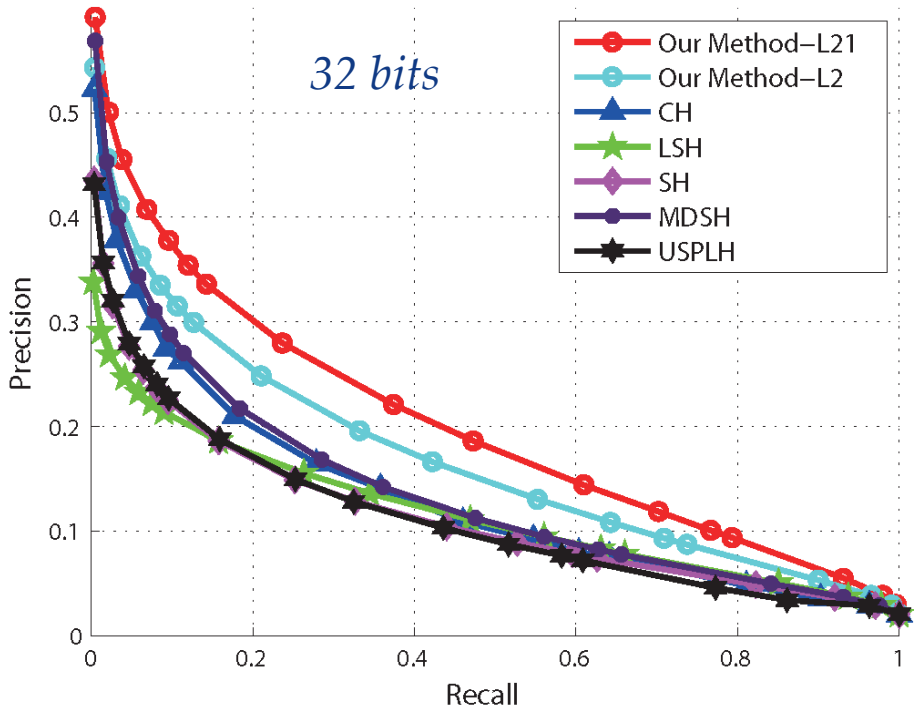
LSH

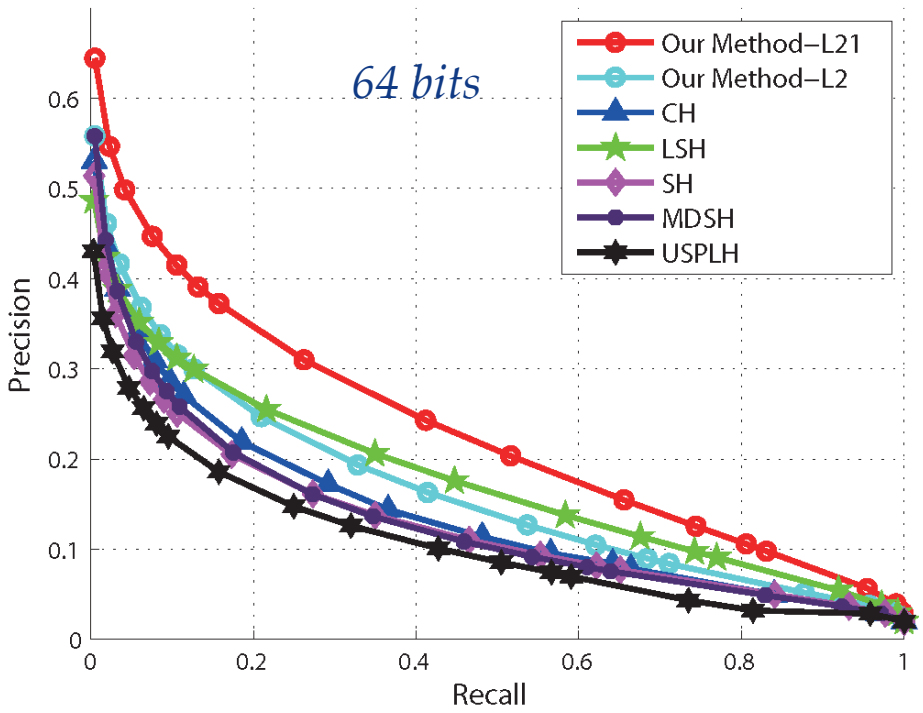
CH

USPLH

L21 Norm VS L2

Cifar-10 precision recall curve





Roadmap

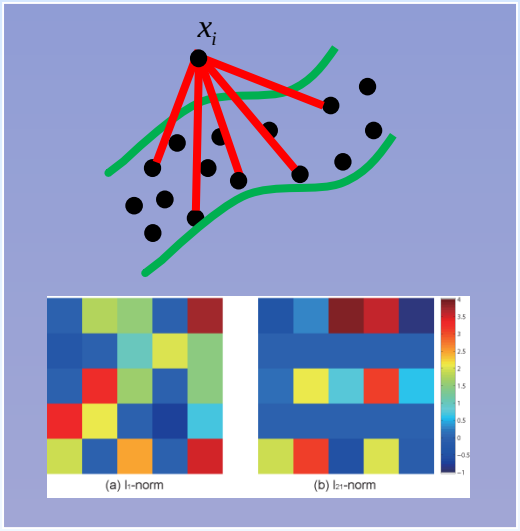
- *Overview*
- *Method*
- *Experiments*
- *Summary*

Summary

➢ Contributions

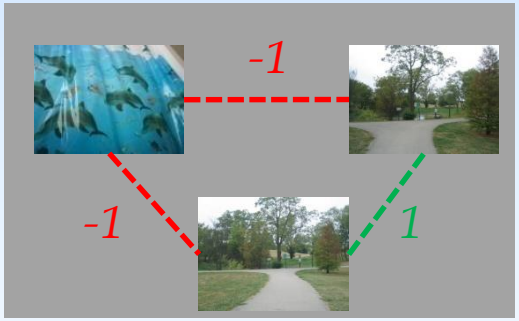
A **novel sparse reconstruction** *is proposed by introducing the* $l_{2,1}$*-norm into SSP framework to preserve* **potential discrimination information** *of hash functions.*

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.*



Thank You

Q & A