Hashing Techniques

윤성의 (Sung-Eui Yoon)
Professor
KAIST

http://sgvr.kaist.ac.kr

Student Presentation Guidelines

- **Good summary, not full detail, of the paper**
  - **Talk about motivations of the work**
  - **Give a broad background on the related work**
  - **Explain main idea and results of the paper**
  - **Discuss strengths and weaknesses of the method**

- **Prepare an overview slide**
  - **Talk about most important things and connect them well**

High-Level Ideas

- Deliver most important ideas and results
  - Do not talk about minor details
  - Give enough background instead
- Deeper understanding on a paper is required
  - Go over at least two related papers and explain them in a few slides
- Spend most time to figure out the most important things and prepare good slides for them

Deliver Main Ideas of the Paper

- **Identify main ideas/contributions of the paper and deliver them**

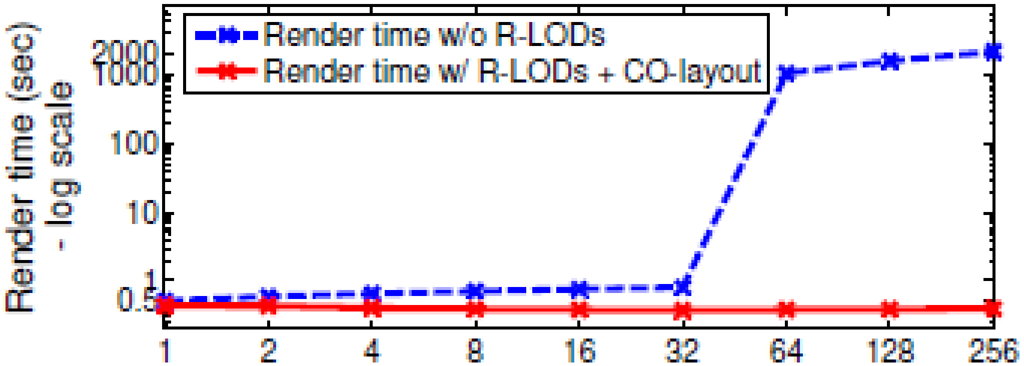
- **If there are prior techniques that you need to understand, study those prior techniques and explain them**
    - **For example, A paper utilizes B's technique in its main idea. In this case, you need to explain B to explain A well.**

Be Honest

- Do not skip important ideas that you don't know
  - Explain as much as you know and mention that you don't understand some parts
- If you get questions you don't know good answers, just say it
- In the end, you need to explain them before the semester ends at KLMS board

Result Presentation

- **Give full experiment settings and present data with the related information**
  - **What does the x-axis mean in the below image?**



- **After showing the data, give a message that we can pull of the data**
- **Show images/videos, if there are**

Utilizing Existing Resources

- **Use author's slides, codes, and video, if they exist**
- **Give proper credits or citations**
  - **Without them, you are cheating!**

Audience feedback form

Date:
Talk title:
Speaker:

1. Was the talk well organized and well prepared?
5: Excellent 4: good 3: okay 2: less than average 1: poor

2. Was the talk comprehensible? How well were important concepts covered?
5: Excellent 4: good 3: okay 2: less than average 1: poor

Any comments to the speaker

Prepare Quiz

- **Review most important concepts of your talk**
  - **Prepare two multiple-choices questions**

- **Example: What is the biased algorithm?**
  - **A: Given N samples, the expected mean of the estimator is I**
  - **B: Given N samples, the exp. Mean of the estimator is I + e**
  - **C: Given N samples, the exp. Mean of the estimator is I + e, where e goes to zero, as N goes to infinite**

- **Grade them in the scale of 0 to 10 and send it to TA**

Class Objectives

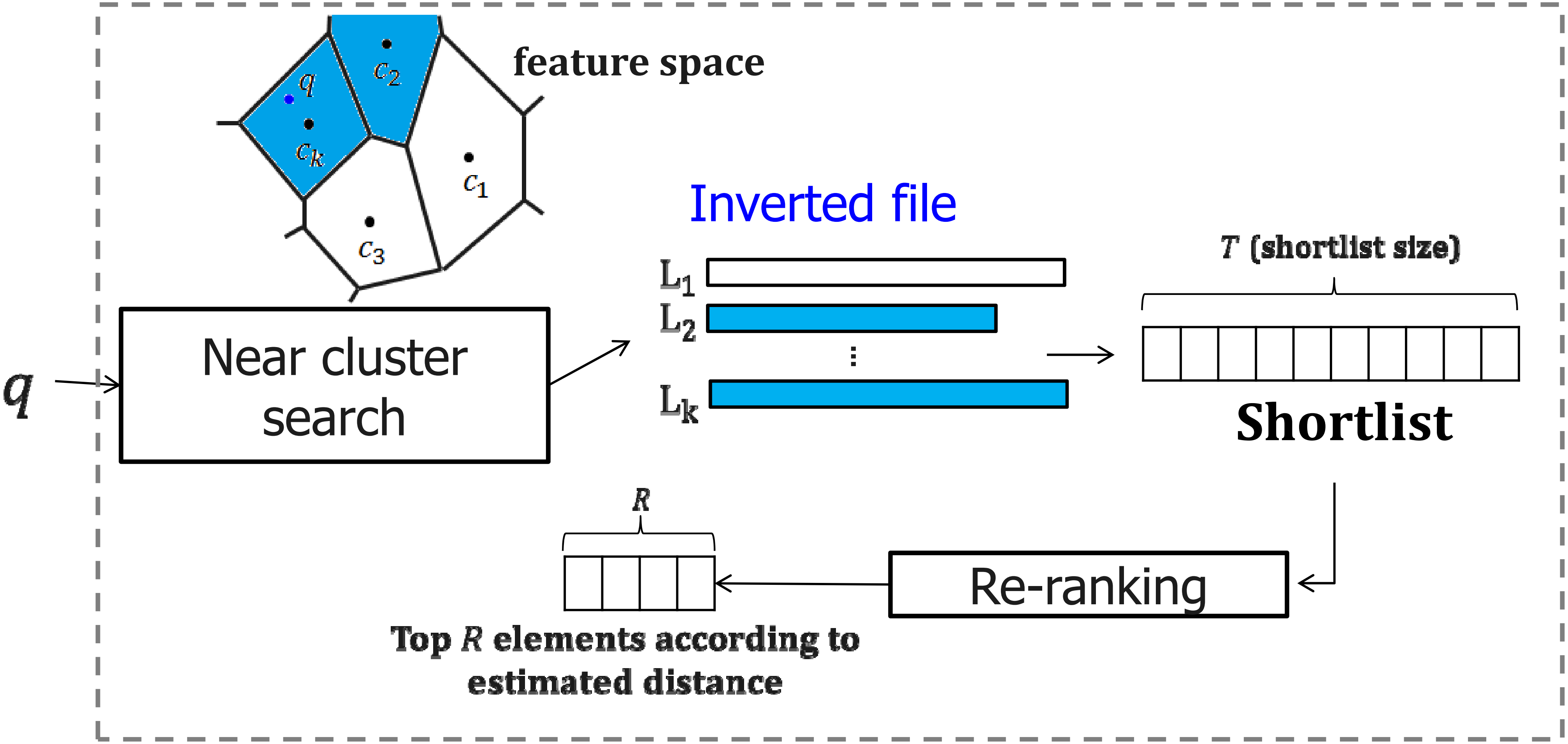
- **Understand the basic hashing techniques based on hyperplanes**
  - **Unsupervised approach**
- **Supervised approach using deep learning**

- **At the last class:**
  - **Discussed re-ranking methods: spatial verification and query expansion**
  - **Talked about inverted index**

Questions

- **When we talk about accuracy, I don't understand why we only think about the accuracy of matching victual point/patch/features. I think we should also concern about finding images with similar style, images with similar emotion, images reflecting similar activity...**

Review of Basic Image Search



Ack.: Dr. Heo

KAIST

Image Search

Finding visually similar images

Image Descriptor

High dimensional point

(BoW, GIST, Color Histogram, etc.)

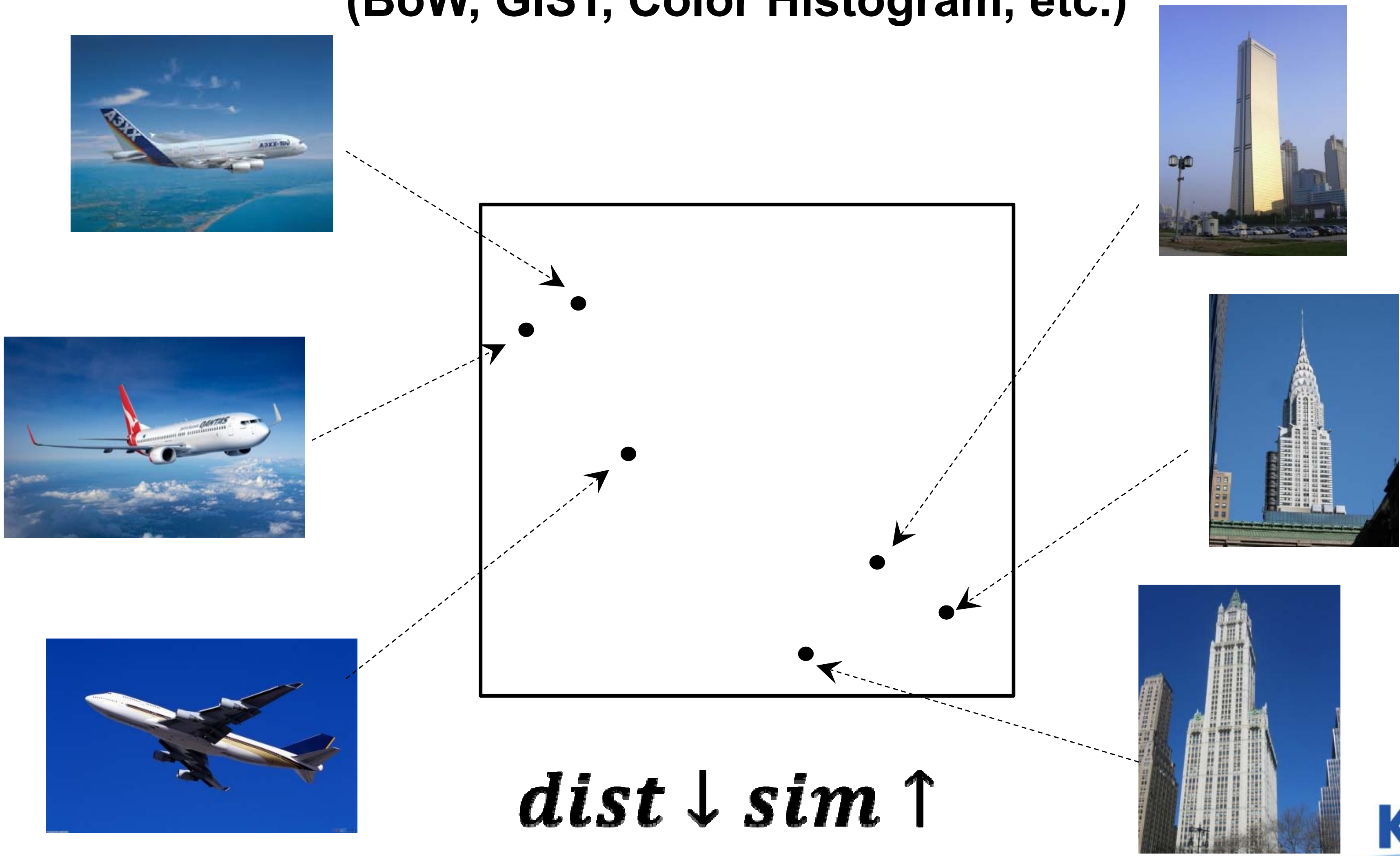
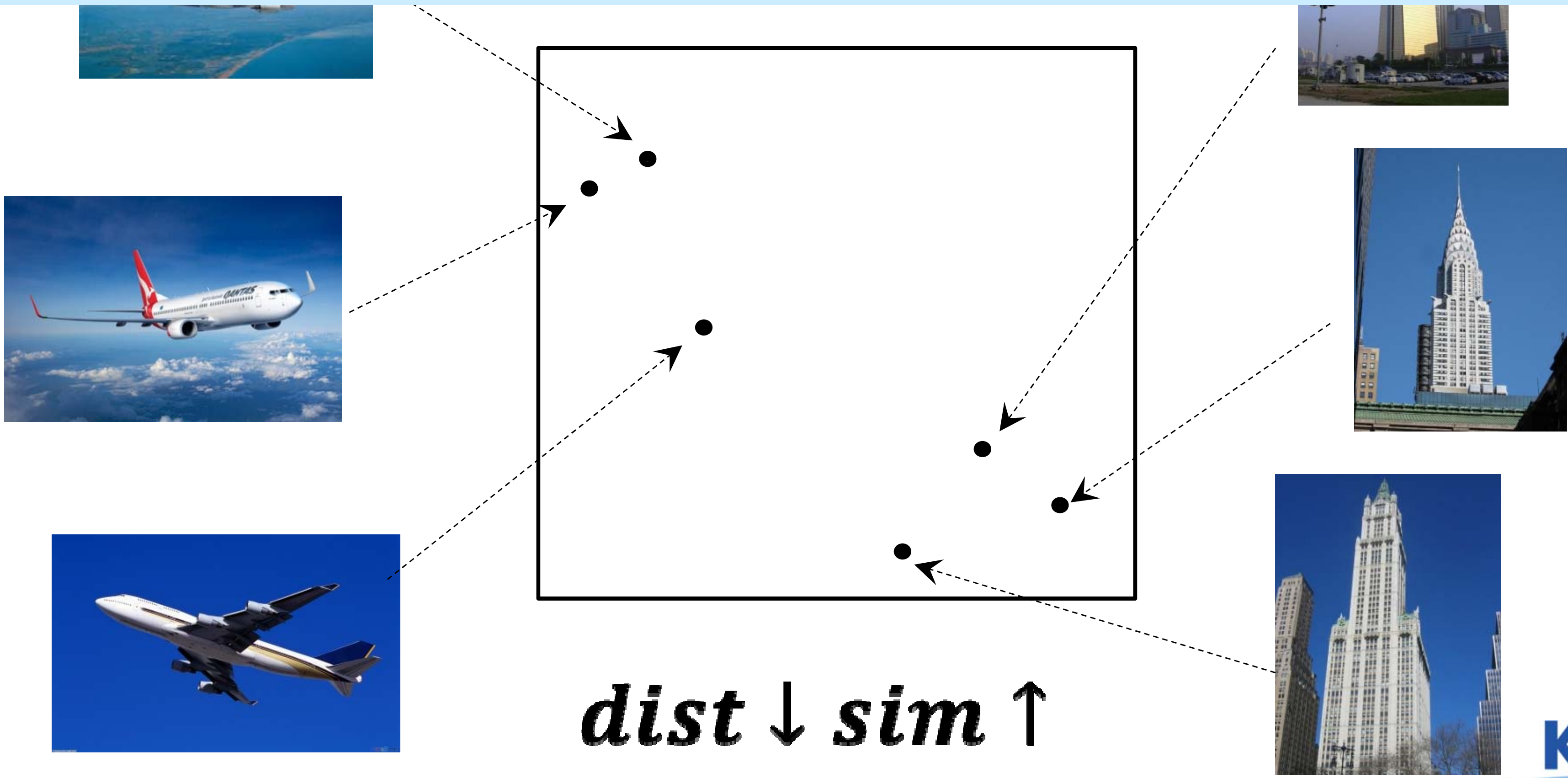


Image Descriptor

High dimensional point

Nearest neighbor search (NNS) in high dimensional space



$dist \downarrow sim \uparrow$

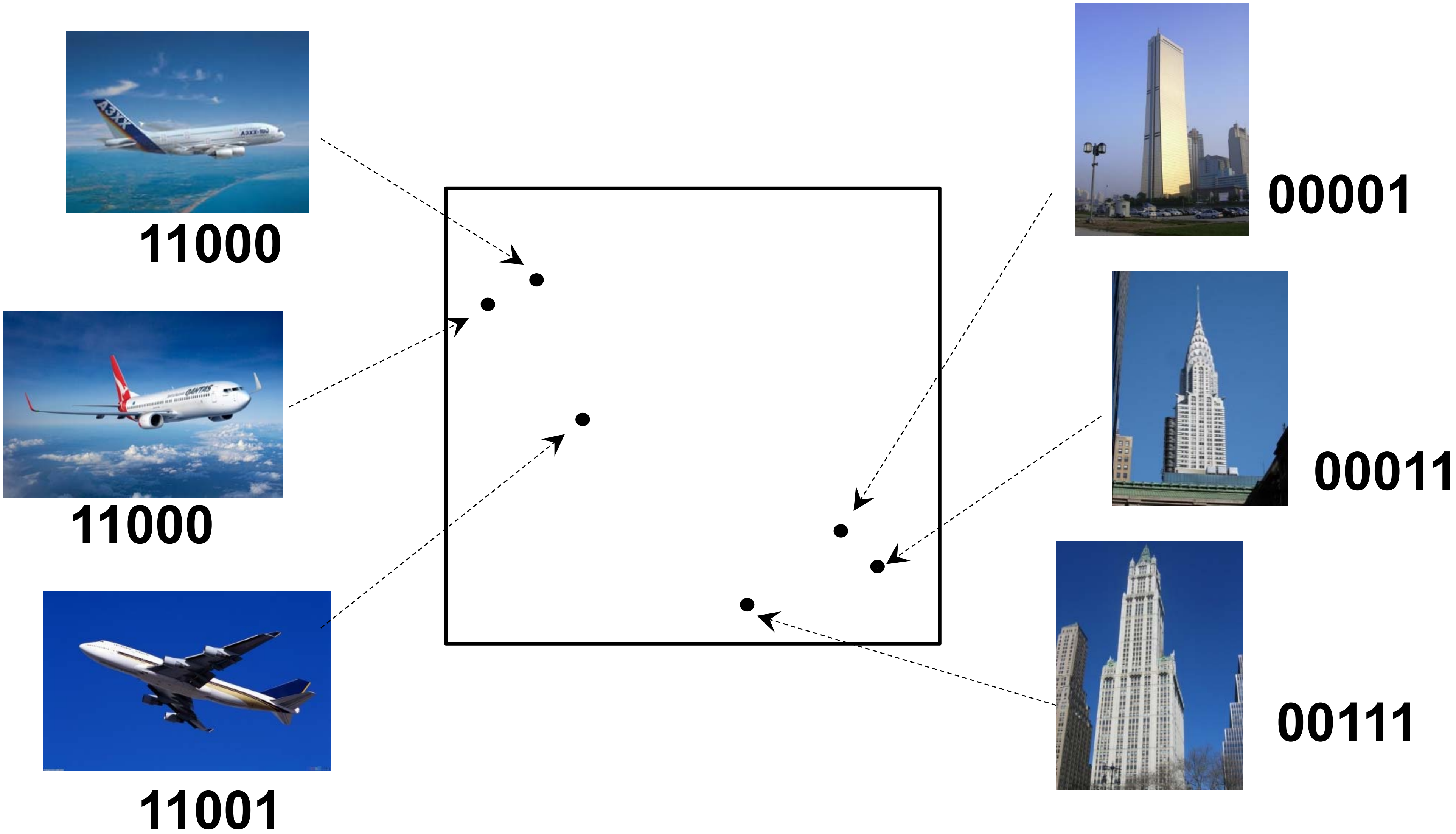
Challenge

	BoW	CNN
Dimensions	1000+	4000+
1 image	4 KB+	16 KB+
1B images	4 TB+	16 TB+

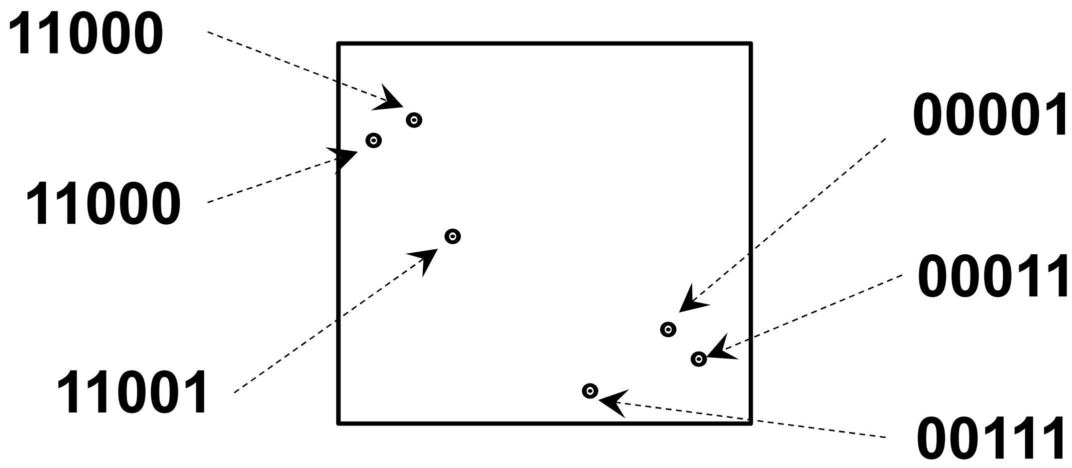
$$\frac{144\ GB\ memory}{1\ billion\ images} \approx \frac{128\ bits}{1\ image}$$

KAIST

Binary Code



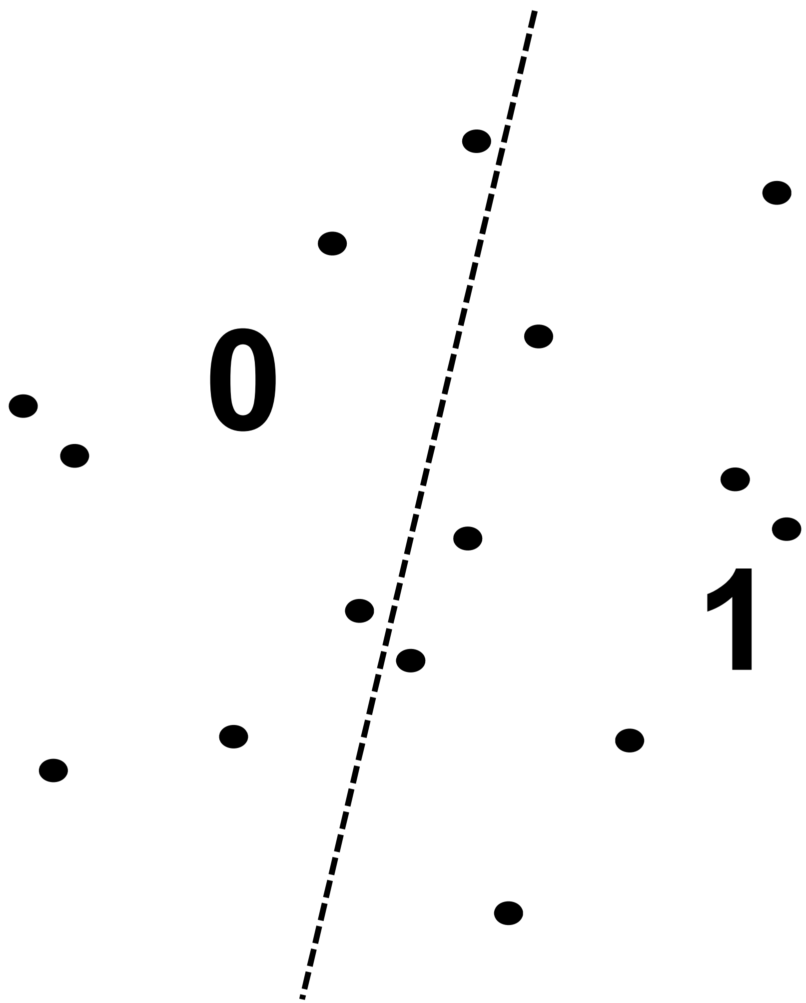
Binary Code



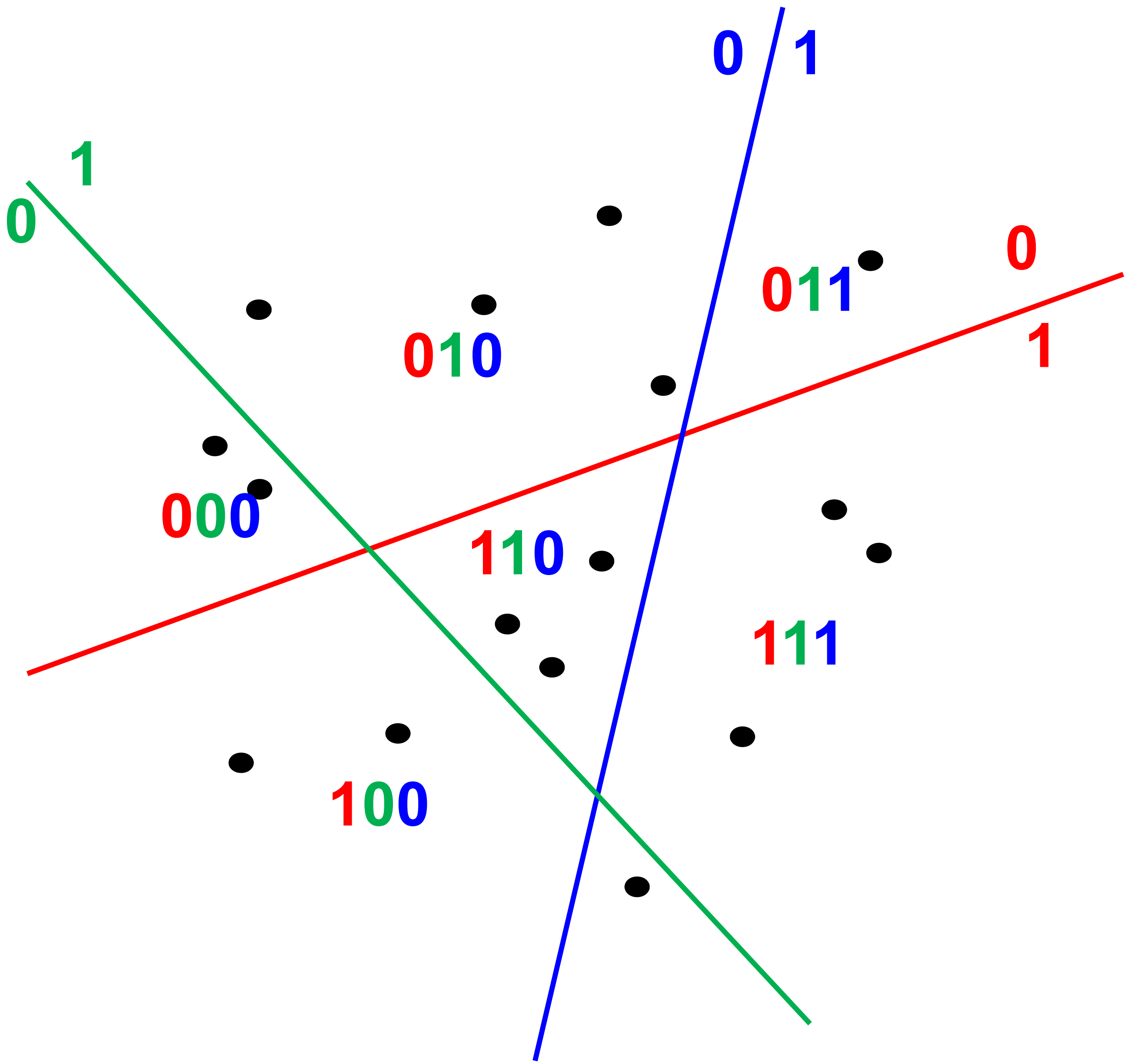
* Benefits

- Compression
- Very fast distance computation (Hamming Distance, XOR)

Hyper-Plane based Binary Coding



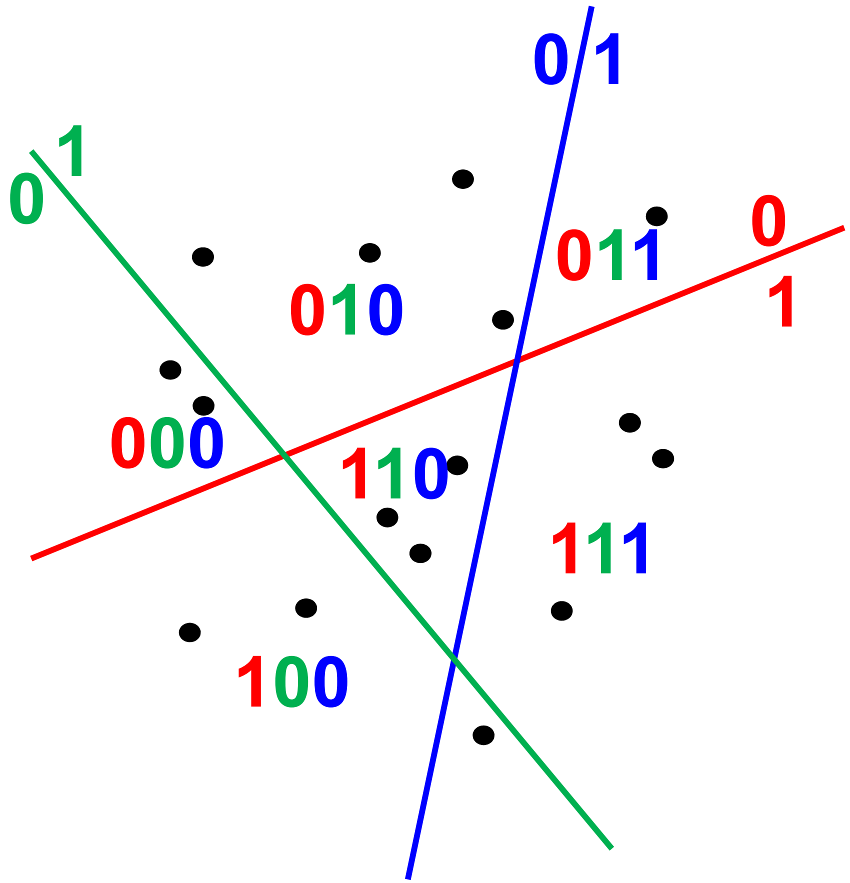
Hyper-Plane based Binary Coding



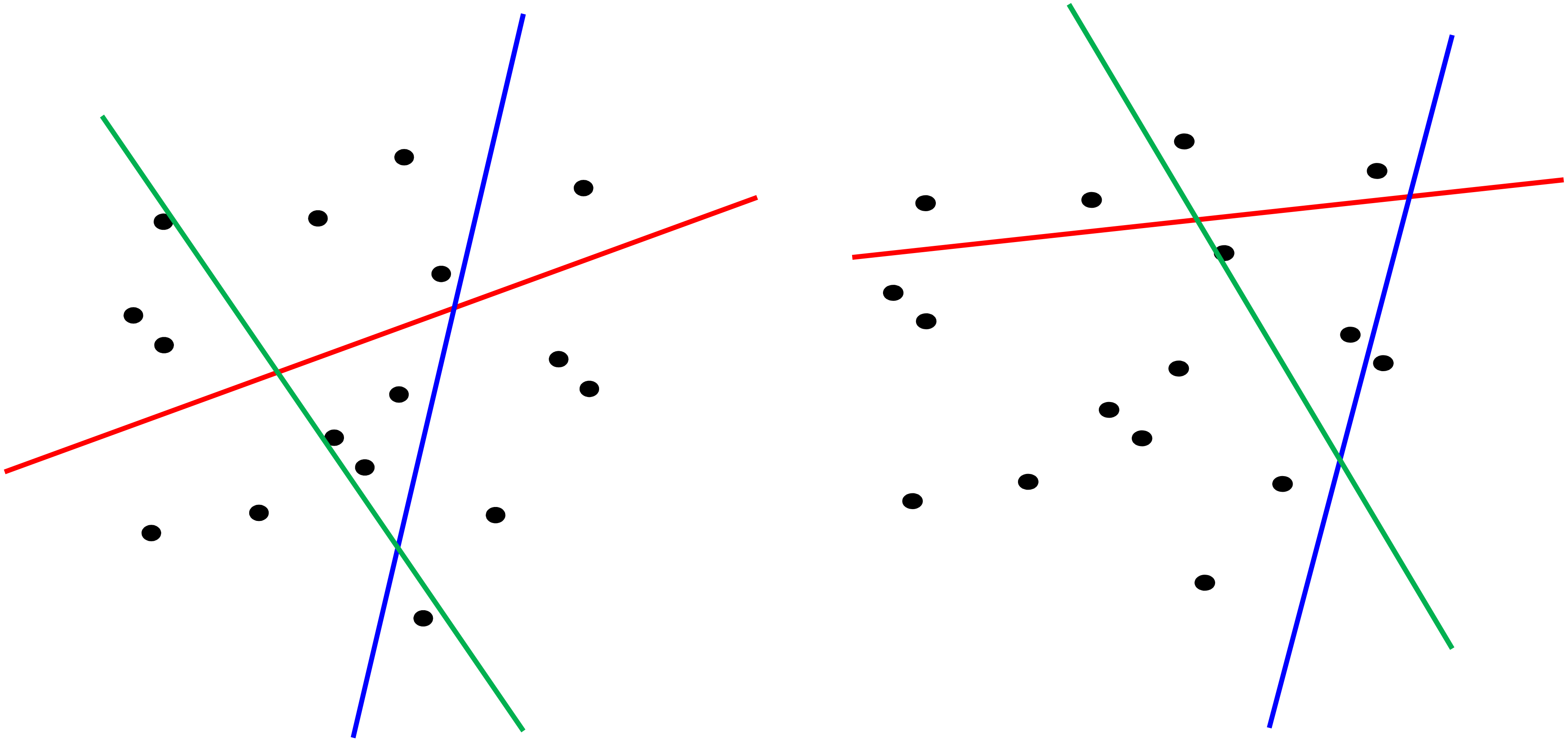
Distance between Two Points

- Measured by bit differences, known as Hamming distance
- Efficiently computed by XOR bit operations

$$d_{hd}(b_i, b_j) = |b_i \oplus b_j|$$



Good and Bad Hyper-Planes



Previous work focused on how to determine good hyper-planes

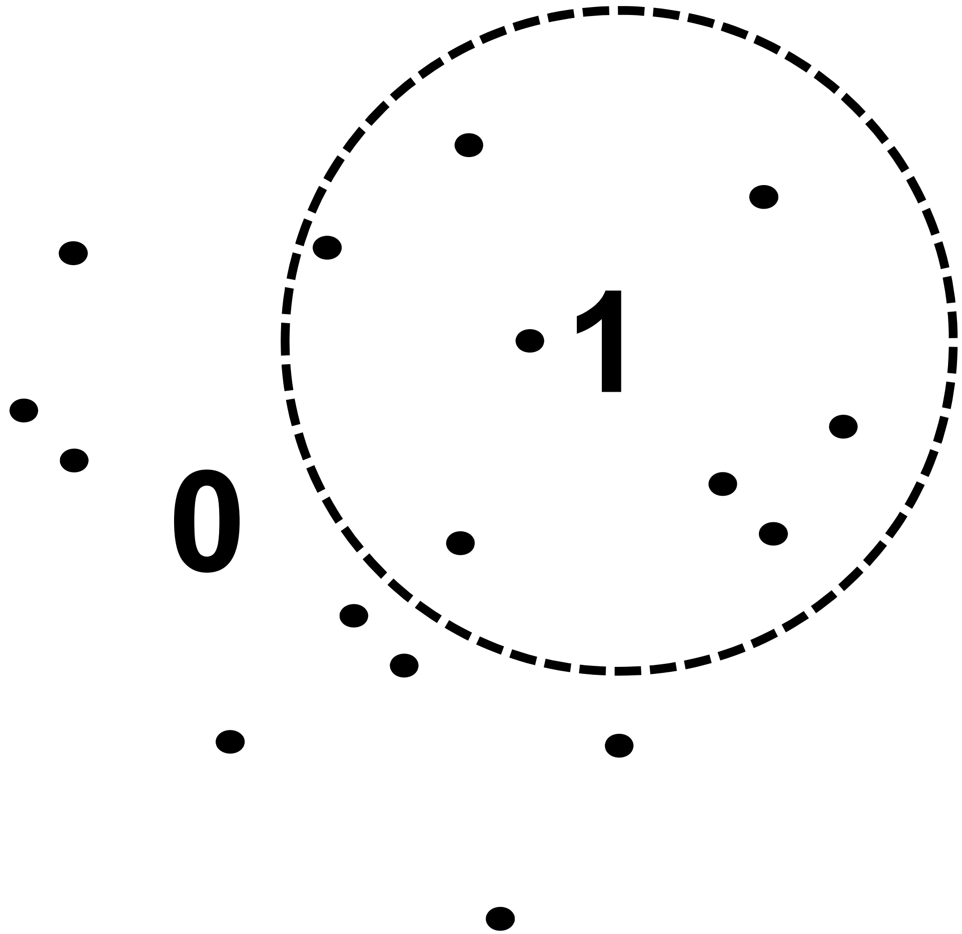
Components of Spherical Hashing

- Spherical hashing
- Hyper-sphere setting strategy
- Spherical Hamming distance

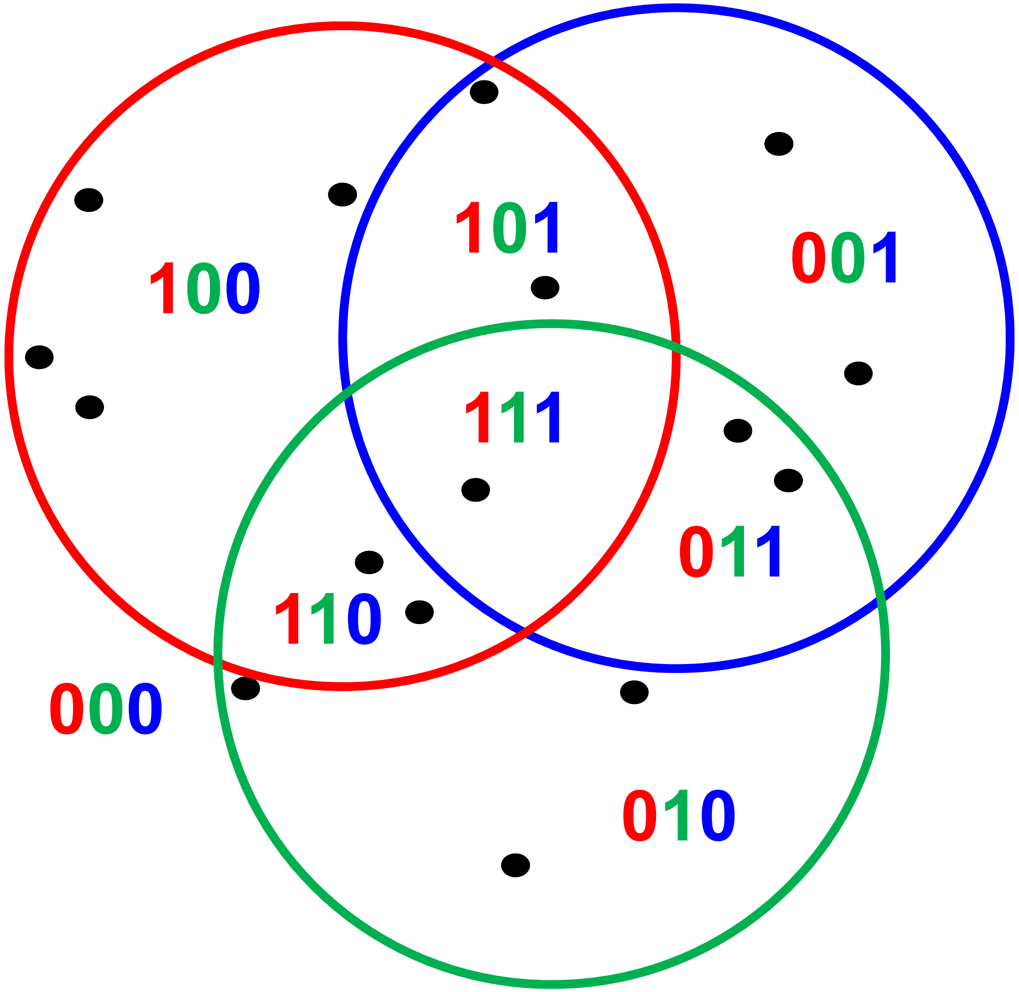
Components of Spherical Hashing

- **Spherical hashing**
- Hyper-sphere setting strategy
- Spherical Hamming distance

Spherical Hashing [Heo et al., CVPR 12]

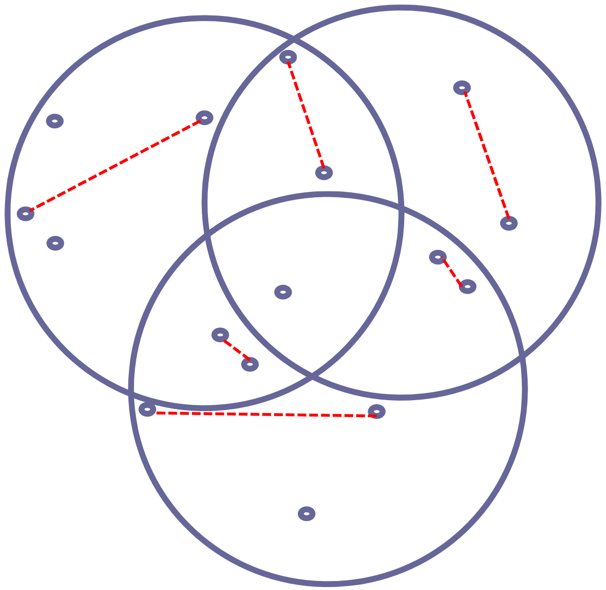


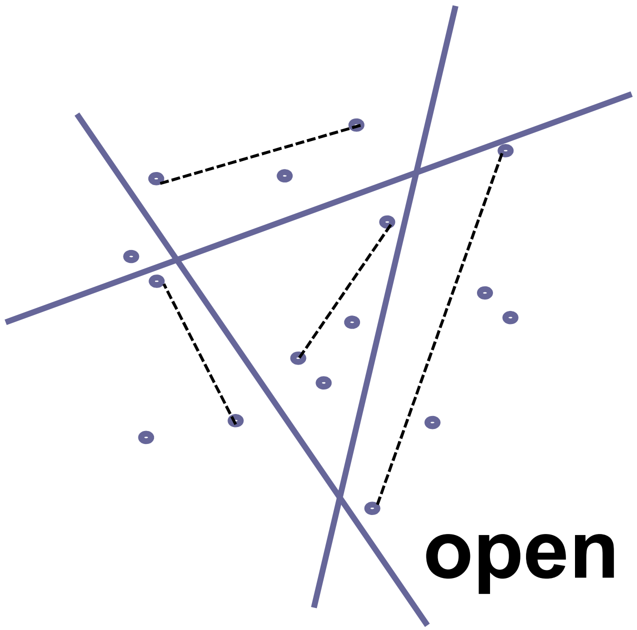
Spherical Hashing [Heo et al., CVPR 12]

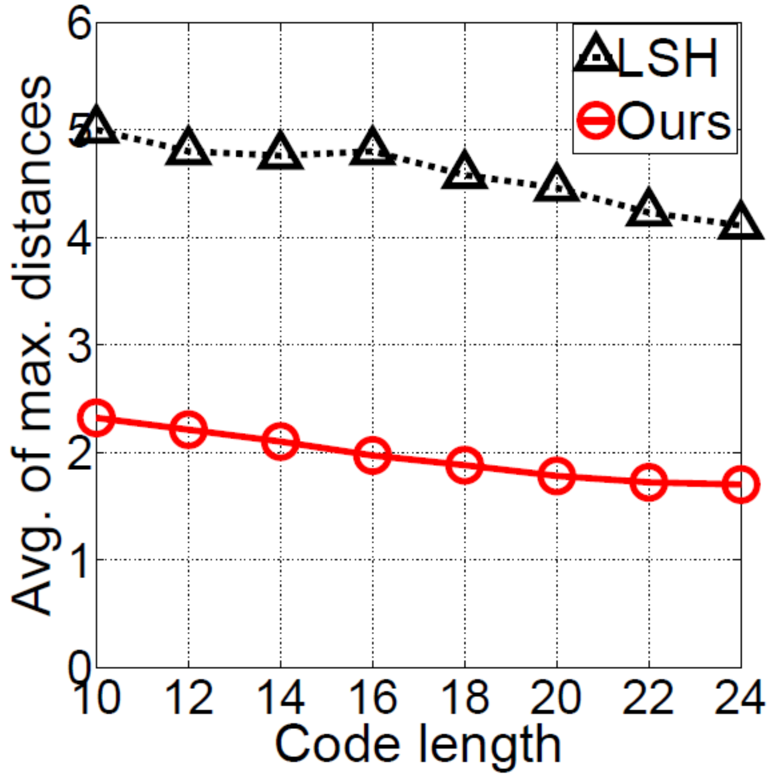


KAIST

Hyper-Sphere vs Hyper-Plane







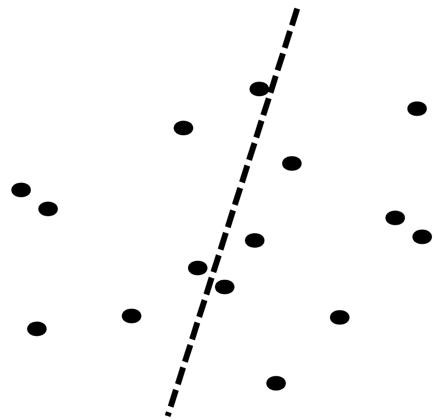
Average of maximum distances within a partition:
- Hyper-spheres gives tighter bound!

Components of Spherical Hashing

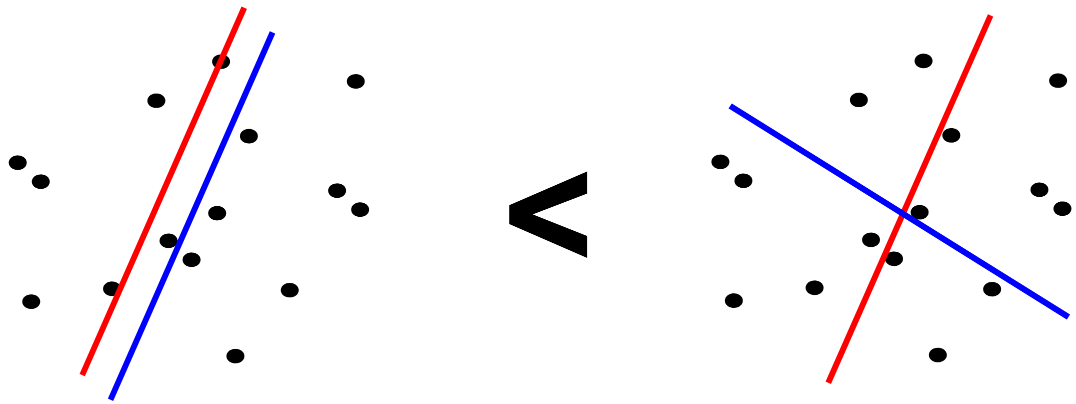
- Spherical hashing
- **Hyper-sphere setting strategy**
- Spherical Hamming distance

Good Binary Coding [Yeiss 2008, He 2011]

1. Balanced partitioning

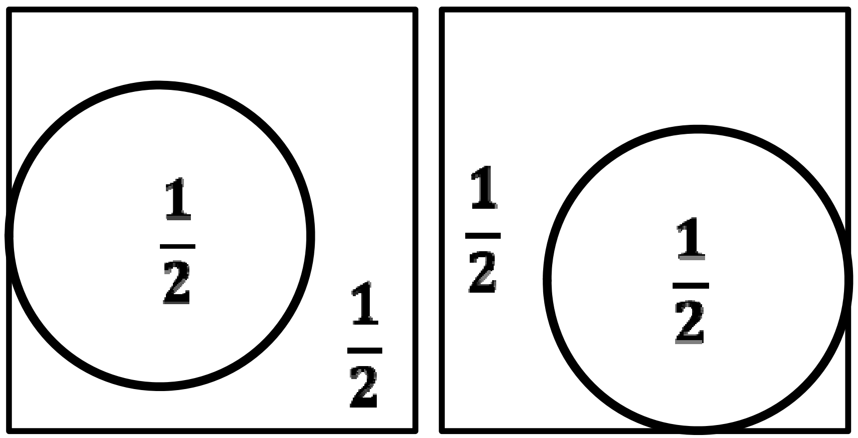


2. Independence

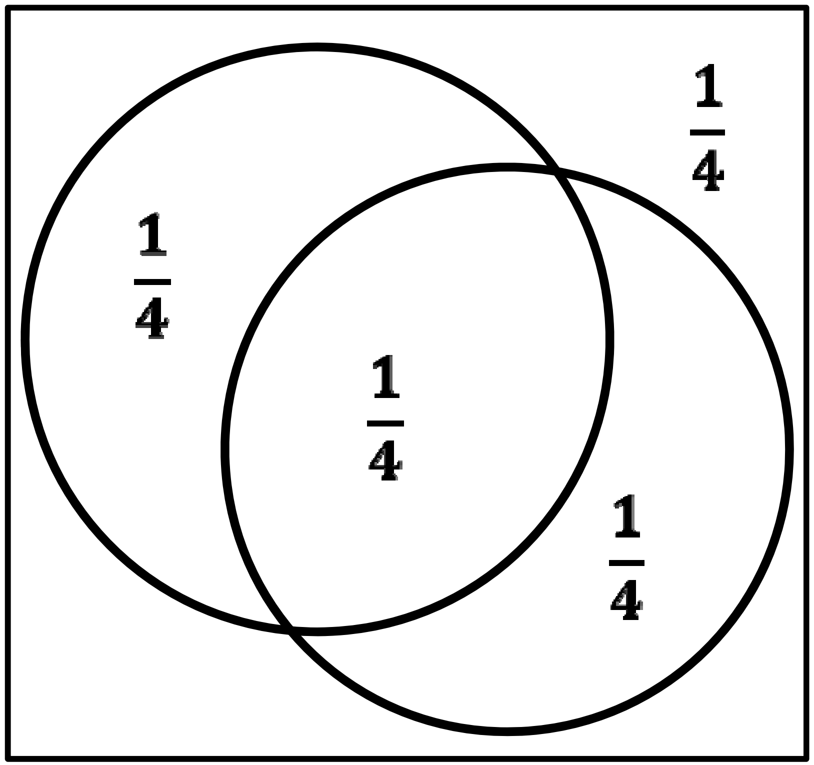


Intuition of Hyper-Sphere Setting

1. Balance

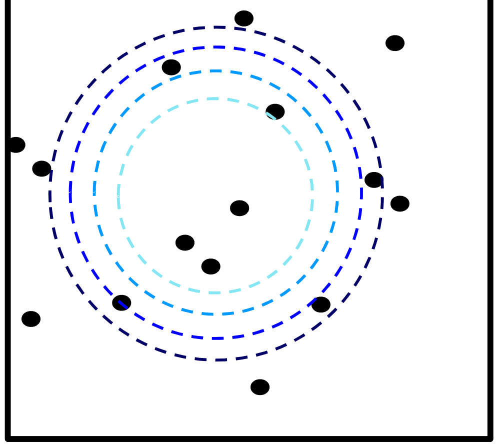


2. Independence

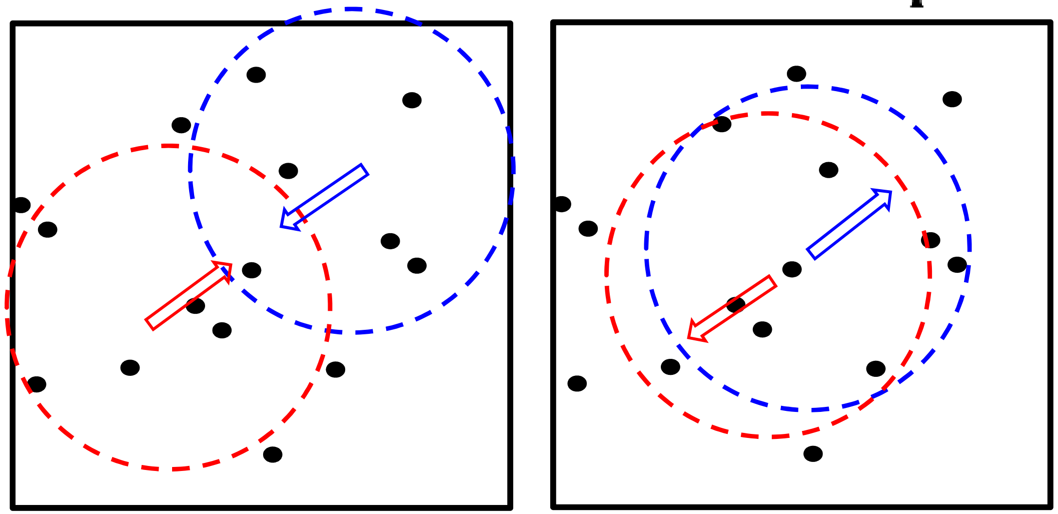


Hyper-Sphere Setting Process

1. Balance
- by controlling radius for $n(S) = \frac{N}{2}$



2. Independence
- by moving two hyper-spheres for $n(S_1 \cap S_2) = \frac{N}{4}$

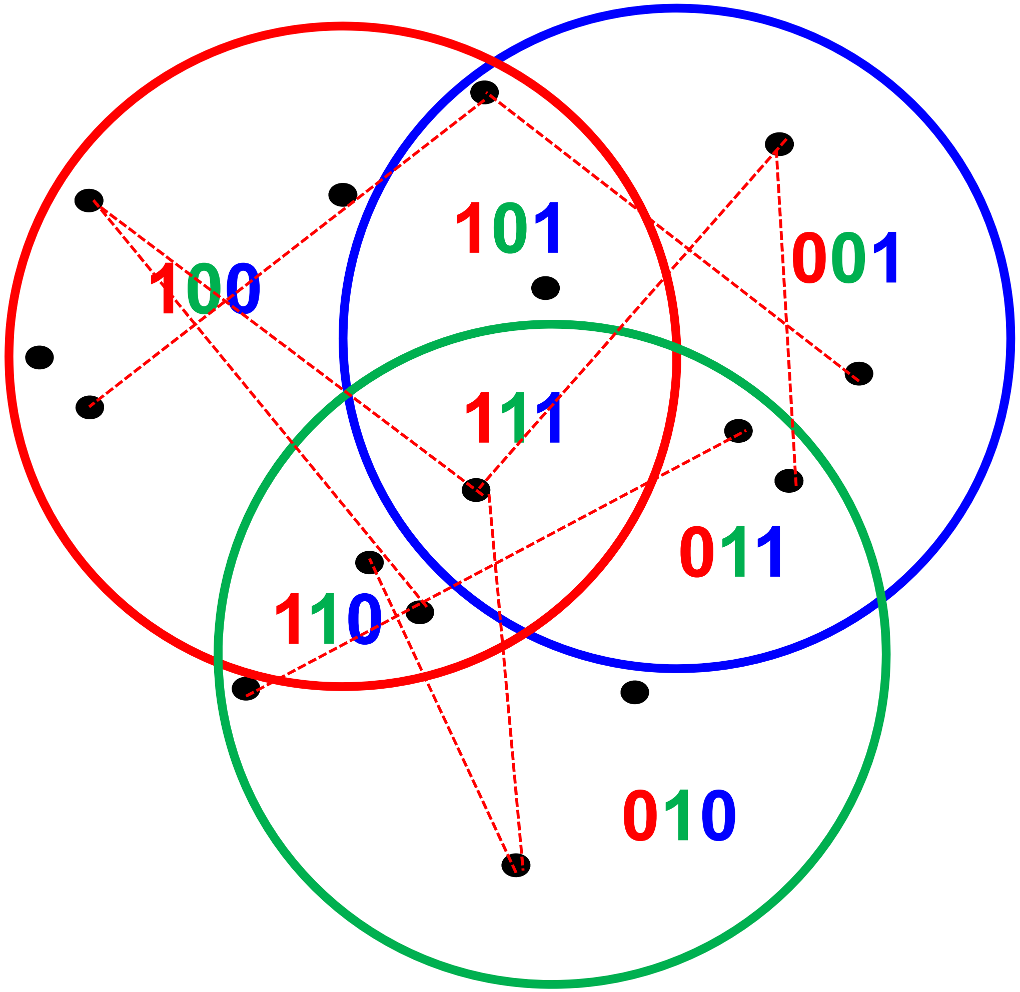


Iteratively repeat step 1, 2 until convergence.

Components of Spherical Hashing

- Spherical hashing
- Hyper-sphere setting strategy
- **Spherical Hamming distance**

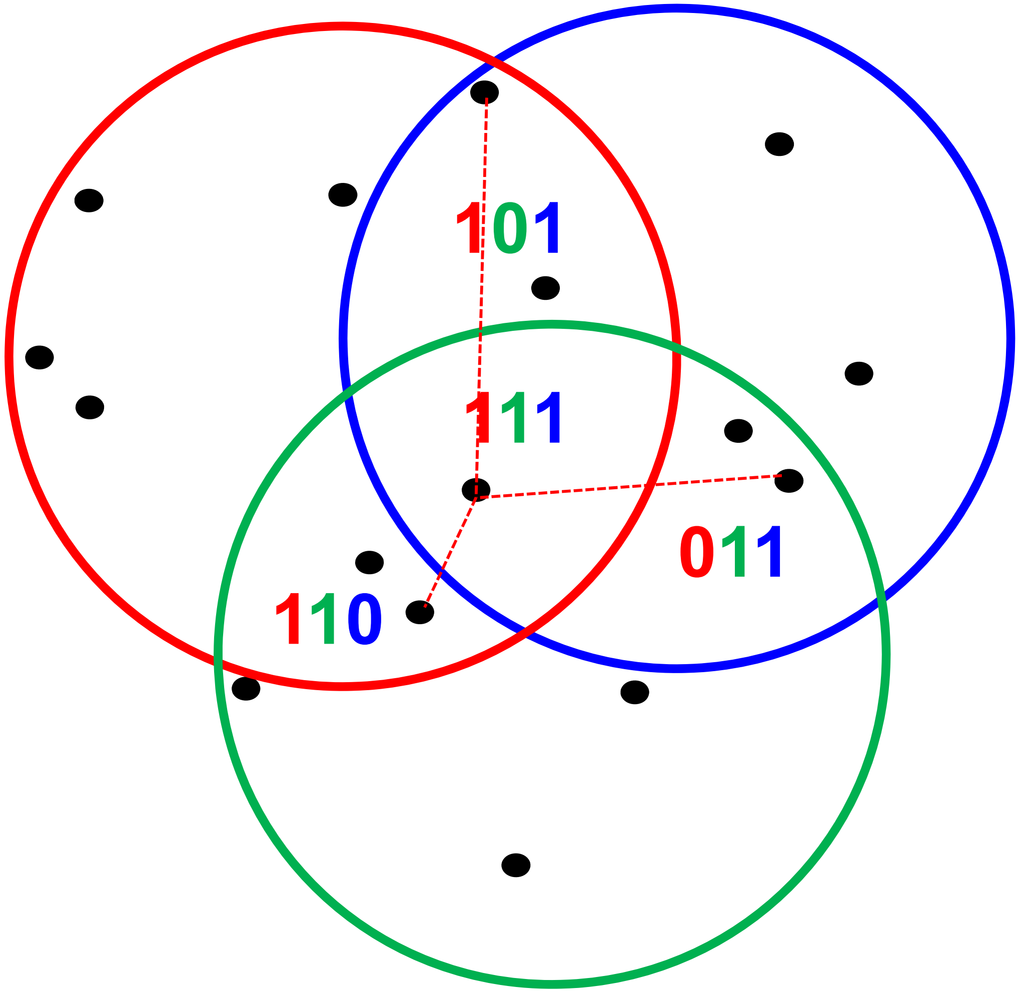
Max Distance and Common ‘1’



Common ‘1’s

: 1

Max Distance and Common '1'

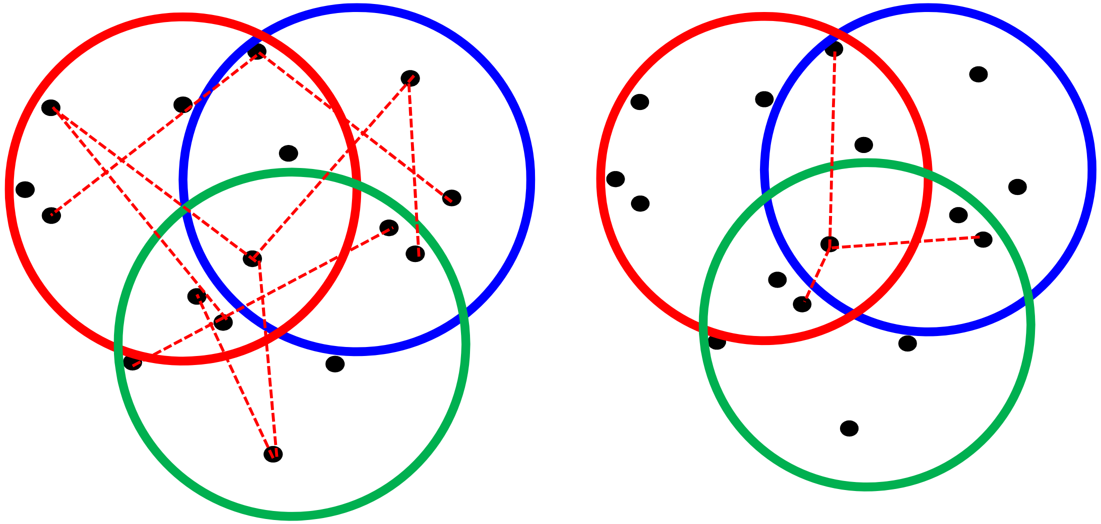


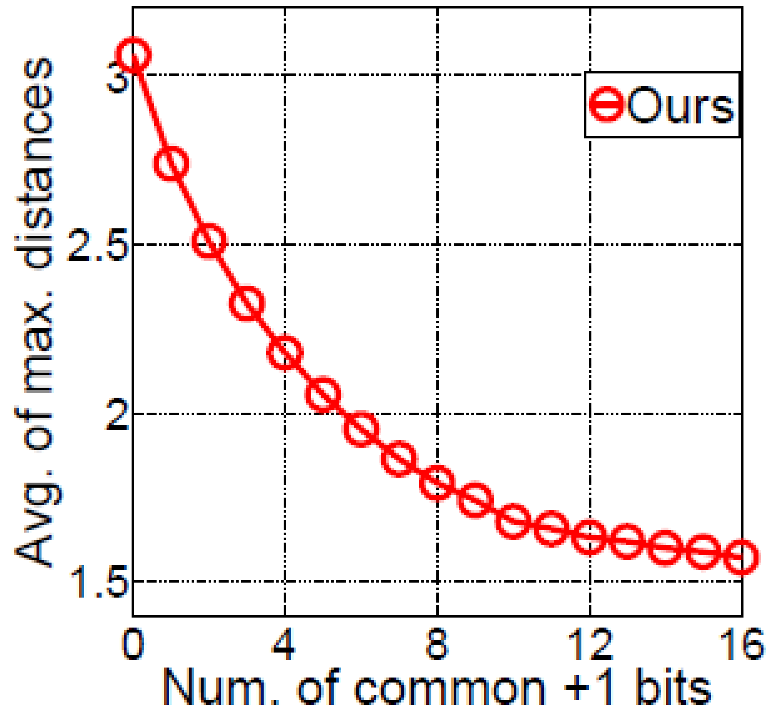
Common '1's

: **2**

Max Distance and Common ‘1’

Common ‘1’s: **1** Common ‘1’s: **2**





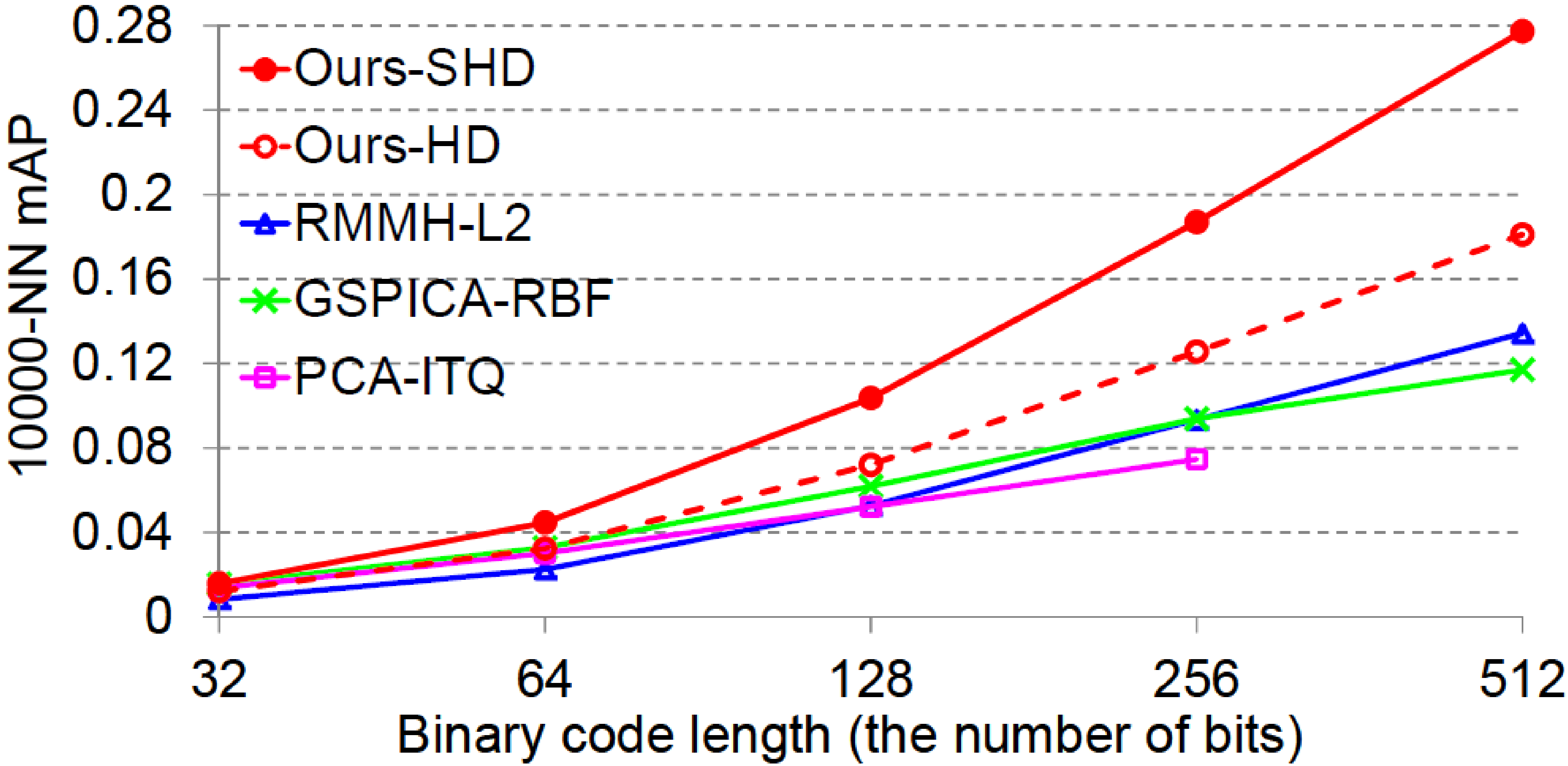
Average of maximum distances between two partitions: decreases as number of common ‘1’

Spherical Hamming Distance (SHD)

$$d_{shd}(b_i, b_j) = \frac{|b_i \oplus b_j|}{|b_i \wedge b_j|}$$

SHD: Hamming Distance divided by the number of common '1's.

Results

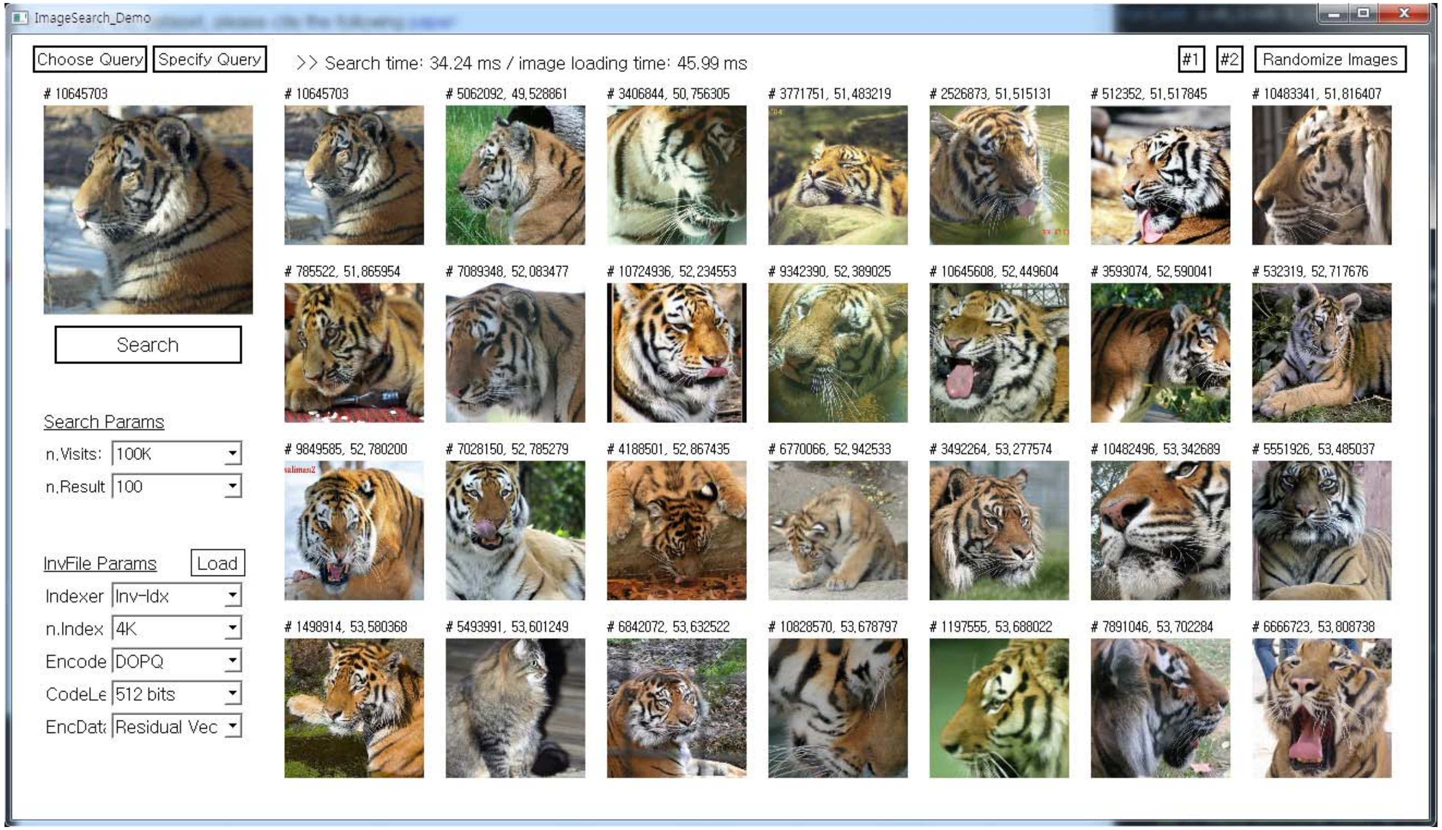


384 dimensional 75 million GIST descriptors

KAIST

Results of Image Retrieval

- **Collaborated with Adobe**
  - **11M images**
  - **Use deep neural nets for image representations**
  - **Spend only 35 ms for a single CPU thread**



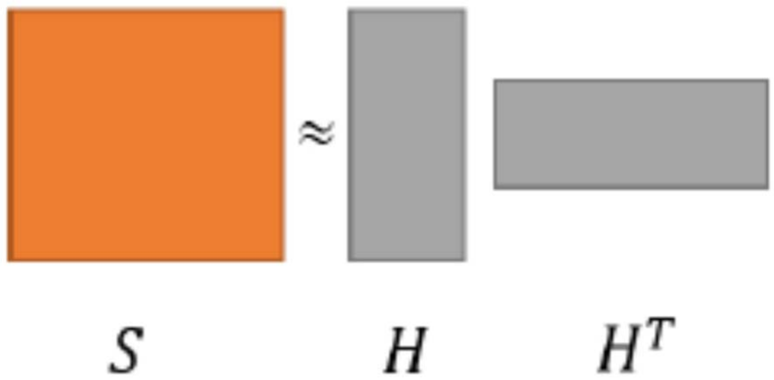
KAIST

Supervised Hashing

- **Utilize image labels**
  - **Conducted by using deep learning**

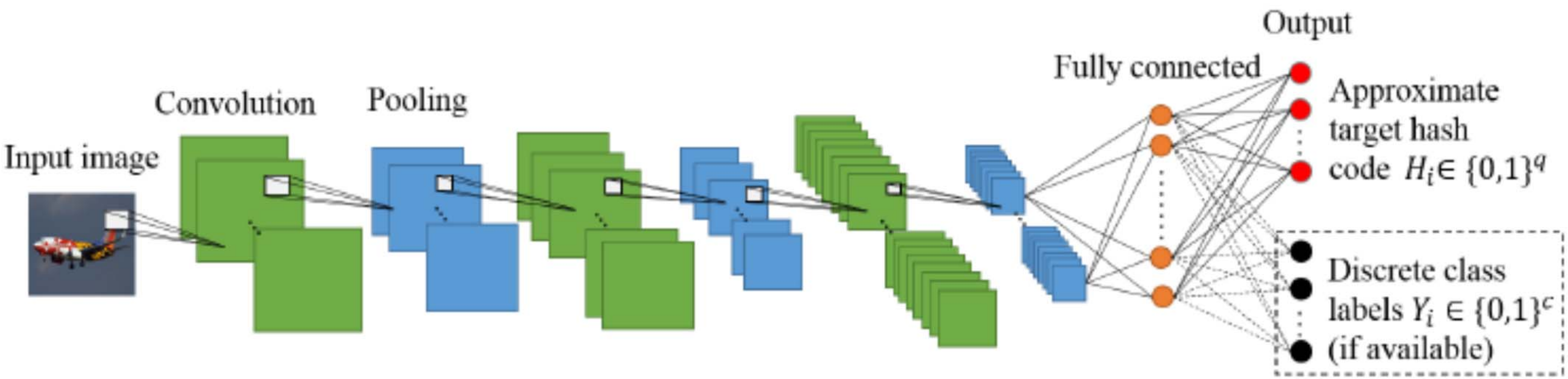
Supervised hashing for image retrieval via image representation learning, AAA 14

- **First step: approximate hash codes**
  - **S (similarity matrix, i.e., 1 when two images i & j have same label)**
  - **H (Hamming embedding, binary codes): dot products between two similar codes gives 1**
  - **Minimize the reconstruction error between S and similarity between codes**



Supervised hashing for image retrieval via image representation learning, AAA 14

- **Second step: learning image features and hash functions**
  - **Use Alexnet by utilizing approximate target hash codes and optionally class labels**
  - **Once the network is trained, it is used for test images**



Class Objectives were:

- **Understand the basic hashing techniques based on hyperplanes**
  - **Unsupervised approach**
- **Supervised approach using deep learning**

- **Codes are available**

  http://sglab.kaist.ac.kr/software.htm

Homework for Every Class

- **Go over the next lecture slides**
- **Come up with one question on what we have discussed today**
  - **Write questions three times**
- **Go over recent papers on image search, and submit their summary before Tue. class**

Next Time…

- **CNN based image search techniques**

Fig