CS588: Deep learning Image Search

Re-Ranking and Inverted Index

Sung-Eui Yoon
(윤성의)

Course URL:
http://sgvr.kaist.ac.kr/~sungeui/IR

Class Objectives

- **Discuss re-ranking (i.e., post-processing) for achieving higher accuracy**
  - **Spatial verification**
  - **Query expansion**
- **Understand approximate nearest neighbor search**
  - **Inverted index and inverted multi-index**

- **At the last class:**
  - **CNN based image descriptors**
  - **Training losses, and data**
  - **Benchmarks**

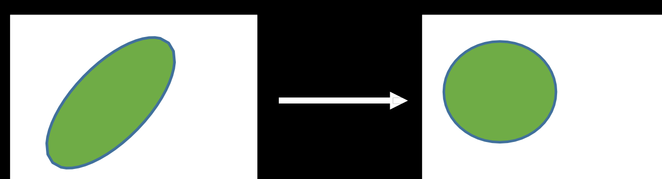
Descriptor processing with PCA

$$\hat{\mathbf{x}} = P^\top(\mathbf{x} - \mu)$$

$P \in \mathbb{R}^{d \times d}$ eigen-vectors as columns

$\mu \in \mathbb{R}^d$ mean vector

$\mathbf{x} \in \mathrm{R}^d$ global descriptor



[Jegou & Chum, ECCV'12]

Post-processing with whitening

$\mathcal{X} = f($  $)$

$\mathcal{X} = \{\mathbf{x}\}, \ \mathbf{x} \in \mathbb{R}^d$

process & aggregate

post-processing (whitening)

local descriptor set

global descriptor

global descriptor

learned end-to-end

Post-processing with whitening

$\mathcal{X} = f($  $)$

$\mathcal{X} = \{\mathbf{x}\},\ \mathbf{x} \in \mathbb{R}^d$ → process & aggregate → FC (whitening)

local descriptor set | global descriptor | global descriptor

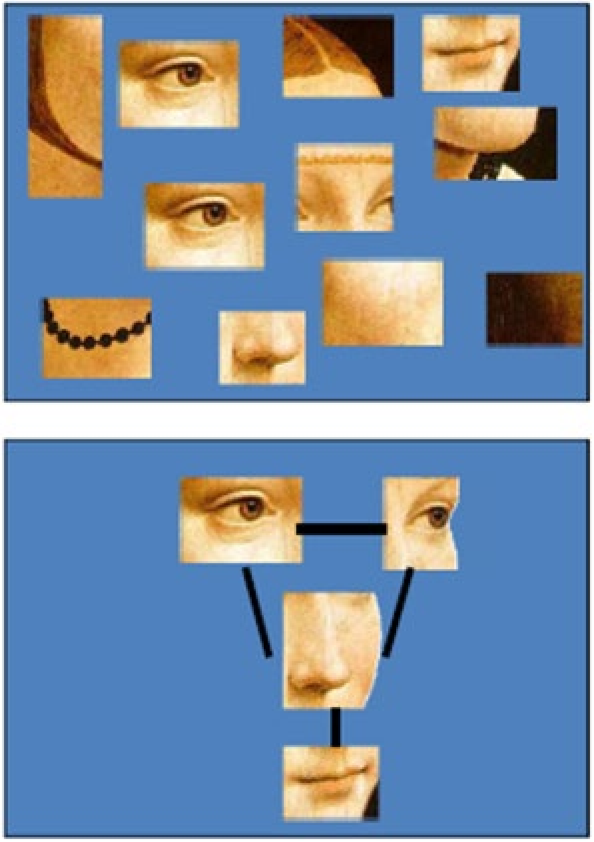
learned end-to-end

https://github.com/filipradenovic/cnnimageretrieval-pytorch

Problems of BoW or Global Descriptors

- **No spatial relationship between words**

- **How can we do better?**

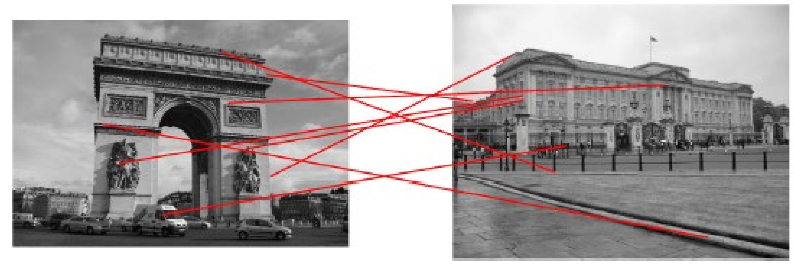


Ack.: Fei-Fei Li

Post-Processing or Reranking

Post-Processing

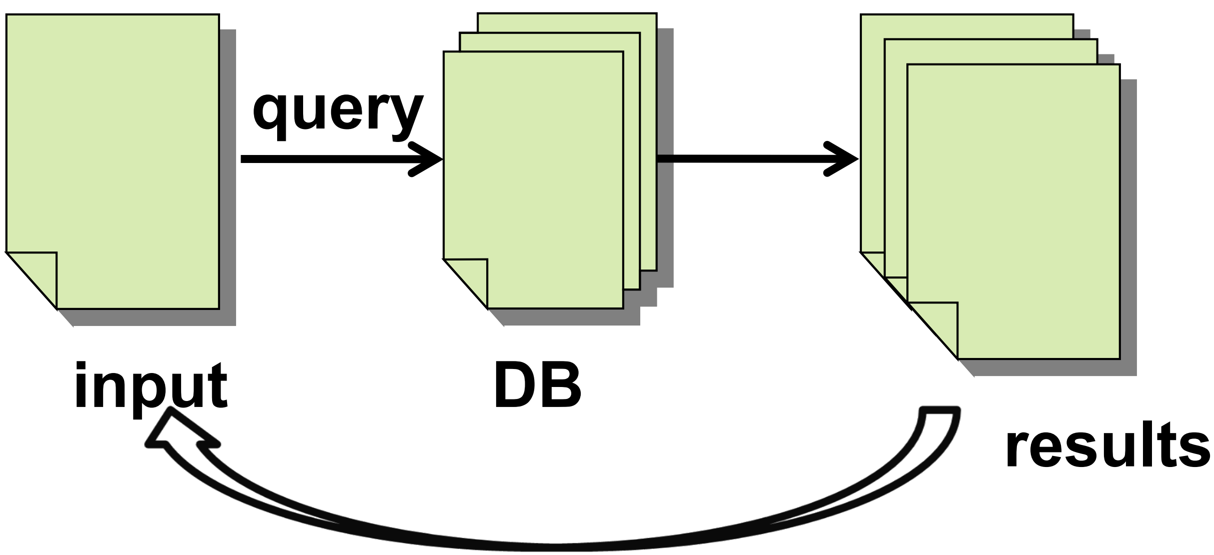
- Geometric verification
  - RANSAC



Matching w/o spatial matching

(Ack: Edward Johns et al.)

- Query expansion



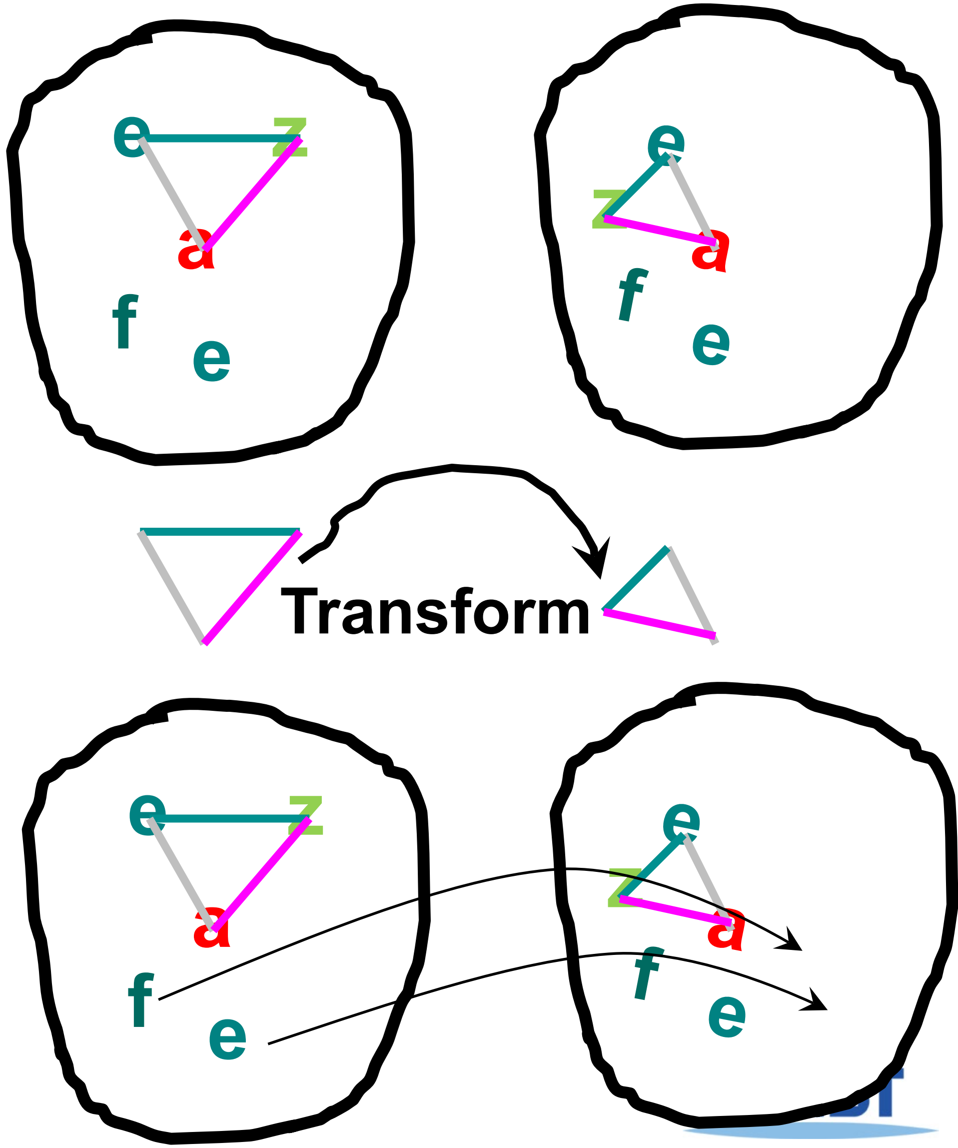
Geometric Verification using RANSAC

Repeat N times:

- Randomly choose 4 matching pairs

- Estimate transformation
    - Assume a particular transformation (Homography)

- Predict remaining points and count "inliers"

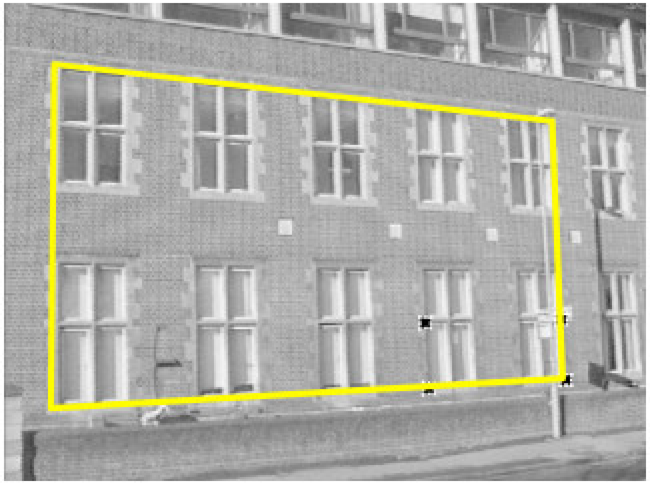


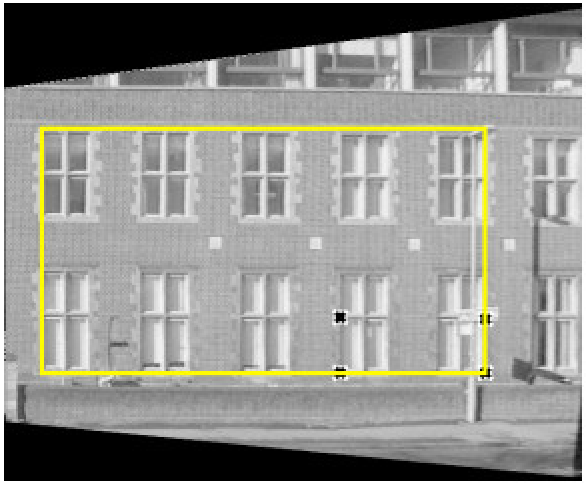
Ack.: Derek Hoiem (UIUC)

Homography

- **Transformation, H, between two planes**
  - **8 DoF due to normalization to 1**

$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

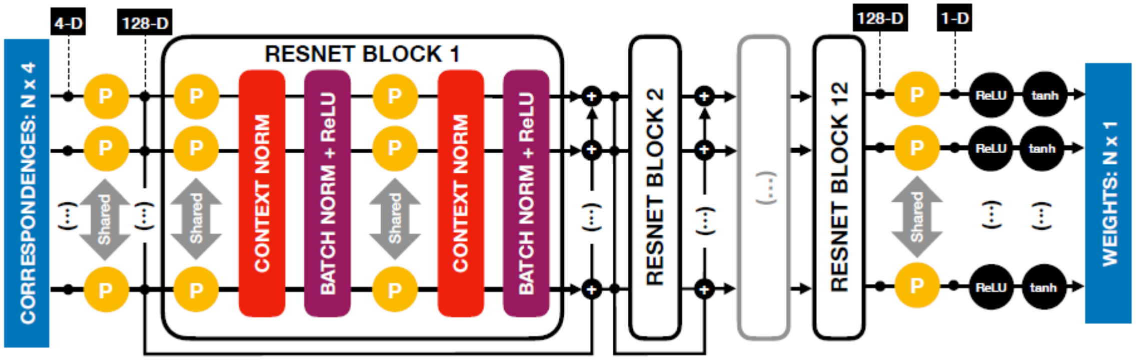


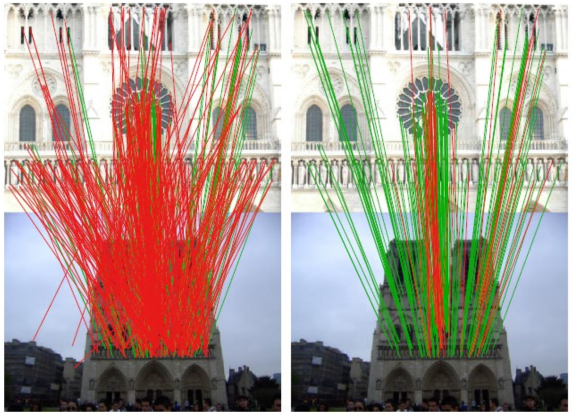


from Hartley & Zisserman

Learning to Find Good Correspondences, CVPR 18

- **Given two sets of input features (e.g., SIFTs), return a prob. of being inliers for each feature**
  - **Adopt the classification approach being inlier or not**
  - **Consider the relative motion between two images for the loss function**





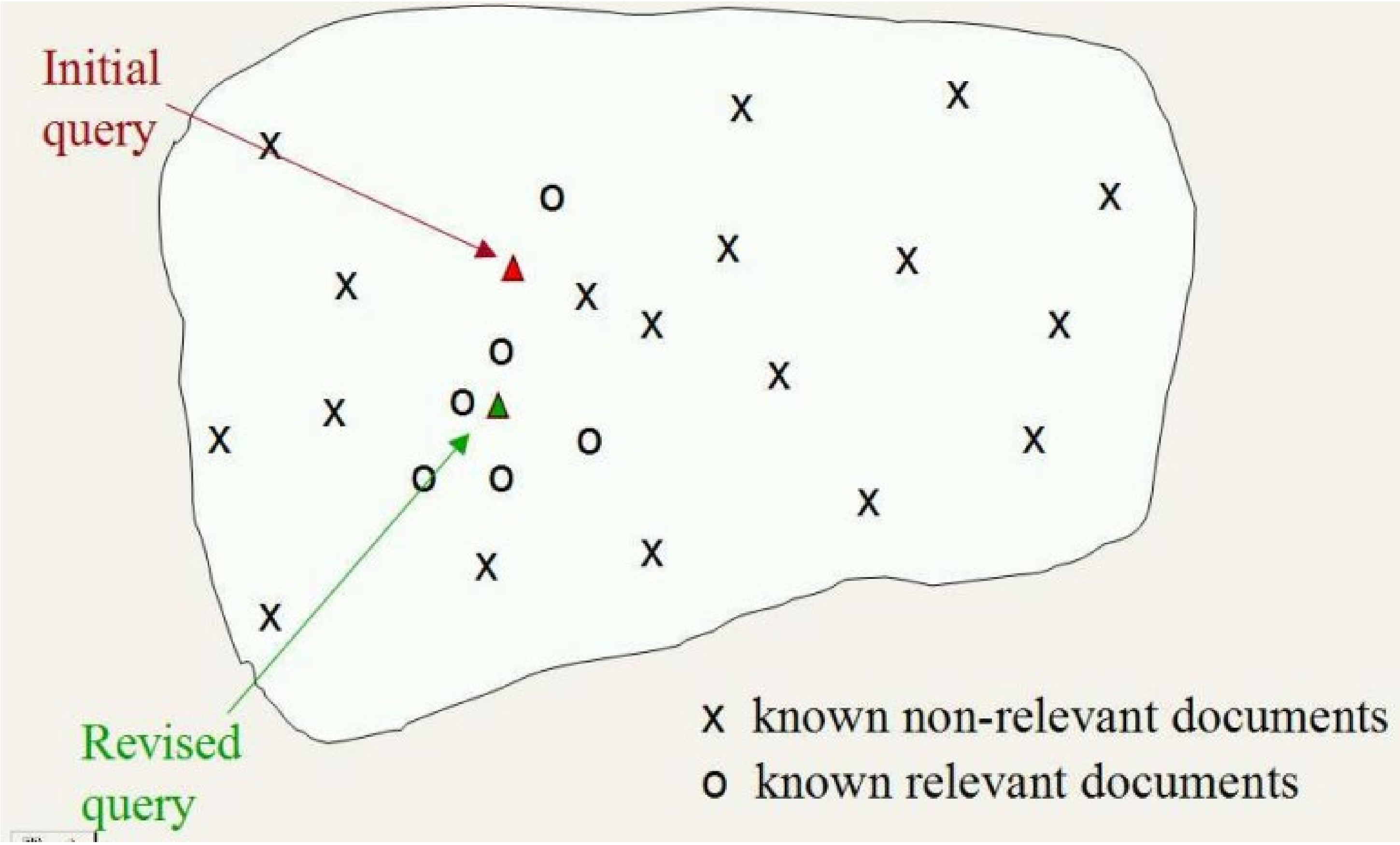
(a) RANSAC (b) Our approach

Pattern matching

- Drones surveying city
  - Identify a particular car
- Related to tracking

Query Expansion

- **Use NN information to get more confident query**



[Chum et al. ICCV'07]

Query Expansion

Original query

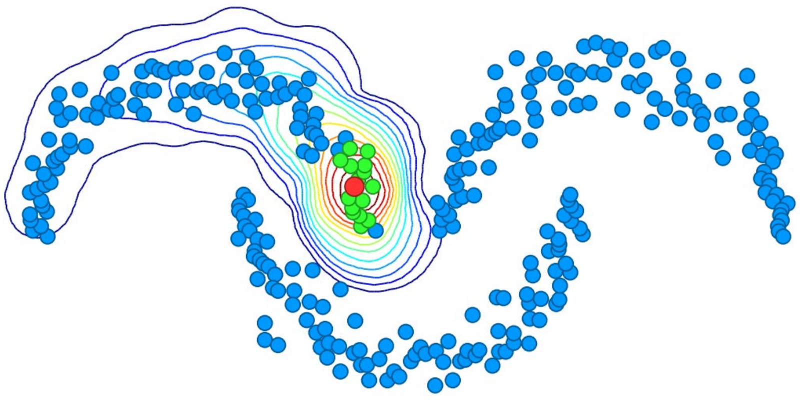
Top 4 images

Expanded results that were not identified by the original query

Diffusion (random walk) on Feature Space

- **High dimensional feature is likely to have a manifold shape**
- **Identify related images by the diffusion process, i.e., random walks**
  - **Perform random walks based on the similarity between a pair of images**

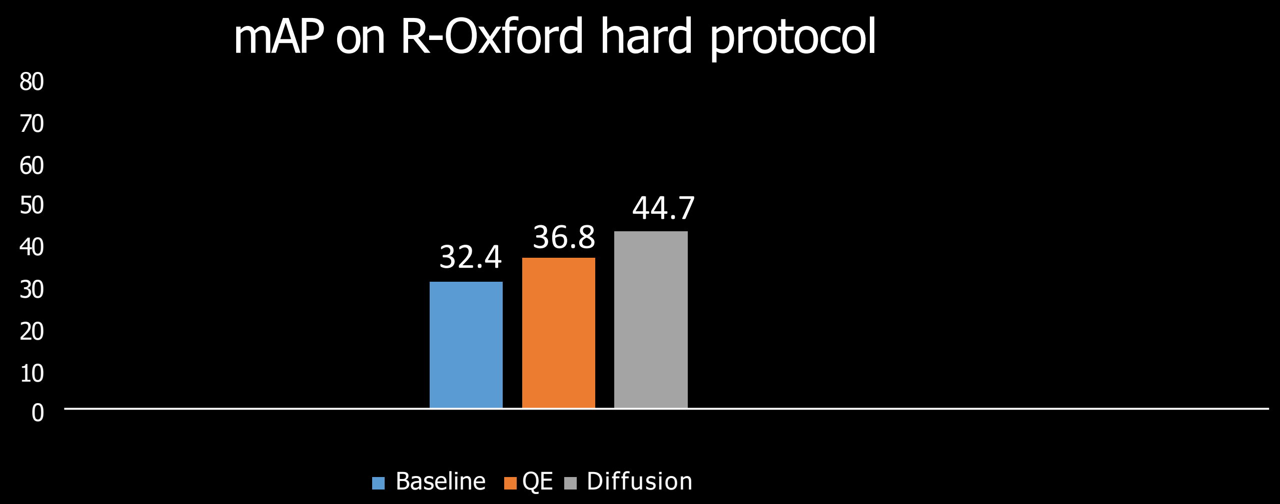
- **Utilize k-Nearest Neighbor (NNs) of the query images**



[Iscen et al. CVPR'19]

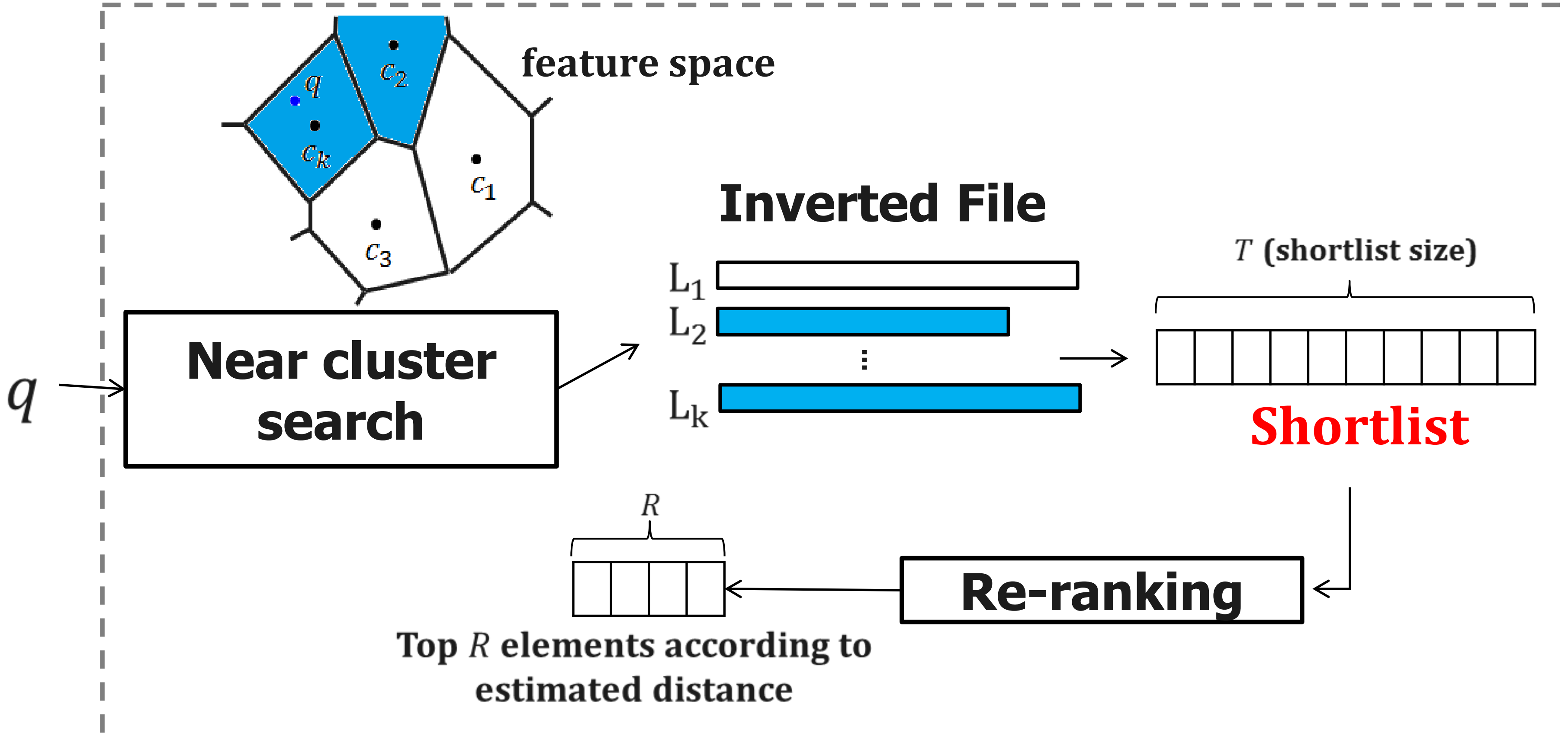
KAIST

Performance comparison



Inverted File or Index for Efficient Search

- **For each word, list images containing the word**



Ack.: Dr. Heo

Inverted Index

Construction time:

- Generate a codebook by quantization
  - e.g. k-means clustering

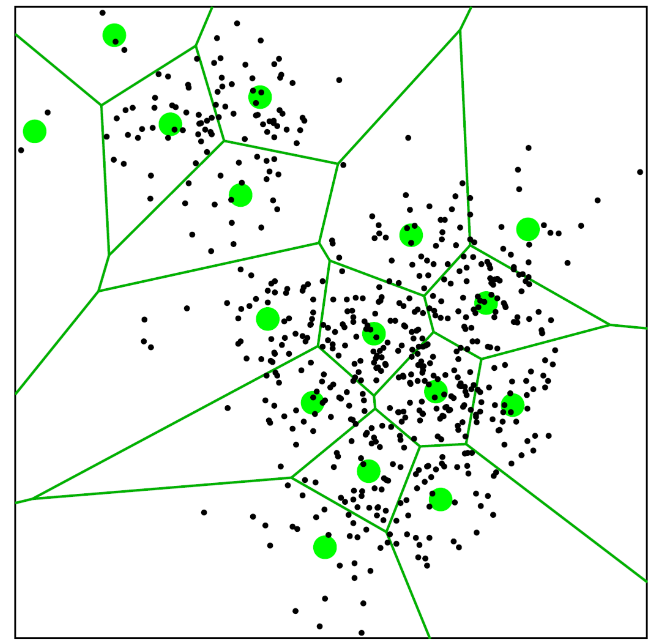
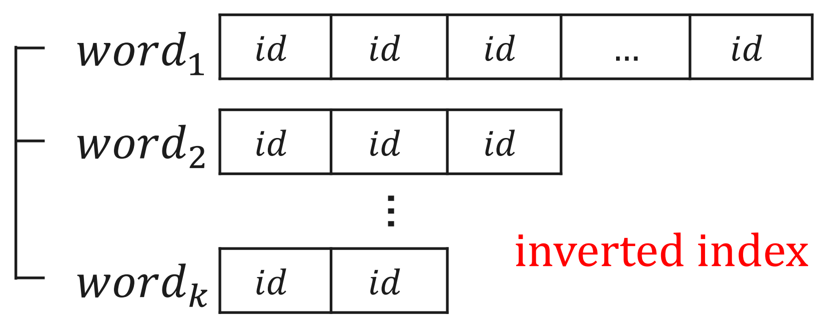


Figure from Lempitsky's slides

- Build an inverted index
  - Quantize each descriptor into the closest word
  - Organize desc. IDs in terms of words

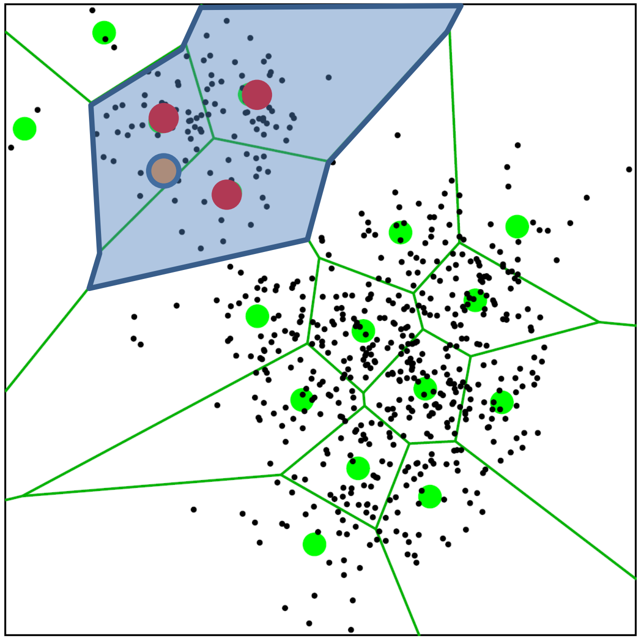


Ack.: Zhe Lin

Inverted Index

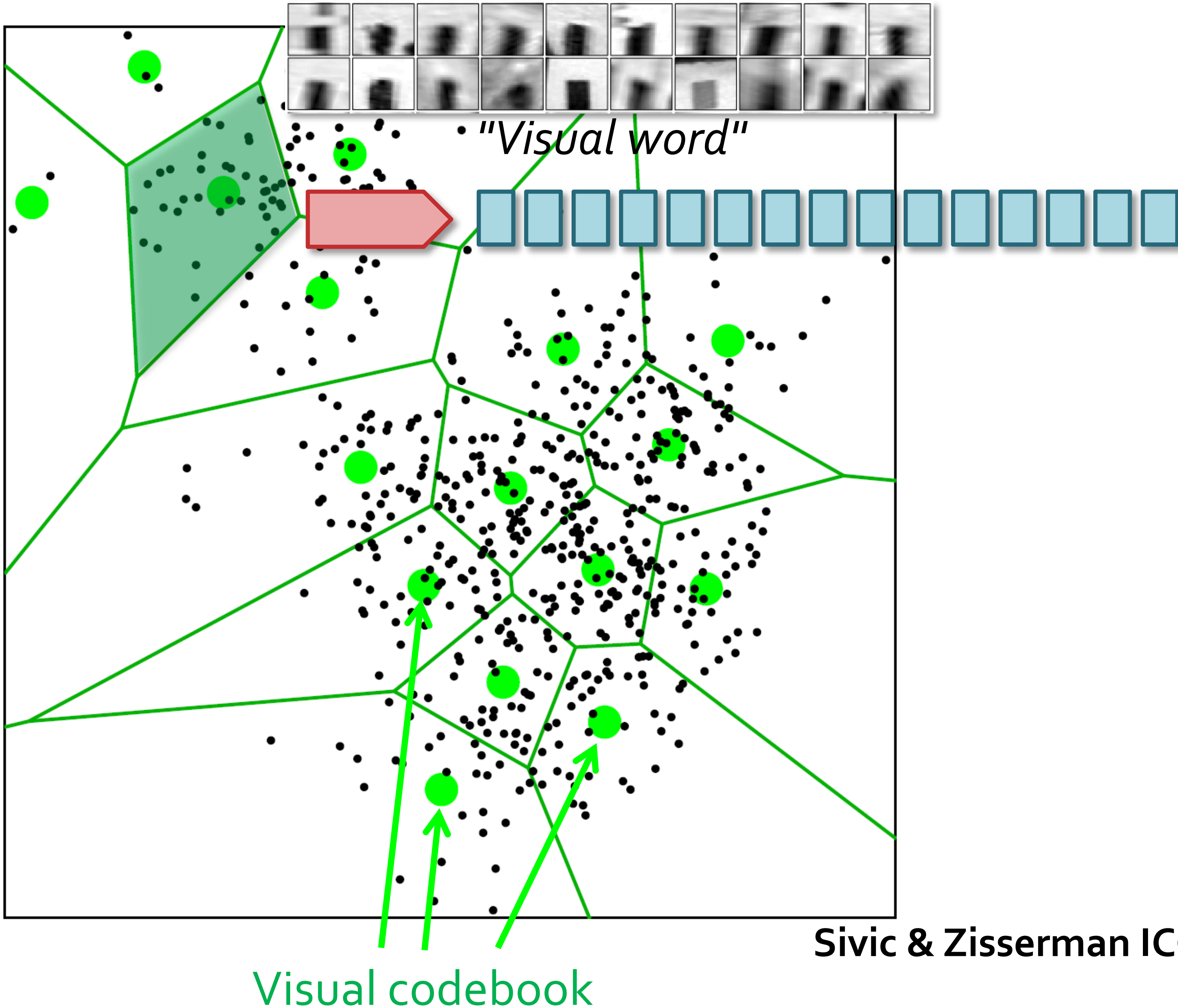
Query time:

- Given a query,
  - Find its K closest words
  - Retrieve all the data in the K lists corresponding to the words
- Large K
  - Low quantization distortion
  - Expensive to find kNN words



Ack.: Zhe Lin

The inverted index



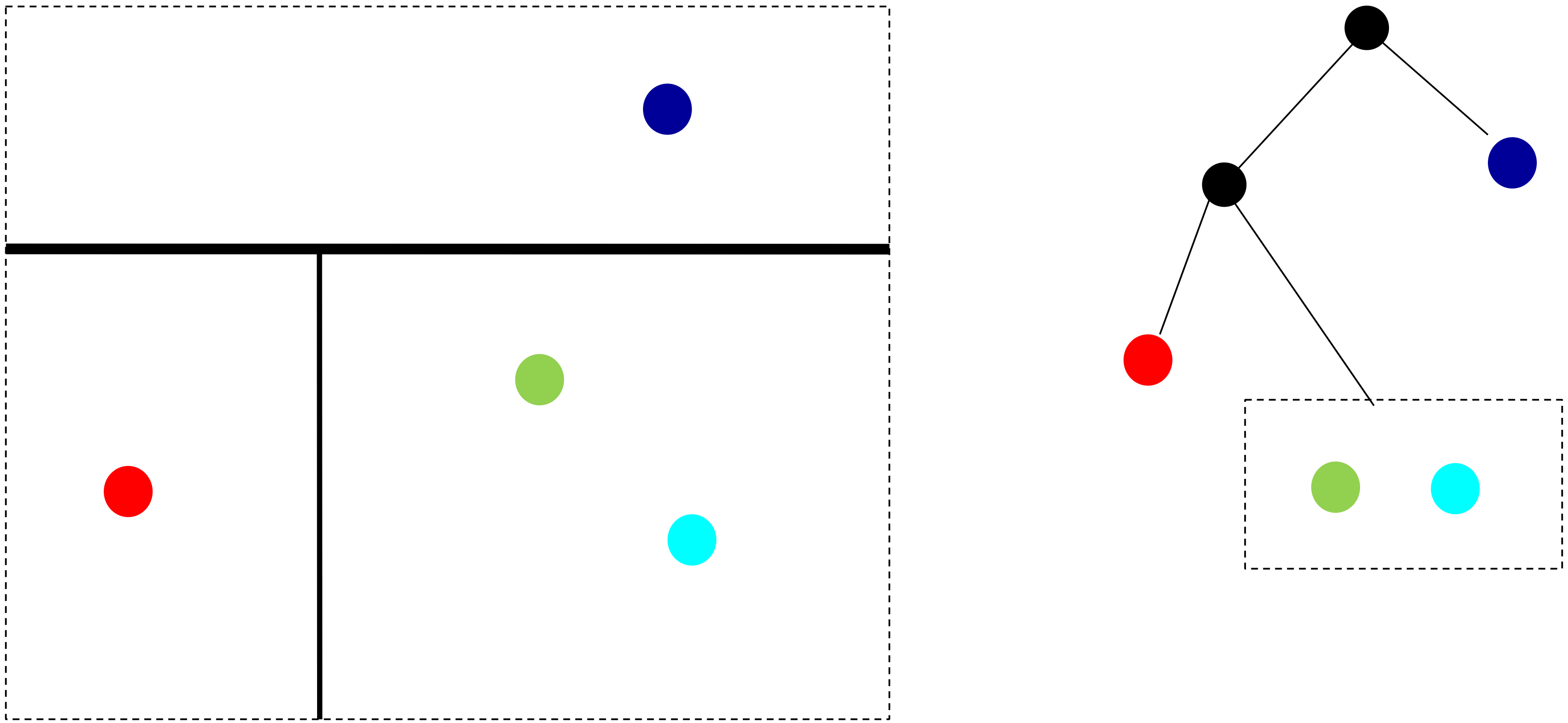
Sivic & Zisserman ICCV 2003

Approximate Nearest Neighbor (ANN) Search

- For large K
  - Takes time to find clusters given the query
  - Use those ANN techniques for efficiently finding near clusters

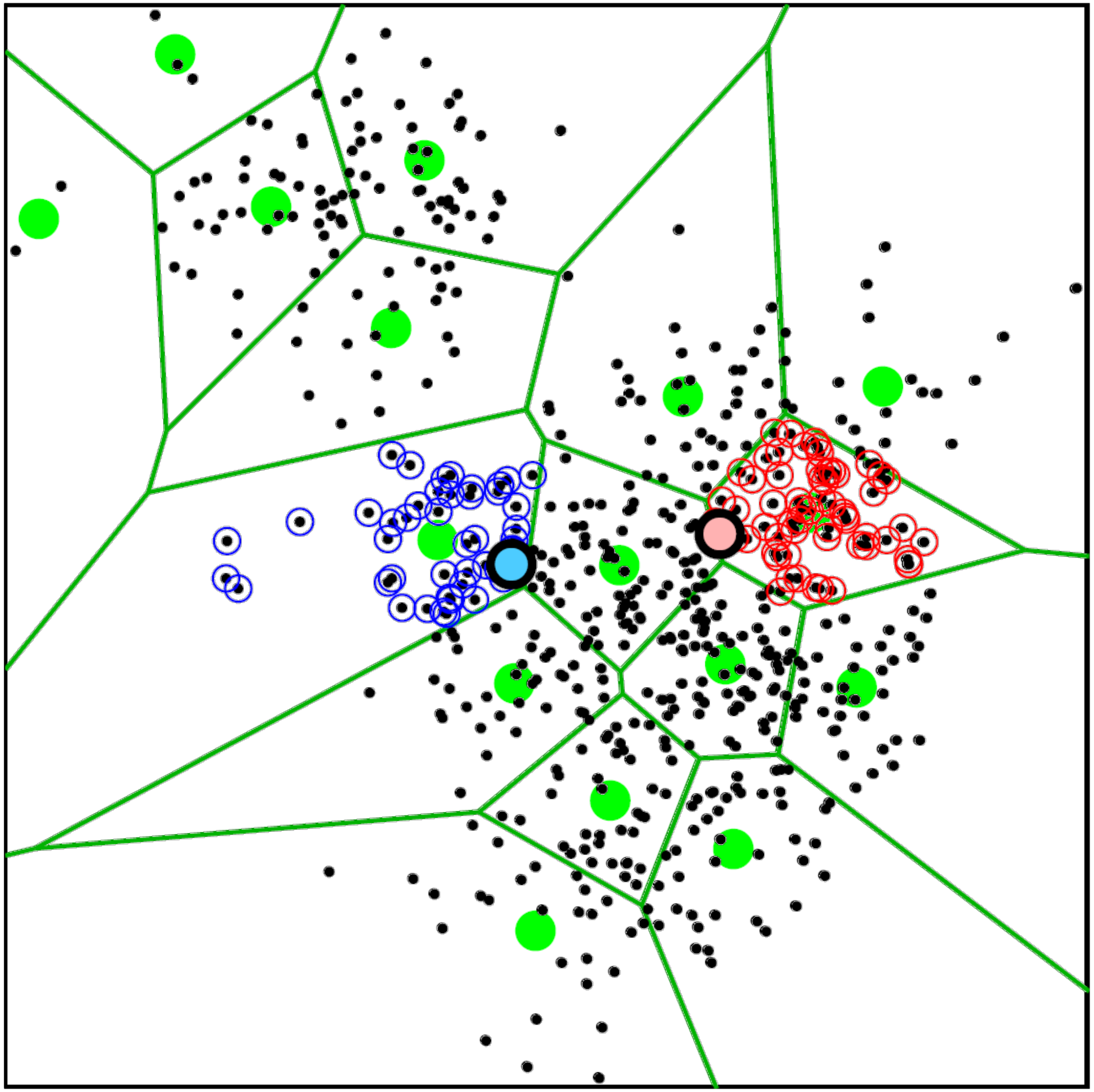
- ANN search techniques
  - kd-trees: hierarchical approaches for low-dimensional problems
  - Hashing for high dimensional problems; will be discussed later with binary code embedding
  - Quantization (k-means cluster and product quantization)

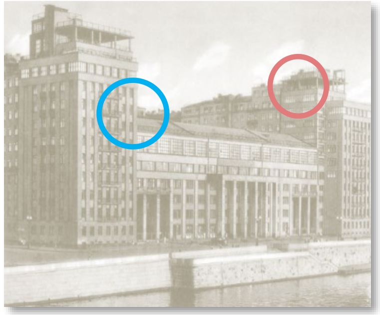
kd-tree Example



- **Many good implementations (e.g., vl-feat)**

Querying the inverted index



Query: 

- Have to consider several words for best accuracy
- Want to use as big codebook as possible

conflict

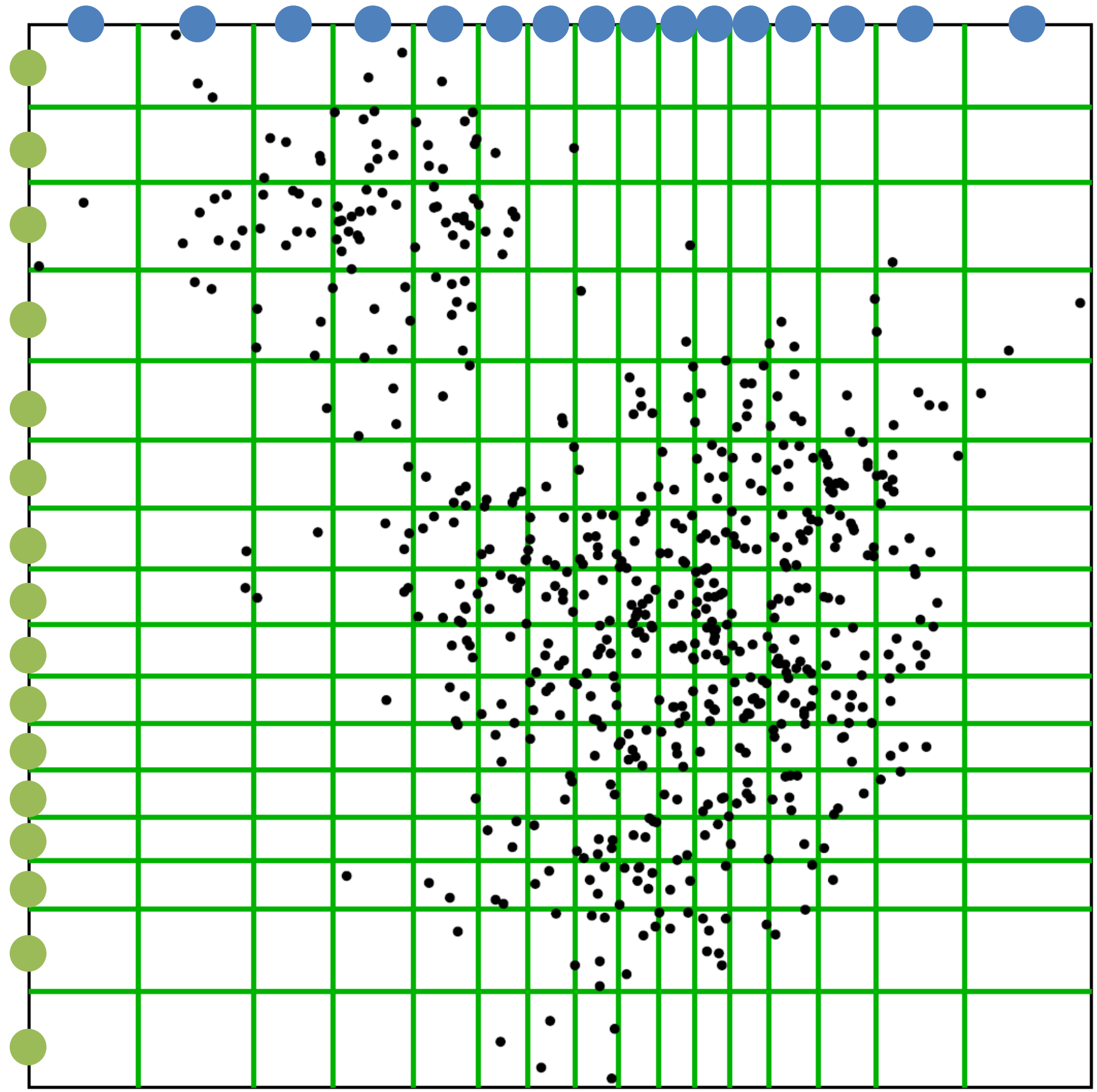
- Want to spend as little time as possible for matching to codebooks

Ack.: Lempitsky

Inverted Multi-Index [Babenko and Lempitsky, CVPR 2012]

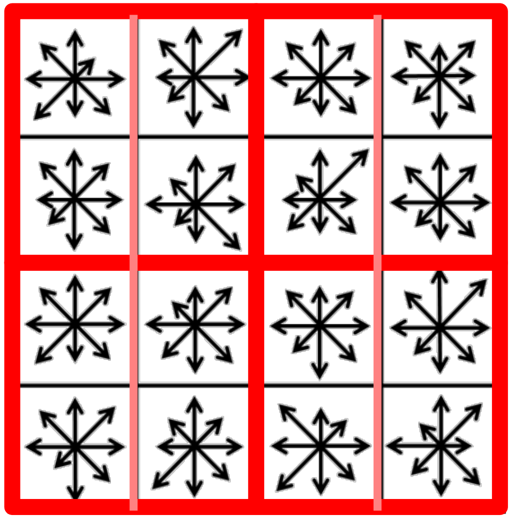
- **Product quantization for indexing**

- **Main advantage:**
  - For the same K, much finer subdivision
  - Very efficient in finding kNN codewords



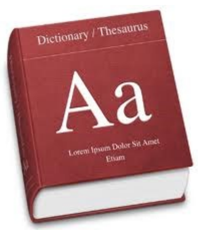
Ack.: Lempitsky

Product quantization



1. Split vector into correlated subvectors
2. use separate small codebook for each chunk

Quantization vs. Product quantization:

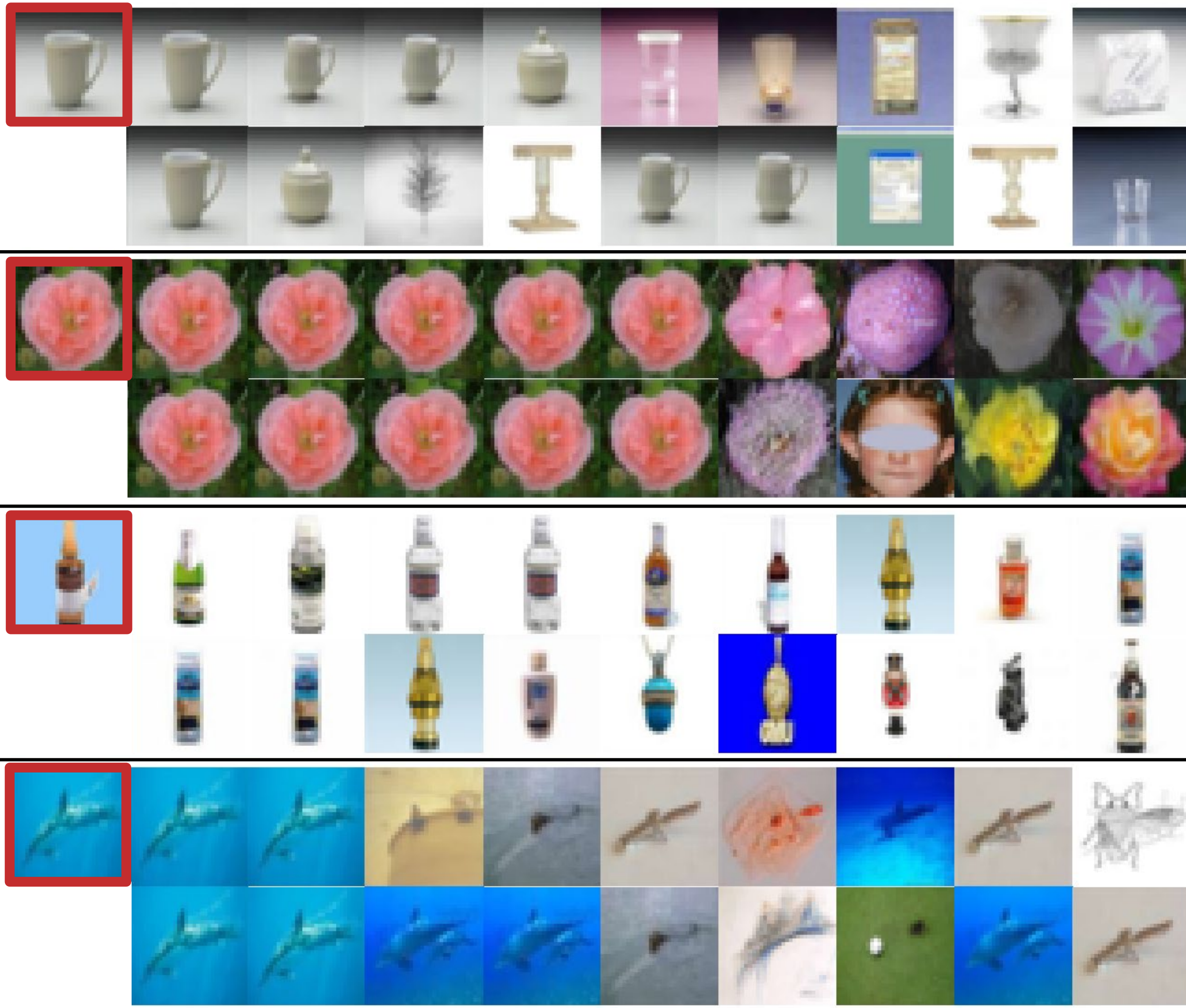


For a budget of 4 bytes per descriptor:

1. Use a single codebook with 1 billion codewords or	many minutes	128GB
2. Use 4 different codebooks with 256 codewords each	< 1 millisecond	32KB

Ack.: Lempitsky

Retrieval examples



Exact NN
Uncompressed GIST

Multi-D-ADC
16 bytes

Exact NN
Uncompressed GIST

Multi-D-ADC
16 bytes

Exact NN
Uncompressed GIST

Multi-D-ADC
16 bytes

Exact NN
Uncompressed GIST

Multi-D-ADC
16 bytes

Ack.: Lempitsky

Scalability

- Issues with billions of images?
  - Searching speed → inverted index
  - Accuracy → larger codebooks, spatial verification, expansion, features
  - Memory → compact representations
  - Easy to use?
  - Applications?
  - A new aspect?

Class Objectives were:

- **Discuss re-ranking (i.e., post-processing) for achieving higher accuracy**
  - **Spatial verification**
  - **Query expansion**
- **Understand approximate nearest neighbor search**
  - **Inverted index and inverted multi-index**

Next Time…

- **Hashing techniques**

Homework for Every Class

- **Go over the next lecture slides**
- **Come up with one question on what we have discussed today**
  - **1 for typical questions (that were answered in the class)**
  - **2 for questions with thoughts or that surprised me**

- **Write questions 3 times**