Similarity search with polysemous codes

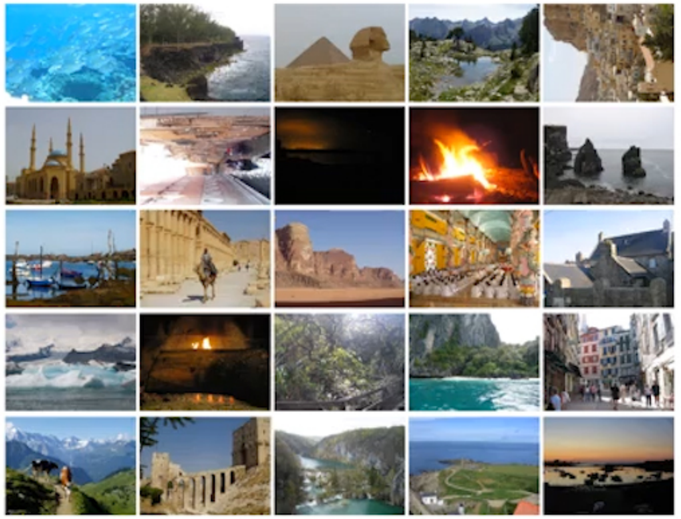
Matthijs Douze, Hervé Jégou, Florent Perronnin

Facebook AI Research

ECCV'2016

Problem setup

Build index for a collection:



Images from Inria Holidays
Copyright Jégou ©2005-2008

Media Description (CNN)

$y_1, y_2, \ldots, y_n \in \mathbb{R}^d$

Indexing

Index in RAM

Query:

$x \in \mathbb{R}^d$

Result: $k - \text{argmin}_{i=1..n} \|x - y_i\|^2$

Approximate search

Criteria: compact, fast, accurate

Polysemous codes, Douze, Jégou, Perronnin

Binary codes versus Product Quantization

Seen as concurrent methods in the literature

[Iterative quantization: A procrustean approach to learning binary codes, Gong, Lazebnik, CVPR'11]

Binarisation (ITQ)	PQ
comparison is context-free	need quantizer centroids
1190M comparisons / s	222M comparisons / s
precision=0.143	precision=0.442
Multi-index hashing high memory overhead	low memory overhead With an inverted file

[Fast search in Hamming space with multi-index hashing, Norouzi, Pubjabi, Fleet, CVPR'12]

How to get the best of both worlds?

Polysemous codes, Douze, Jégou, Perronnin

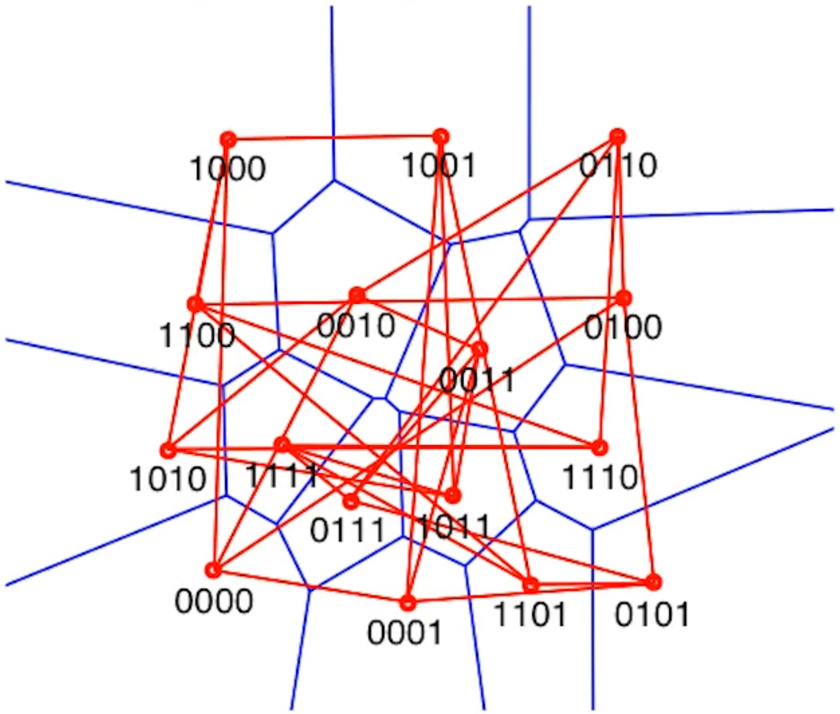
Optimize the order of PQ centroids: Index assignment

Given a k-means quantizer,

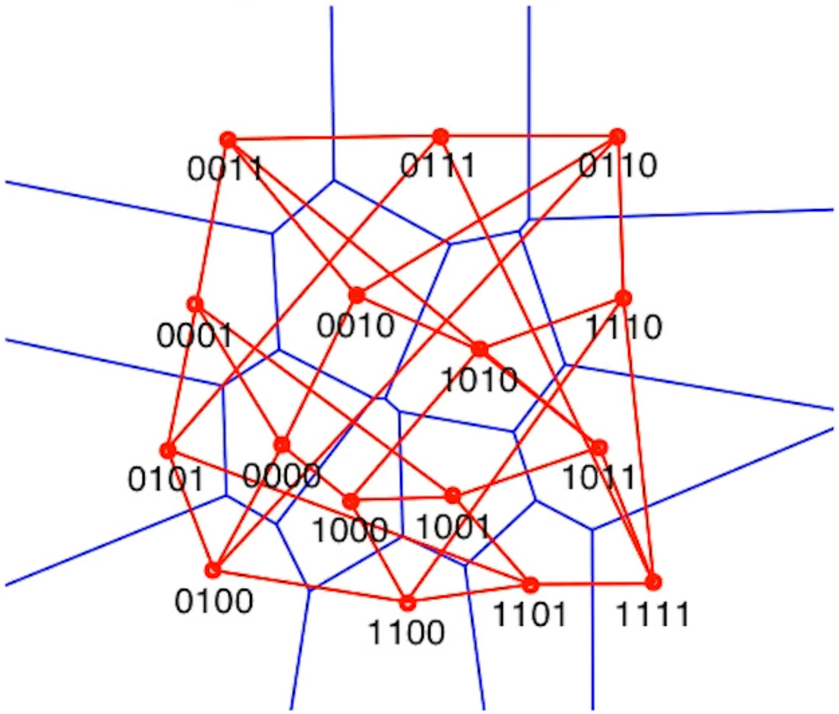
learn a permutation

So that binary comparison ≈ centroid distances

Regular PQ codes

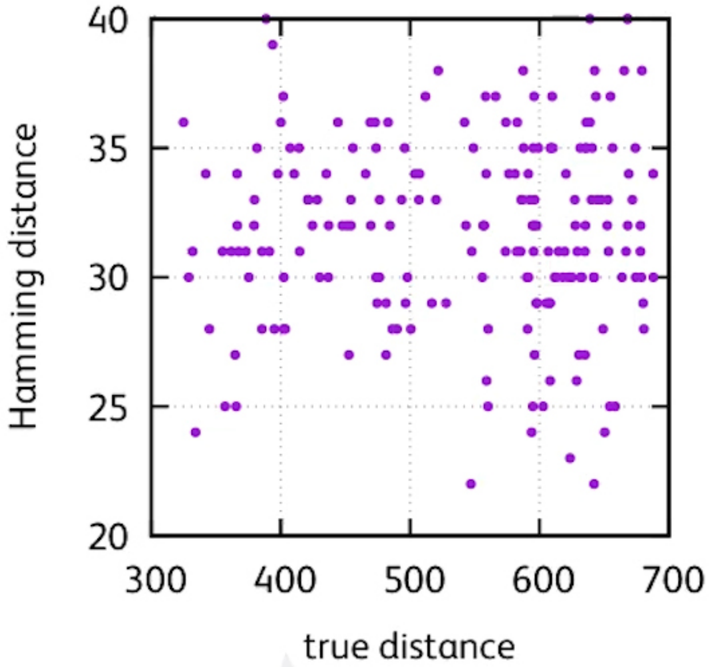


Polysemous PQ codes

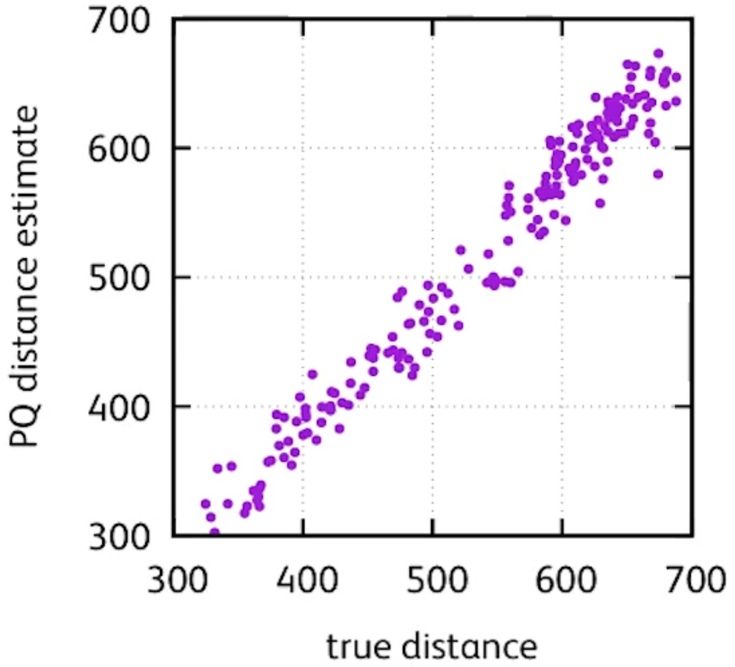


Polysemous codes, Douze, Jégou, Perronnin

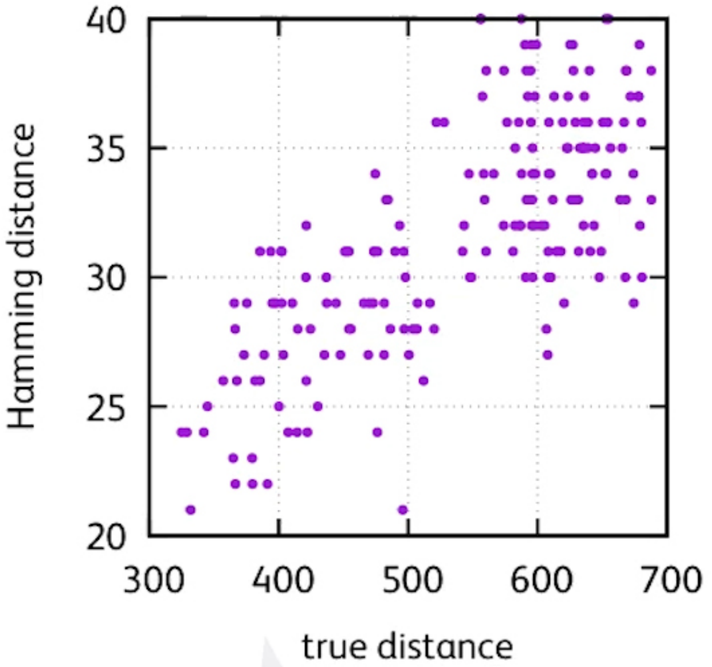
Optimize the order of PQ centroids: Index assignment



Before optimization



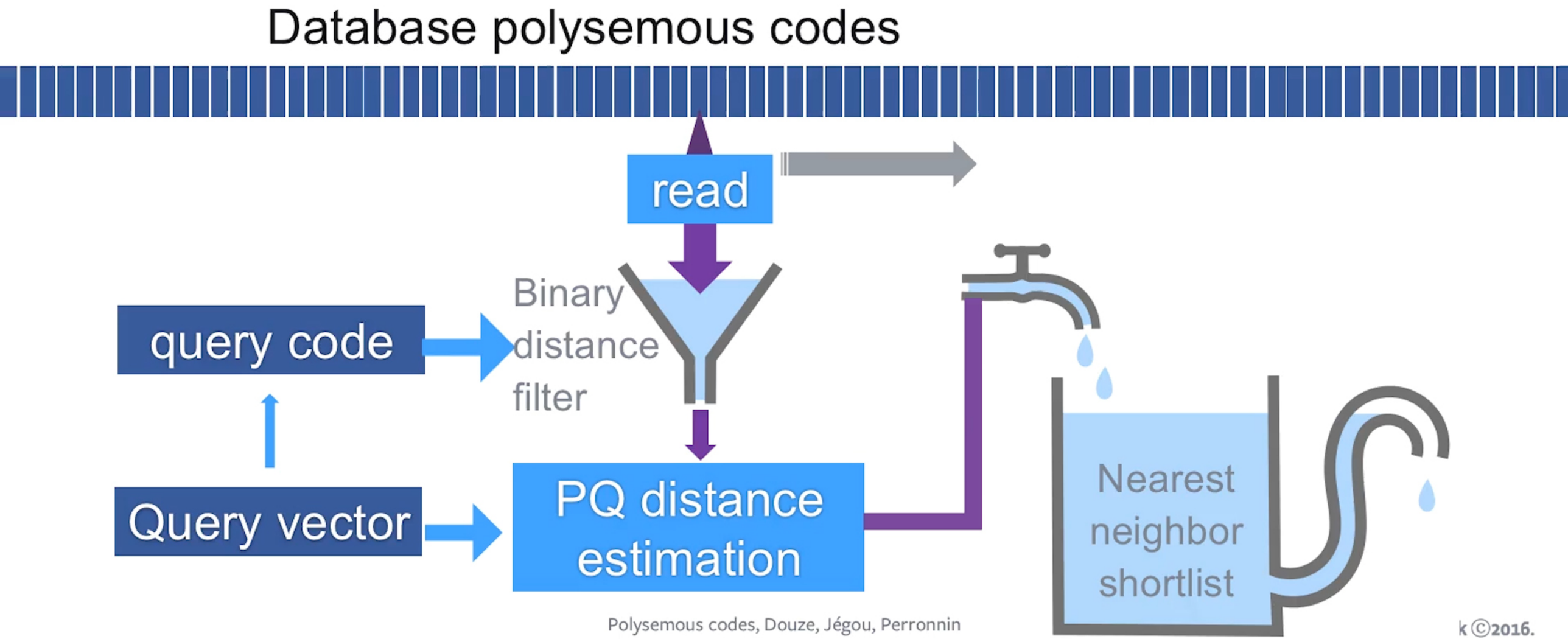
Optimization target



After optimization

Polysemous codes, Douze, Jégou, Perronnin

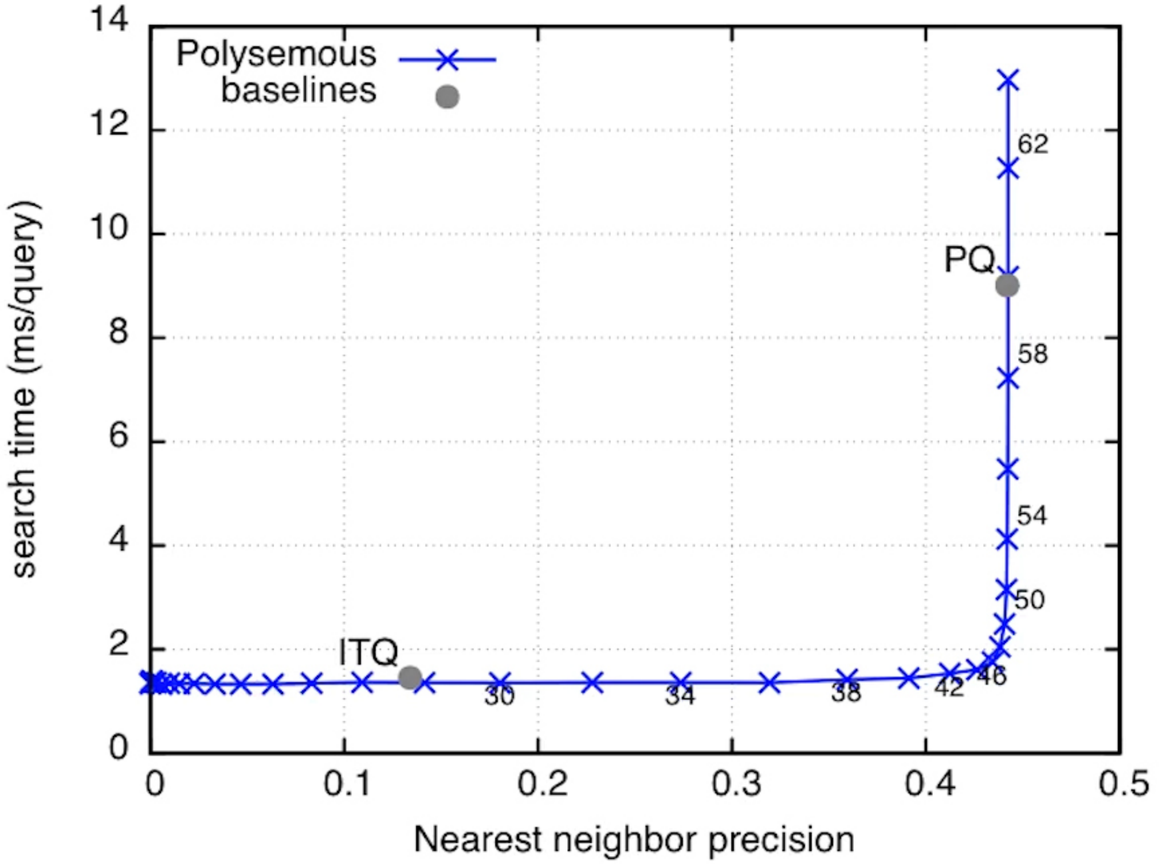
Binary codes for pre-filtering



Polysemous codes, Douze, Jégou, Perronnin

Results on datasets of 1M-1G vectors

Memory usage *is the same*: focus on speed-accuracy tradeoff



Combines with non-exhaustive search: BIGANN (1G vectors)

2-2.5x faster, 0.5 ms per query for 1 thread