Milvus Paper sharing

Product quantization for nearest neighbor search

godchen fishpenguin

ZILLIZ

Content

1.Background

2.Product quantizers

3.ADC vs SDC

4.IVFADC

5.Experiments

background

K Dimension-tree

KD-tree [5] have been proposed to reduce the search time. However, for high dimensions it turns out [6] that such approaches are not more efficient than the brute-force exhaustive distance calculation, whose complexity is O(nD).

LSH(locality sensitive hashing)

In the case of E2LSH, the memory usage may even be higher than that of the original vectors. Moreover, E2LSH need to perform a final re-ranking step based on exact L2 distances, which requires the indexed vectors to be stored in main memory if access speed is important. It performs well in small and middle size datasets(up to tens of millions)

Product quantizers

i. Vector quantization

Purpose: reduce the cardinality of the representation space, in particular when the input data is real-valued

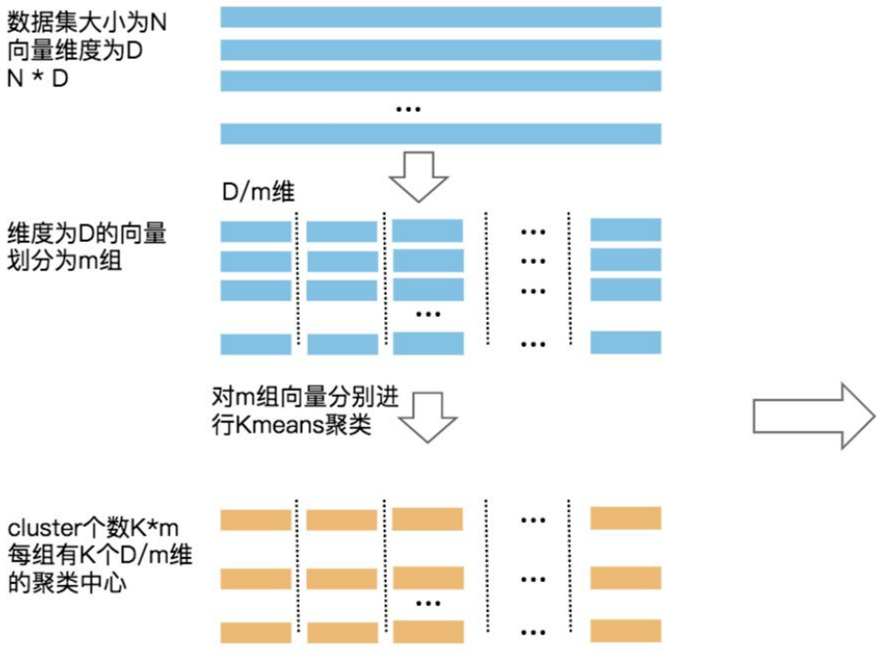
K-Means

Lloyd optimality conditions:
(1) a vector x must be quantized to its nearest codebook centroid;
(2) the reconstruction value must be the expectation of the vectors lying in the Voronoi cell.

ii. Product quantizers

Trade off between search quality and memory requirements.

Product quantizers



每个子向量，用其所属的cluster_id表示
压缩前数据大小为N*D*4*8bits (float)
压缩后 压缩后 N * $\log_2 K$ * m

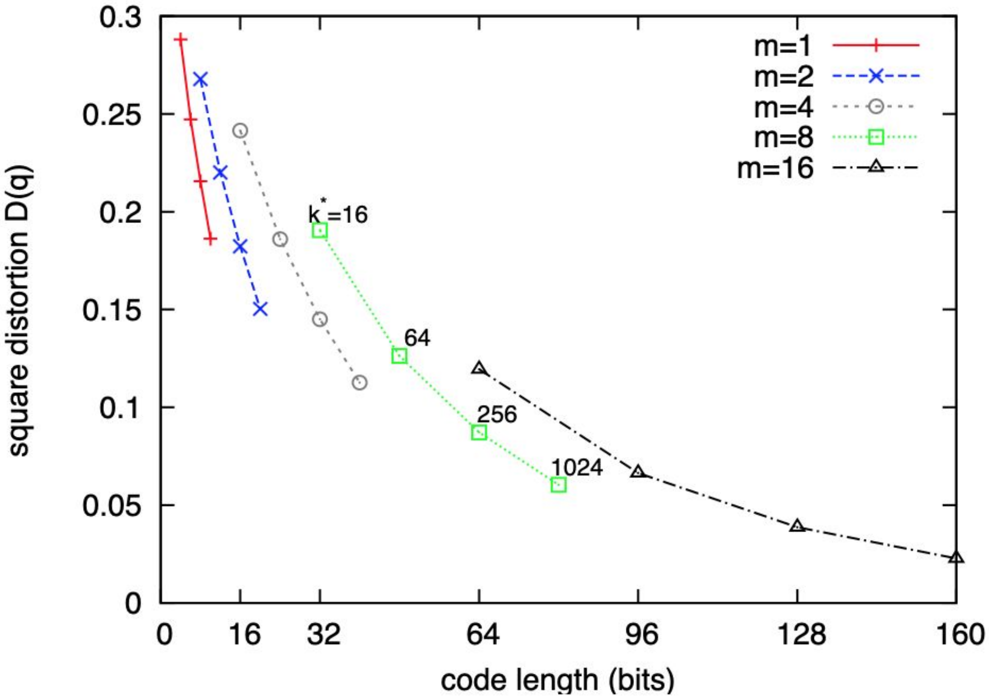
1	2	2	...	3
208	99	91	...	128
211	108	79	...	139
...	...	...	...	...
109	28	65	...	89

k❌D

(k*)^m = k

m❌k*❌D
* =k*❌D
=D❌k^1/
m

Product quantizers



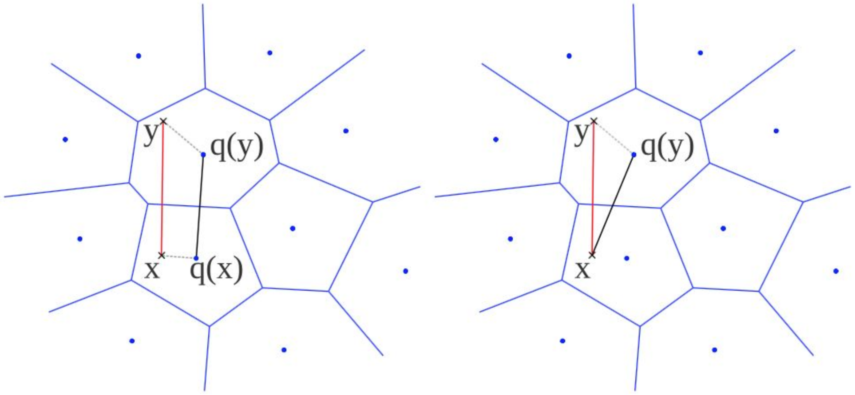
SIFT: quantization error associated with the parameters m and k .

ZILLIZ

ADC vs SDC

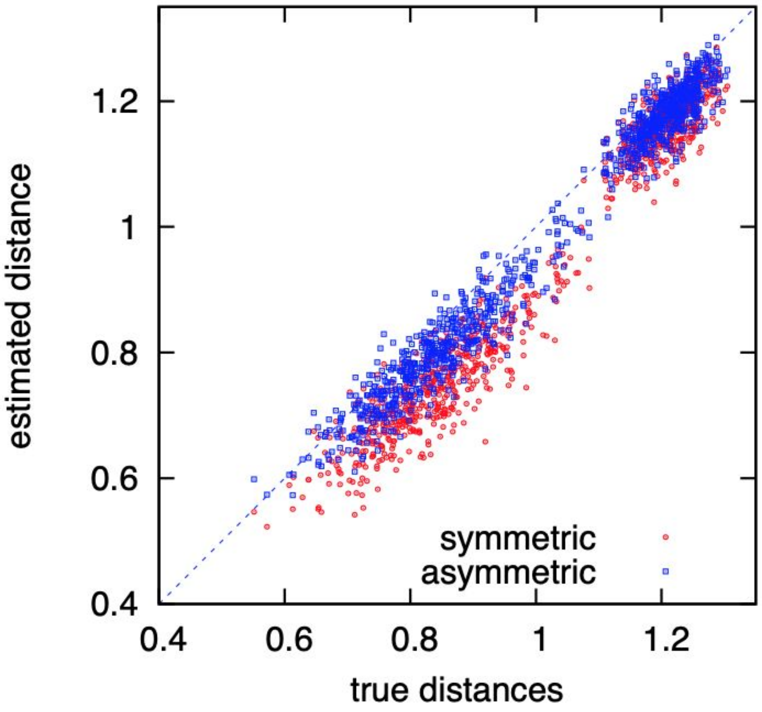
SDC: $$\hat{d}(x, y) = d\big(\boxed{q(x)}, q(y)\big) = \sqrt{\sum_j d\big(q_j(x), q_j(y)\big)^2}, \quad (12)$$

ADC: $$\tilde{d}(x, y) = d\big(\boxed{x}, q(y)\big) = \sqrt{\sum_j d\big(u_j(x), q_j(u_j(y))\big)^2}, \quad (13)$$



symmetric case　　asymmetric case

ADC vs SDC



	SDC	ADC
encoding x	$k^* D$	0
compute $d(u_j(x), c_{j,i})$	0	$k^* D$
for $y \in \mathcal{Y}$, compute $\hat{d}(x,y)$ or $\tilde{d}(x,y)$	$n\,m$	$n\,m$
find the k smallest distances	$n + k \log k \log\log n$	

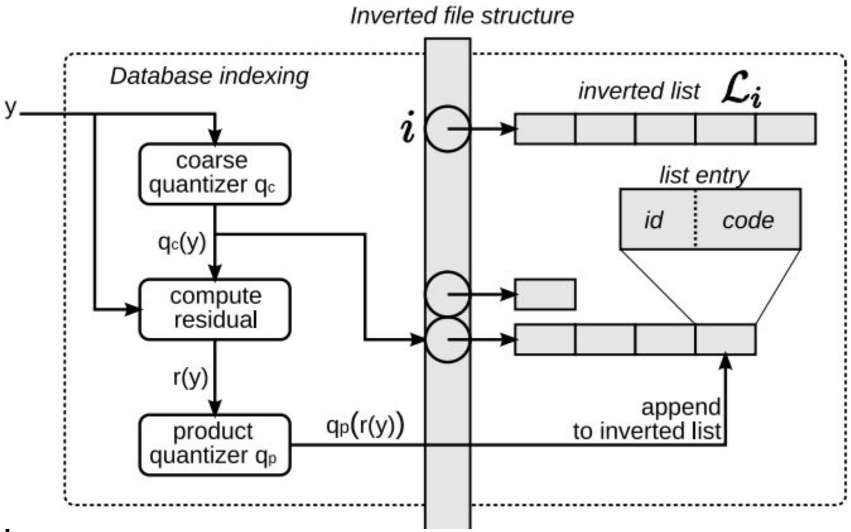
1.When n is large (n > k*D), the most consuming operations are the summations in Equations 12 and 13.

2.The only advantage of SDC over ADC is to limit the memory usage associated with the queries, as the query vector is defined by a code.

Another problem: If n is large, an exhaustive search is prohibitive, as we need to index billions of descriptors and to perform multiple queries

ZILLIZ

IVFADC



Indexing a vector y proceeds as follows:

1. quantize y to qc(y)

2. compute the residual r(y) = y − qc(y)

3. quantize r(y) to qp(r(y)), which, for the product quantizer, amounts to assigning uj (y) to qj (uj (y)), for j = 1 . . . m.

4. add a new entry to the inverted list corresponding to qc(y). It contains the vector (or image) identifier and the binary code (the product quantizer's indexes).

IVFADC

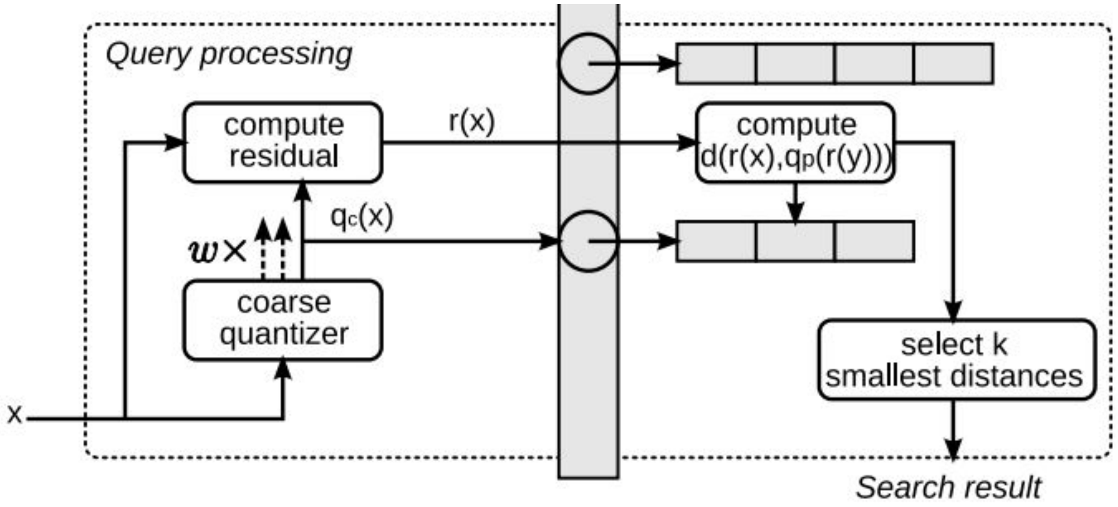
Searching:

1. coarse quantizer

2. compute the residual r(x)

3. compute d(x, y)

4. select the k nearest neighbors



Experiments

vector dataset:	SIFT	GIST
descriptor dimensionality d	128	960
learning set size	100,000	100,000
database set size	1,000,000	1,000,991
queries set size	10,000	500

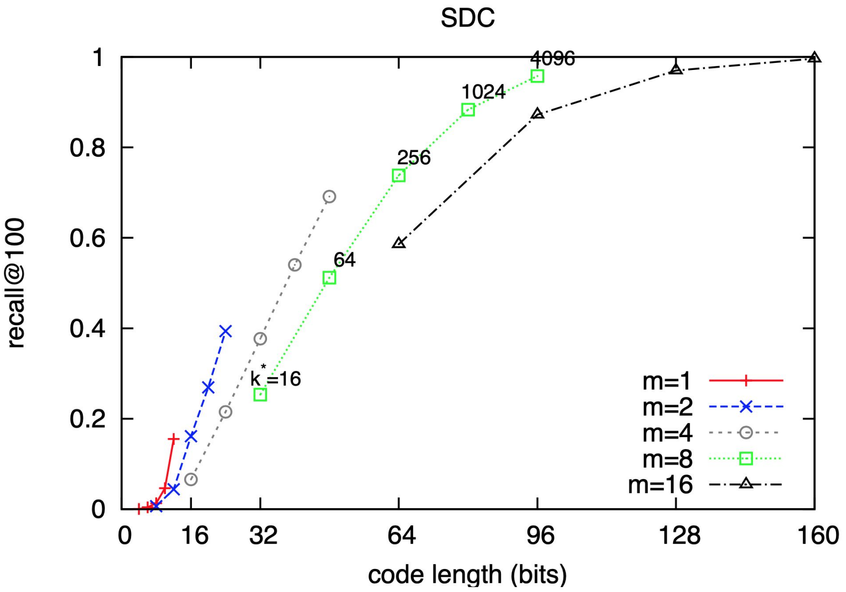
TABLE III

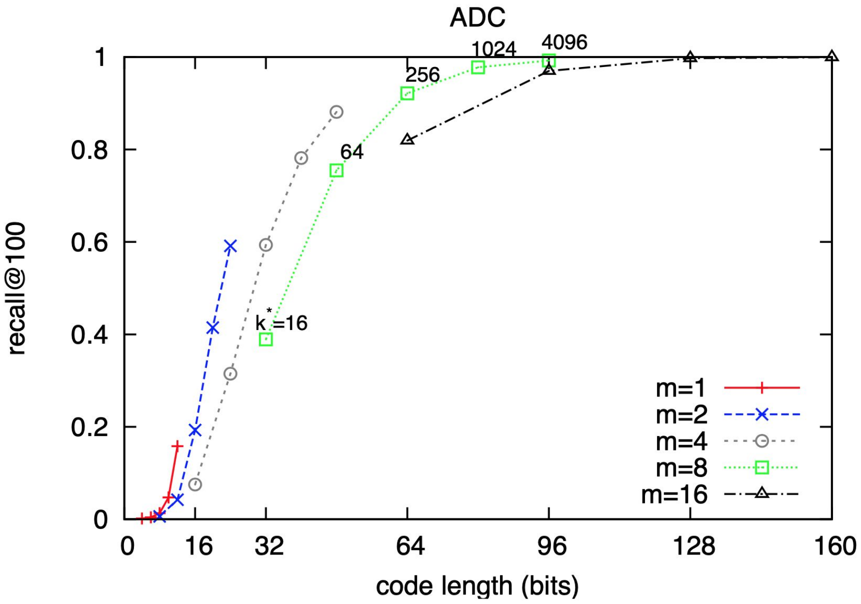
SUMMARY OF THE SIFT AND GIST DATASETS.

ZILLIZ

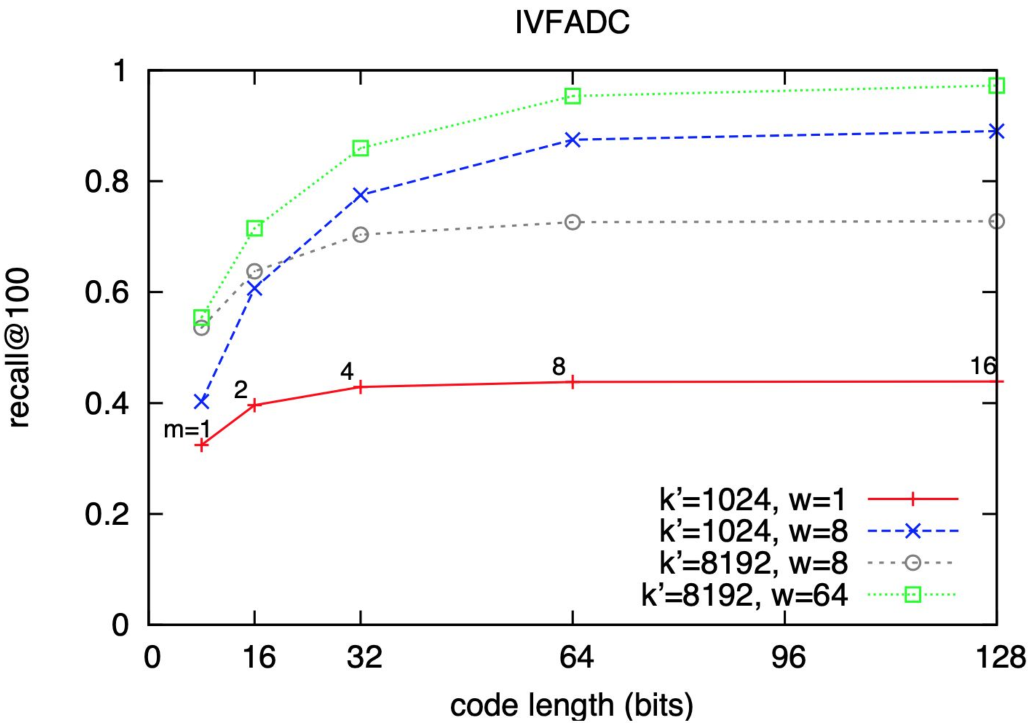
Experiments

SDC vs ADC



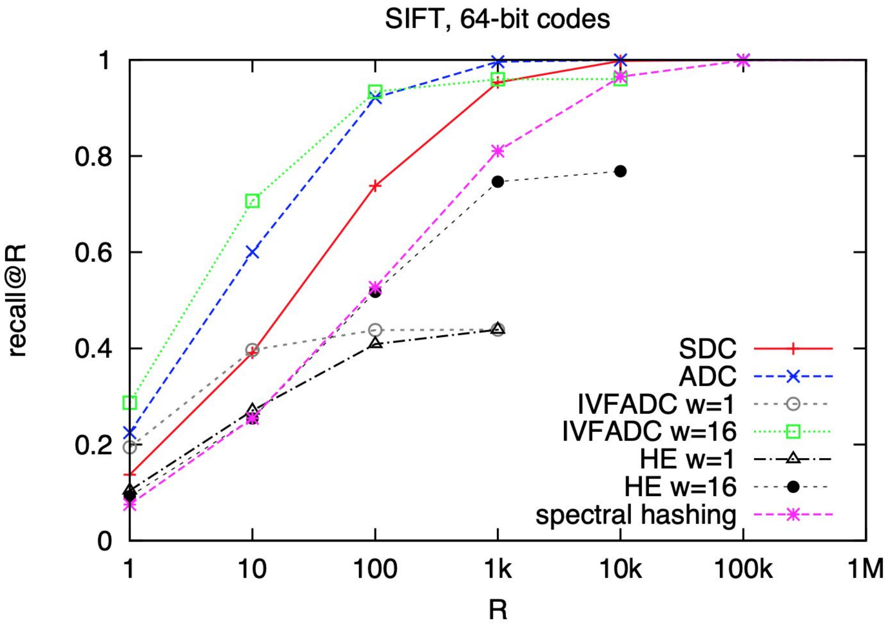


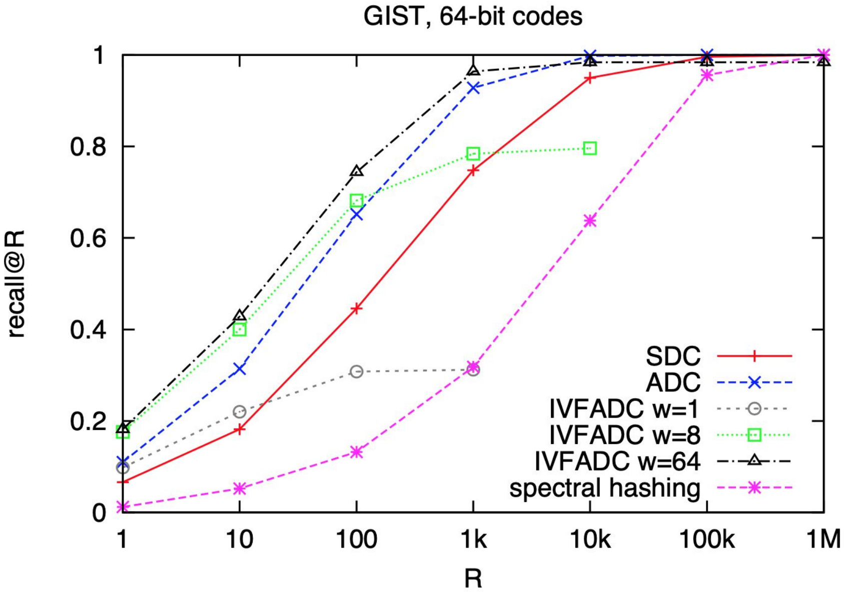
Experiments



Experiments

Compare with state of art method:
spectral hashing and **hamming embedding**

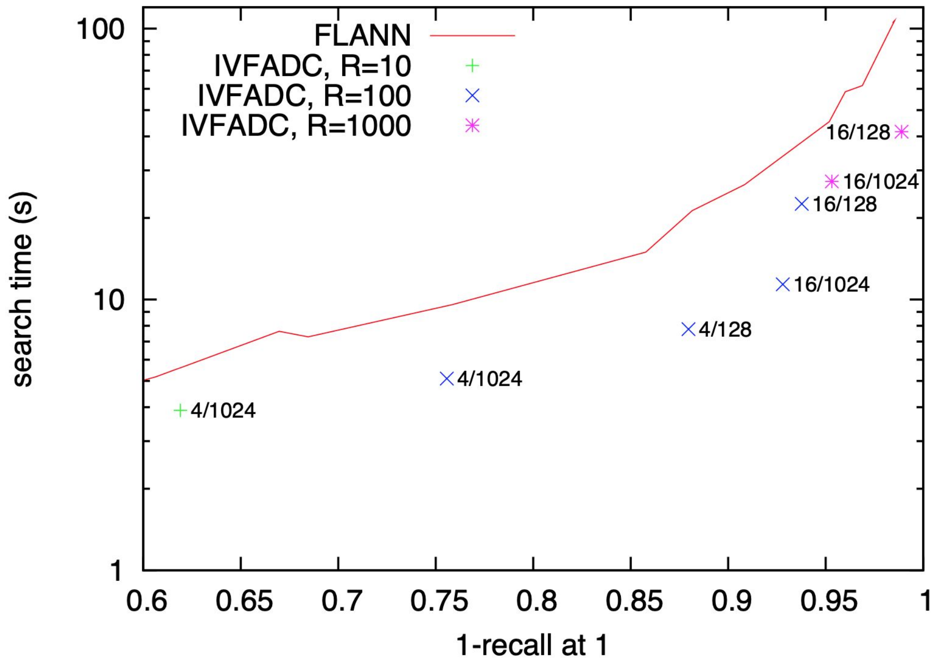




ZILLIZ

Experiments

Compare with state of art method:
FLANN



ZILLIZ

Experiments

Complexity and speed

method	parameters	search time (ms)	average number of code comparisons	recall@100
SDC		16.8	1 000 991	0.446
ADC		17.2	1 000 991	0.652
IVFADC	$k'=1\,024$, w=1	1.5	1 947	0.308
	$k'=1\,024$, w=8	8.8	27 818	0.682
	$k'=1\,024$, w=64	65.9	101 158	0.744
	$k'=8\,192$, w=1	3.8	361	0.240
	$k'=8\,192$, w=8	10.2	2 709	0.516
	$k'=8\,192$, w=64	65.3	19 101	0.610
SH		22.7	1 000 991	0.132

Experiments

large scale

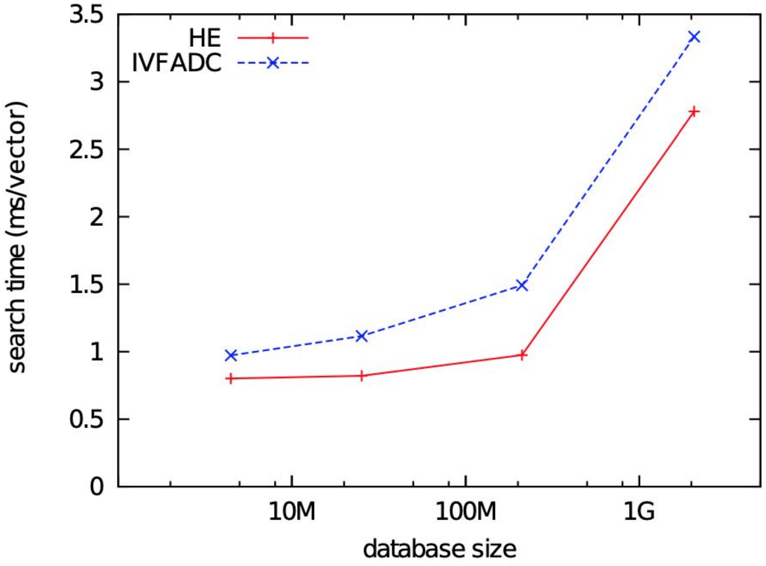


Fig. 11. Search times for SIFT descriptors in datasets of increasing sizes, with two search methods. Both use the same 20 000-word codebook, $w = 1$, and 64-bit signatures.

Experiments

Image search

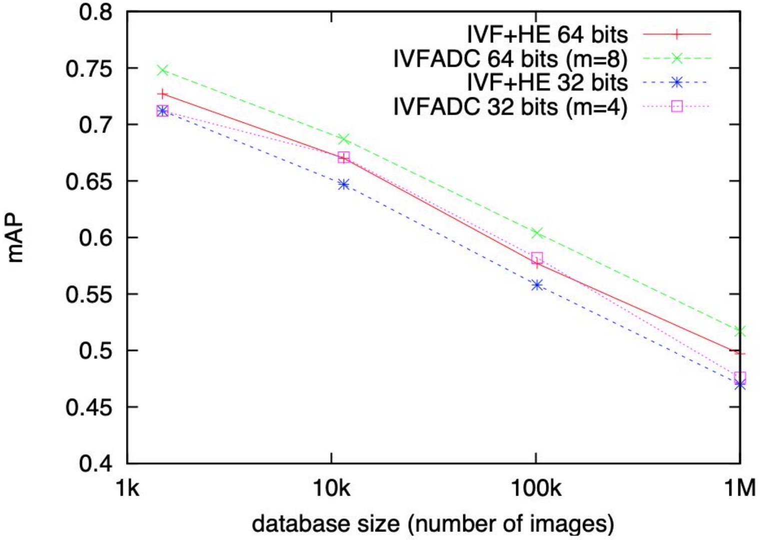


Fig. 12. Comparison of IVFADC and the Hamming Embedding method of [20]. mAP for the Holidays dataset as function of the number of distractor images (up to 1 million).

Thanks