Hamming Embedding and Weak Geometry Consistency for Large Scale image Search

Hervé Jégou, Matthijs Douze, Cordelia Schmid
INRIA Grenoble, LEAR, LJK

ONLINE DEMO:
http://lear.inrialpes.fr/

Contributions

Refinements of the Bag-of-features (BOF) representation:

✔ embedding of local descriptors in the Hamming space
⇒ compact representation, high comparison efficiency

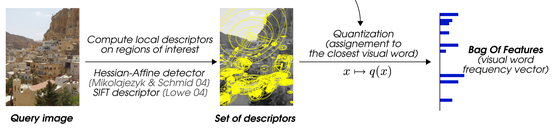
✔ geometrical information introduced into the inverted file structure, i.e. within the BOF dissimilarity measure
⇒ geometry is efficiently exploited for *all* images

Results:
✔ state-of-the-art results on two datasets (INRIA Holidays and Oxford building datasets), *and* very fast retrieval

✔ This is the core of our video matching system, which won the TRECVID'2008 copyright detection task

Starting point: Video-Google

(Zivic & Zisserman 2003)



ANN interpretation of Bag-of-features

✔ Descriptor matching function:

$$f_q(x,y) = \begin{cases} 1 & \text{if } q(x) = q(y) \\ 0 & \text{otherwise} \end{cases}$$

✔ Score for comparison of two images represented by their sets of descriptors $\{x_i\}_{i=1..n}$ and $\{y_j\}_{j=1..m}$ derived as

$$s(\mathrm{im}_1, \mathrm{im}_2) = \frac{1}{n}\frac{1}{m}\sum_i \sum_j f_q(x_i, y_j)$$

✔ This is the (standard) cosine similarity between the BOF representations

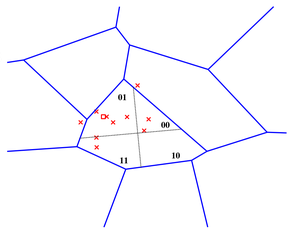
✔ In the following: we use the matching interpretation to improve the image comparison

Hamming Embedding

Motivation: refine the representation of SIFT descriptors to improve the matching function

✔ add a binary signature per descriptor: $b(x) \in \{0,1\}^{d_b}$

✔ Learning: define d_b orthogonal hyperplanes per quantization cell



✔ Signature:
$b_i(x)$ = 1 iff x "on the left" of hyperplane i

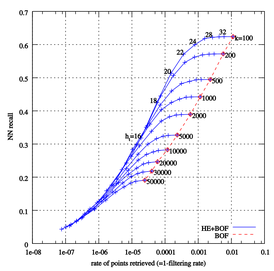
✔ new representation: (q(x),b(x))

✔ corresponding matching function:

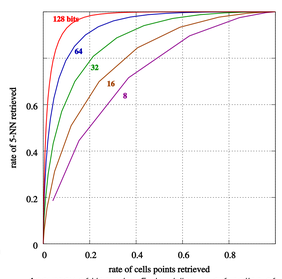
$$f_{\mathrm{HE}}(x,y) = \begin{cases} 1 & \text{if } q(x) = q(y) \text{ and } h\left(b(x), b(y)\right) \le h_t \\ 0 & \text{otherwise} \end{cases}$$

where:
- h_t is a threshold
- h(...) is the Hamming distance between signatures

High dimensional point search results

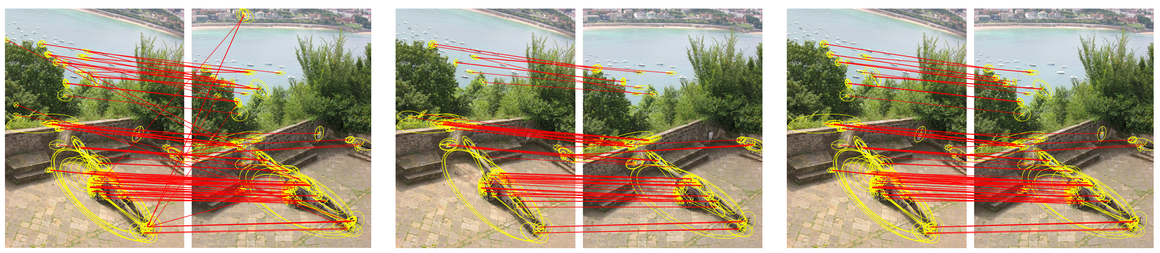


Evaluation of Approximate nearest neighbor search using Hamming Embedding. We plot the probability that the true nearest neighbor is found (recall) against the rate of points of the dataset that are returned.
Red curve: matching based on quantized indexes only (i.e., standard BOF).
Blue curves: performance of Hamming Embedding for different vocabulary sizes and different thresholds (d_b=64 bit)



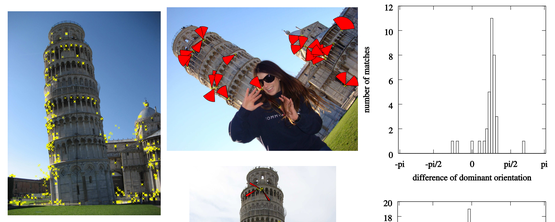
Accuracy of Hamming Embedding as a function of the length of the binary signatures. Each curve is generated by sweeping h_t from 0 to d_b. The higher d_b, the better the approximation of the Euclidean neighborhood by the Hamming neighborhood. We choose 64 bits, a good compromise between retrieving the Euclidean nearest neighbors and the memory usage (8 bytes thresholds (d_b=64 bit)

k=20000 | k=200000 | k=20000 with HE

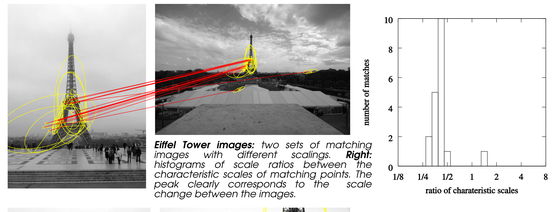


Weak Geometry Consistency

Motivation: use (partial) geometric information for all the images = not only for a short-list of a few hundred images!



Pise Tower images: *query image and two shots taken under different angles. The red slices show the difference between the dominant orientations of matching points. Matches are obtained using our Hamming Embedding method.*
Histograms on the right: *histograms of difference between the dominant orientations. The peaks corresponds to the global angle variation. Incorrect matches can therefore be filtered using angle consistency.*



Eiffel Tower images: *two sets of matching images with different scalings.* ***Right:*** *histograms of scale ratios between the characteristic scales of matching points. The peak clearly corresponds to the scale change between the images.*

Key ideas:
✔ do not try to estimate a full affine transform
✔ use the region scale and dominant angle information
✔ integrate this (quantized) information into the inverted file
✔ produce a score (derived from BOF inner product) which only integrates the matches that are consistent with the main orientation and angle (maximum of histograms)

Advantages:
✔ query scores on output from the inverted file include the angle and scale information (against no geometry at all)
✔ the number of inverted file elements accessed is almost the same as for a standard inverted file
✔ no need to map the points from an image to another: matches are intrinsically filtered when not consistent

⇒ Very efficient for 1 million images!

Efficiency and memory usage

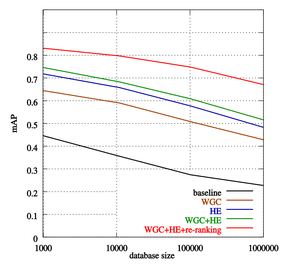
		time per image k=20000	time per image k=200000	size per descriptor
compute descriptors		0.88 s		
quantization		0.36 s	0.60 s	
search	baseline	2.74 s	0.62 s	4 bytes
search	WGC	10.19 s	2.11 s	4 bytes
search	HE	1.16 s	0.20 s	12 bytes
search	HE+WGC	1.82 s	0.65 s	12 bytes

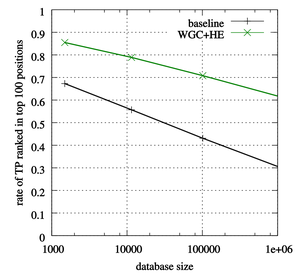
Experiments

		Holidays		Oxford5k	
k=		20000	200000	20000	200000
mAP	baseline	0.4463	0.5488	0.3854	0.3950
mAP	HE h_t=24	0.6947	0.7115	0.4906	0.4585
mAP	WGC	0.6446	0.6859	0.4375	0.4602
mAP	HE+WGC h_t=24	0.7507	0.7439	0.5397	0.5252

Large scale search

We use the Holidays base + up to 1 million distractors





The Holidays database

http://lear.inrialpes.fr/~jegou/data.php

✔ personal holiday photos.
✔ 1491 images, 500 distinct scenes, 4.4 million descriptors
✔ scene types: natural/man-made, water/fire effects, close-ups, landscapes, persons,...
✔ transformations: rotations, viewpoint and illumination changes, blurring,...
✔ from Flickr: 60,000-image independent learning set, 1-million image distractor set