A Scalable Pattern Spotting System for Historical Documents

Sovann EN

Under the supervision of

Prof. Laurent Heutte, Prof. Frédéric Jurie
Dr. Stéphane Nicolas and Dr. Caroline Petitjean

LITIS, University of Rouen
GREYC, University of Caen Basse-Normandie

Introduction: Pattern Spotting

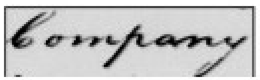
- A lot of interest in historical document image management systems

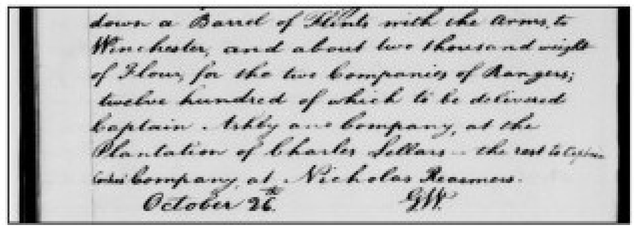
- Indexing plateform:
  - Word spotting

What about graphical object spotting ? (pattern spotting)

Introduction: Pattern Spotting

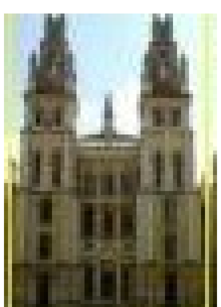
- Word spotting




down a Barrel of Flints with the Arms, to
Winchester, and about two thousand weight
of Flour, for the two Companies of Rangers;
twelve hundred of which to be delivered
Captain Ashby and Company, at the
Plantation of Charles Sellars – the rest to Captain
Cocke's Company, at Nicholas Reasmers.
October 26. GW

- Pattern spotting

- (sub-)Image retrieval

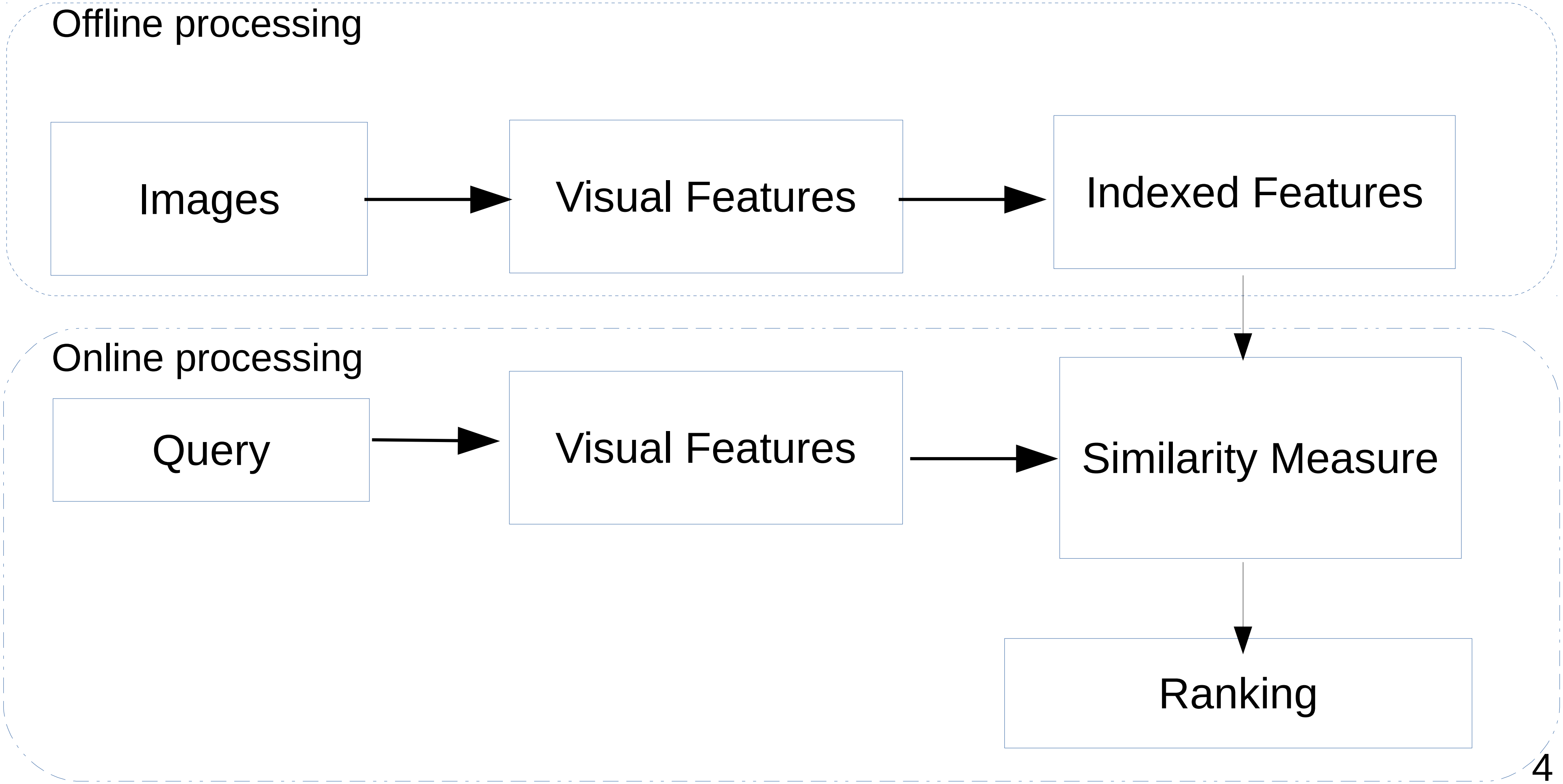


Key Factors:

- Prior knowledge
- Query size
- Image quality

Introduction: Pattern Spotting

- Global Processing Chain



Content

1. Introduction
2. Pattern Spotting System
   - processing pipeline
   - corpus preparation
   - non-exhaustive search
3. Experimentation
4. Future works and perspective

Processing Pipeline

- Constraints:
  - The targeted patterns can appear everywhere
  - The query size is relatively small (20*20 pixels to 500*500 pixels) compared to the Image

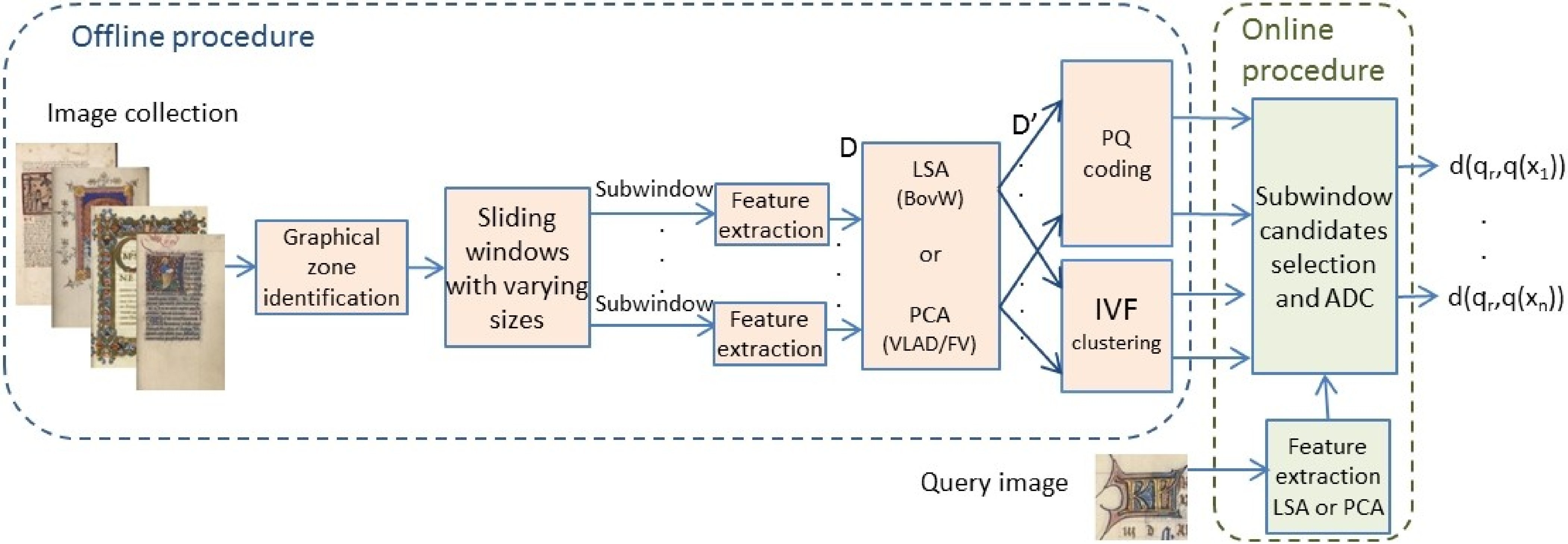
→ sliding windows is needed, but not an exhaustive way

Processing Pipeline

Key ingredients:

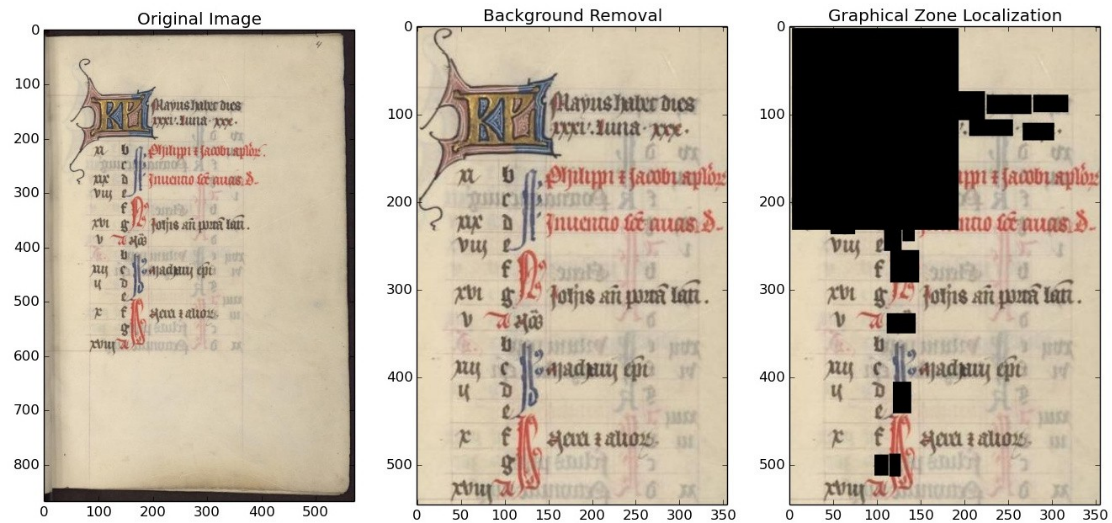
- background removal: gradient information
- graphical zone identification:
  - Scribo module (https://olena.lrde.epita.fr/demos/historical_document_layout_analysis.php)
- restricted sliding windows:
  - quantized sizes, {20, 40, 80, 160, 320} pixels
- Non-exhaustive search:
  - Inverted file structure + distance approximation
- feature compression
  - dimensionality reduction and product quantization

Processing Pipeline



Corpus Preparation: preprocessing

- Preprocessing:
  1. binarisation
  2. find the biggest region for which all the pixels to the border are white
  3. Extract graphical zones



Corpus Preparation: feature extraction

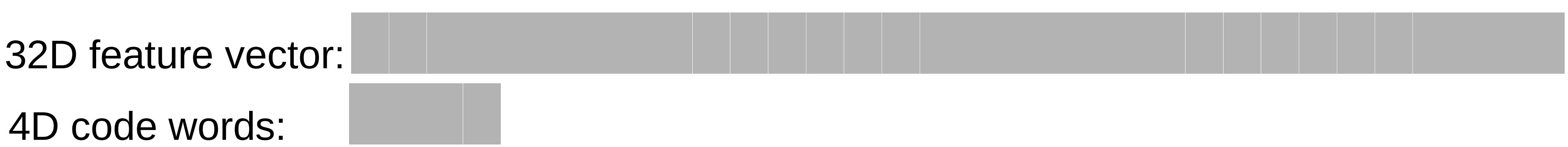
- Multi-scale sliding windows is used (20, 40, 80, 160, 320 pixels)
- Feature extraction:
  - dense sampling
  - bag of visual word + tf.idf
  - fisher vector [Sánchez et al., 2013]
  - vector locally aggregated descriptor (VLAD) [Jégou et al., 2012]

Image classification with the fisher vector: Theory and practice, IJCV'13, Sánchez et al.

Aggregating local descriptors into a compact image code, CVPR'12, Jégou et al.

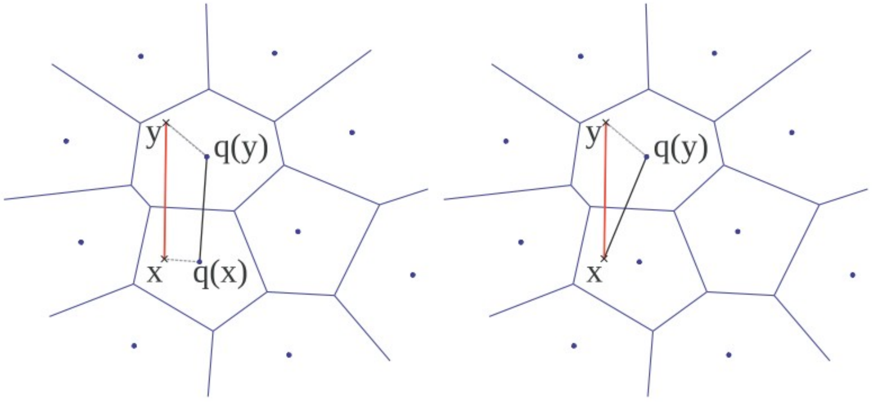
Corpus Preparation: feature compression

- Dimensionality reduction
  - latent semantic indexing (LSI)
  - principle component analysis (PCA)
- Compression
  - Product quantization (PQ)
    - 1:32 compression ratio, *m=8*



Non-exhaustive search

- Avoid exhaustive search
  - Inverted File Structure (IVF)
- Avoid exhaustive distance computation
  - Assymetric distance computation (ADC)



symmetric case   asymmetric case

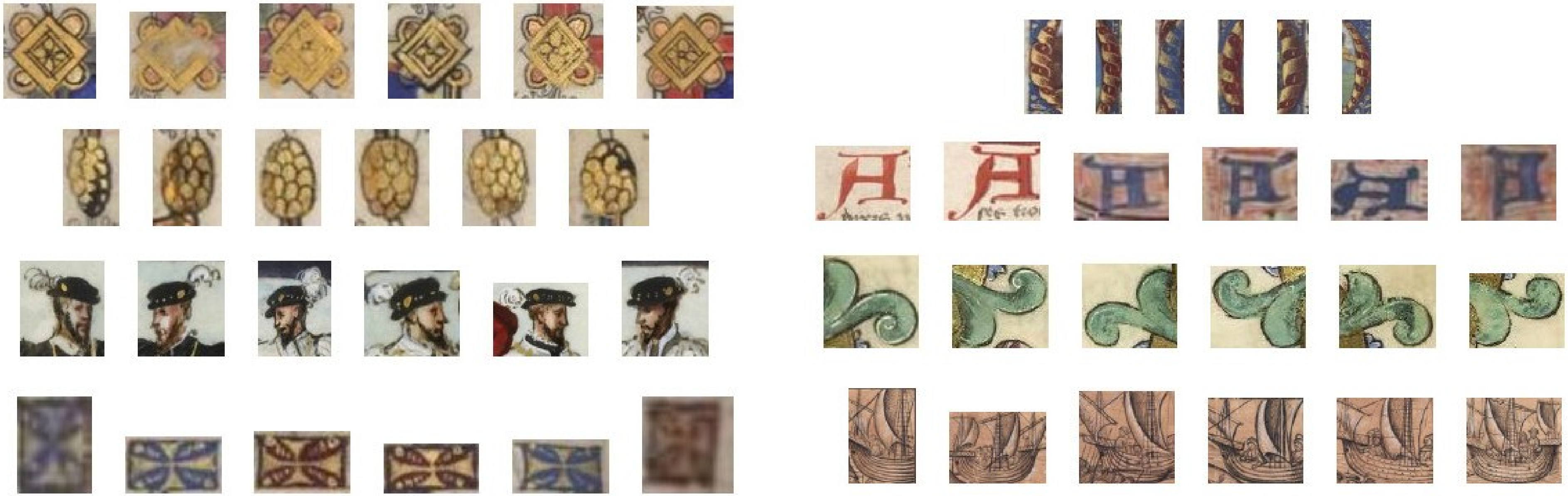
- Distance measure: Cosine, Euclidean, Dot product

Experimentation: dataset

- DocExplore (http://www.docexplore.eu/) Dataset:
  - 1597 medieval manuscript images
    - maximum size: 1024 * 1024 pixels
  - 1097 queries with groundtruth annotations
    - varying size: 20*20 pixels to 500*500 pixels
  - performance is measured as mAP

Experimentation: dataset

- Illustration of variability of the queries



Experimentation: results

- Background removal Vs Graphical Zone Identification

Codebook size	500	1k	5k	10k	#subw.
Background removal	0.189	0.190	0.186	0.171	14.5M
Scribo module	0.195	0.204	0.212	0.212	2.1M

- Feature used: BoVW

Experimentation: results

- A unified benchmarking: image representation and distance measures

K	BoVW			K	VLAD			FV		
	Dot	Cos	Euc		Dot	Cos	Euc	Dot	Cos	Euc
500	**0.195**	**0.195**	**0.195**							
1k	0.203	**0.204**	**0.204**	4	**0.251**	0.250	0.250	**0.216**	0.215	**0.216**
5k	**0.213**	0.212	**0.213**	16	0.319	**0.320**	0.319	0.261	0.276	**0.277**
10k	**0.212**	**0.212**	**0.212**	64	**0.295**	**0.295**	**0.295**	**0.348**	0.310	0.315

- Distances used: cosine, dot product and euclidean
- K stands for codebook size

Experimentation: results

- Scalable search: BoVW

codebook size	D	D'=512		D'=256		D'=128	
		Exh	N-Exh	Exh	N-Exh	Exh	N-Exh
500	0.195	-	-	0.198	0.156	0.193	0.168
1k	0.203	**0.195**	0.138	0.191	0.154	0.189	0.162
5k	0.213	0.181	0.100	0.177	0.145	0.171	0.149
10k	0.212	0.172	0.095	0.170	0.137	0.165	0.138

Exh and N-Exh stands for exhaustive search and non-exhaustive search (PQ+ADC+IVF) respectively. D and D' represents feature dimension and reduced dimension.

Experimentation: results

- Scalable search: VLAD

codebook size	D	D'=512		D'=256		D'=128	
		Exh	N-Exh	Exh	N-Exh	Exh	N-Exh
4	0.251	-		0.269	0.224	0.261	0.217
16	0.319	**0.359**	0.282	0.353	0.306	0.340	0.286
64	0.295	0.357	0.304	0.346	0.306	0.332	0.288

Exh and N-Exh stands for exhaustive search and non-exhaustive search (PQ+ADC+IVF) respectively. D and D' represents feature dimension and reduced dimension.

Experimentation: results

- Scalable search: Fisher Vector

codebook size	D	D'=512		D'=256		D'=128	
		Exh	N-Exh	Exh	N-Exh	Exh	N-Exh
4	0.216	-		0.243	0.186	0.240	0.190
16	0.261	0.287	0.242	0.283	0.242	0.275	0.229
64	0.348	**0.374**	0.342	0.364	0.330	0.350	0.302

Exh and N-Exh stands for exhaustive search and non-exhaustive search (PQ+ADC+IVF) respectively. D and D' represents feature dimension and reduced dimension.

Conclusion & future works

- A memory and computation efficient pattern spotting
  - 14.5M sub-windows Vs 2.1M sub-windows
  - Less than 1 second for a single query
- A unified benchmarking:
  - Features: BoVW, VLAD and Fisher Vector
  - Distances: Cosine, Dot product and Euclidean
- Future work:
  - A better localization or post processing (spatial reenforcement) framework
  - Color feature in combination with the existing pipeline

Thanks for your attention !