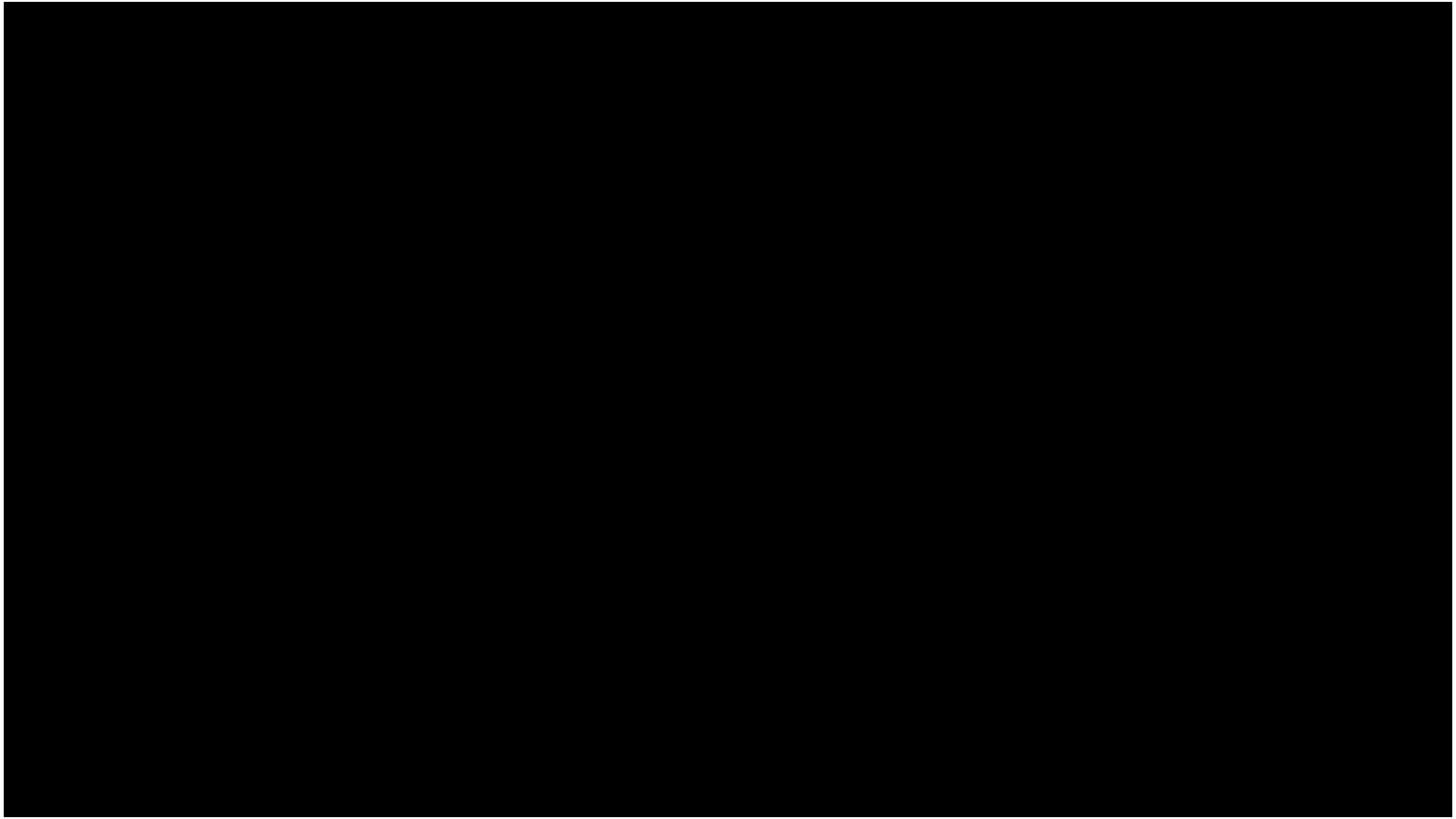
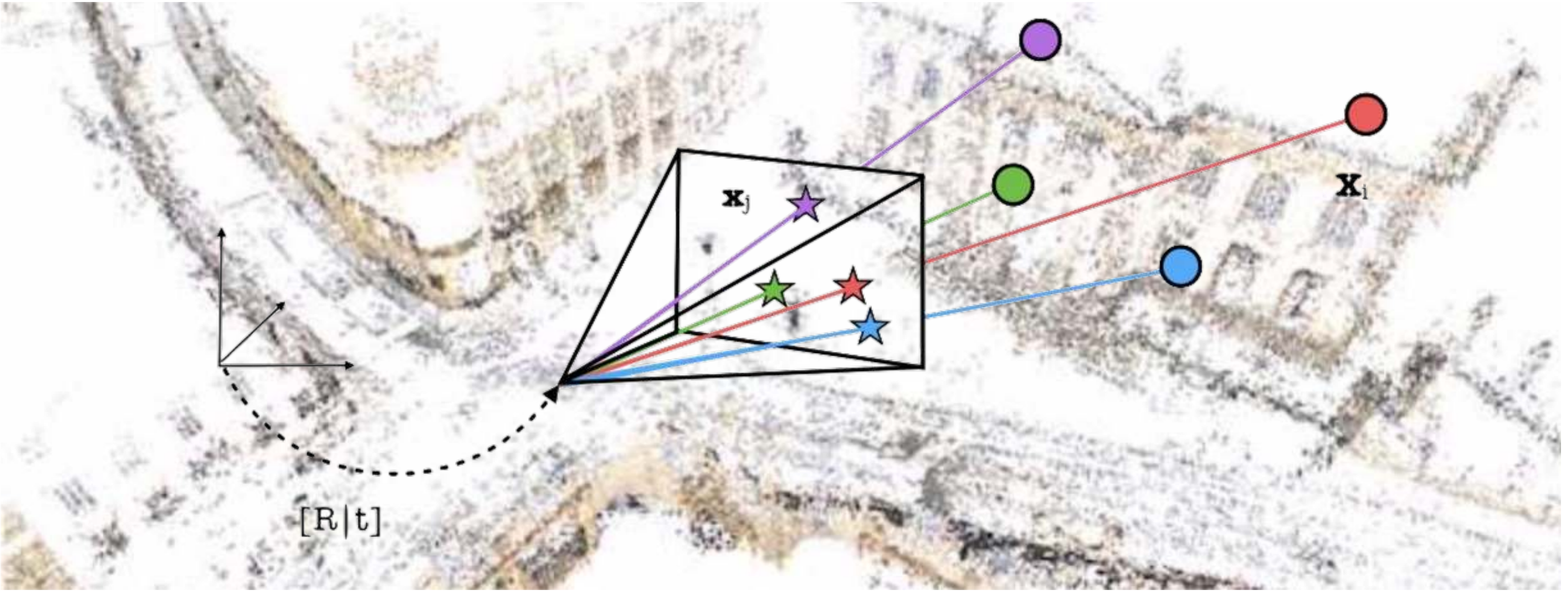
Deep Learning for Augmented Reality

Vincent Lepetit

École des Ponts
ParisTech

Image Retrieval





École des Ponts
ParisTech

Hierarchical Localization

Query Image

Nearest Neighbour Retrieval

Sparse 3D Model

Local 2D-3D Matching

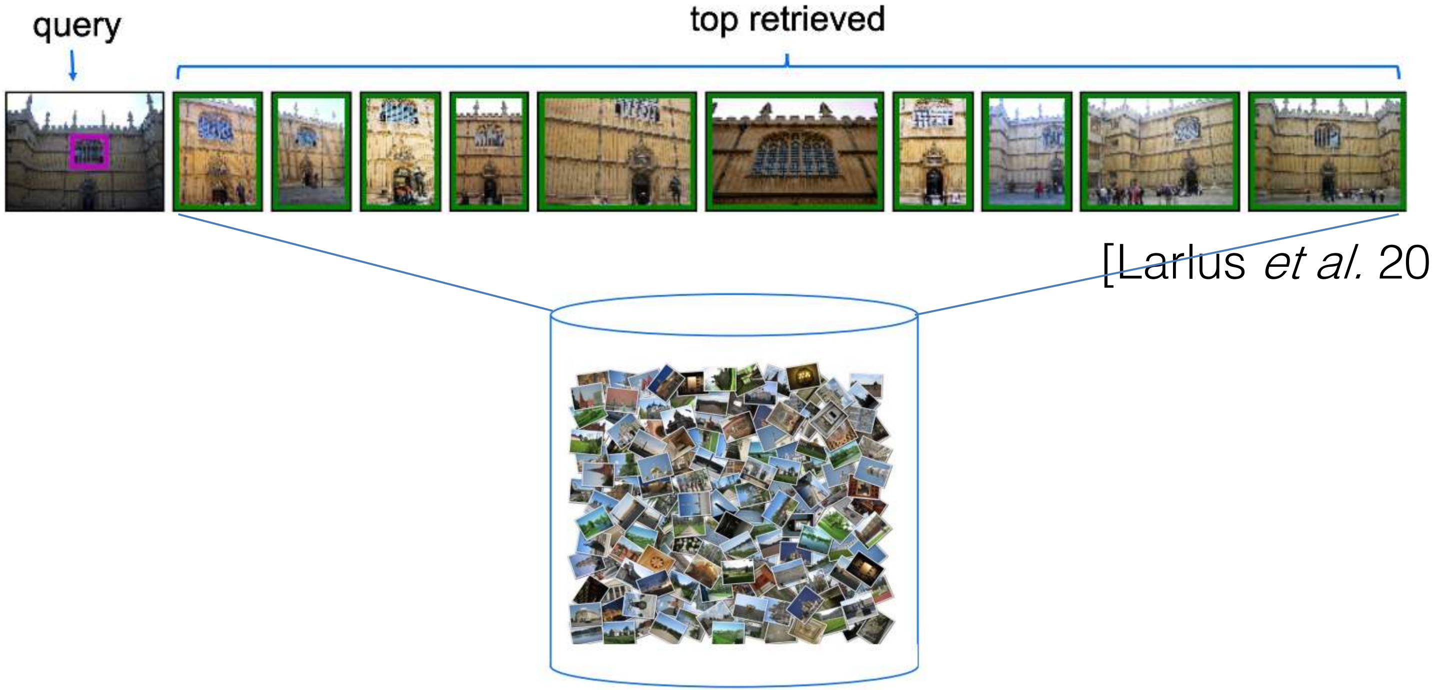
Camera Pose Estimation (P*n*P)

NetVLAD [Arandjelovic *et al.*, CVPR 2016]:

Application to Camera Localization

Localization by Image Retrieval

Matching a query image against a database



[Larlus *et al.* 2017]

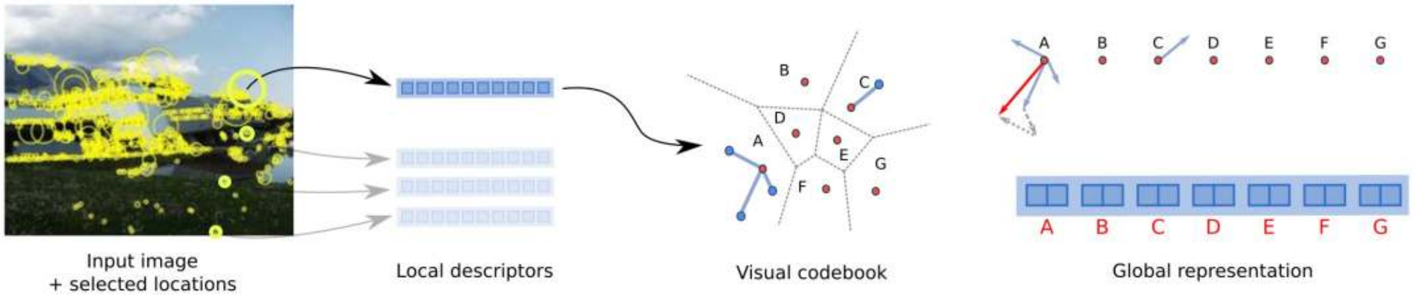
Aggregating Local Representations

A popular approach: VLAD (Vector of Locally Aggregated Descriptors)

- Assign local descriptors to *visual words;*

- Concatenate vectors for individual words by computing *residuals;*

- Store a 2D vector per cluster as part of final descriptor.



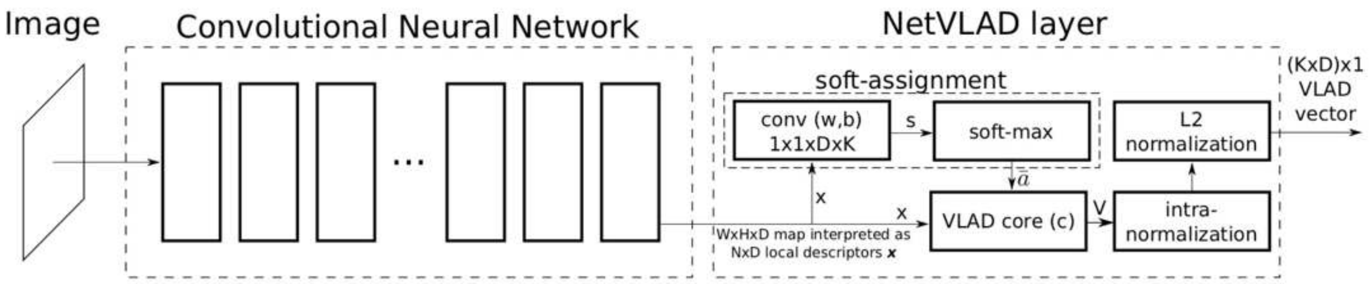
École des Ponts ParisTech

[Jégou et al. CVPR 2010]

Improving VLAD: *Learning* to extract local features and to aggregate them

NetVLAD: Apply VLAD on features learned end-to-end

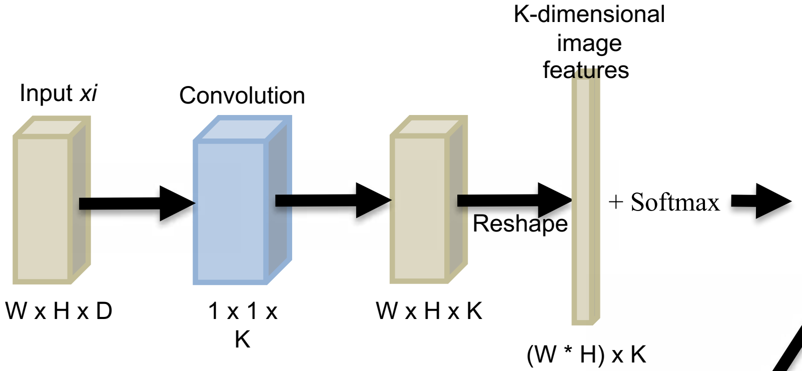
Define a differentiable VLAD layer, append it to a Siamese Network



[Arandjelovic et al. CVPR16]

NetVLAD in practice

$\{\mathbf{w}_k\}$, $\{b_k\}$ and $\{\mathbf{c}_k\}$ are sets of trainable parameters



"Soft" assignment of features to 1 of K cluster centers

$$\bar{a}_k(\mathbf{x}_i) = \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}}$$

Compute the residuals between the features and the centroid of their clusters (as in VLAD)

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}} (x_i(j) - c_k(j))$$

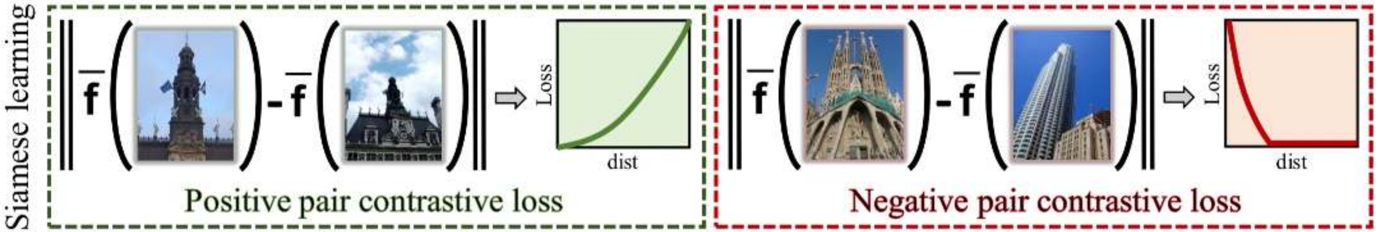
Normalization

$(K \times D) \times 1$ descriptors

Training NetVLAD

The network is trained on pairs of images, either *positive* or *negative;*

- For positive pairs, minimize the distance between the output descriptors.
- For negative pairs, maximize it.



$$\mathcal{L}(i,j) = \begin{cases} \frac{1}{2}||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||^2, & \text{if } Y(i,j) = 1 \\ \frac{1}{2}\left(\max\{0, \tau - ||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||\}\right)^2, & \text{if } Y(i,j) = 0 \end{cases}$$

[Radenovic et al. TPAMI2018]

École des Ponts
ParisTech

NetVLAD: Results

[Arandjelovic et al. CVPR16]

Hierarchical Localization

Query Image

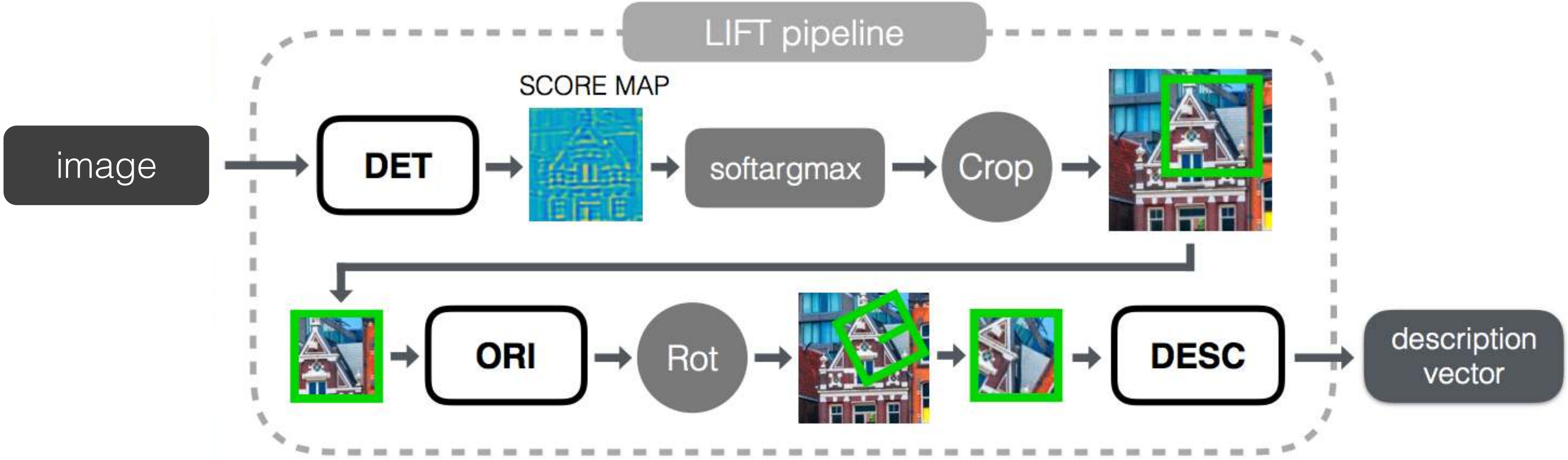
Nearest Neighbour Retrieval

Sparse 3D Model

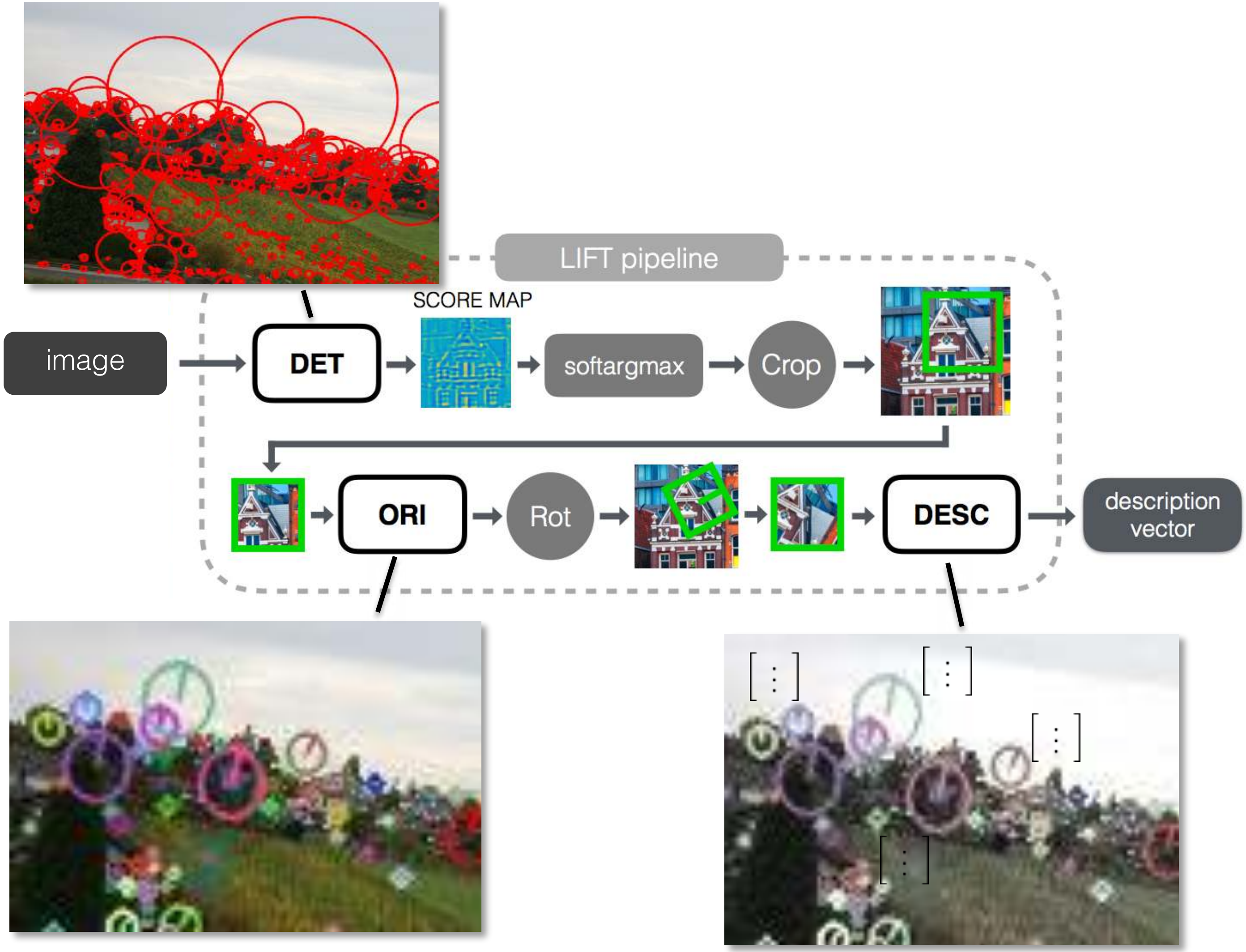
Local 2D-3D Matching

Camera Pose Estimation (P*n*P)

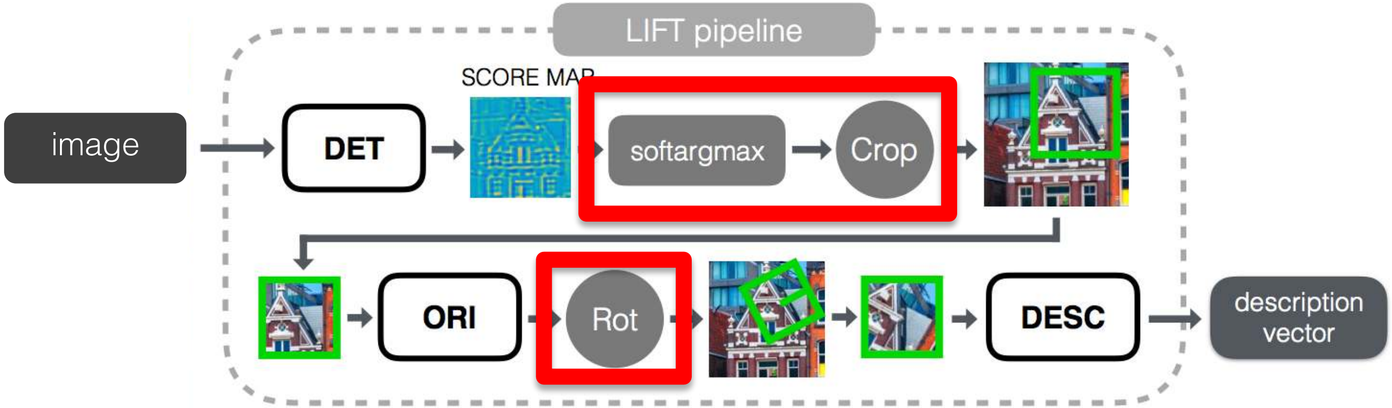
LIFT



LIFT pipeline
SCORE MAP
image
DET
softargmax
Crop
ORI
Rot
DESC
description vector



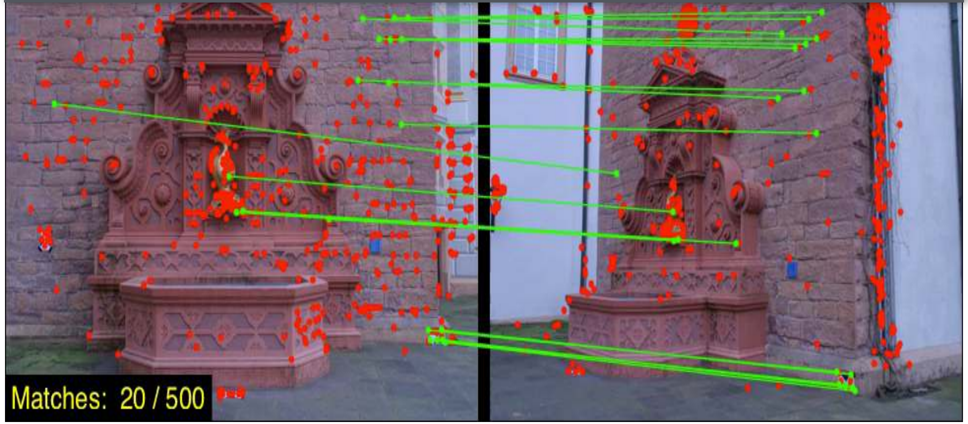
École des Ponts
ParisTech



"Differentiable glue"

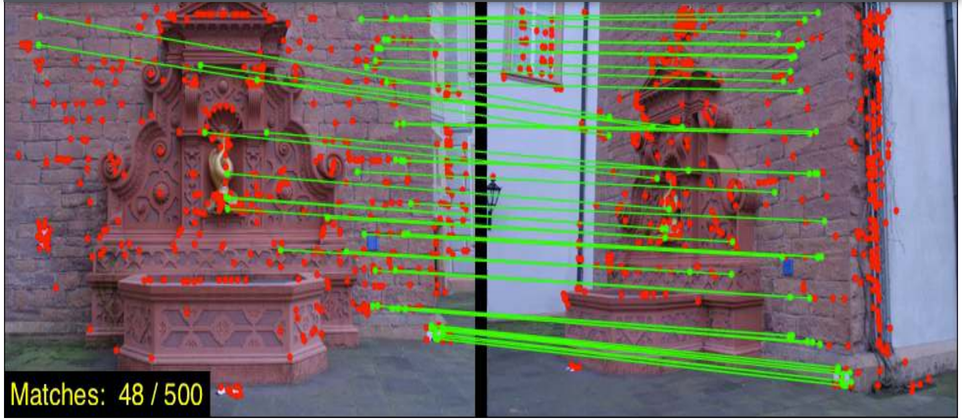
École des Ponts
ParisTech

SIFT. Average: 60.2 matches



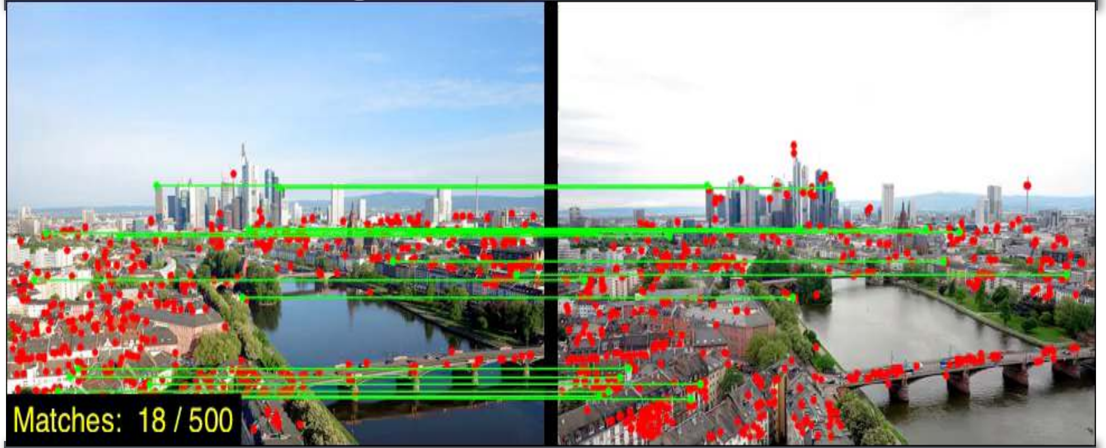
+64%

LIFT (Ours). Average: 98.6 matches



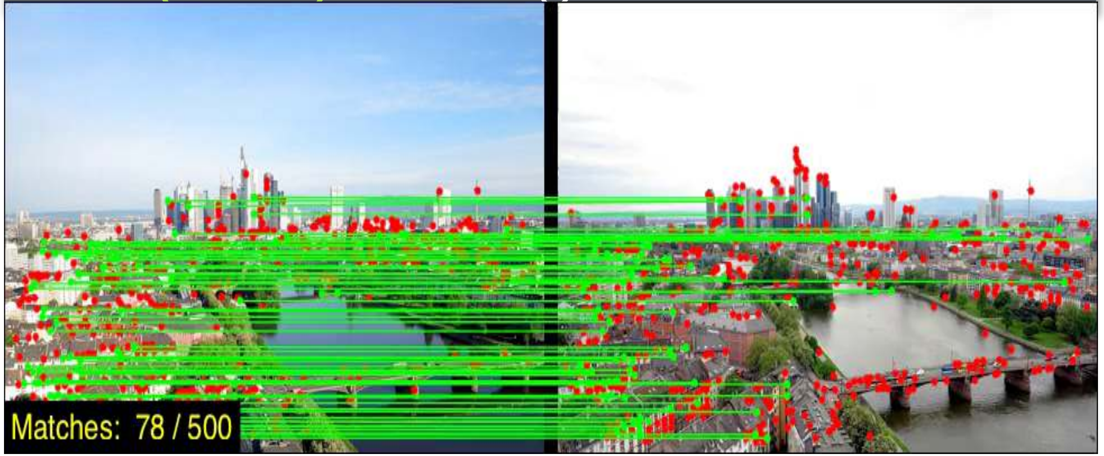
École des Ponts

SIFT. Average: 23.1 matches



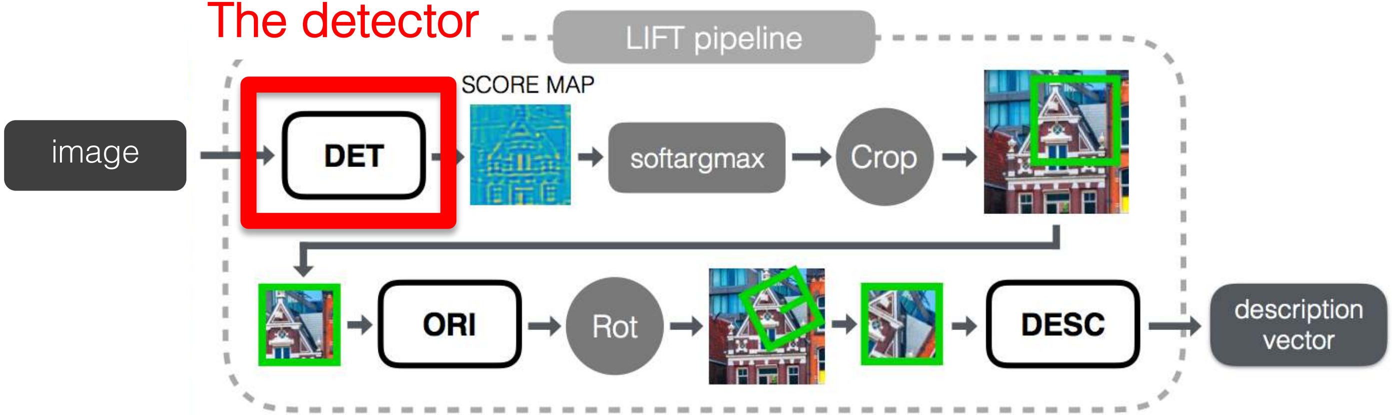
LIFT (Ours). Average: 60.6 matches

+162%



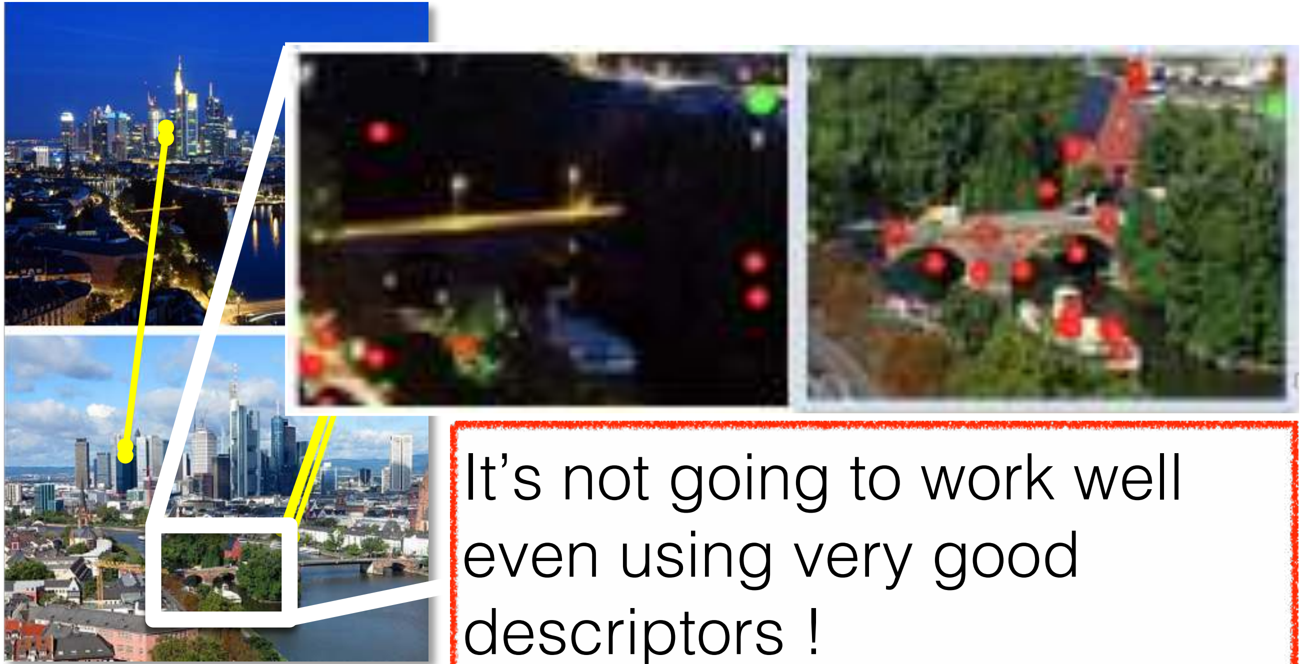
École des Ponts
ParisTech

The detector



Detection under Severe Illumination Changes

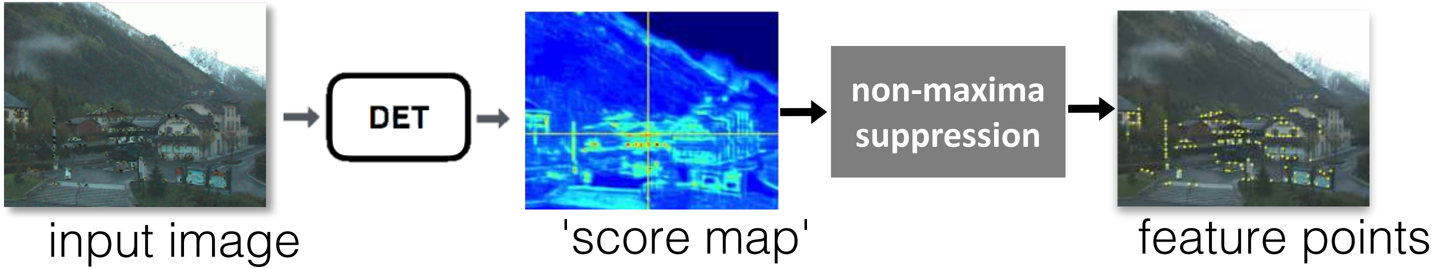
Detection under Severe Illumination Changes



It's not going to work well even using very good descriptors !

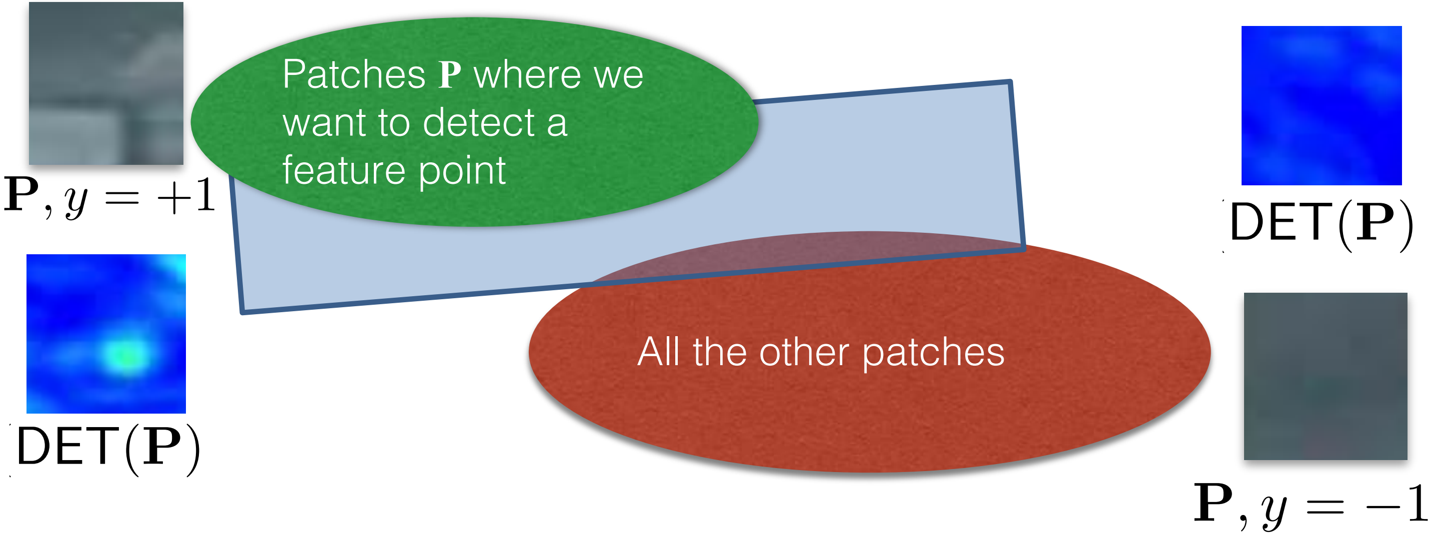
Learning to Detect under Severe Illumination Changes

How the Detector is Used at Run-Time



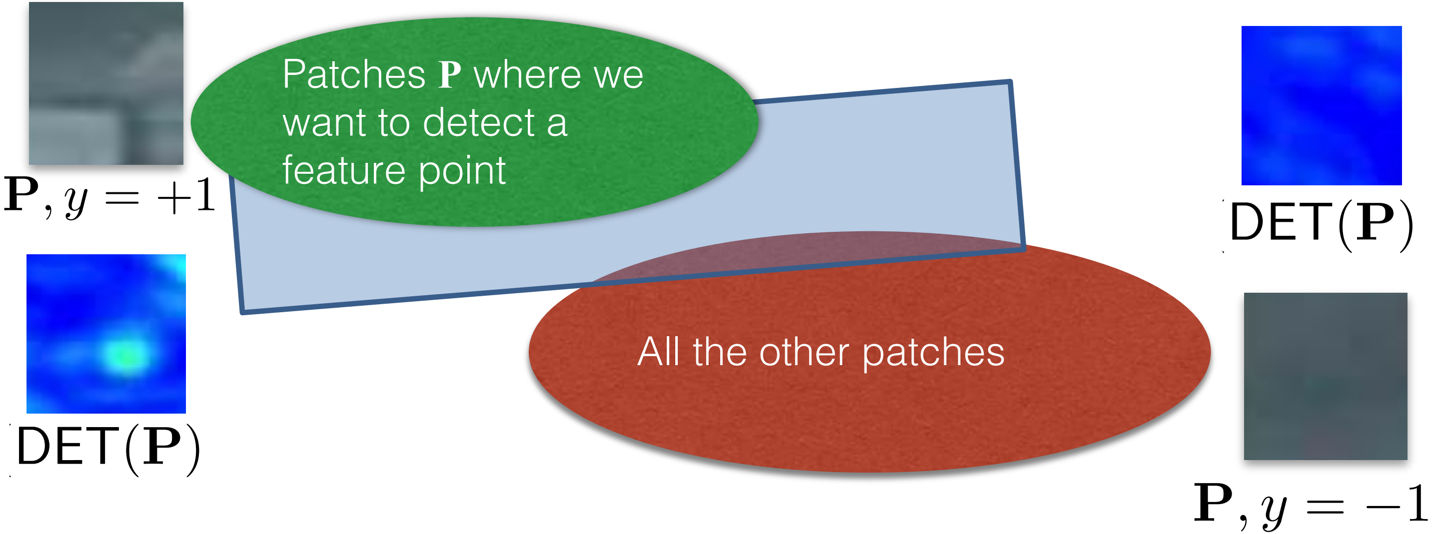
École des Ponts
ParisTech

Cost Function (1)



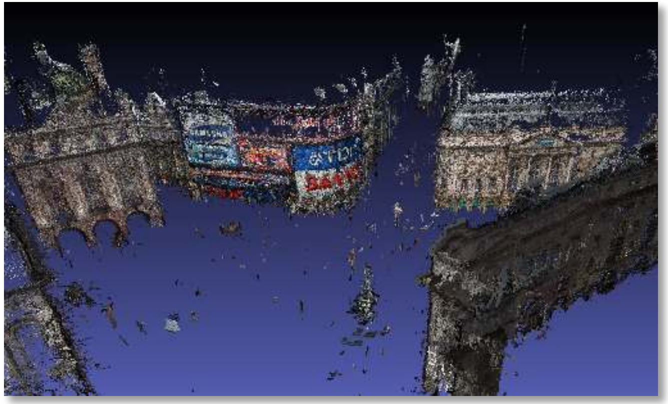
$$\mathcal{L}_{\text{class}}(\mathbf{P}) = \max(0, 1 - y \max(\text{DET}(\mathbf{P})))^2, y \in \{-1, +1\}$$

Cost Function (1)

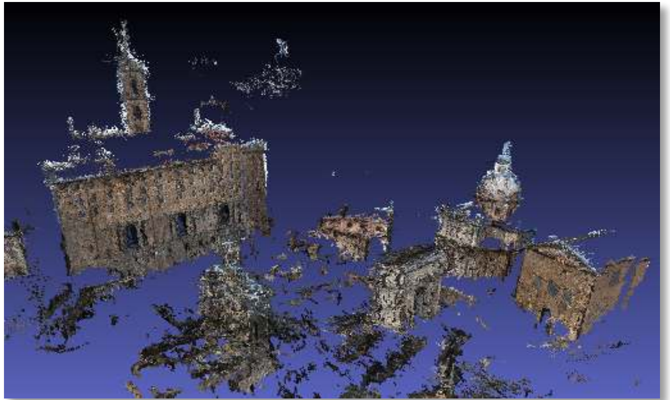


$$\mathcal{L}_{\text{class}}(\mathbf{P}) = \max(0, 1 - y\ \underline{\text{softmax}}(\text{DET}(\mathbf{P})))^2, y \in \{-1, +1\}$$

Training with SfM Keypoints



Piccadilly (pic)



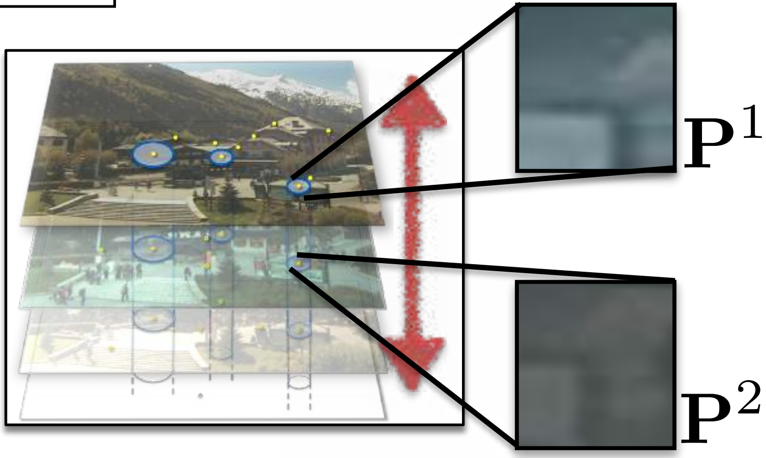
Roman Forum (rf)

- We need variability (illumination, perspective, etc). We build SfM reconstructions from **photo-tourism sets.**

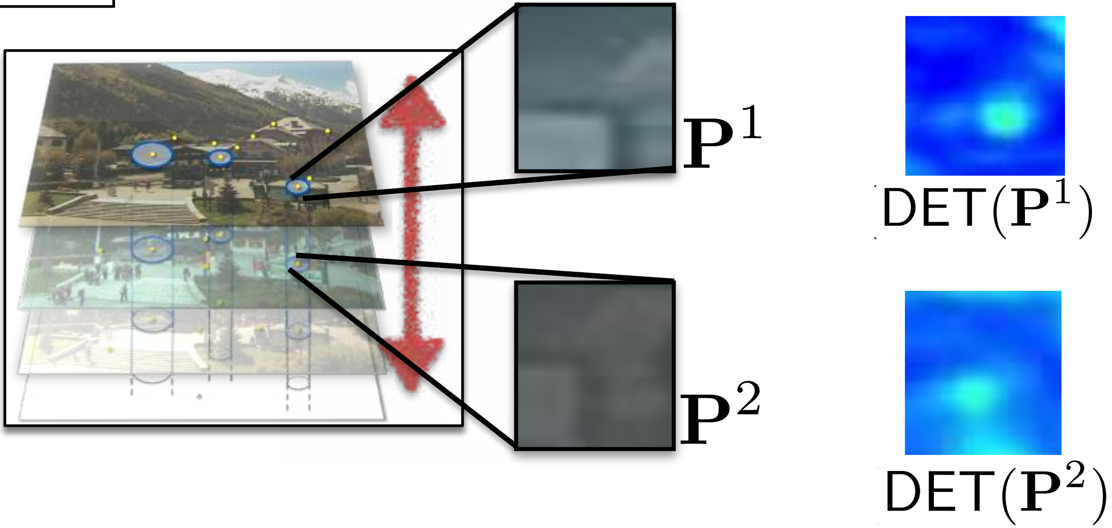
- We keep only **points with SfM correspondences** as positive examples, that is, we **learn to find repeatable points.**

École des Ponts
ParisTech

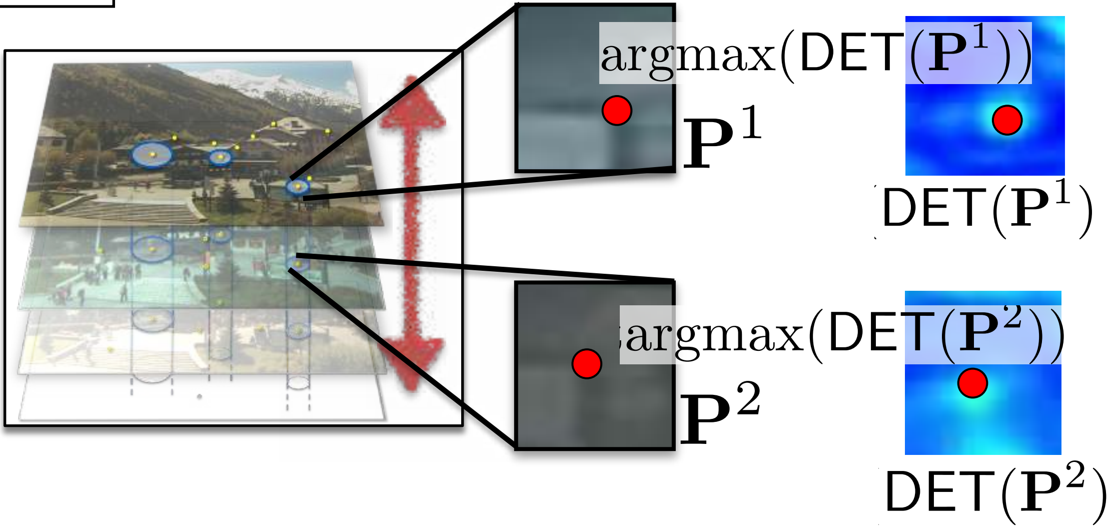
Cost Function (2)



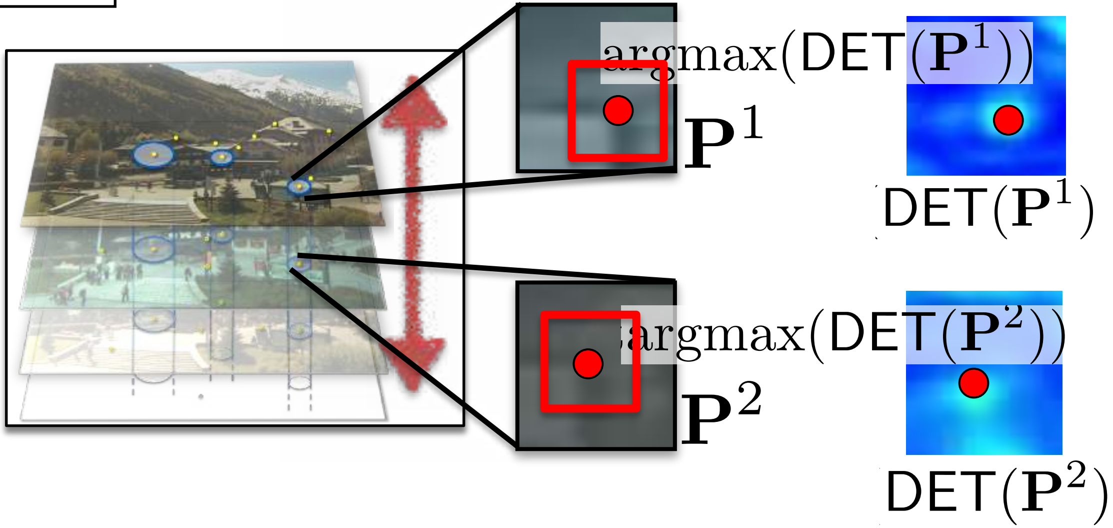
Cost Function (2)



Cost Function (2)

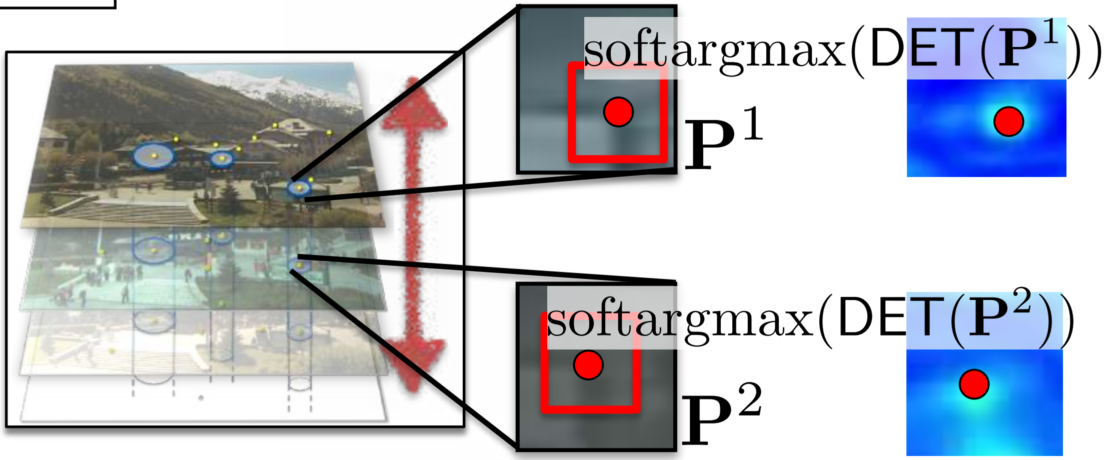


Cost Function (2)



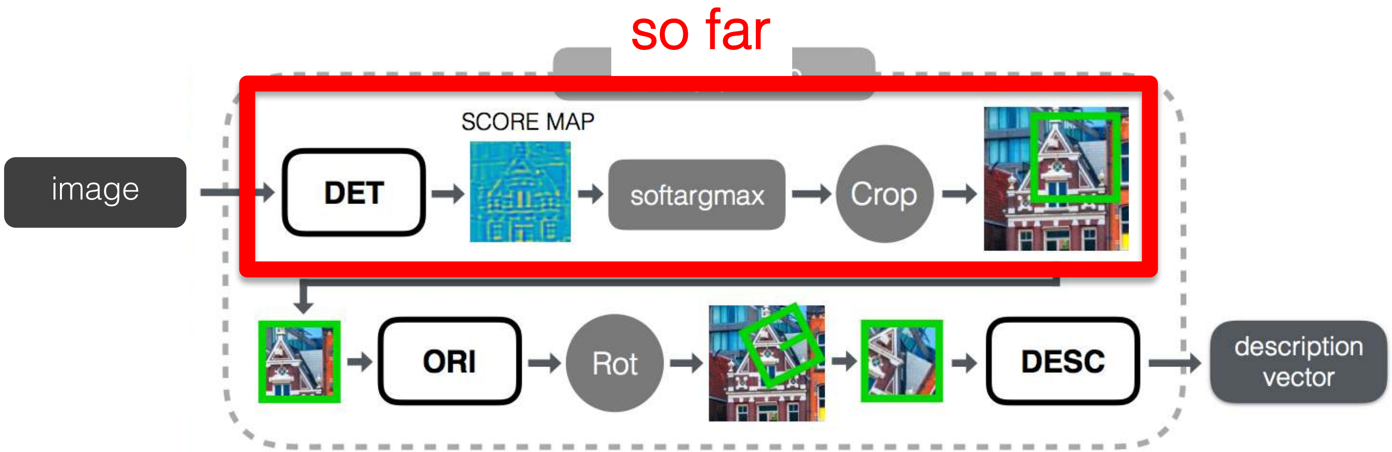
$$\mathcal{L}_{\text{pair}}(\mathbf{P}^1,\mathbf{P}^2) = \| \ \text{DESC}(\text{Crop}(\mathbf{P}^1, \text{argmax}(\text{DET}(\mathbf{P}^1)))) - \text{DESC}(\text{Crop}(\mathbf{P}^2, \text{argmax}(\text{DET}(\mathbf{P}^2)))) \ \|^2$$

Cost Function (2)

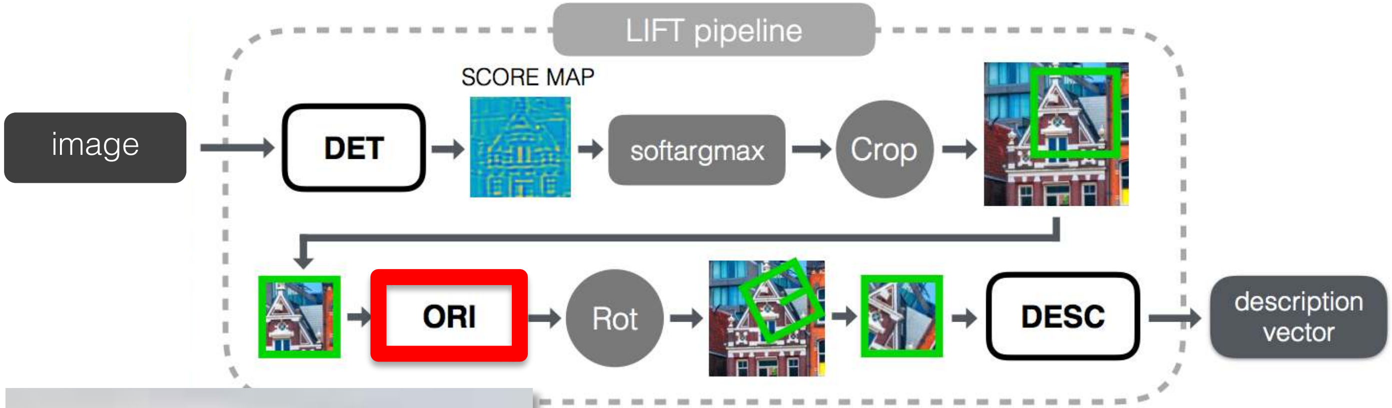


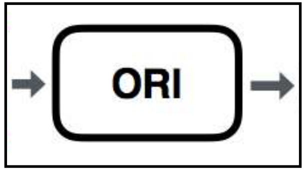
$$\mathcal{L}_{\text{pair}}(\mathbf{P}^1, \mathbf{P}^2) = \| \quad \text{DESC}(\text{Crop}(\mathbf{P}^1, \underline{\text{softargmax}}(\text{DET}(\mathbf{P}^1)))) - \text{DESC}(\text{Crop}(\mathbf{P}^2, \underline{\text{softargmax}}(\text{DET}(\mathbf{P}^2)))) \quad \|^2$$

$$\text{softargmax}(\mathbf{S}) = \frac{\sum_{\mathbf{x}} \exp(\beta \mathbf{S}(\mathbf{x}))\mathbf{x}}{\sum_{\mathbf{x}} \exp(\beta \mathbf{S}(\mathbf{x}))}$$



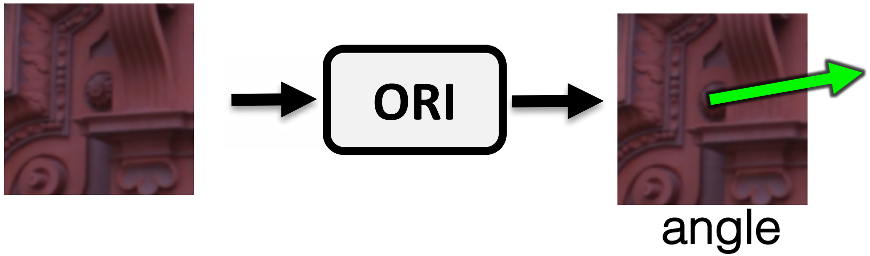
École des Ponts

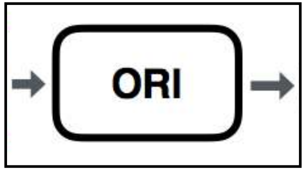

LIFT pipeline
image
DET
SCORE MAP
softargmax
Crop
ORI
Rot
DESC
description vector



Learning Orientations Implicitly

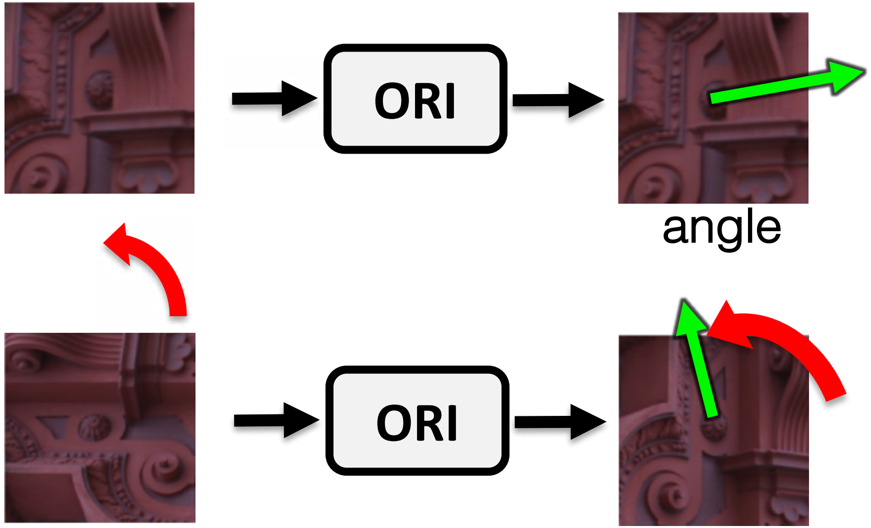
We want the orientation estimator to provide **consistent** results, regardless of imaging changes



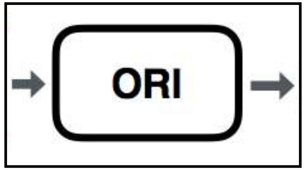


Learning Orientations Implicitly

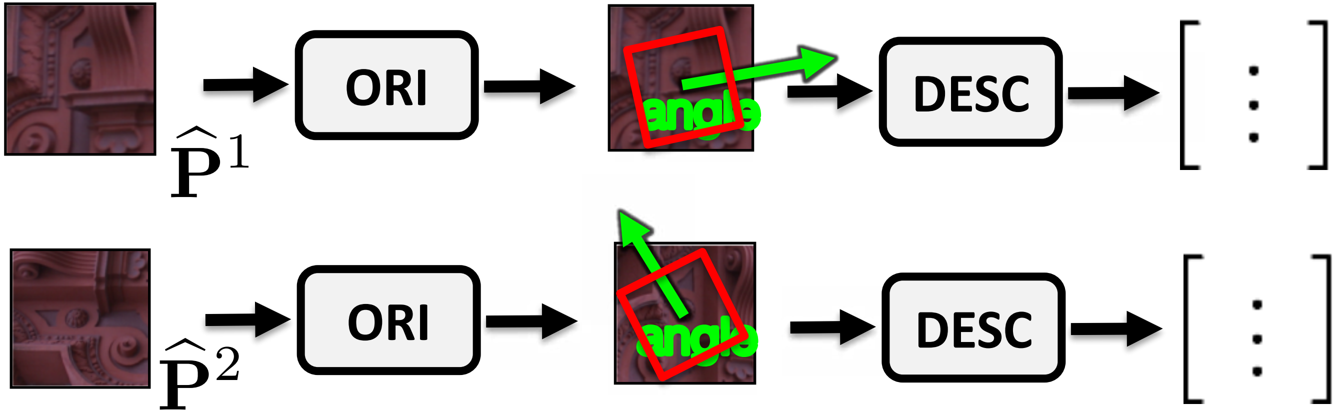
We want the orientation estimator to provide **consistent** results, regardless of imaging changes

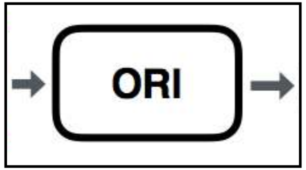


École des Ponts



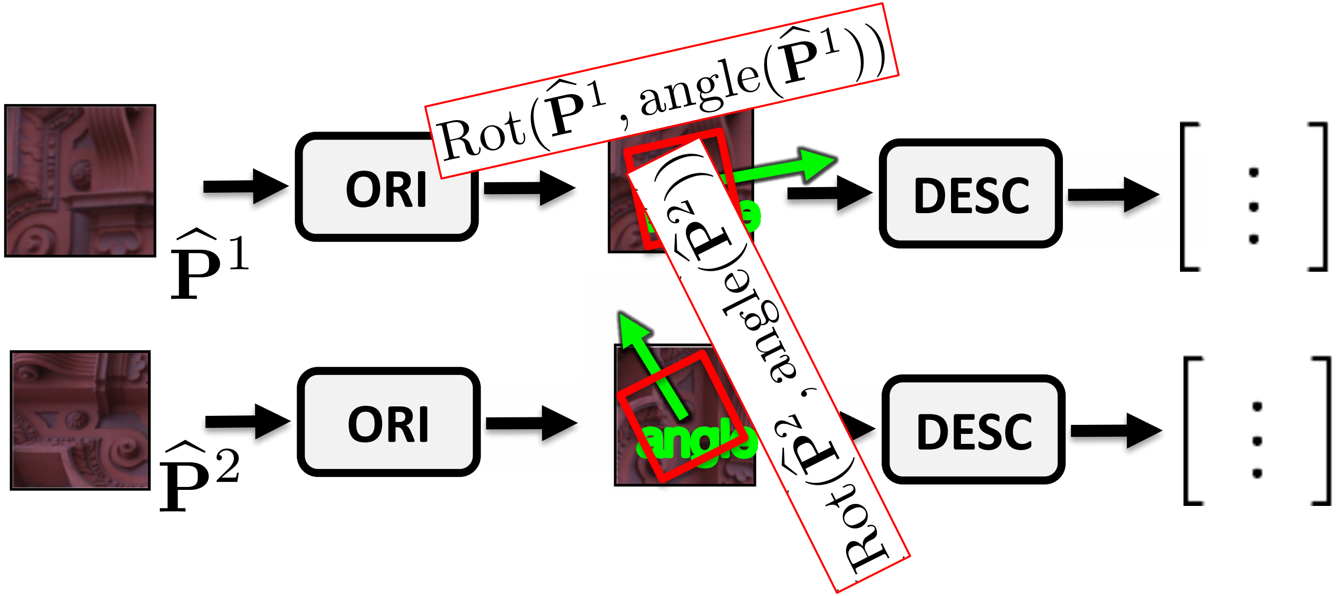
Learning Orientations Implicitly: A Siamese Network with a Twist

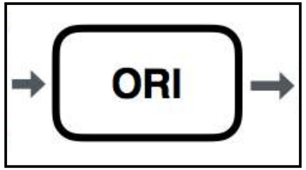




Learning Orientations Implicitly: A Siamese Network with a Twist

$$\mathcal{L}_{\text{pair}}(\widehat{\mathbf{P}}^1, \widehat{\mathbf{P}}^2) = \| \ \text{DESC}(\text{Rot}(\widehat{\mathbf{P}}^1, \text{angle}(\widehat{\mathbf{P}}^1))) - \text{DESC}(\text{Rot}(\widehat{\mathbf{P}}^2, \text{angle}(\widehat{\mathbf{P}}^2))) \ \|^2$$

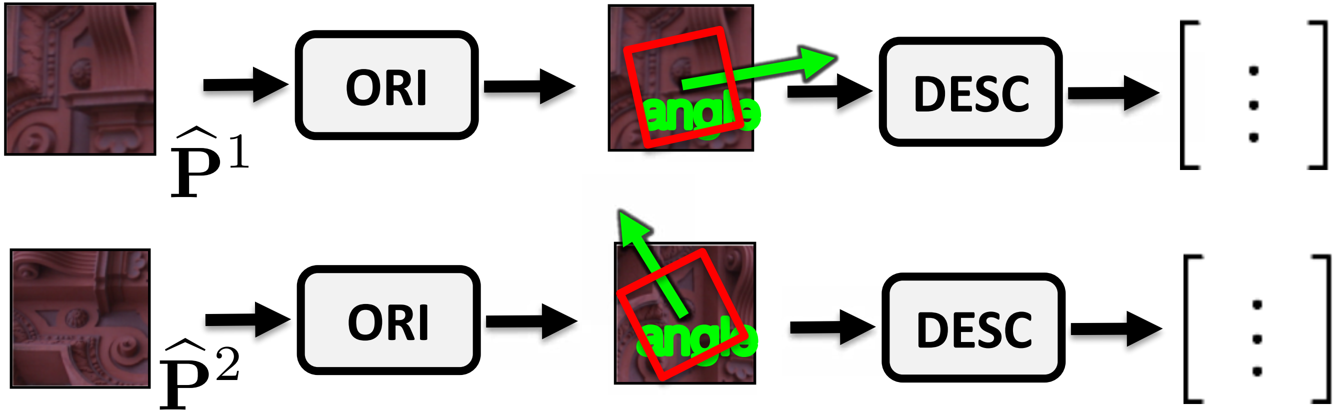


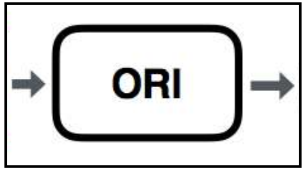


Learning Orientations Implicitly: A Siamese Network with a Twist

$$\mathcal{L}_{\text{pair}}(\widehat{\mathbf{P}}^1, \widehat{\mathbf{P}}^2) = \| \ \mathsf{DESC}(\text{Rot}(\widehat{\mathbf{P}}^1, \text{angle}(\widehat{\mathbf{P}}^1))) - \mathsf{DESC}(\text{Rot}(\widehat{\mathbf{P}}^2, \text{angle}(\widehat{\mathbf{P}}^2))) \ \|^2$$

$$\text{with angle}(\widehat{\mathbf{P}}) = \text{arctan2}(\mathsf{ORI}(\widehat{\mathbf{P}})^{[1]}, \mathsf{ORI}(\widehat{\mathbf{P}})^{[2]})$$



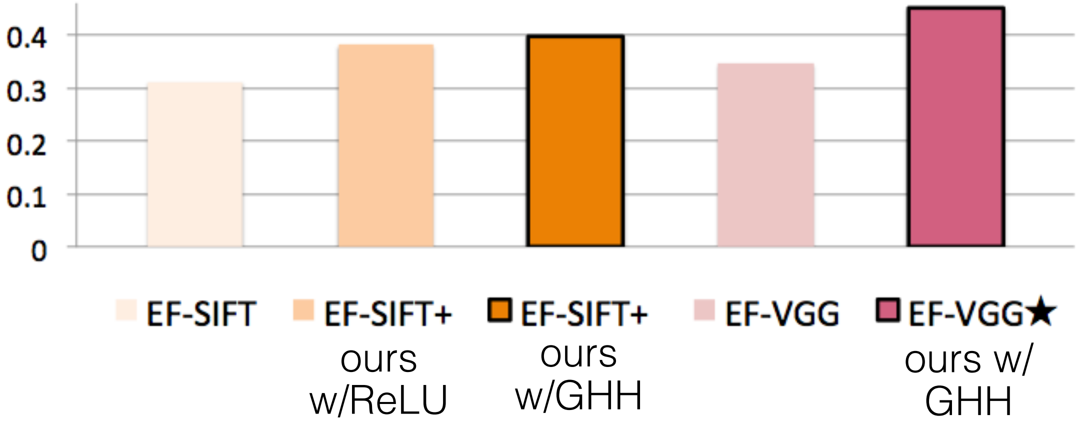


Performance Gain with Learned Orientations

mAP

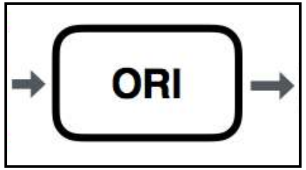
(higher is better)

Average performance

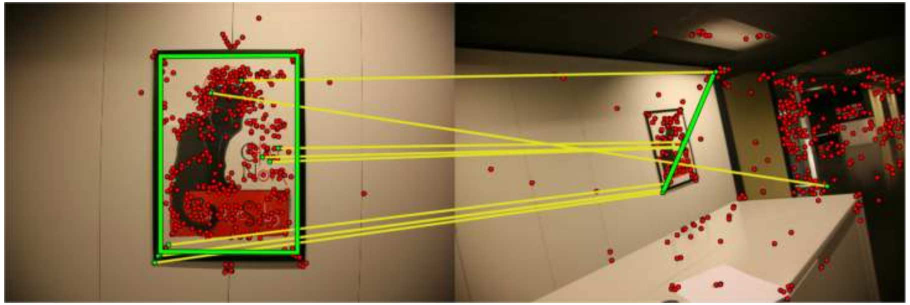


Descriptor matching performances (mAP) with nearest neighbor matching (Mikolajczyk & Schmid, IJCV'04).

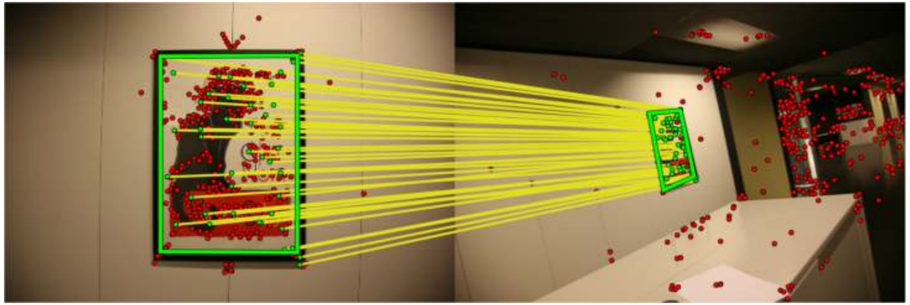
École des Ponts ParisTech



Learned Orientations

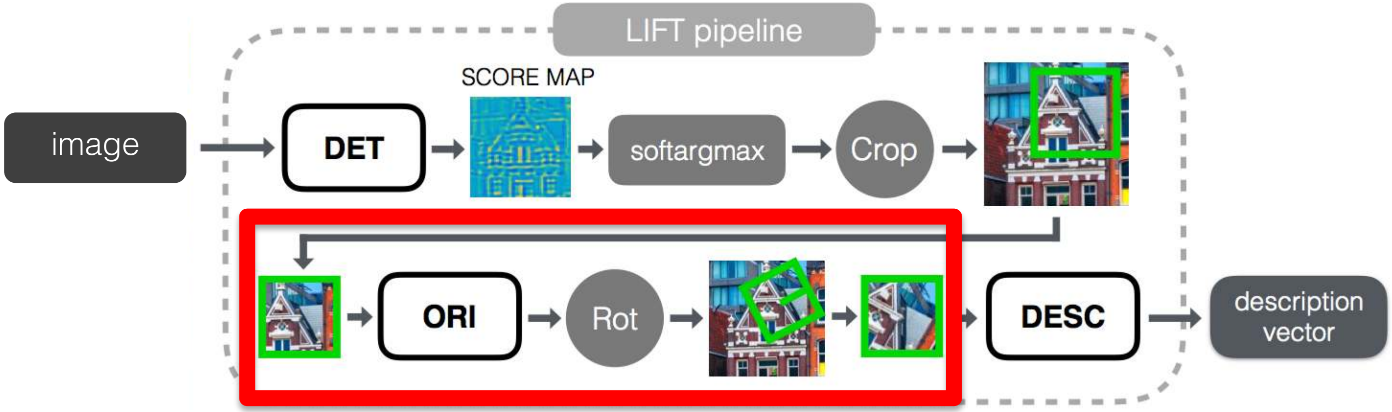


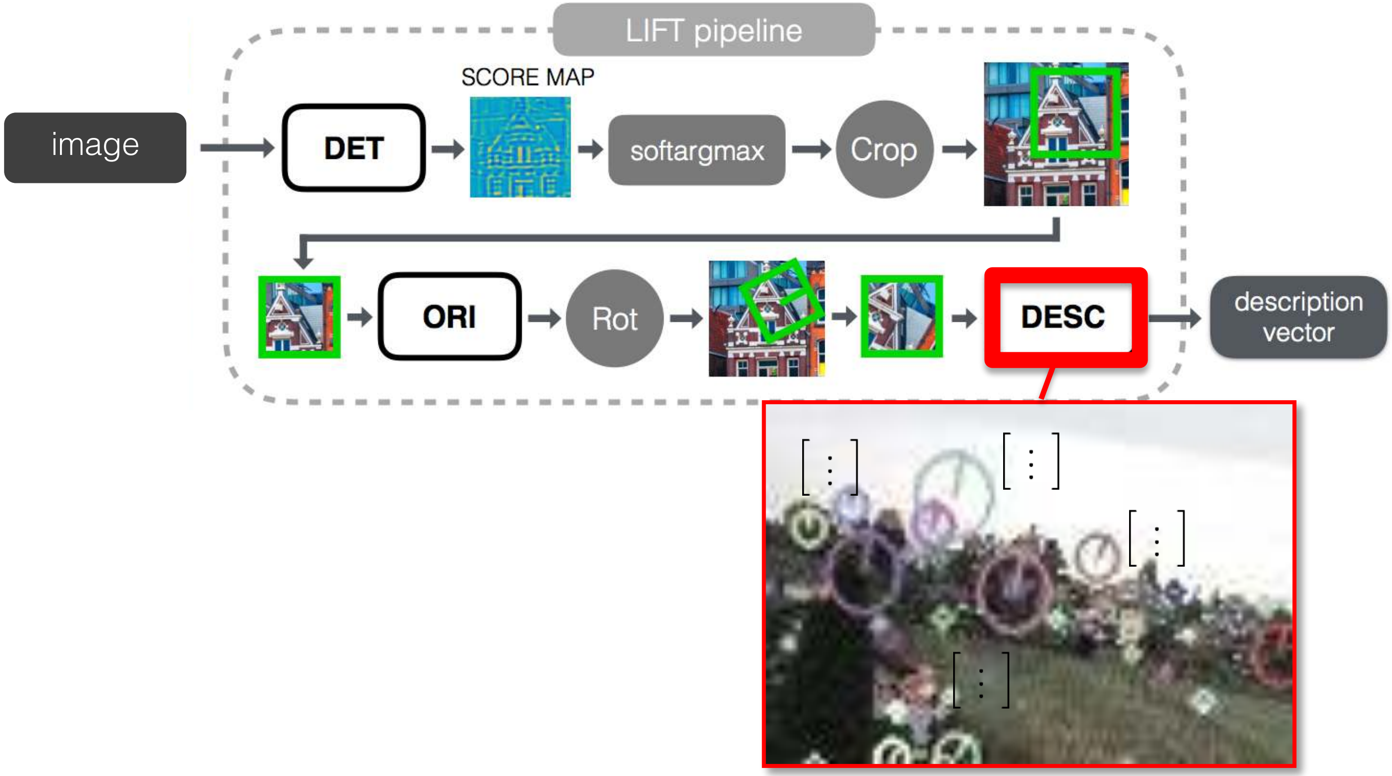
Dominant Gradient Orientations



Our Learned Orientations

École des Ponts ParisTech


LIFT pipeline
image
DET
SCORE MAP
softargmax
Crop
ORI
Rot
DESC
description vector



École des Ponts
ParisTech

Learning the Descriptor

- Positive pairs:  $\widehat{\mathbf{P}}^1$  $\widehat{\mathbf{P}}^2$

$$\mathcal{L}_{\text{pos}}(\widehat{\mathbf{P}}^1, \widehat{\mathbf{P}}^2) = \|\text{DESC}(\widehat{\mathbf{P}}^1) - \text{DESC}(\widehat{\mathbf{P}}^2)\|^2$$

Learning the Descriptor

- Positive pairs:

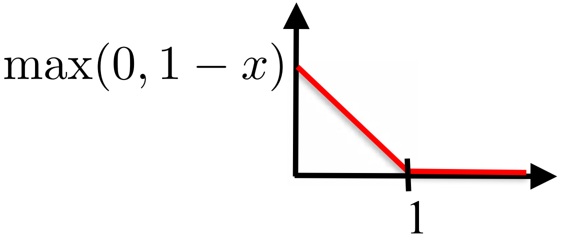
$\widehat{\mathbf{P}^1}$

$\widehat{\mathbf{P}^2}$

$$\mathcal{L}_{\text{pos}}(\widehat{\mathbf{P}^1}, \widehat{\mathbf{P}^2}) = \|\text{DESC}(\widehat{\mathbf{P}^1}) - \text{DESC}(\widehat{\mathbf{P}^2})\|^2$$

- Negative pairs:

$$\mathcal{L}_{\text{neg}}(\widehat{\mathbf{P}^1}, \widehat{\mathbf{P}^3}) = \max(0, 1 - \|\text{DESC}(\widehat{\mathbf{P}^1}) - \text{DESC}(\widehat{\mathbf{P}^3})\|^2)$$



$\widehat{\mathbf{P}^1}$

$\widehat{\mathbf{P}^3}$

Hard example mining is very important for training

A Single, Global Cost Function

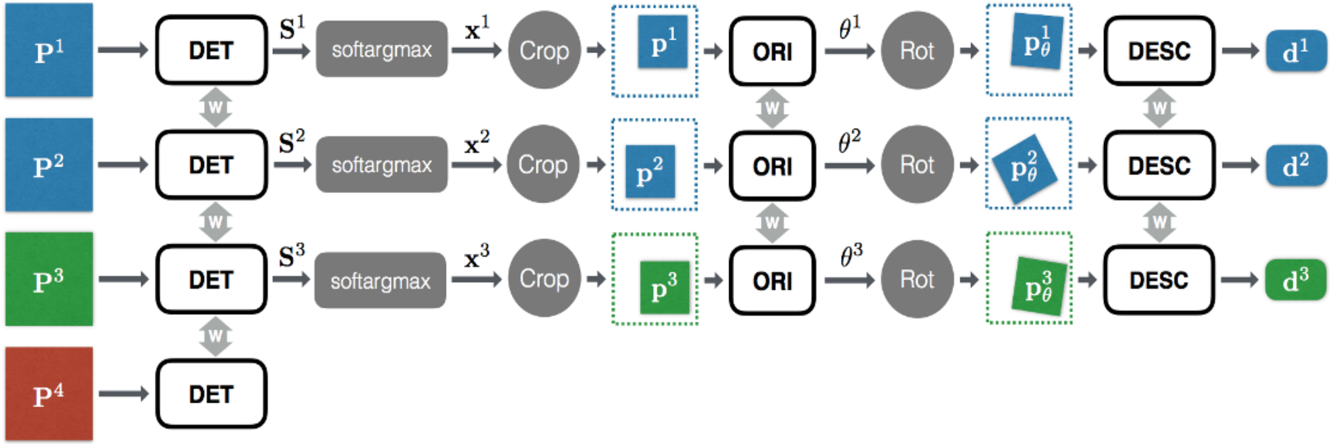
$$\min_{\{\mathsf{DET},\mathsf{ORI},\mathsf{DESC}\}} \sum_{\{(\mathbf{P},y)\}} \max(0, 1 - y \text{ softmax}(\mathsf{DET}(\mathbf{P})))^2 +$$

$$\sum_{(\mathbf{P}_1,\mathbf{P}_2)} \|\mathsf{DESC}\left(G(\mathbf{P}^1, \text{softargmax}(\mathsf{DET}(\mathbf{P}^1))\right) - \mathsf{DESC}\left(G(\mathbf{P}^2, \text{softargmax}(\mathsf{DET}(\mathbf{P}^2))\right)\|^2 +$$

$$\sum_{(\mathbf{P}_1,\mathbf{P}_3)} \max(0, 1 - \|\mathsf{DESC}\left(G(\mathbf{P}^1, \text{softargmax}(\mathsf{DET}(\mathbf{P}^1))\right) - \mathsf{DESC}\left(G(\mathbf{P}^3, \text{softargmax}(\mathsf{DET}(\mathbf{P}^3)))\right)\|^2)$$

$$G(\mathbf{P}, \mathbf{x}) = \text{Rot}\left(\mathbf{P}, \mathbf{x}, \text{angle}_{\mathsf{ORI}}(\text{Crop}(\mathbf{P}, \mathbf{x}))\right)$$

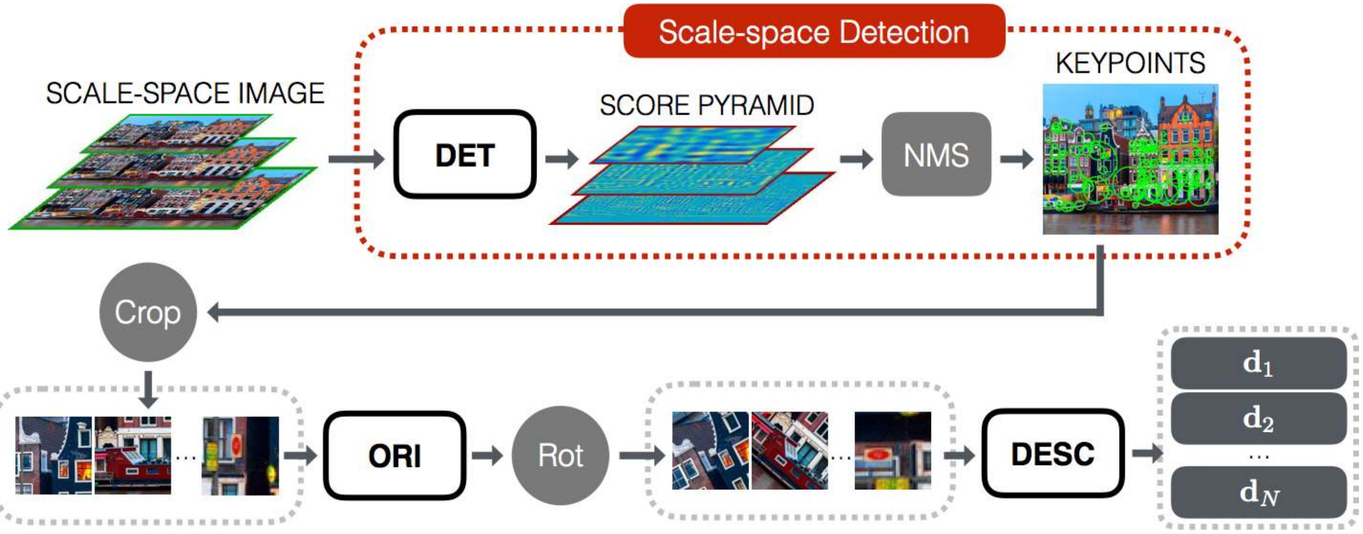
Problem-Specific Training



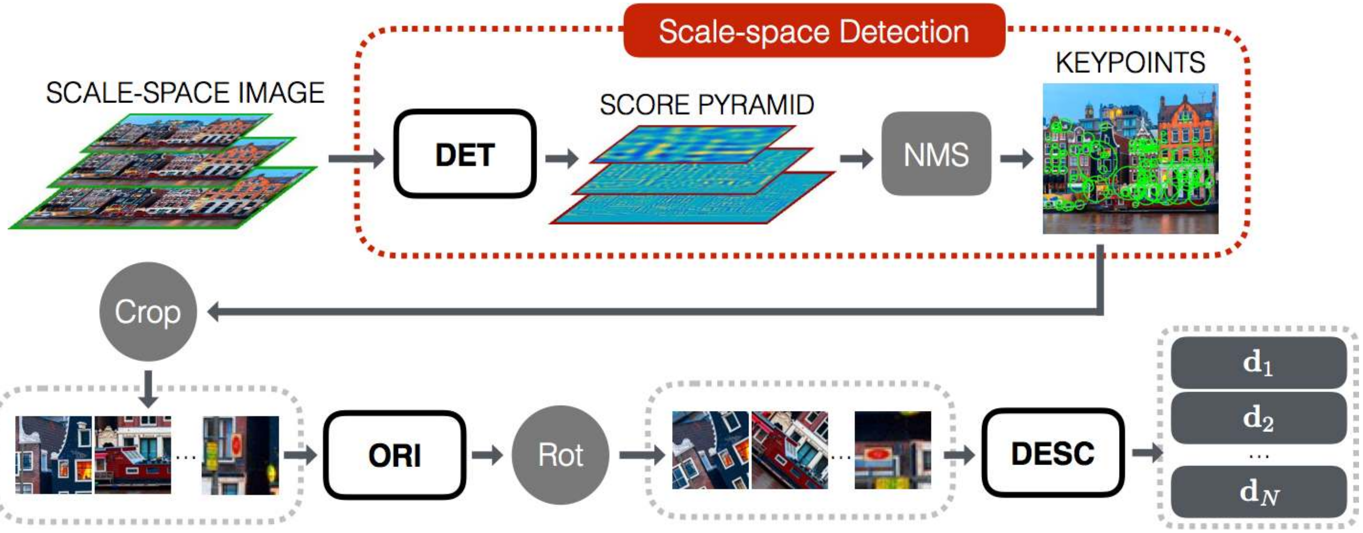
Run-Time Pipeline

Detector

- is purely convolutional, efficiently applied to the whole image;
- works in scale-space.



Run-Time Pipeline



The orientation estimator and the descriptor are applied only to keypoints.

École des Ponts
ParisTech