TRaitement des Images pour la Vision Artificielle

Vincent Lepetit

slides in part based on material from Mathieu Aubry, Andrew Zisserman, David Lowe

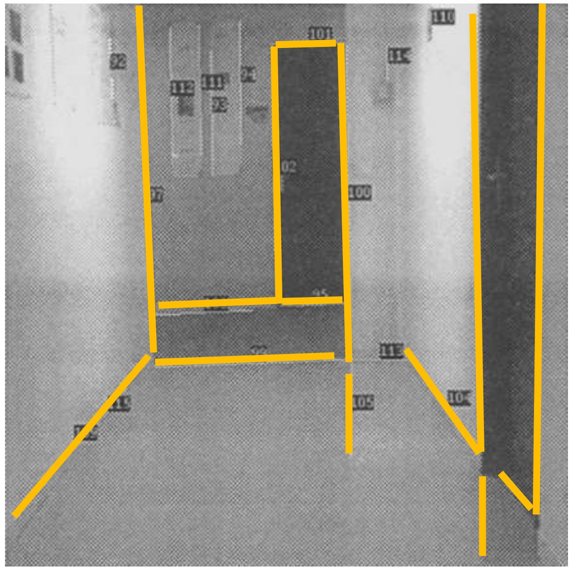
LIGM-Imagine Lab

École des Ponts
ParisTech

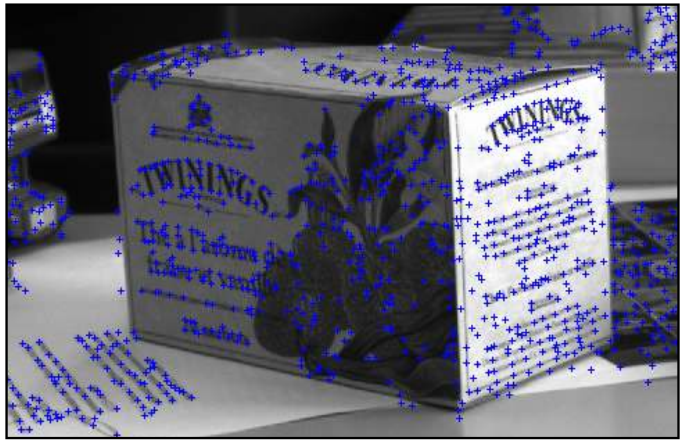
Today: Interest Points

Purpose

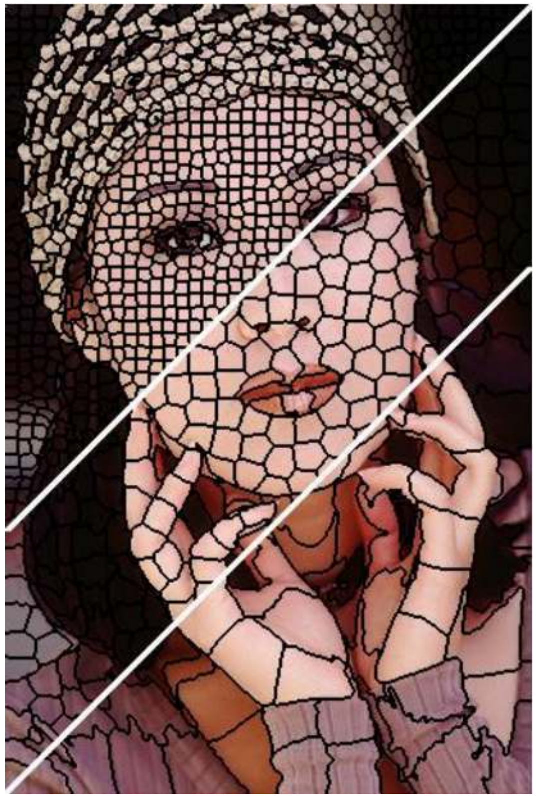
an image contains a lot of redundant information



straight lines segments



interest points

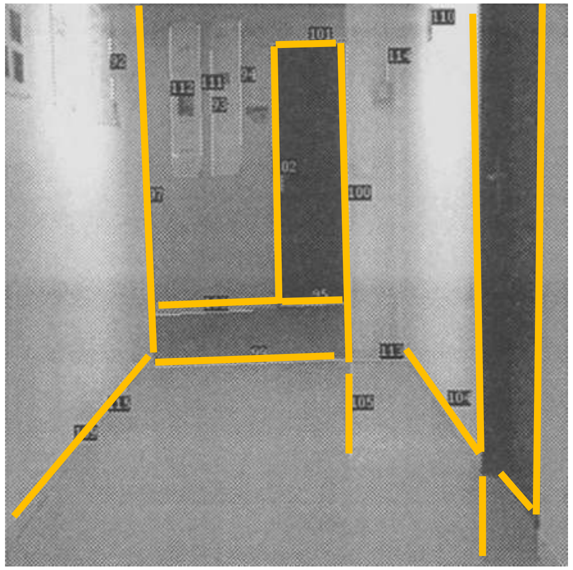


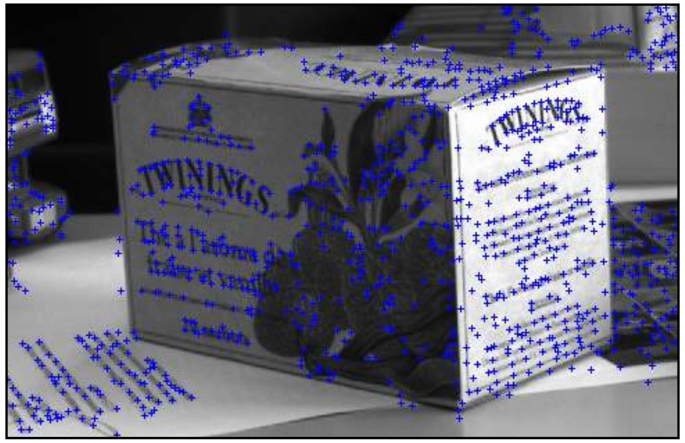
superpixels

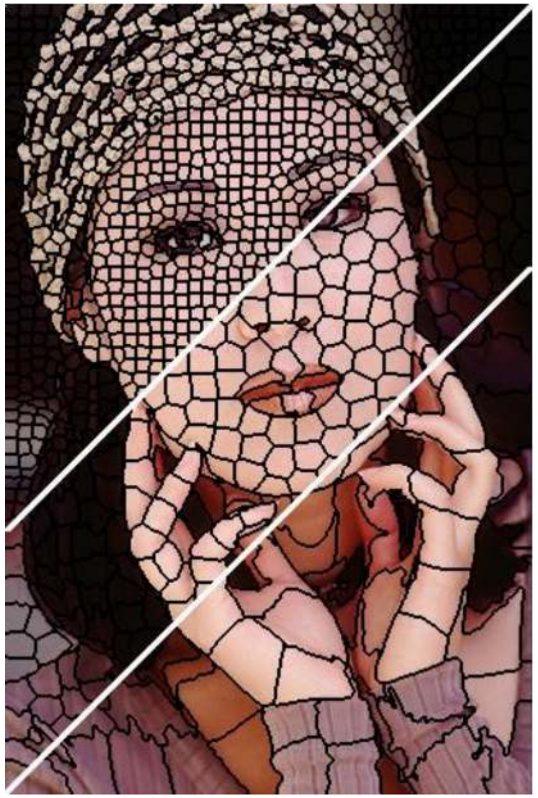
Features (~ *caractéristiques*): “Summary” of image content useful for computer vision applications

Purpose

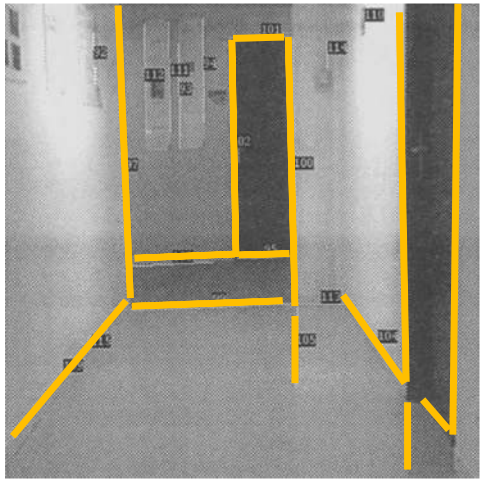
an image contains a lot of redundant information

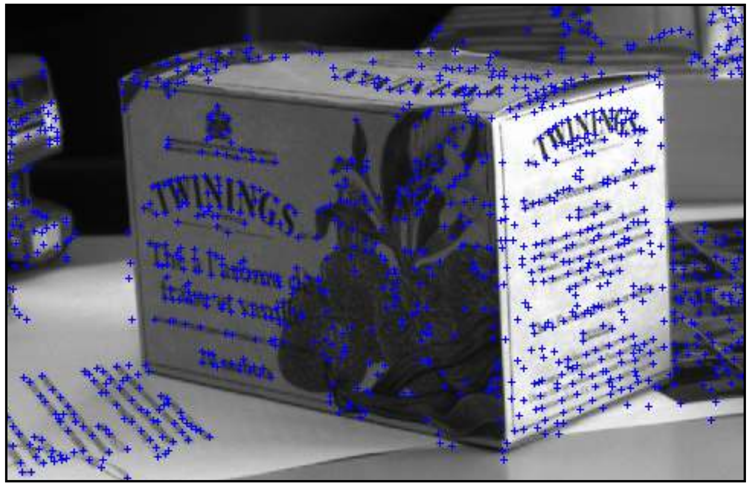

straight lines segments

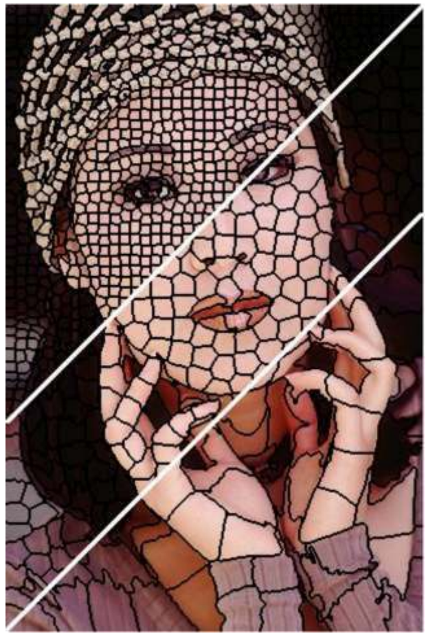

interest points


superpixels

Features (~ *caractéristiques*): "Summary" of image content useful for computer vision applications

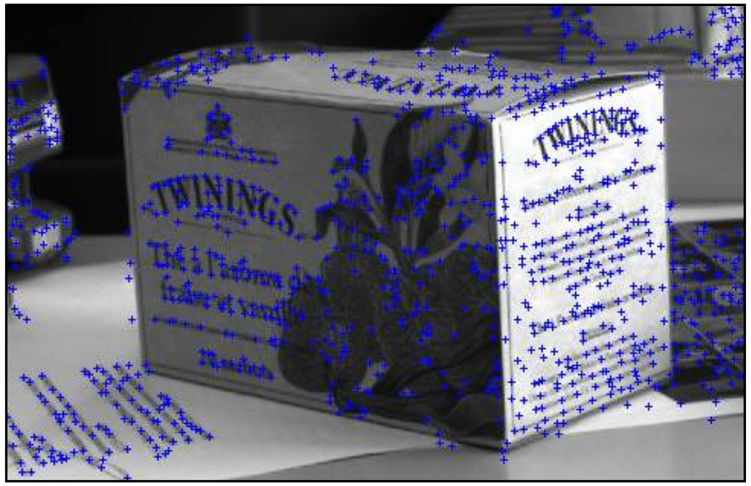

straight lines segments


interest points


superpixels

Features (~ *caractéristiques*): "Summary" of image content useful for computer vision applications.

!! different from features in machine learning, and image features computed by deep networks


interest points

Also called feature points, keypoints, corners, corner points, ..

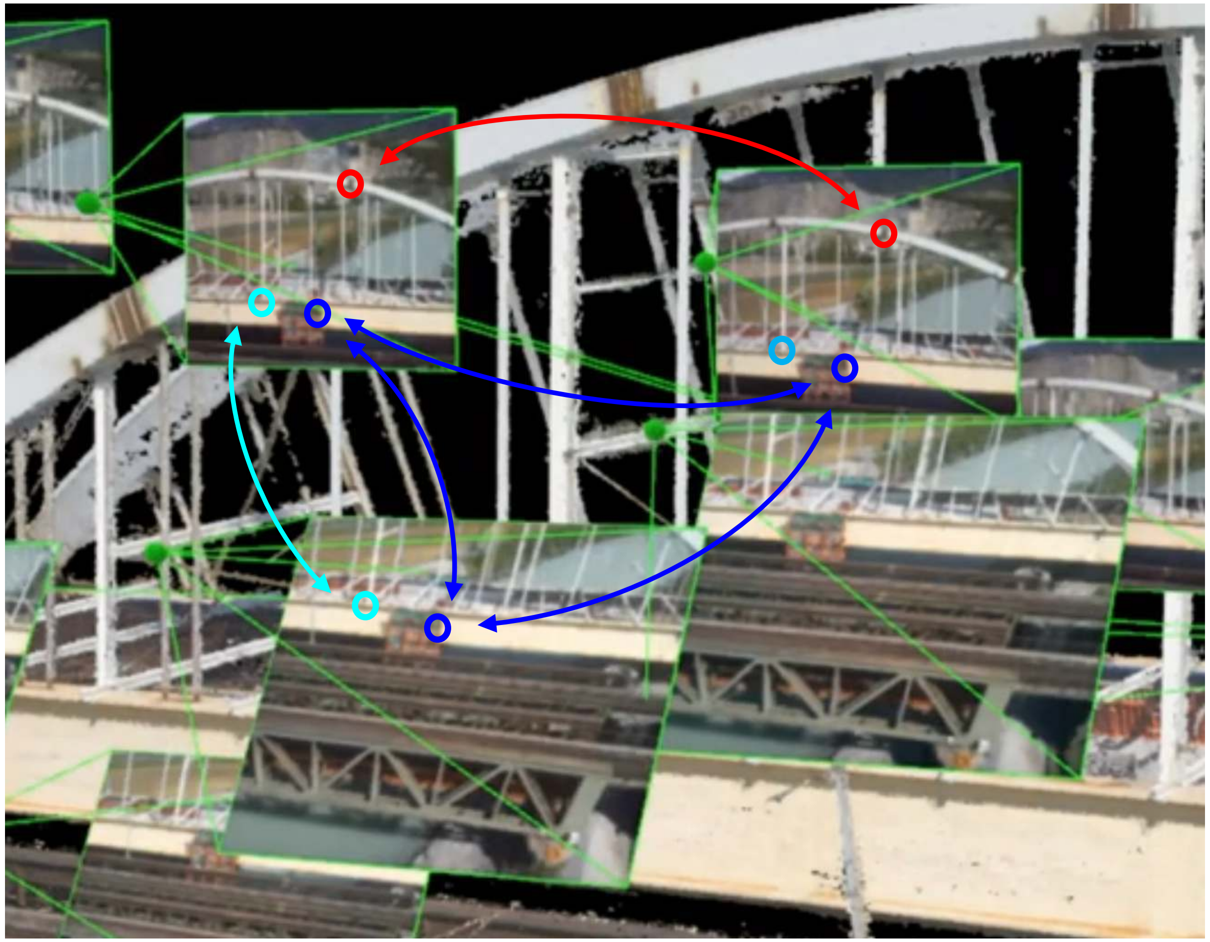
École des Ponts ParisTech

Applications

Pix4D
Pix4Dmapper

3D Reconstruction / Photogrammetry

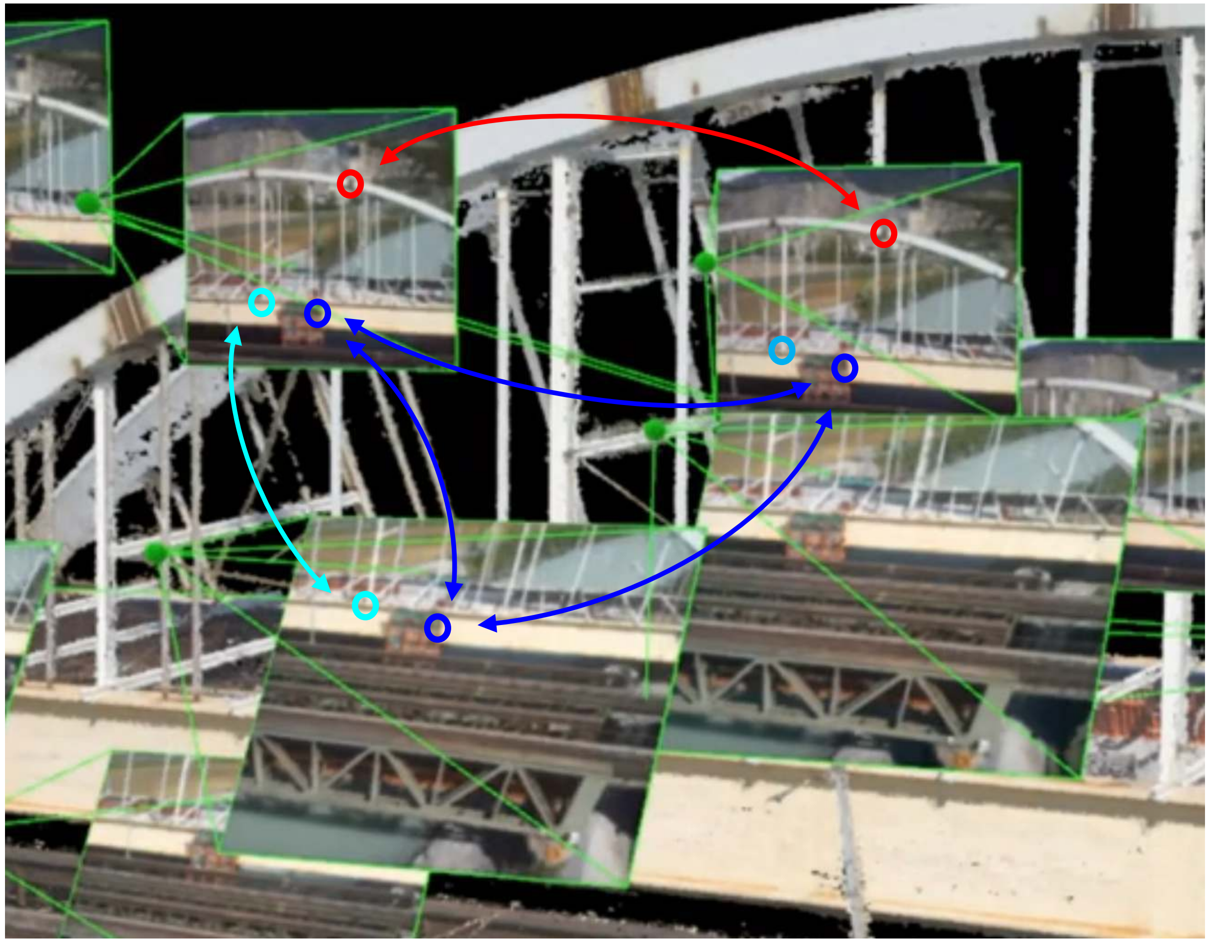
Interest Points for Photogrammetry



1. Detect interest points;
2. Match interest points across images;
3. Use these matches to find the positions and orientations of the cameras, and the 3D locations of the points.

École des Ponts
ParisTech

Interest Points for Photogrammetry



Today

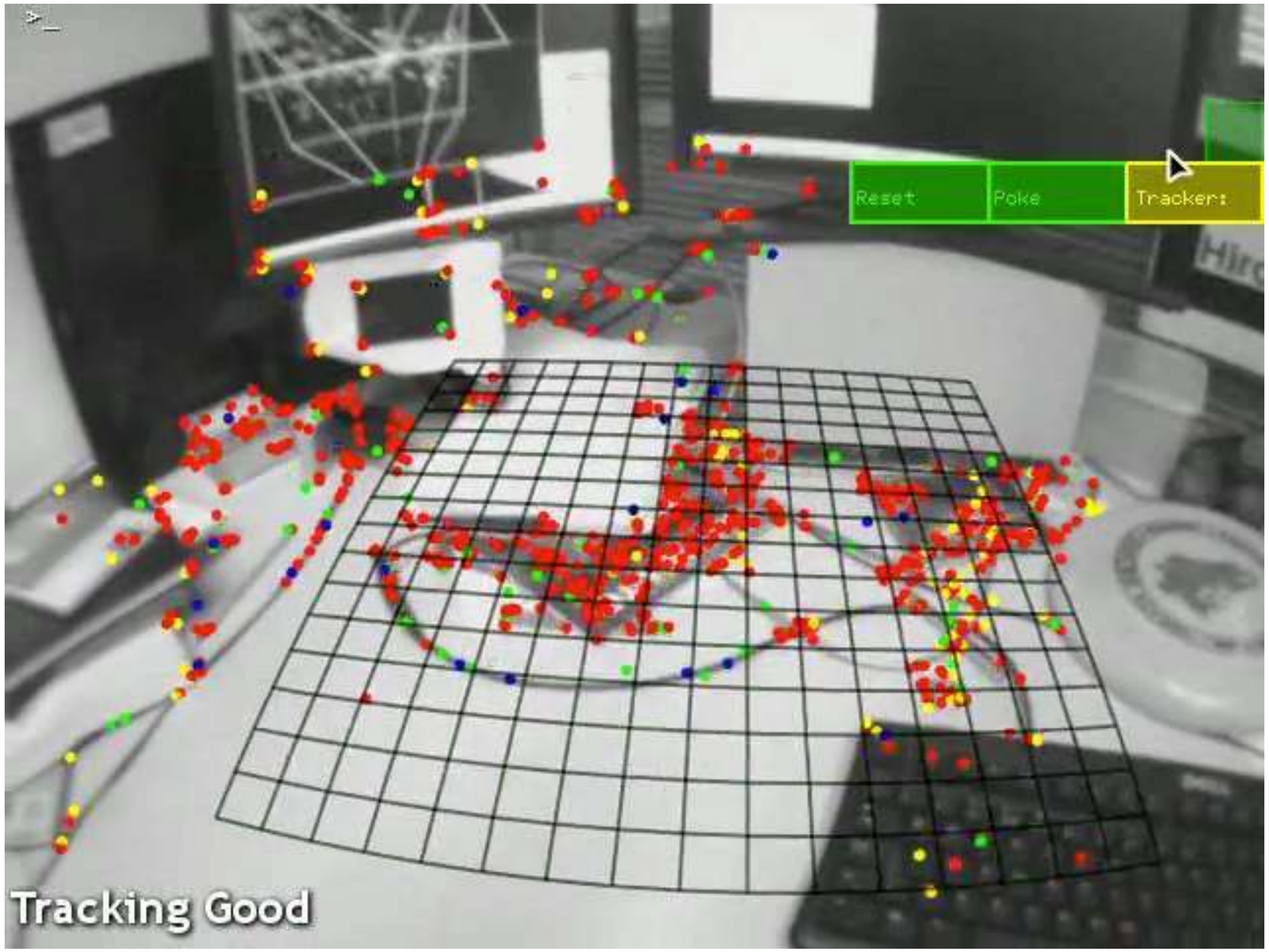
1. Detect interest points;
2. Match interest points across images;

3. Use these matches to find the positions and orientations of the cameras, and the 3D locations of the points.

Lectures #4 and #6

École des Ponts ParisTech

SLAM



École des Ponts ParisTech

Image Mosaicing

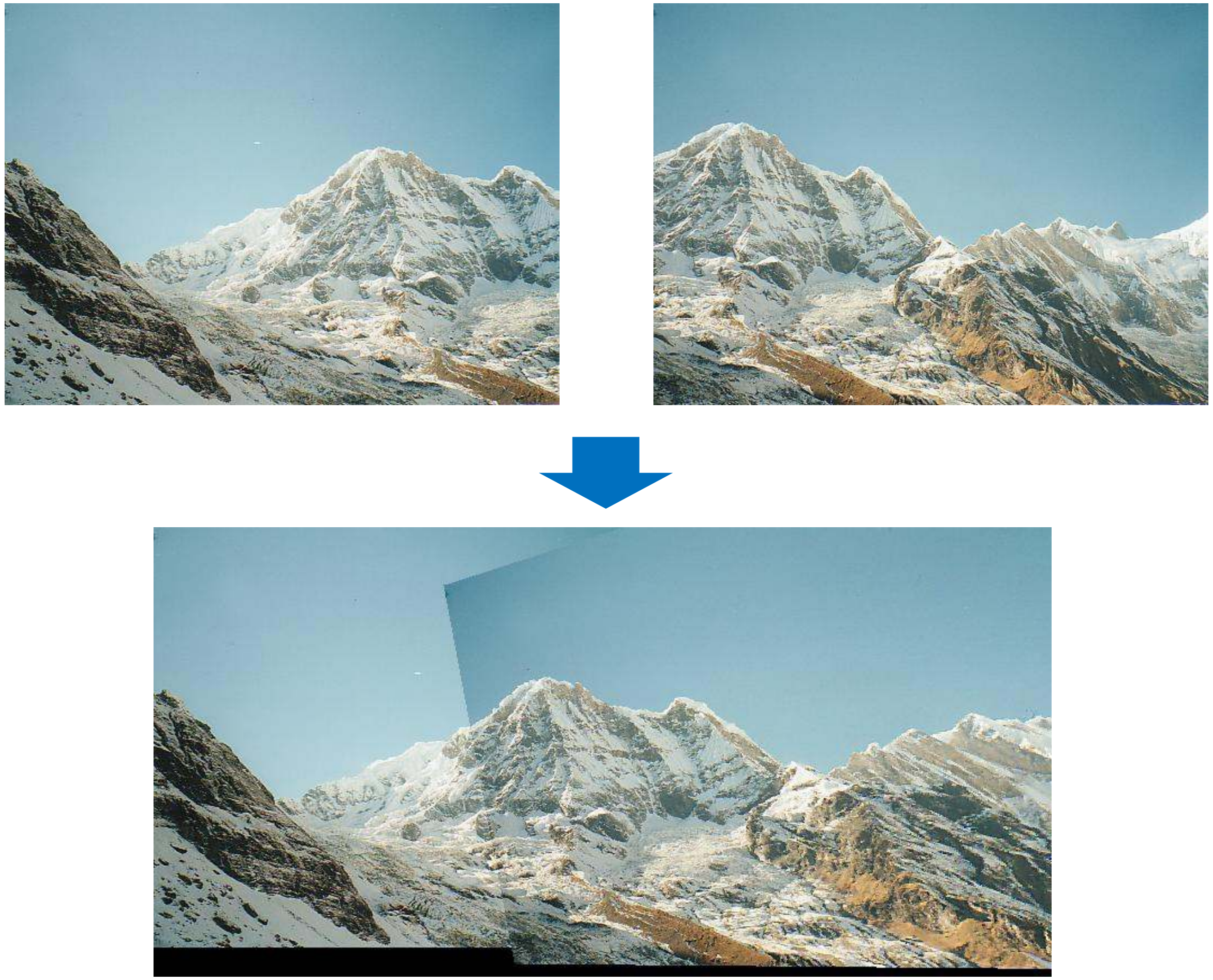
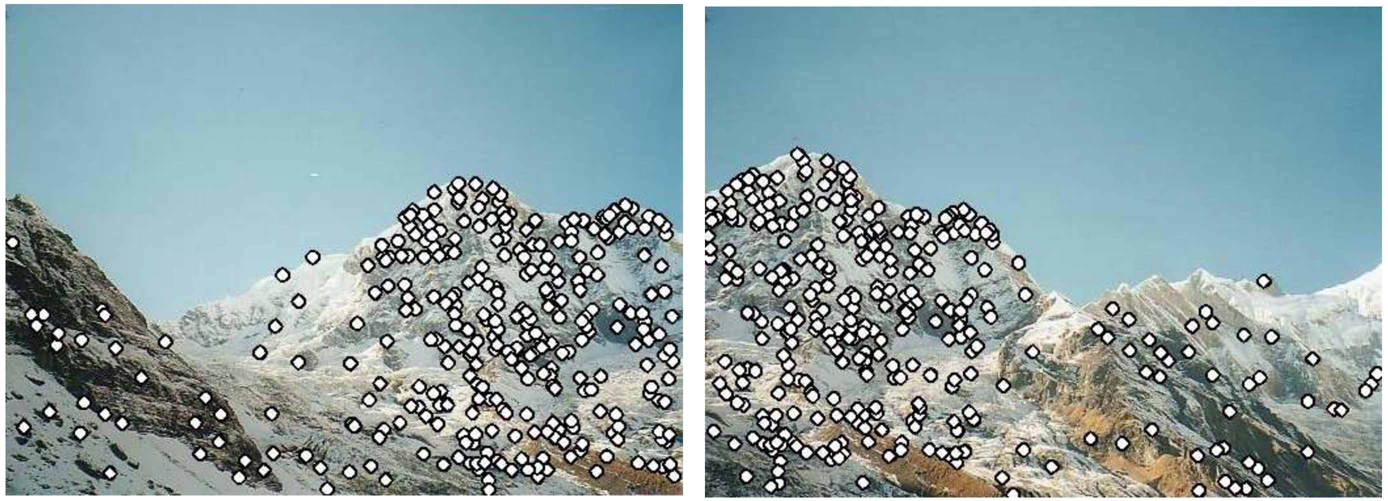
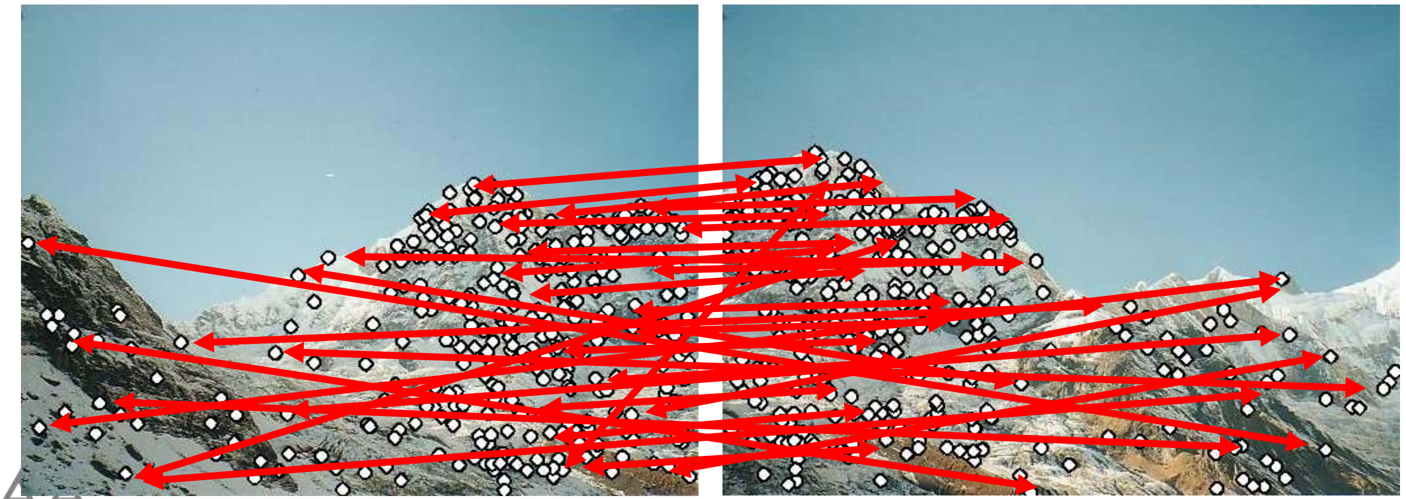


Image Mosaicing

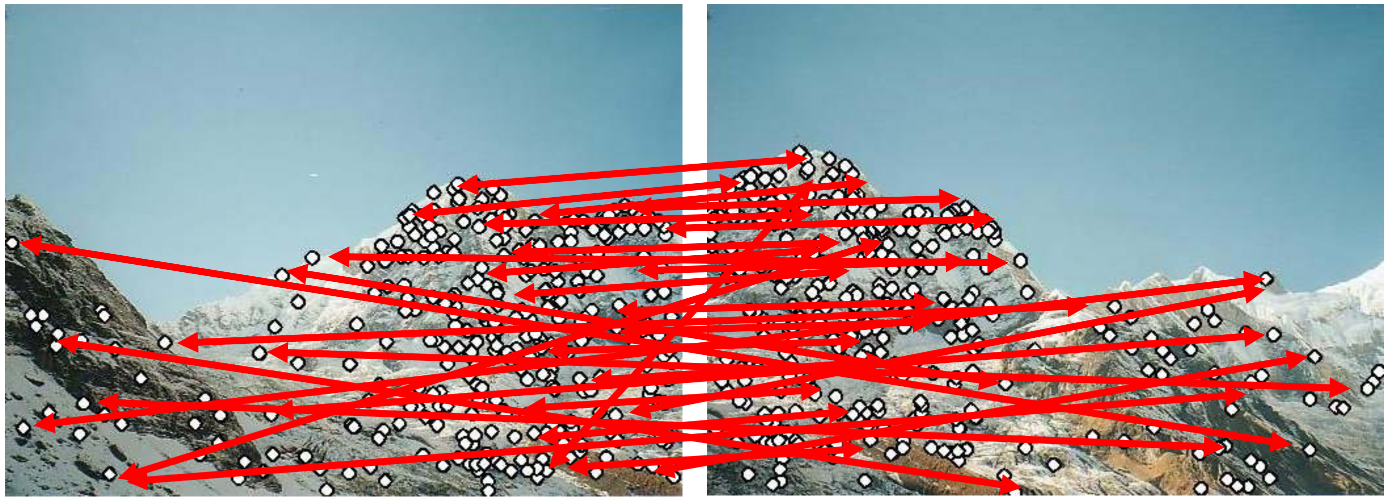


Detect interest points

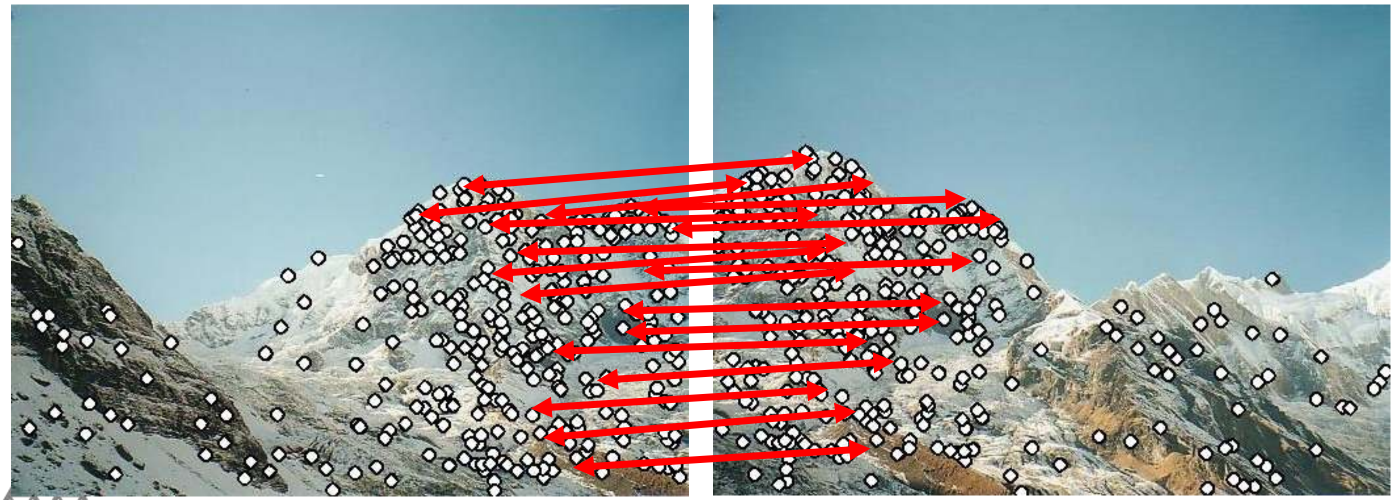


Match interest points

Image Mosaicing

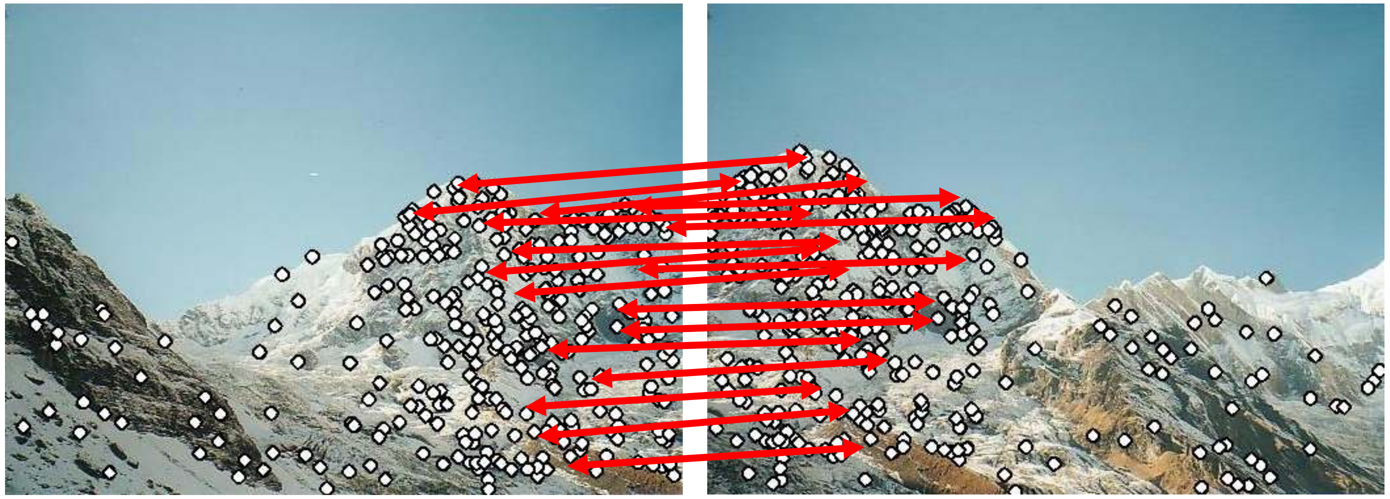


Match interest points

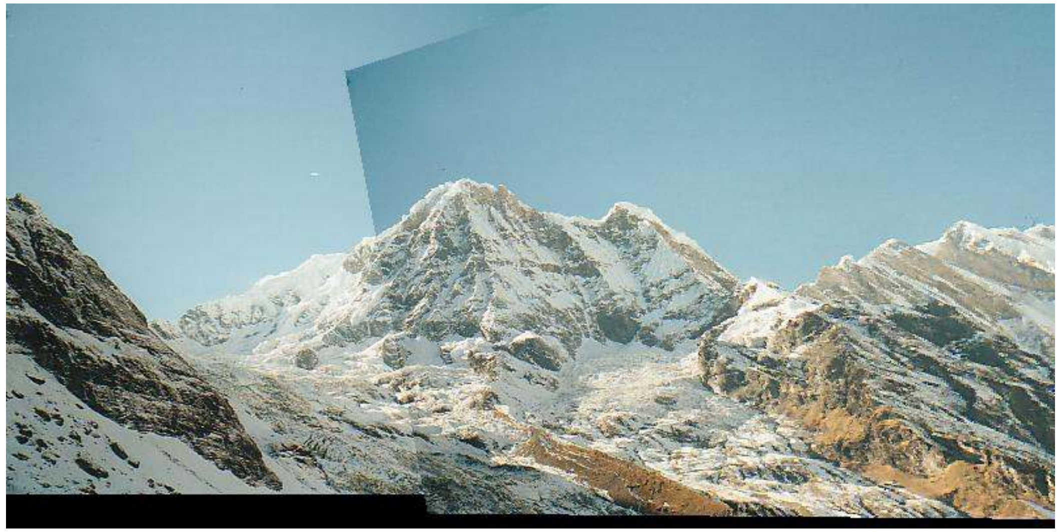


Identify correct matches

Image Mosaicing



Identify correct matches (using geometric constraints, Lecture #4)



Compute the (2D) motion between the 2 images using the correct matches

École des Ponts
ParisTech

Applications

3D object detection

Camera Localisation

Scenario:

1. we capture many images of a scene.
2. we use these images to reconstruct a 3D model of the scene. We also store the images in a database.

3. we capture a new image of the scene and we want to estimate where the camera is.

?

Camera Localisation

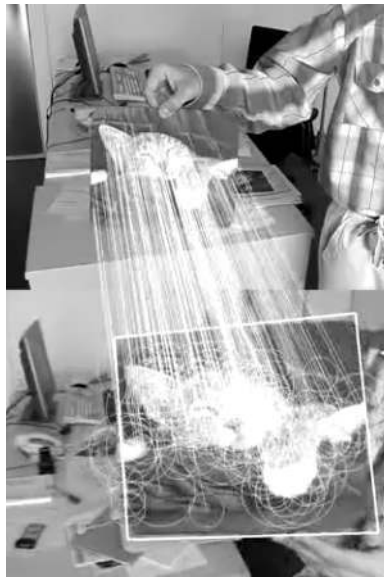
Idea:
we look for an image in the database similar to the captured image,
we match interest points to compute the position and orientation of the camera.

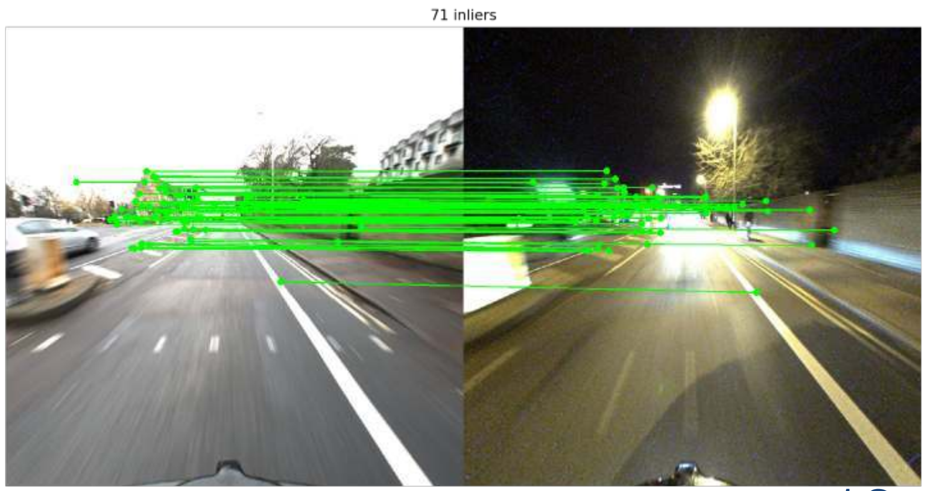
École des Ponts
ParisTech

Desired Properties & Challenges

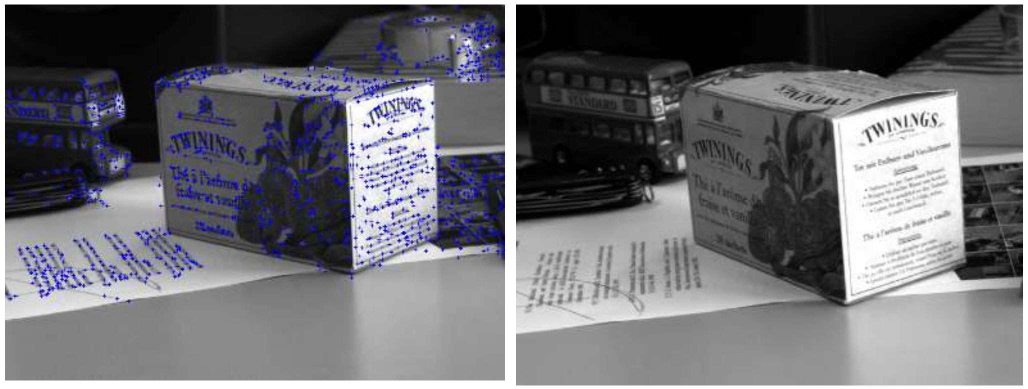
We want interest points that can be matched reliably:

1. We want to detect the same points in different images, even if the rotation, scale, perspective, lighting changed.
2. We want to correctly match the points, even if the rotation, scale, perspective, lighting changed.

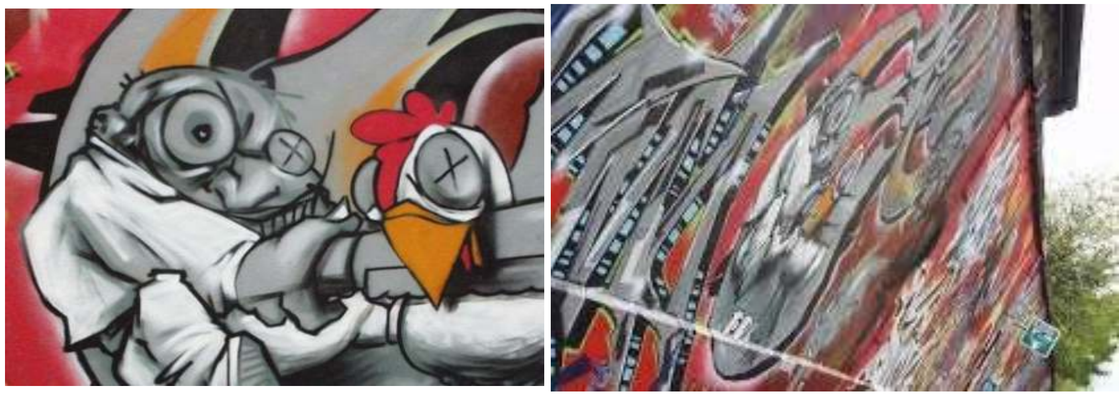




Different Expectations for Different Applications



Short Baseline



Wide Baseline

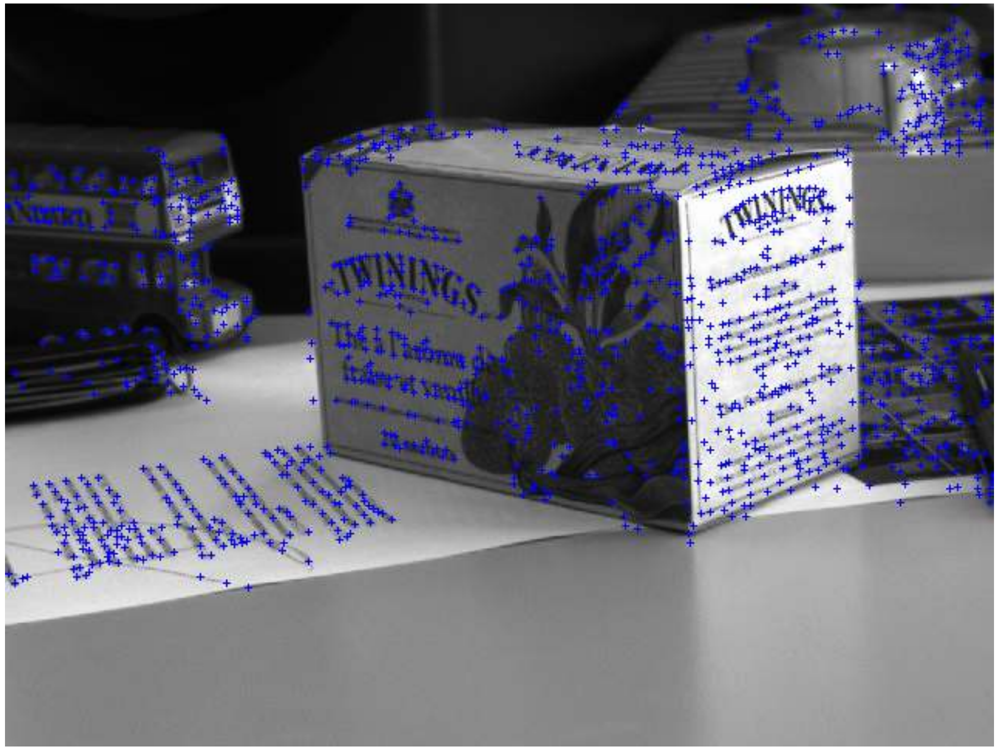
École des Ponts
ParisTech

Interest Points for Short Baseline Problems:

Detection and Matching

École des Ponts
ParisTech

Feature Point Extraction



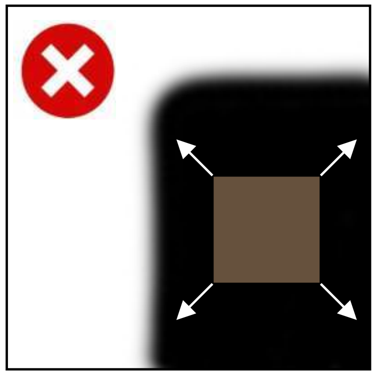
How can we extract the same physical points in the two images?

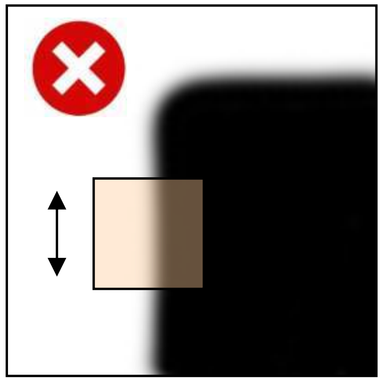
École des Ponts ParisTech

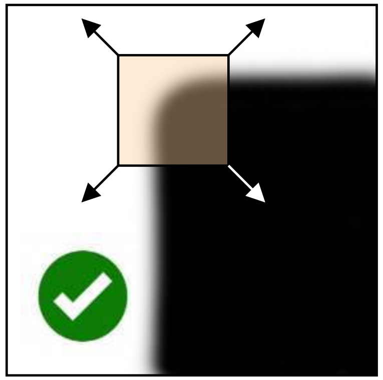
Harris Corners

Introduced by Harris and Stephens in 1988.

Basic assumption:
Shifting a local patch in *any direction* should give *a large change* in intensity.

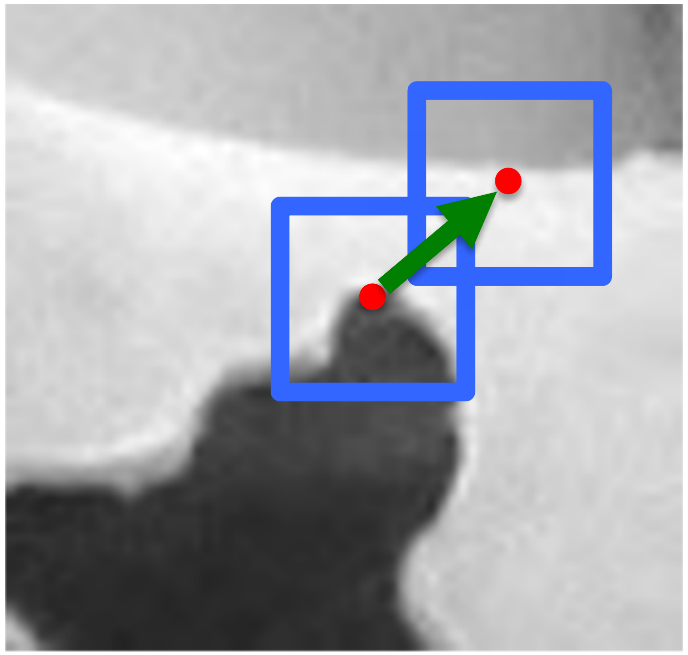






Formalisation

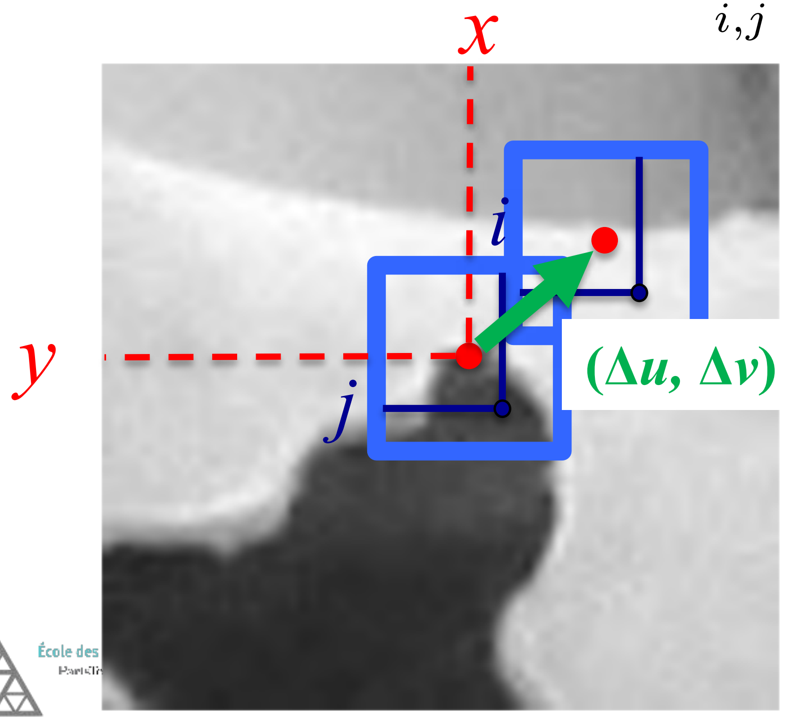
Assumption: shifting a local patch in *any direction* should give *a large change* in intensity



Formalisation

Assumption: shifting a local patch in *any direction* should give *a large change* in intensity

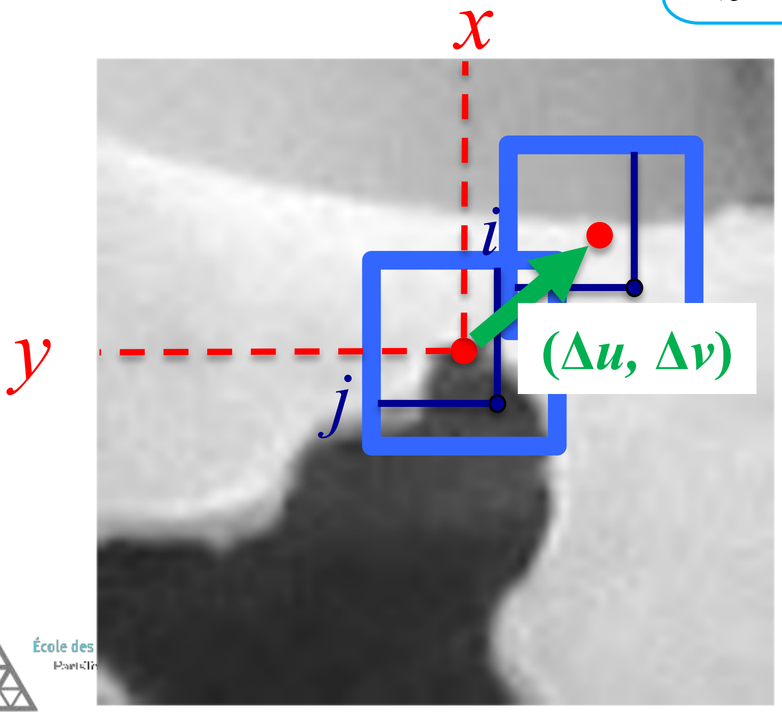
$$E(x, y; \Delta u, \Delta v) = \sum_{i,j} \left[I\left(x+i+\Delta u, y+j+\Delta v\right) - I\left(x+i, y+j\right) \right]^2$$



Formalisation

Assumption: shifting a local patch in *any direction* should give *a large change* in intensity

$$E(x,y;\Delta u,\Delta v) = \underbrace{\sum_{i,j} \left[I\left(x+i+\Delta u, y+j+\Delta v\right) - I\left(x+i, y+j\right)\right]^2}$$



Correlation between image patch centered on (x, y) and image patch centered on $(x+\Delta u, y+\Delta v)$

We are looking for (x, y) images locations such that the correlation

$$E(x, y; \Delta u, \Delta v)$$

is large for *all* directions $(\Delta u, \Delta v)$

Small Motion Assumption

Taylor Series expansion:

$$I(x+\Delta u, y+\Delta v) = I(x,y) + \frac{\partial I}{\partial x}(x,y)\Delta u + \frac{\partial I}{\partial y}(x,y)\Delta v + \text{higher order terms}$$

If the motion is small, then the higher order terms can be ignored:

$$I(x+\Delta u, y+\Delta v) \approx I(x,y) + \begin{bmatrix} \frac{\partial I}{\partial x}(x,y) & \frac{\partial I}{\partial y}(x,y) \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$$

$$E(x, y; \Delta u, \Delta v) \quad = \quad \sum_{i,j} \left(I\left(x{+}i{+}\Delta u, y{+}j{+}\Delta v\right) - I\left(x{+}i, y{+}j\right)\right)^2$$

$$\underbrace{\hphantom{xxxxxxxxxxxxxxxxxxxx}}_{Q\text{: “auto-correlation matrix”}}$$

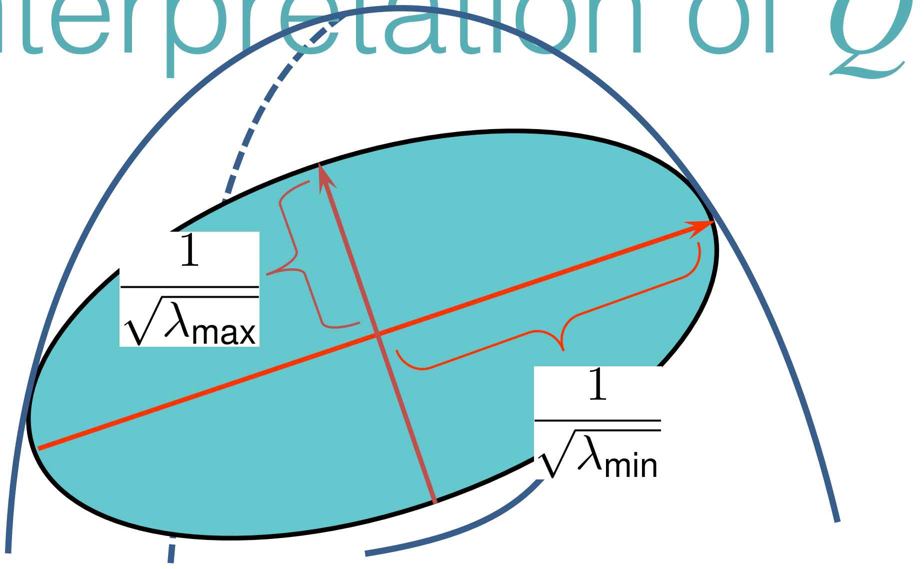
$$E(x,y;\Delta u,\Delta v) \approx \begin{bmatrix} \Delta u & \Delta v \end{bmatrix} \underbrace{\left(\sum_{i,j} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right)}_{Q} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$$

We are looking for (x, y) images locations such that $E(x, y; \Delta u, \Delta v)$ is large for all directions $[\Delta u, \Delta v]$

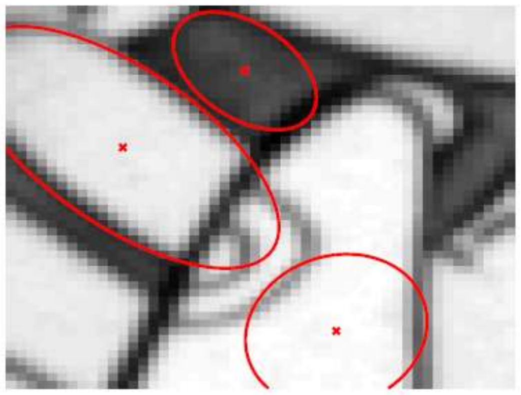
Eigenvalues $\lambda_{\max}$ and $\lambda_{\min}$ of Q reveal the amount of intensity change in the two principal orthogonal gradient directions within the patch

École des Ponts ParisTech

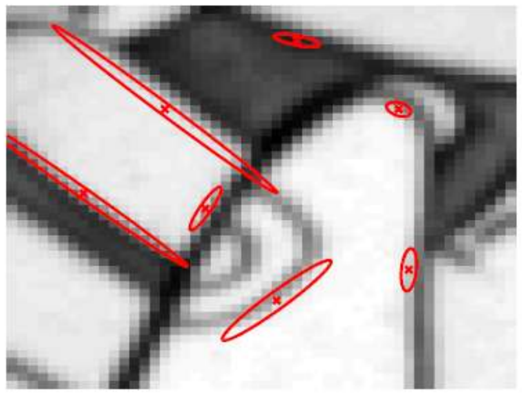
Geometric Interpretation of Q



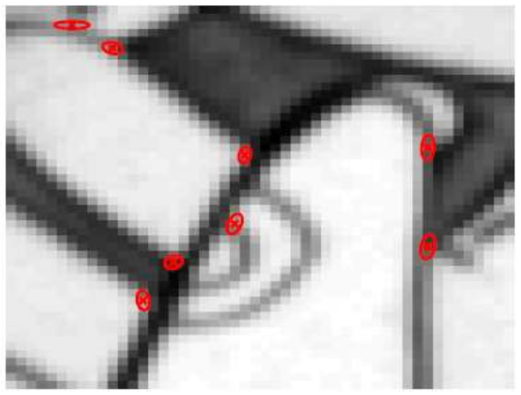
$\frac{1}{\sqrt{\lambda_{\max}}}$
$\frac{1}{\sqrt{\lambda_{\min}}}$



flat region
both eigenvalues small



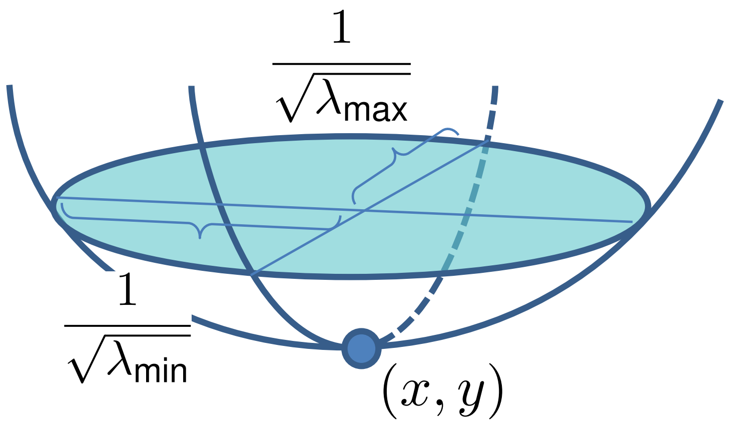
edge
one small, one large



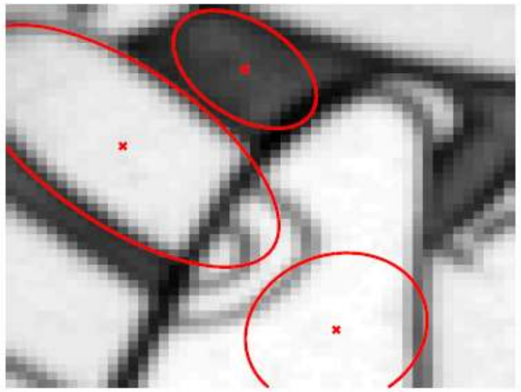
corner
both eigenvalues large

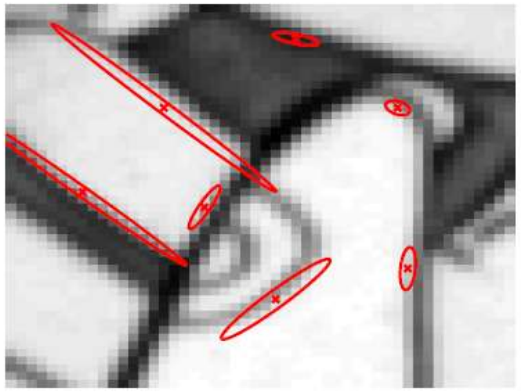
Geometric Interpretation of Q

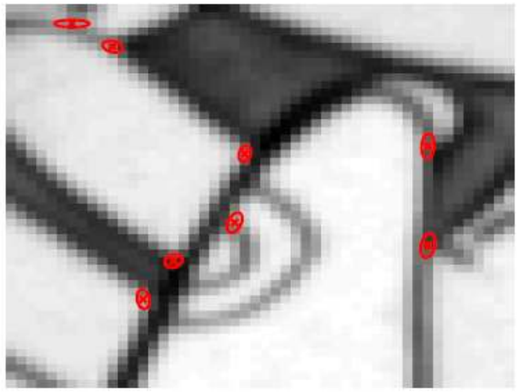
$$E(x,y;\Delta u,\Delta v) = \sum_{i,j} \left[I\left(x{+}i{+}\Delta u, y{+}j{+}\Delta v\right) - I\left(x{+}i, y{+}j\right)\right]^2 \approx \begin{bmatrix} \Delta u & \Delta v \end{bmatrix} Q \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$$



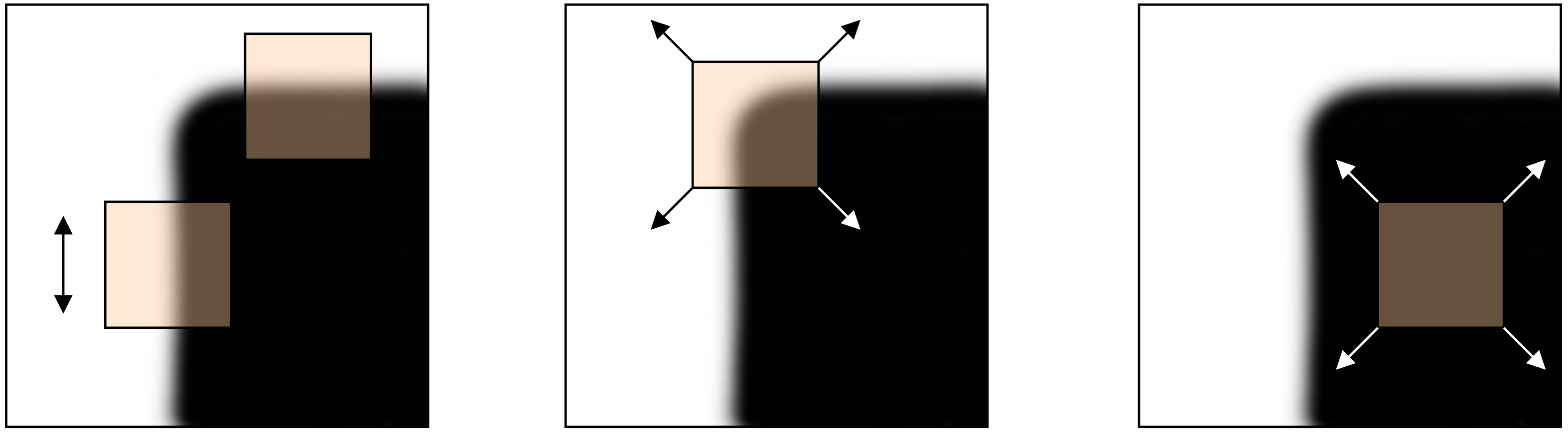
$\frac{1}{\sqrt{\lambda_{\max}}}$
$\frac{1}{\sqrt{\lambda_{\min}}}$
(x,y)


flat region
both eigenvalues small


edge
one small, one large


corner
both eigenvalues large

Recall: Corners as distinctive interest points



"edge":
$\lambda_1 >> \lambda_2$ or
$\lambda_2 >> \lambda_1$

"corner":
λ_1 and λ_2 are large,
$\lambda_1 \sim \lambda_2$

"flat" region
λ_1 and λ_2 are small

Ways to measure the "cornerness": $\min(\lambda_1, \lambda_2)$, or $\dfrac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \dfrac{|Q|}{\operatorname{tr} Q}$

École des Ponts ParisTech

How to Compute Image Gradients?

Approximation by finite differences: $\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$

Convolution $[+1 \ \ -1] * I$ will shift the derivative by ½ pixel.

Convolution $[+1 \ 0 \ \ -1] * I$ is better.

Convolving the image with a Gaussian kernel will make the computation more robust to noise, and the full operation can be seen as the convolution of the image by the derivative of the Gaussian kernel:

$$[+1 \ 0 \ \ -1] * (G_\sigma * I) = ([+1 \ 0 \ \ -1] * G_\sigma) * I$$

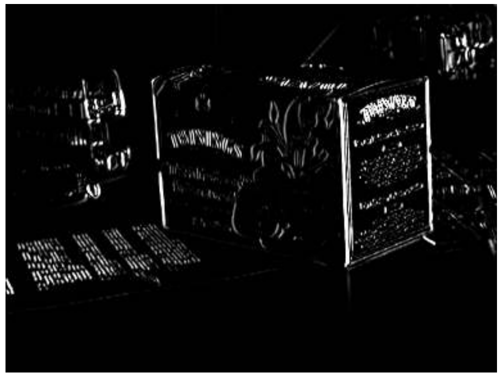
The filter

+1	0	-1
+2	0	-2
+1	0	-1

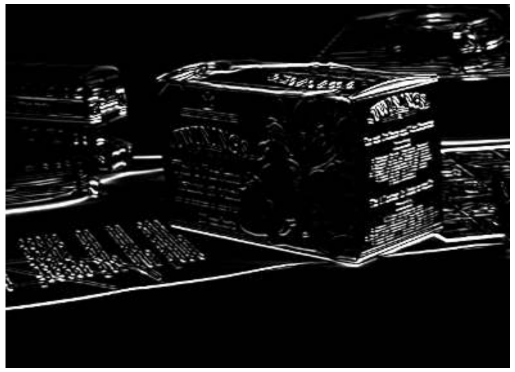
from Lecture #1 is a good approximation of the derivative of a Gaussian kernel.

École des Ponts ParisTech

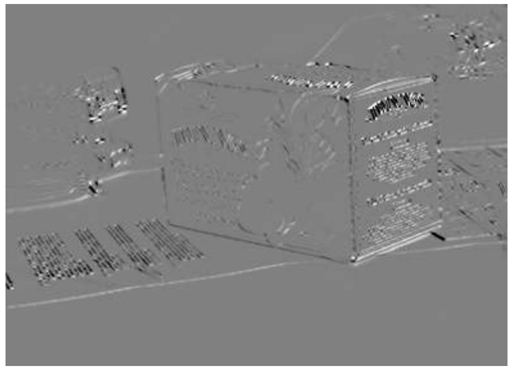
$$Q = \begin{bmatrix} \sum_{i,j} I_x^2 & \sum_{i,j} I_x I_y \\ \sum_{i,j} I_x I_y & \sum_{i,j} I_y^2 \end{bmatrix}$$



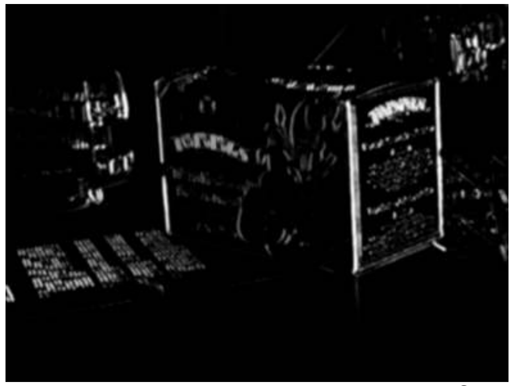
for each pixel: $\left(\frac{\partial I}{\partial x}\right)^2$



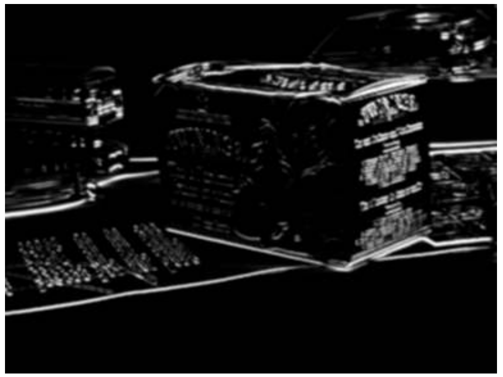
$\left(\frac{\partial I}{\partial y}\right)^2$



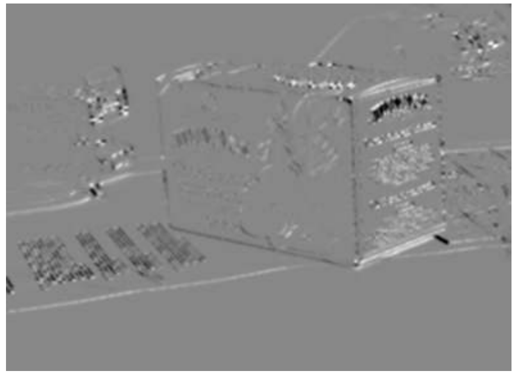
$\left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right)$



for each pixel: $\sum_{i,j} I_x^2$ or $G_\sigma * \left(\frac{\partial I}{\partial x}\right)^2$



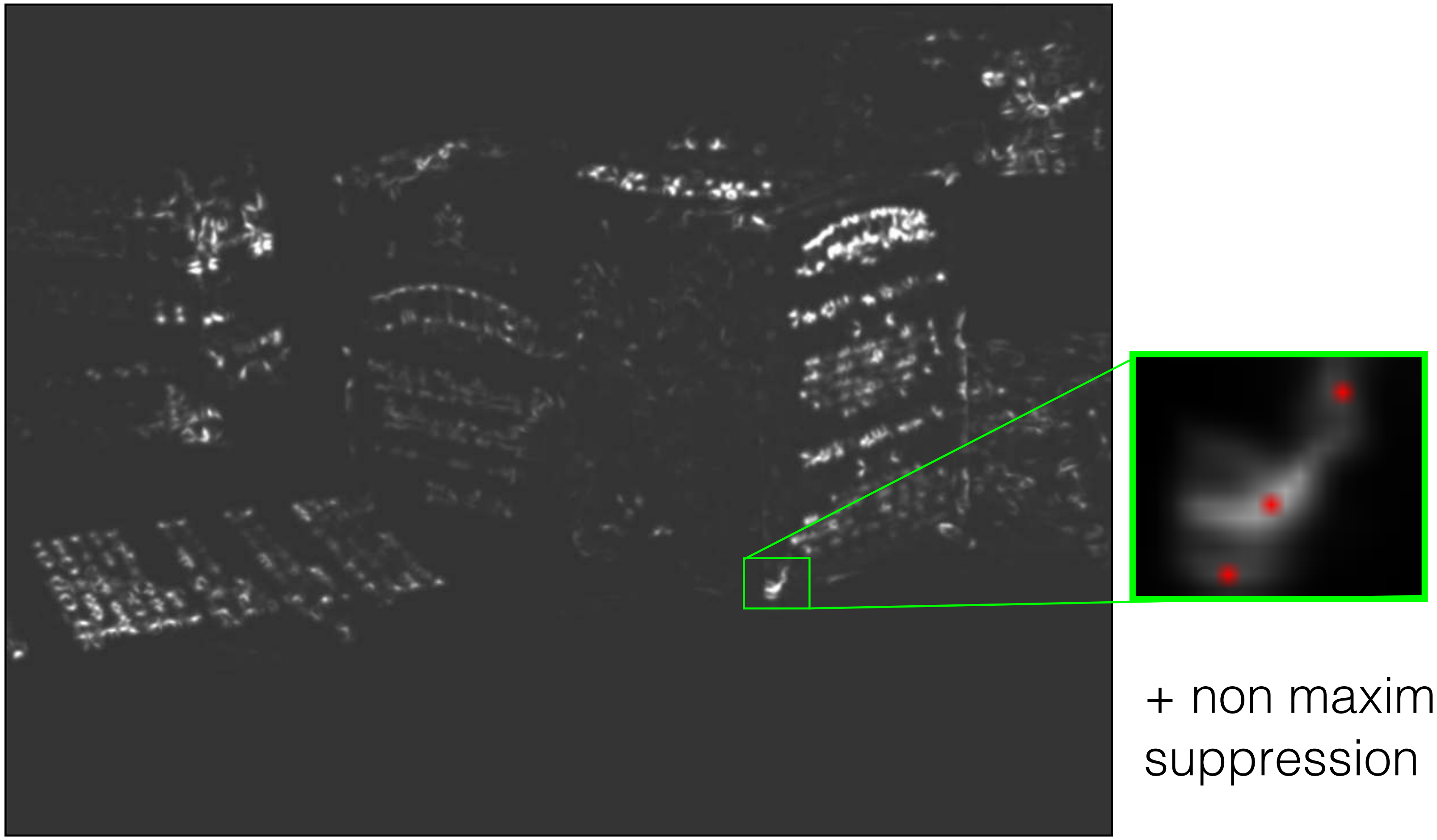
$\sum_{i,j} I_y^2$ or $G_\sigma * \left(\frac{\partial I}{\partial y}\right)^2$



$\sum_{i,j} I_x I_y$ or $G_\sigma * \left(\left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right)\right)$

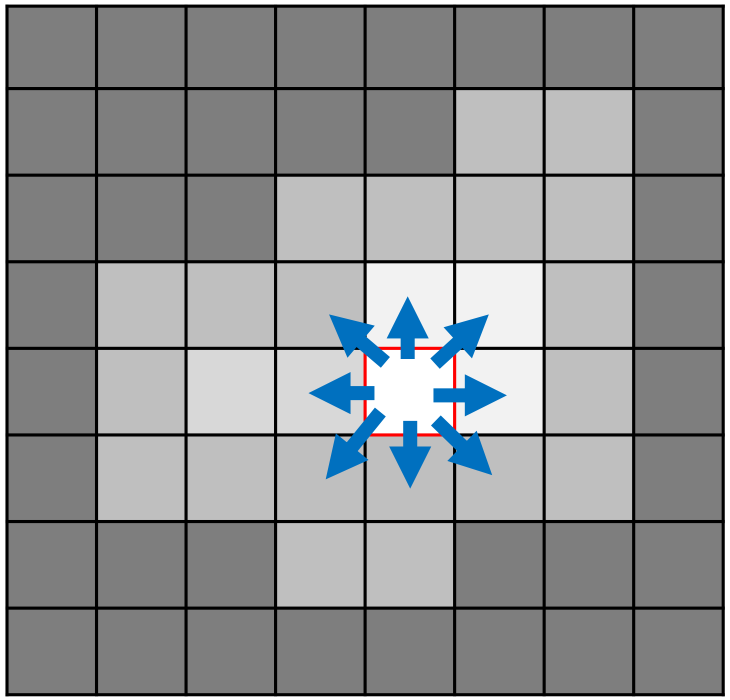
École des Ponts
ParisTech

$\min(\lambda_1, \lambda_2)$



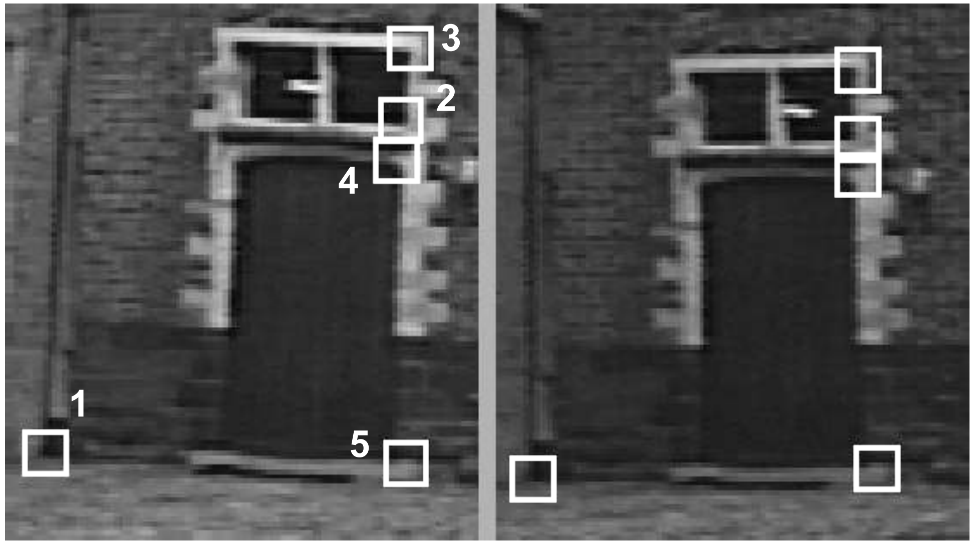
+ non maximum suppression

Non Maximum Suppression



keep only the pixels with cornerness larger than their 3x3 (or 5x5) neighbors as interest points

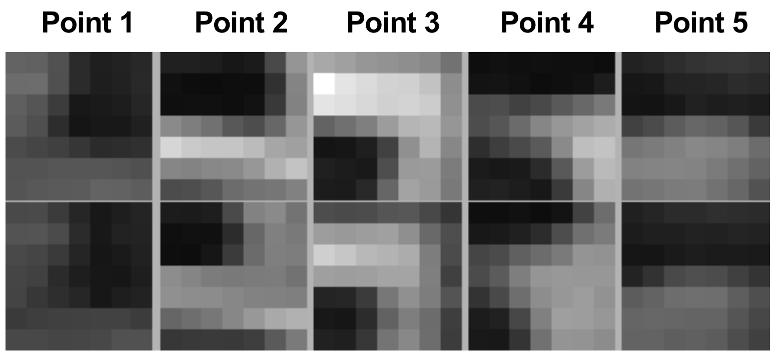
Interest Point Matching (short baseline)

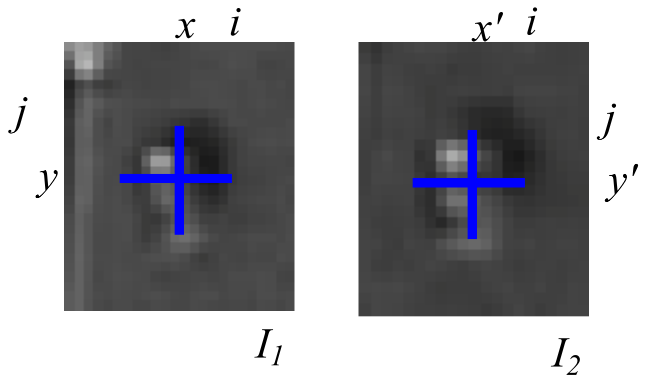


Possible correlation measures:

$$C = \sum_{i,j} \left(I_1(x+i, y+j) - I_2(x'+i, y'+j)\right)^2$$

$$C = \sum_{i,j} \frac{\left(I_1(x+i, y+j) - \overline{I_1}\right)\left(I_2(x'+i, y'+j) - \overline{I_2}\right)}{\sqrt{\sigma_1 \sigma_2}}$$





École des Ponts
ParisTech

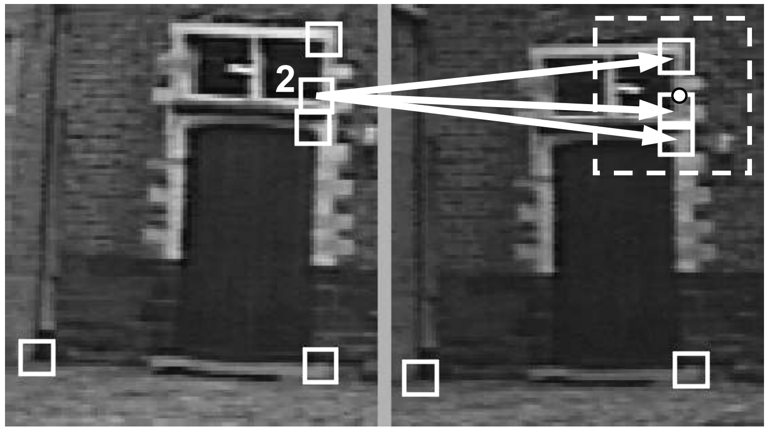
Cross-Correlation Measure

$$C = \sum_{i,j} \frac{\left(I_1(x+i, y+j) - \overline{I_1}\right)\left(I_2(x'+i, y'+j) - \overline{I_2}\right)}{\sqrt{\sigma_1 \sigma_2}}$$

- Invariant to affine changes of the lighting; [Why?]
- Between -1 (completly different patches) and +1 (equal patches); [Why?]

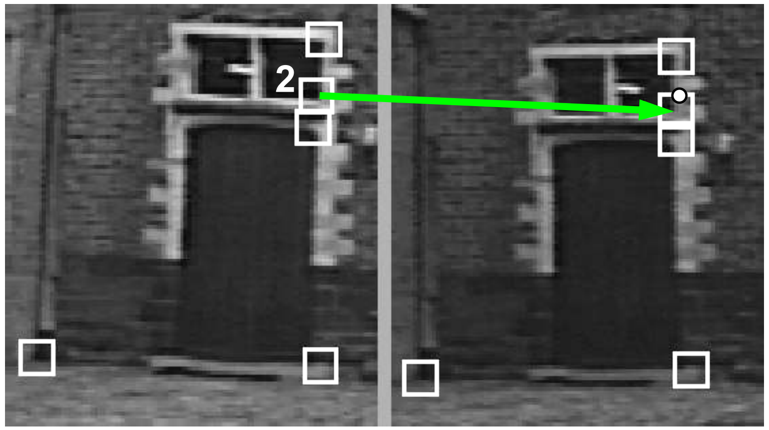
In practice: accept matches when $C > 0.8$

Matching



1. For each point, search for the correspondent that maximizes the correlation. Limit search to a Region of Interest centered on the point.

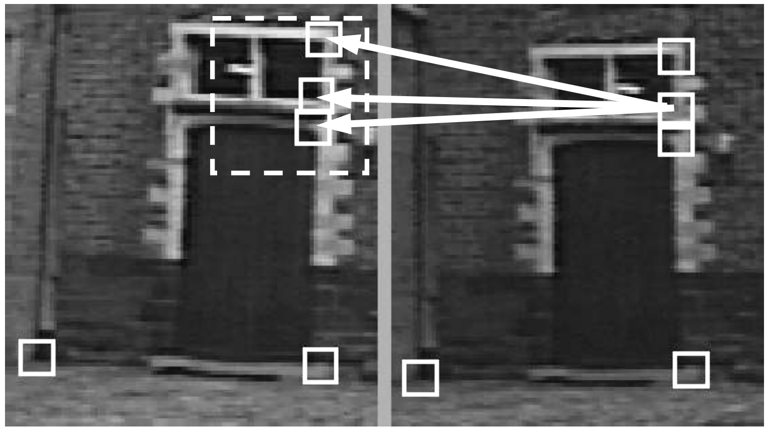
Feature Point Matching



1. For each point, search for the correspondent that maximizes the correlation. Search limited to a Region of Interest.

Keep the best correspondent according to the correlation.

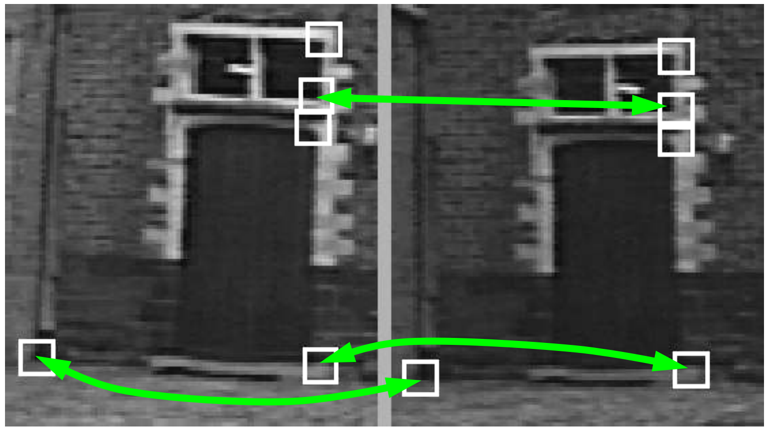
Feature Point Matching



1. For each point, search for the correspondent that maximizes the correlation. Search limited to a Region of Interest.

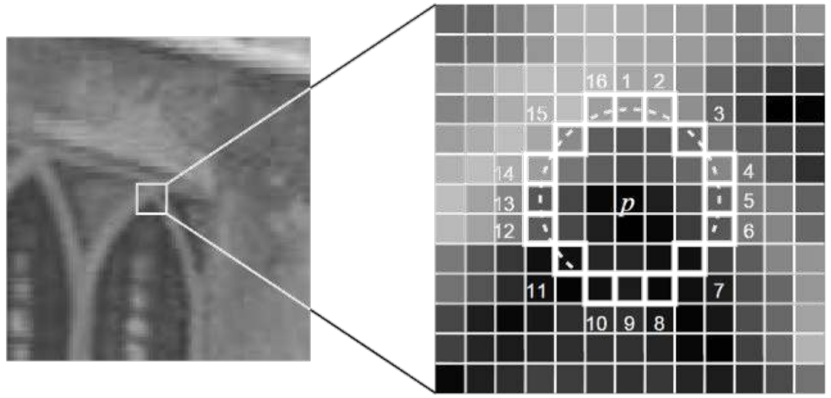
2. Reverse the role of the images.

Feature Point Matching



Keep the points that choose each other.

FAST



Rosten, Edward; Drummond, Tom. "Machine Learning for High-speed Corner Detection". *ECCV 2006*.

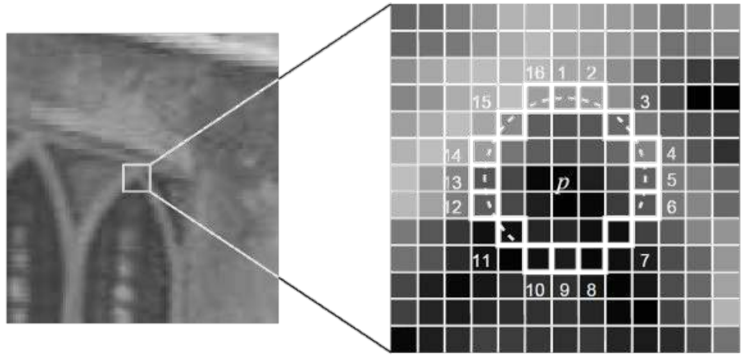
FAST

- Relies on tests

  $$I(\text{center}) > I_k + t$$

  or

  $$I(\text{center}) < I_k - t$$

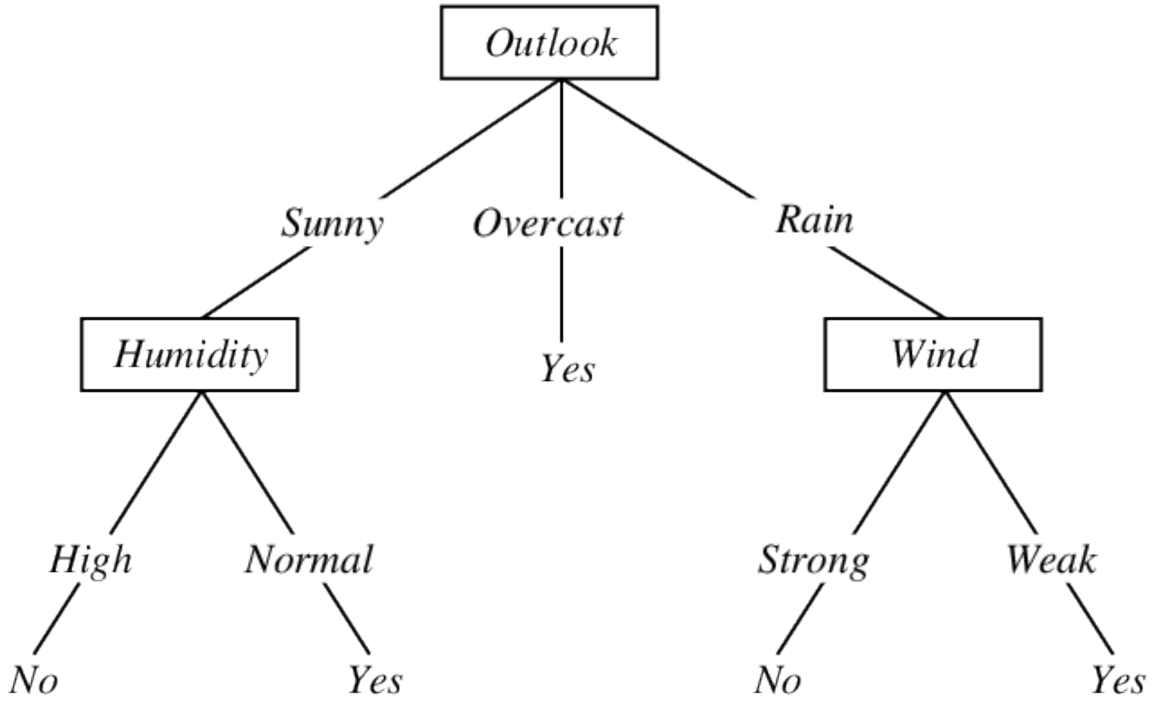


- Create a training set of corners and non-corners using heuristics;
- Use machine learning to find the tests that reject the pixels that are non-corners as fast as possible.

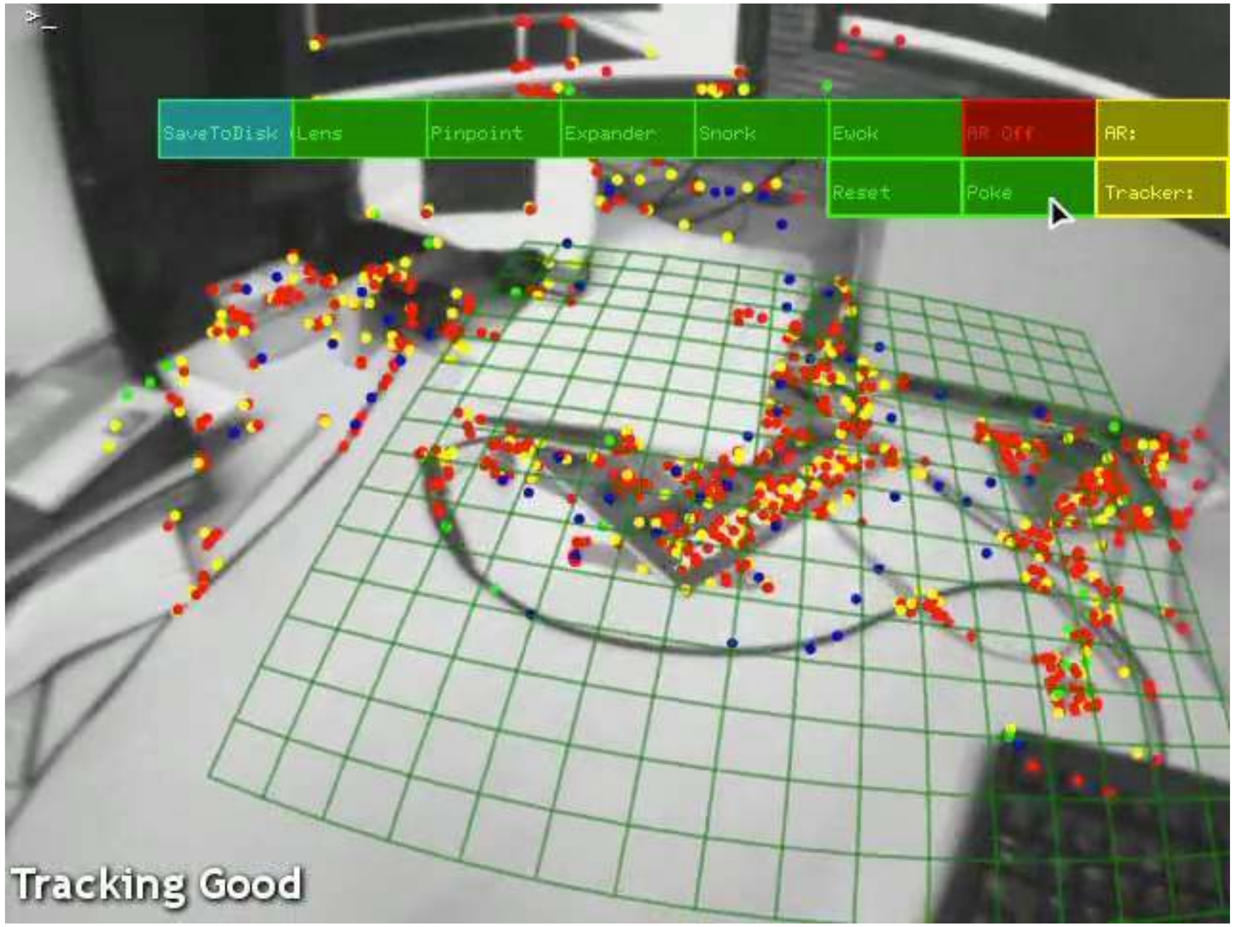
Learning

Decision Tree (ID3) to learn to reject non-corners quickly:

Play
Tennis?



FAST


SaveToDisk
Lens
Pinpoint
Expander
Snork
Ewok
AR Off
AR:
Reset
Poke
Tracker:
Tracking Good

Matching Interest Points based on their “Descriptors”

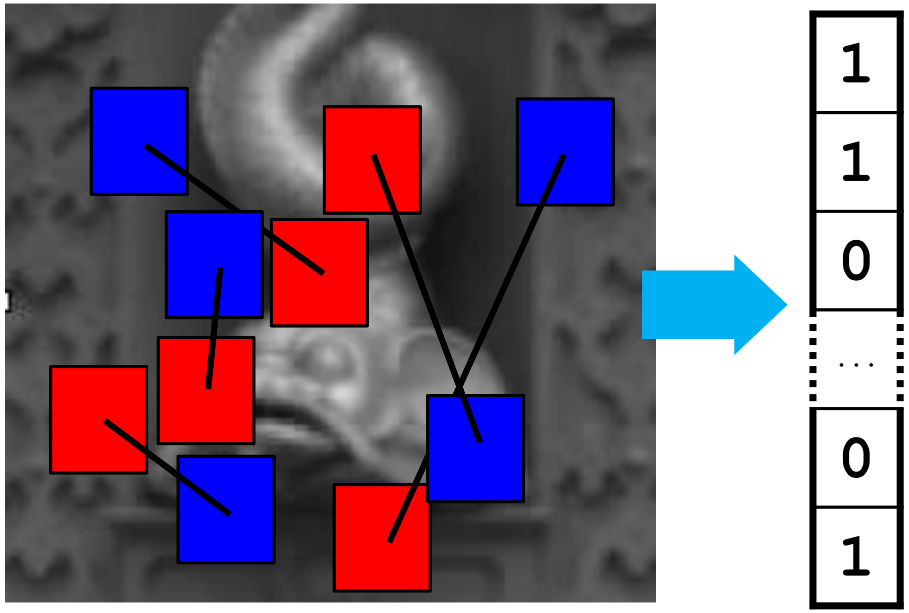
Correlation is the (squared) Euclidean distance between the vector of image intensities around the first point and the vector of image intensities around the second point:

$$\begin{aligned} C &= \textstyle\sum_{i,j} \left(I_1(x+i, y+j) - I_2(x'+i, y'+j)\right)^2 \\ &= \left\| \underbrace{\begin{bmatrix} I_1(x - .., y - ..) \\ \vdots \\ I_1(x + .., y + ..) \end{bmatrix}} - \underbrace{\begin{bmatrix} I_2(x' - .., y' - ..) \\ \vdots \\ I_2(x' + .., y' + ..) \end{bmatrix}} \right\|^2 \end{aligned}$$

description vector or “descriptor”

Can we imagine better vectors / better similarity measures between vectors?

BRIEF Descriptor

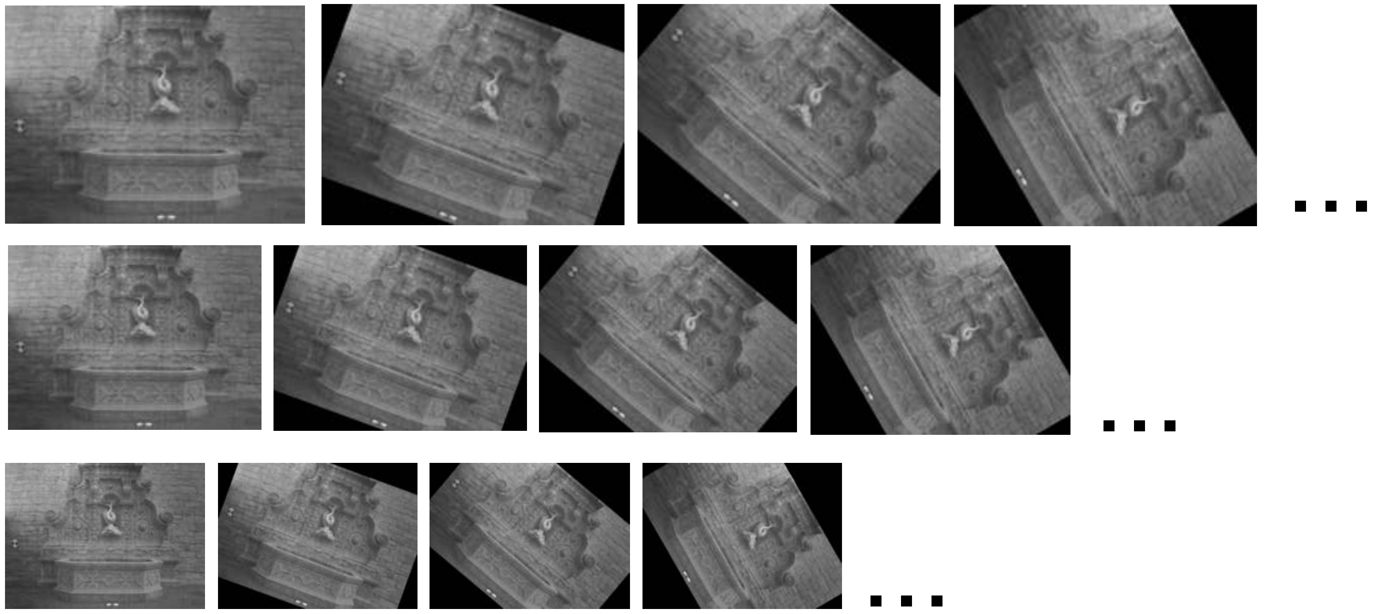


$$= \begin{cases} 1 & \text{if } I(x+i_{1,1}, y+j_{1,1}) > I(x+i_{1,2}, y+j_{1,2}) \\ 0 & \text{otherwise} \end{cases}$$

BRIEF descriptor

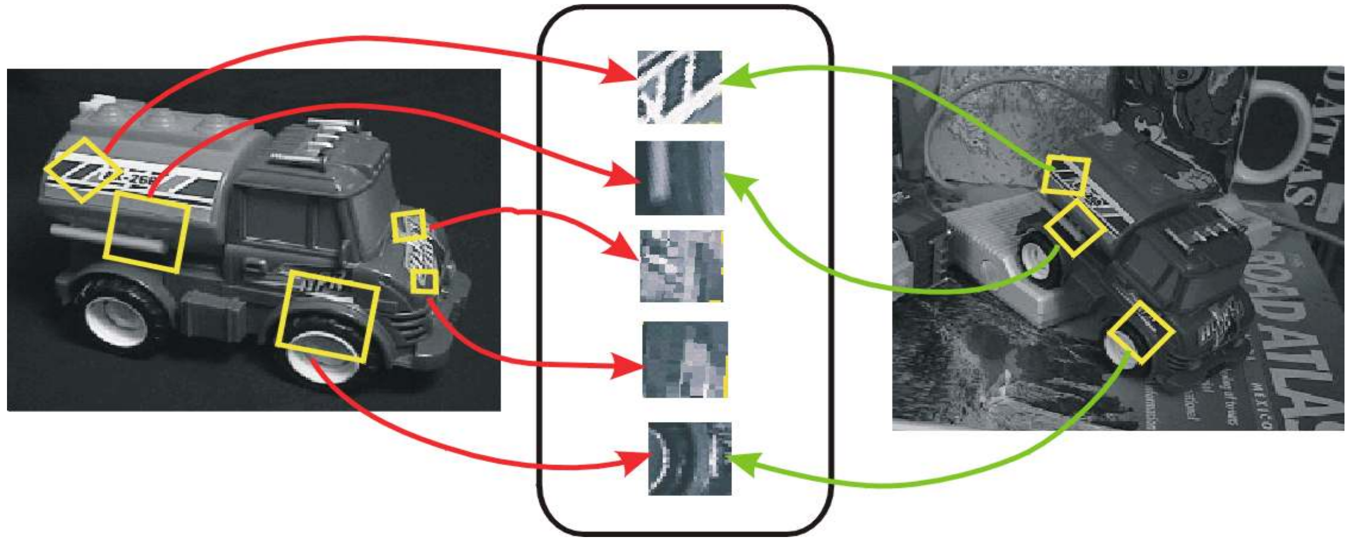
- very robust to light changes;
- computing distance between descriptors is very fast (Hamming distance).

Duplicate the Descriptors: 18 rotations x 3 scales



École des Ponts
ParisTech

Interest Points for Wide Baseline Problems

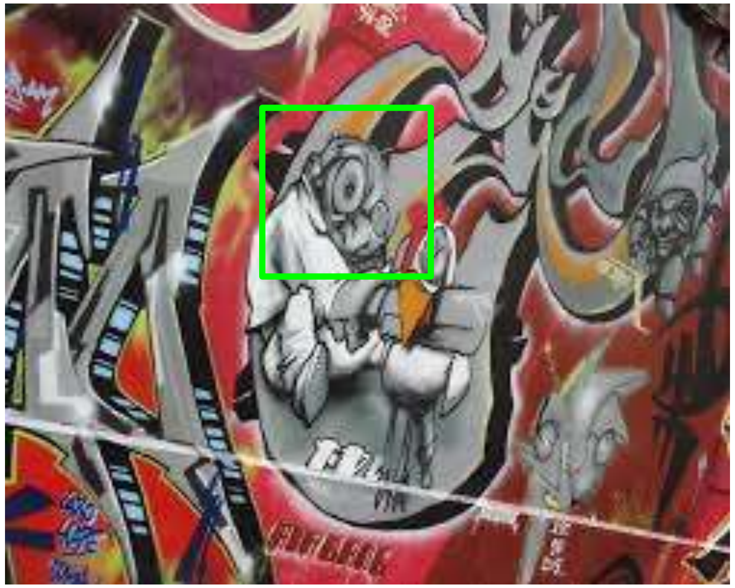


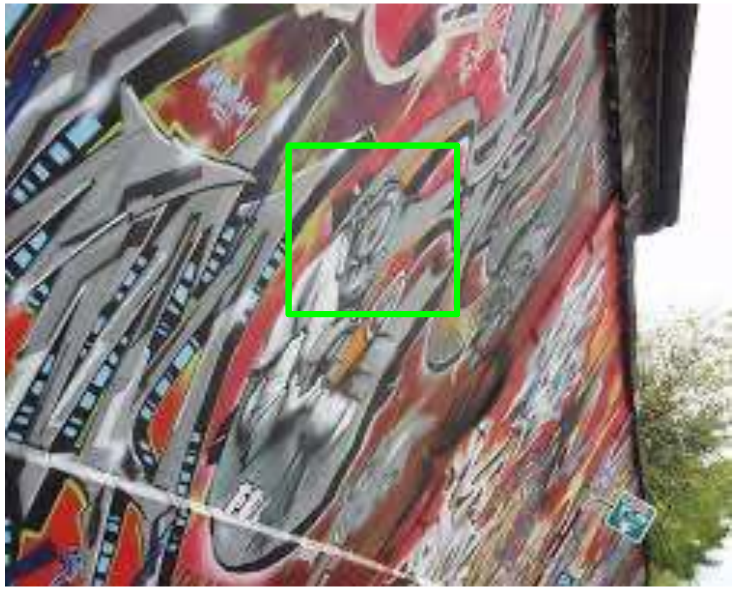
We need

- to detect the same points in the two images, and
- to compute the same descriptors for corresponding points (correlation of image intensities does not work here);

despite scale, rotation, perspective, light changes.

Invariance

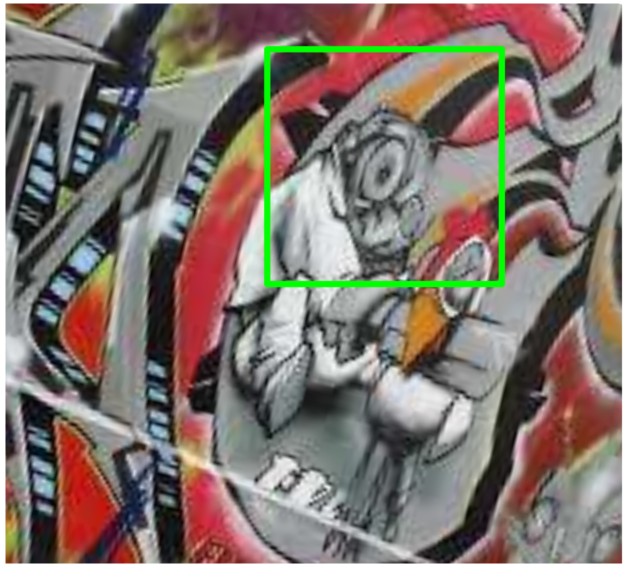




We want the descriptors to be (ideally) *invariant* to scale, rotation, light, perspective changes:

descriptors(transform(image)) = descriptors(image)

Invariance



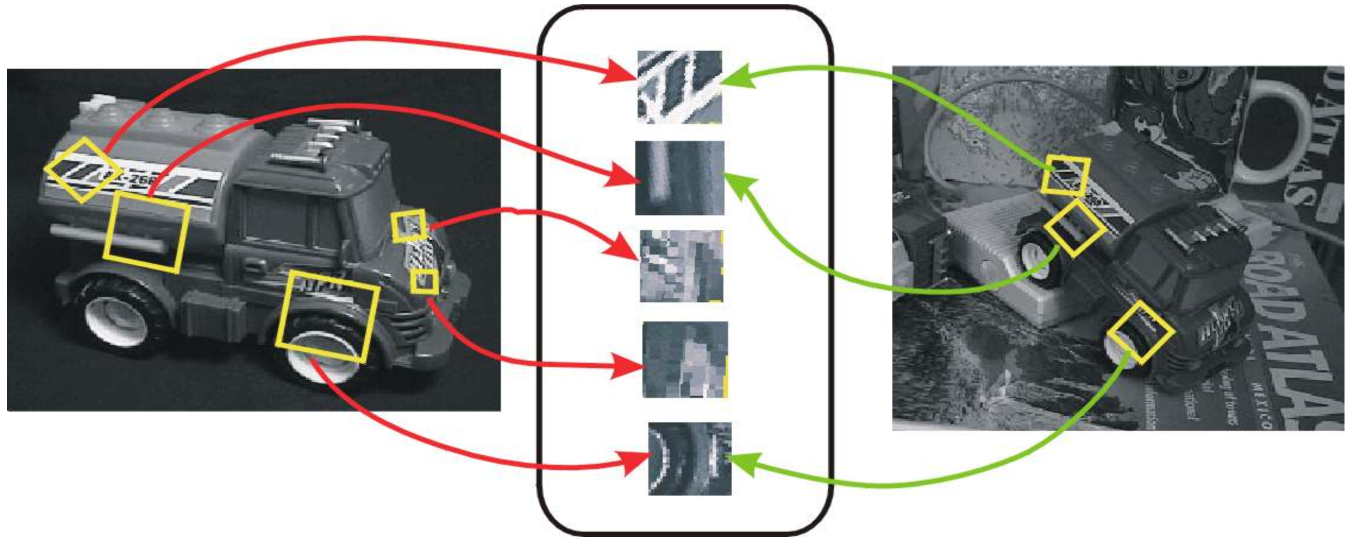
It is possible to have descriptors that are *directly* invariant to scale and rotation:

descriptors(scale+rotation(image)) = descriptors(image)

to some extent.

However, the descriptors then become less discriminant.

Covariance / Invariance



We want the descriptors to be (ideally) *invariant* to scale, rotation, light, perspective changes:

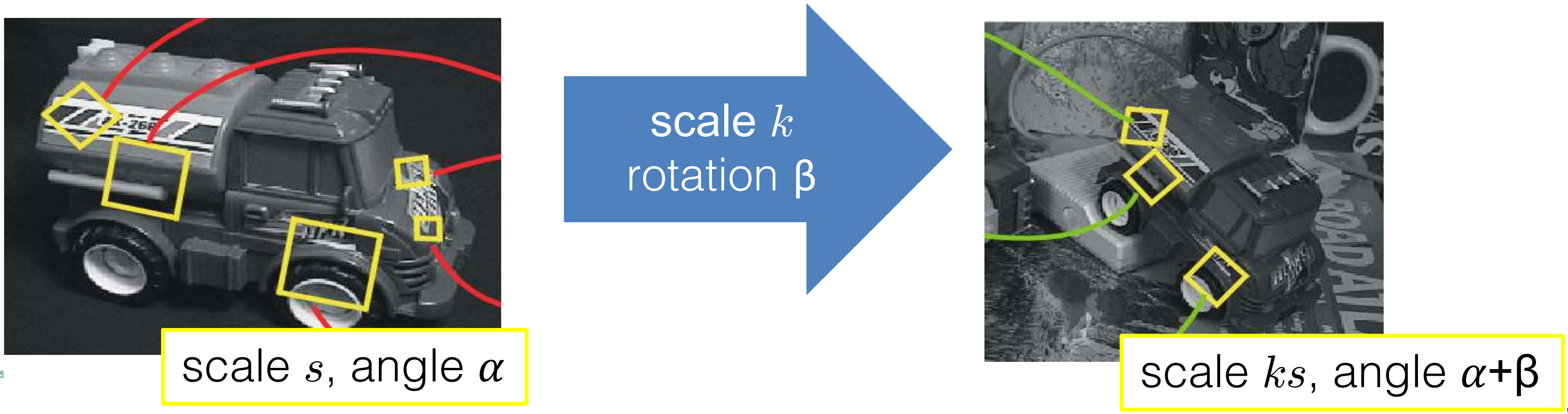
descriptors(transform(image)) = descriptors(image)

A better solution is to use a point detector *covariant* with the image transformation

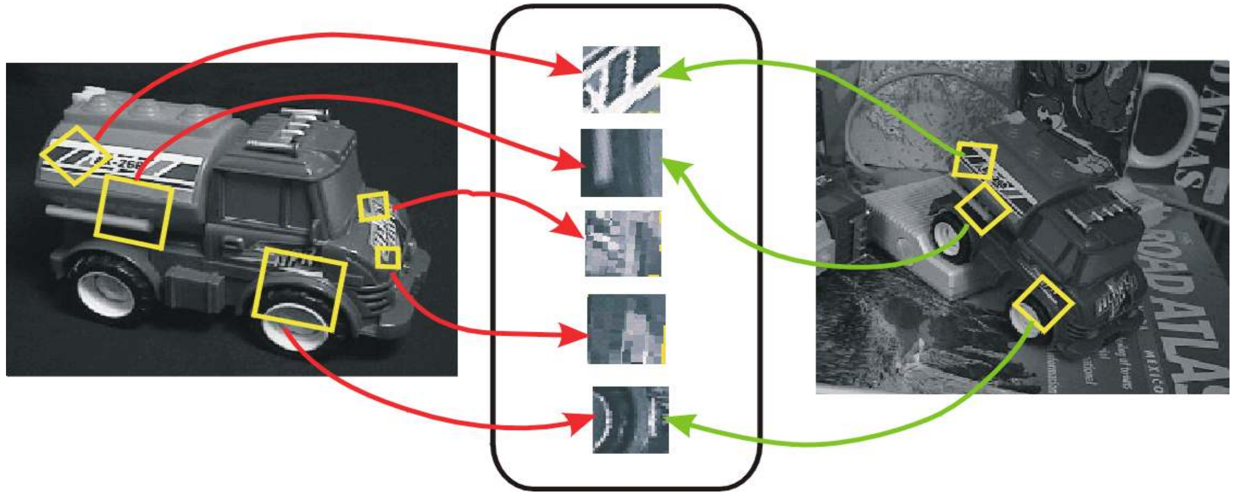
A Rotation and Scale Covariant Point Detector

Point detector covariant to scale and rotation:

- the detector provides a scale and an angle for each keypoint;
- the detector should detect the same points even if the image undergoes a scale and/or a rotation, and
- the scale and angle provided should change with the transformation applied to the image:



SIFT



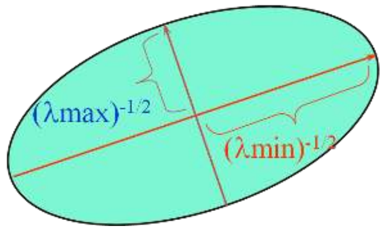
- Scale and rotation covariant detector;
- Descriptor is invariant to affine transformation of image intensities, and more;
- Descriptor is also robust to perspective changes (ie not perfectly invariant to perspective changes).

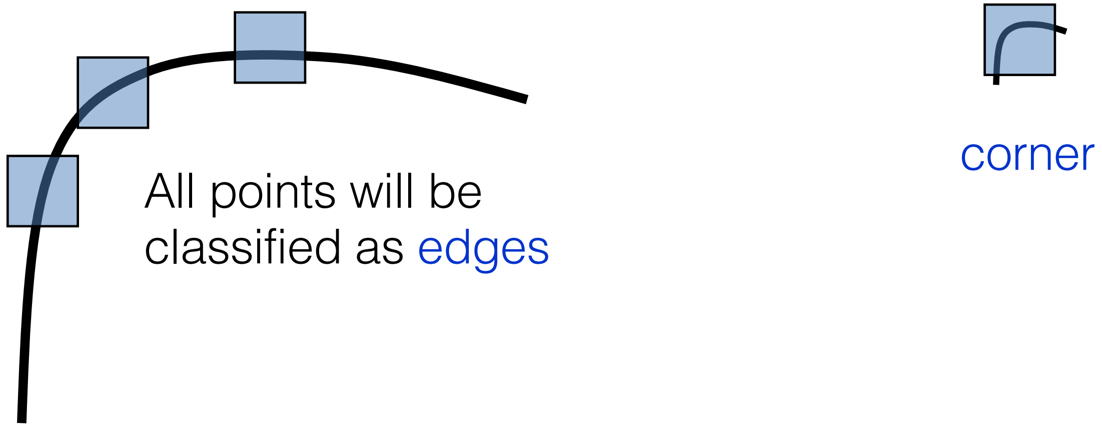
École des Ponts
ParisTech

Scale Covariant Detector

The Harris detector is invariant to rotation, but not to scale:

- "Cornerness" based on the eigenvalues of the auto-correlation matrix → the points detected after rotating the image will be the same.
- Not scale invariant → the points detected after scaling the image will NOT be the same.

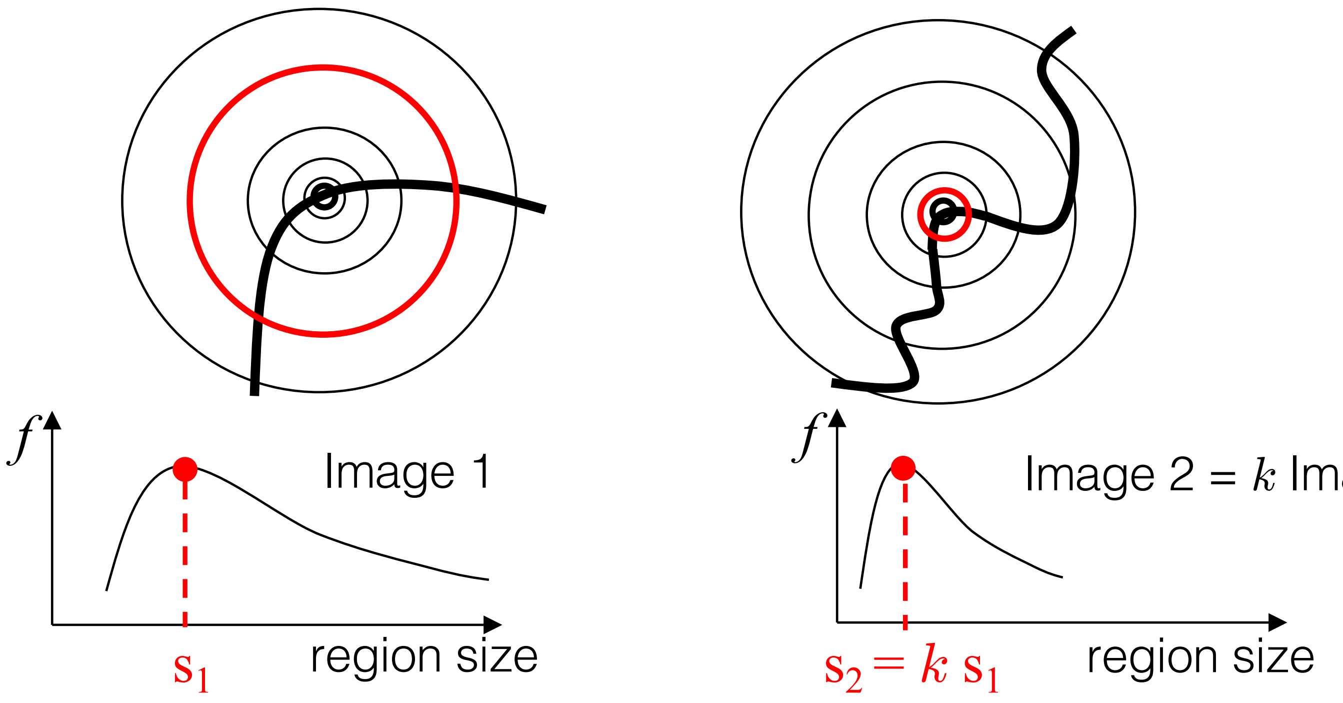




Scale Covariant Detector

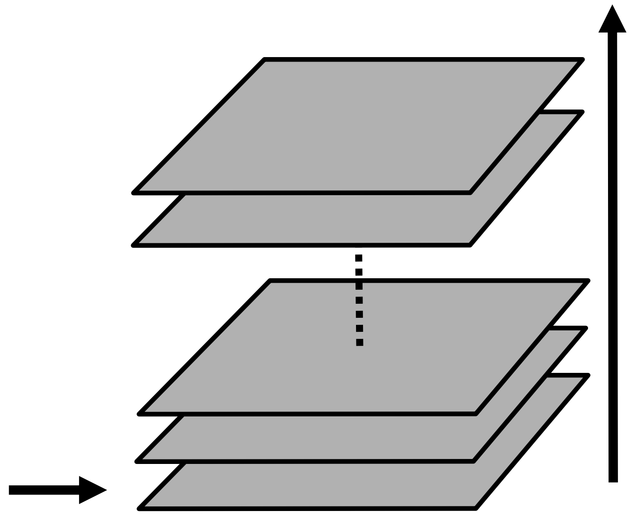
Idea:

Find a function *f* of image position and scale that gives local maxima in both position and scale. The scales of the local maxima should be consistent with the scale changes.



Scale-Space Theory

Original image



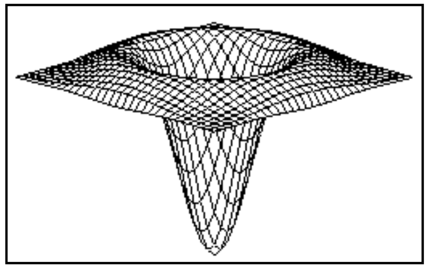
Successive convolutions with a Gaussian filter or Gaussian derivative filter while increasing σ

[Lindeberg 9*]

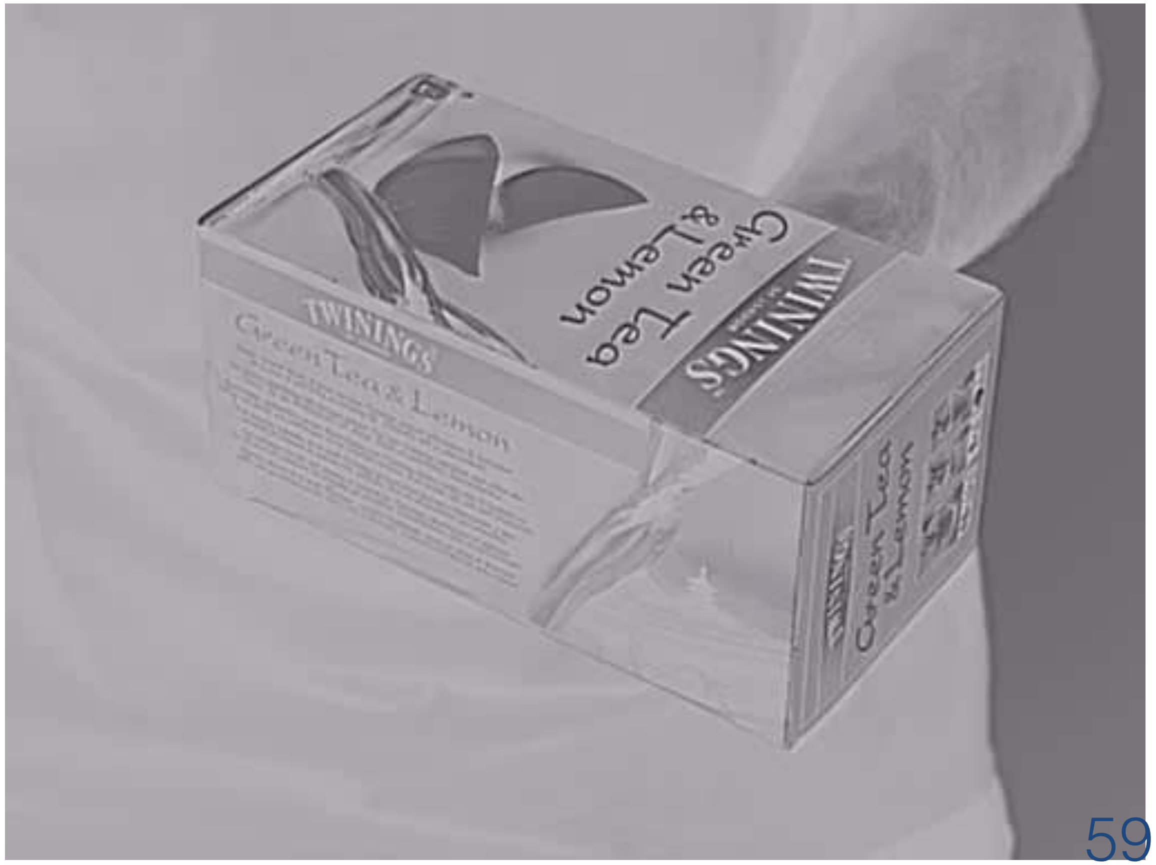
École des Ponts
ParisTech

Laplacian of Gaussian ($G_{xx} + G_{yy}$): scale = σ of the Gaussian kernels

convolution by Laplacian of Gaussian, increasing σ:



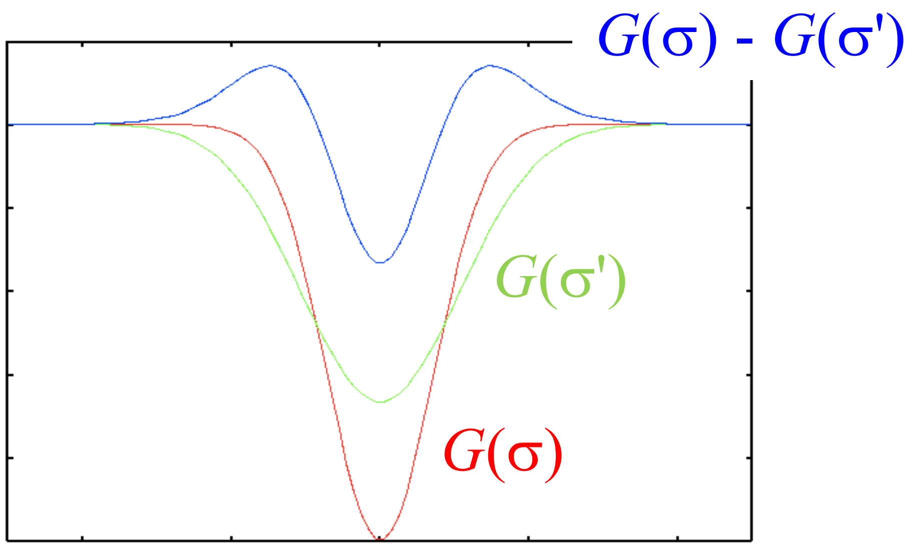
Laplacian operator

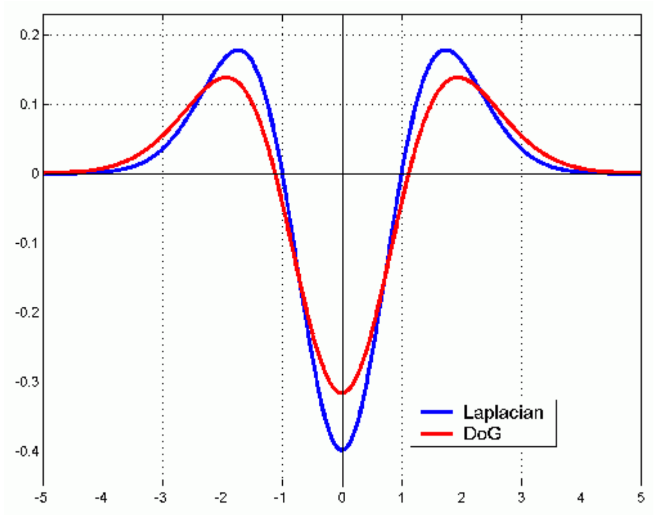


Fast Approximation of the Laplacian of Gaussian

Convolution with Laplacian of Gaussian is slow.

the Laplacian of Gaussian can be approximated by the difference of two Gaussians:

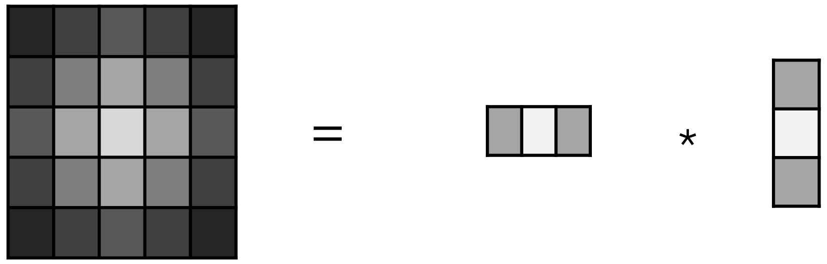




Properties of the Gaussian Convolution

1. The Gaussian kernel is separable:

A 2D Gaussian kernel is equal to the convolution of a 1D horizontal Gaussian kernel and a 1D vertical Gaussian kernel:



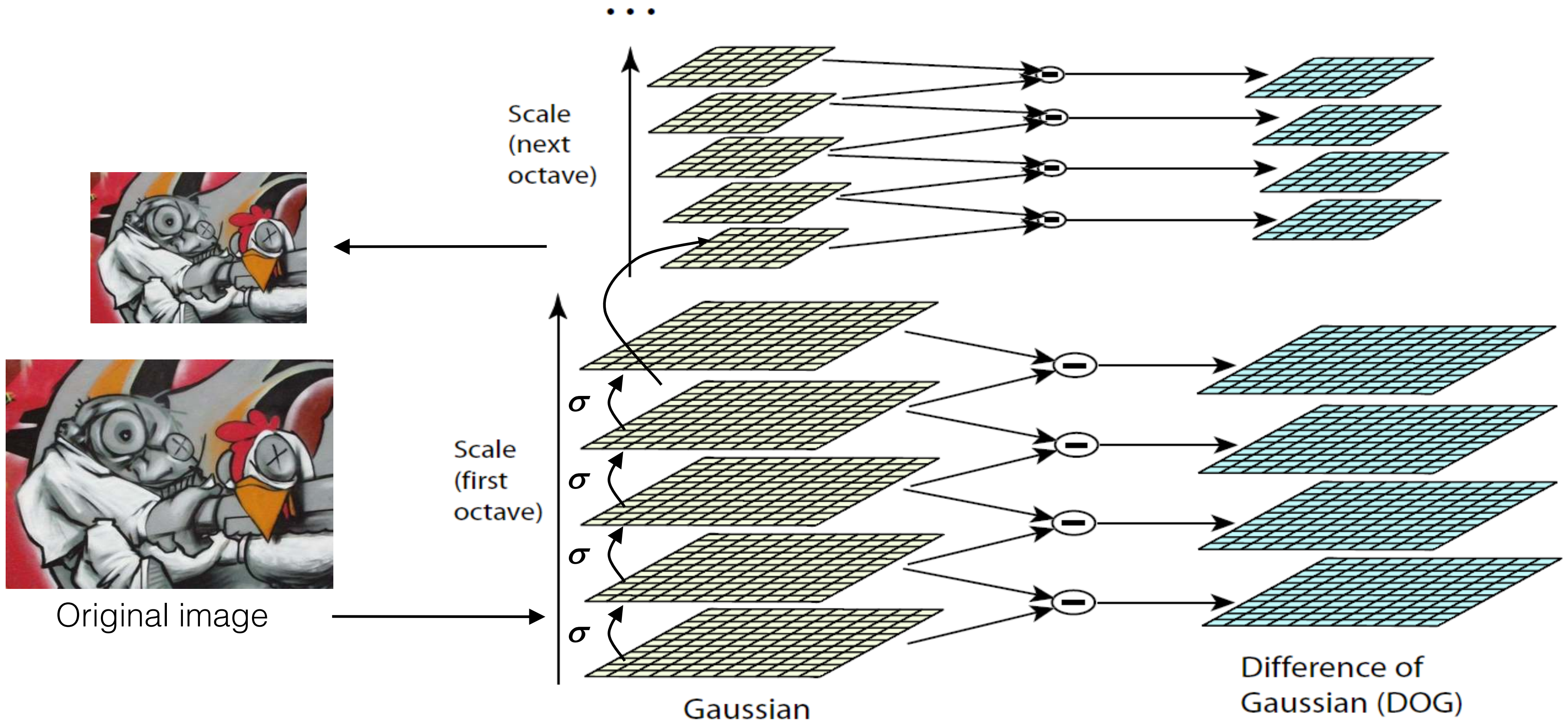
2. The convolution of a Gaussian kernel by a Gaussian kernel is a Gaussian kernel. The variance (σ^2) of the resulting kernel is the sum of the variances.

Why?

Why is it interesting?

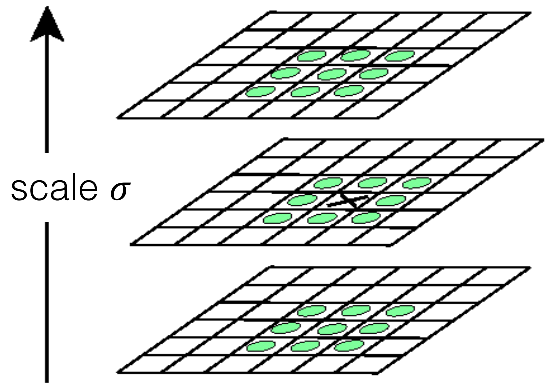
École des Ponts ParisTech

DoG – Efficient Computation



Accurate Keypoint Localization

Keypoint locations: Extrema of Difference-of-Gaussian in scale space



The size of the image region used to compute the descriptor is proportional to the scale of the extremum.

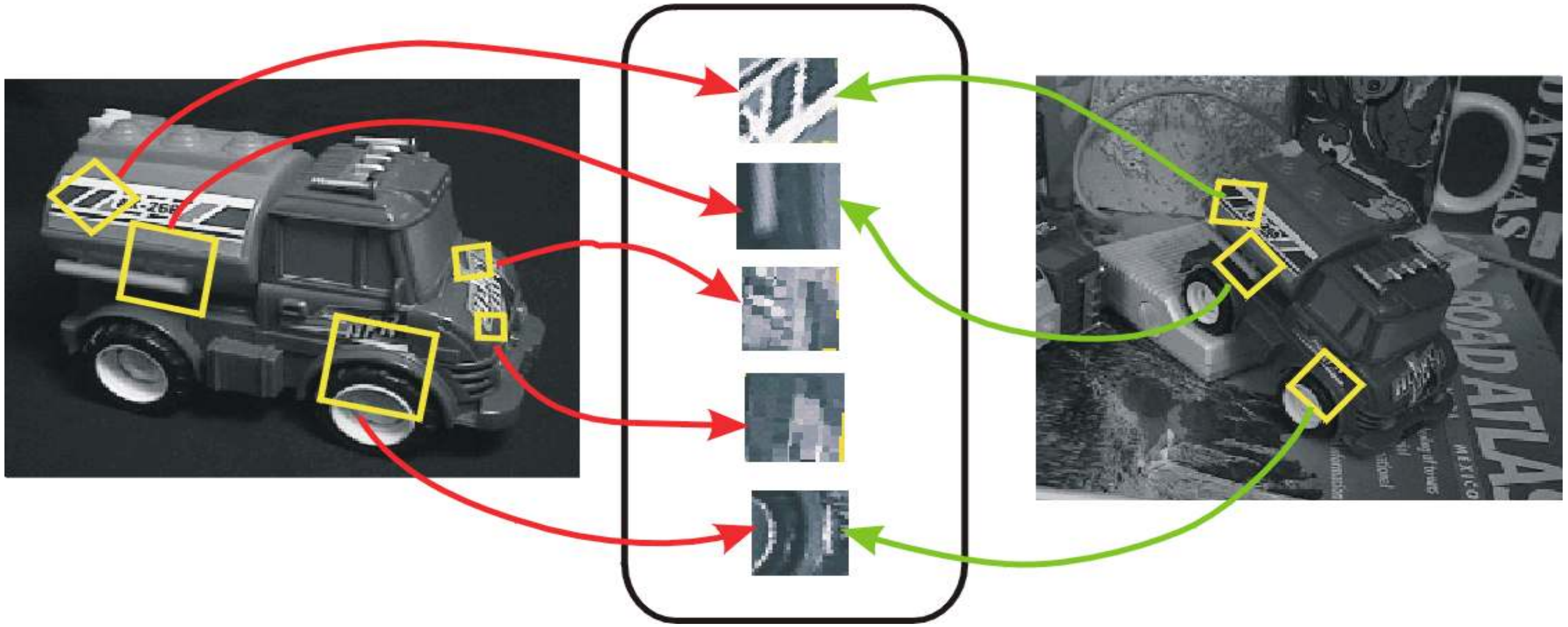
Sub-pixel and sub-scale interpolation. The Taylor expansion around point is:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T} \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

⇒ Offset of extremum (use finite differences for derivatives): $\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$

Keypoint Covariance to Rotation

Idea: Introduce a heuristics to compute a canonical orientation that varies with the image rotation

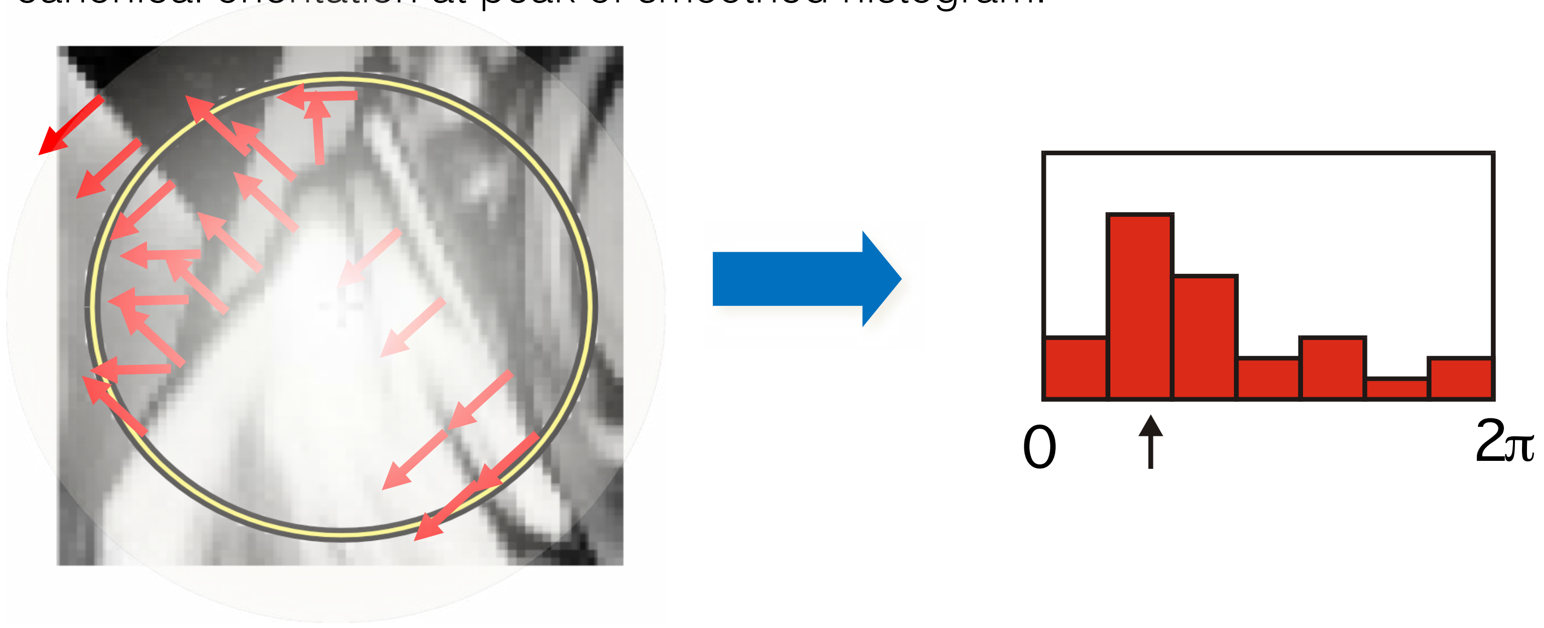


École des Ponts
ParisTech

Keypoint Covariance to Rotation

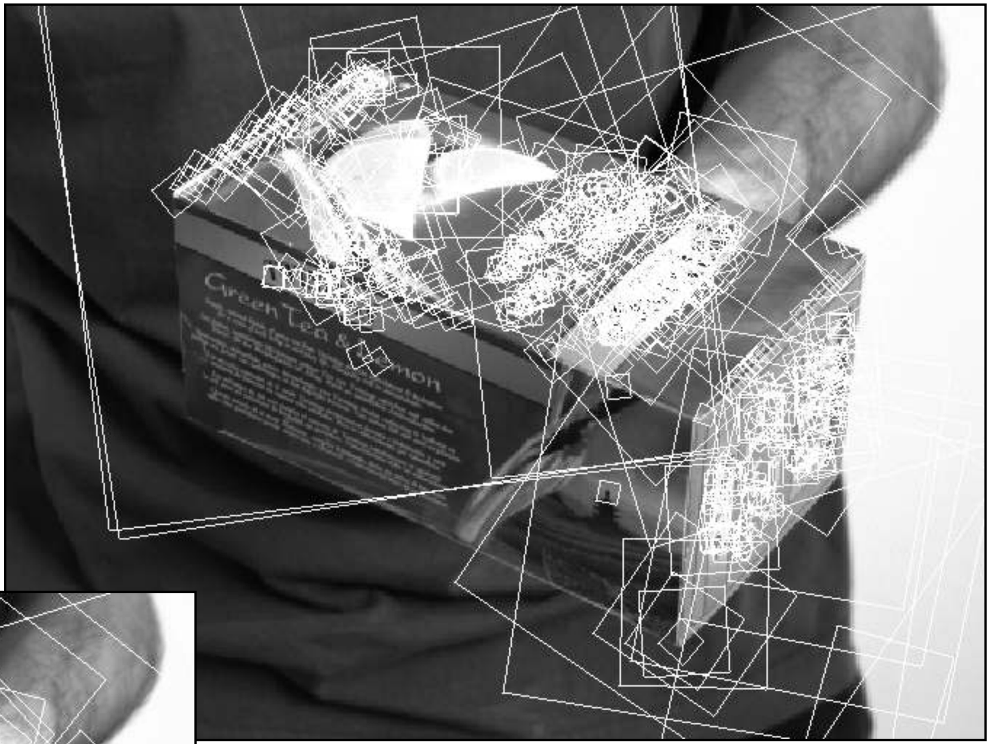
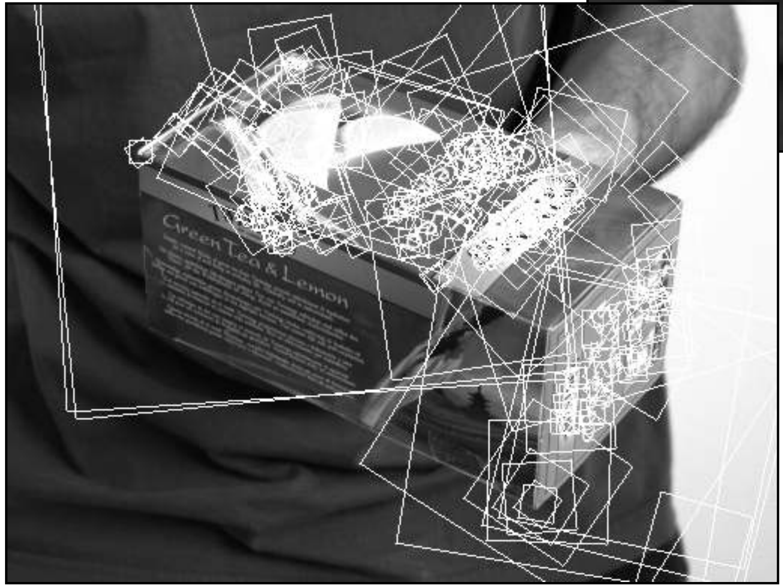
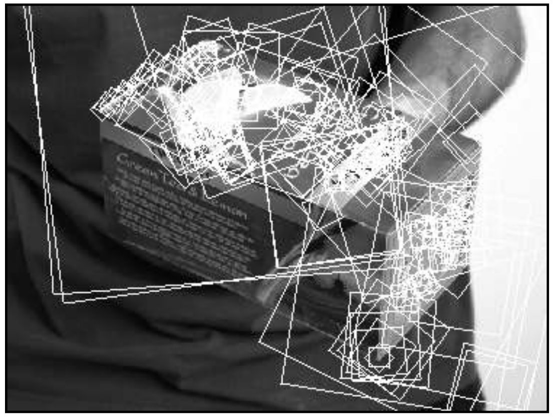
Idea: Introduce a heuristics to compute a canonical orientation that varies with the image rotation

- Create histogram of local gradient directions computed over the image patch;
- Each gradient contributes for its norm, weighted by its distance to the patch center;
- Assign canonical orientation at peak of smoothed histogram.

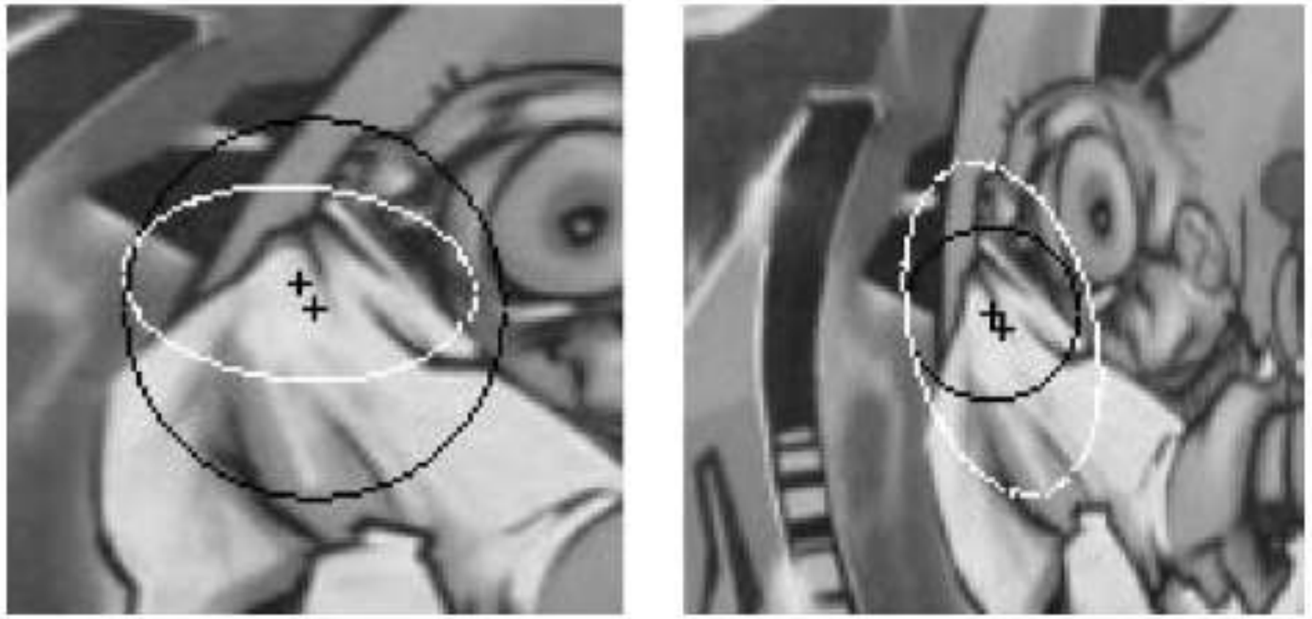


If the image is rotated, the gradient directions will be rotated by the same angle, et so will the canonical orientation.

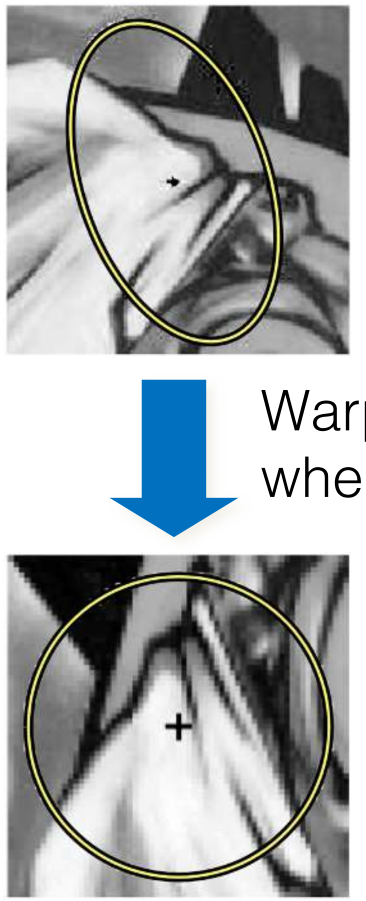
Detection Results

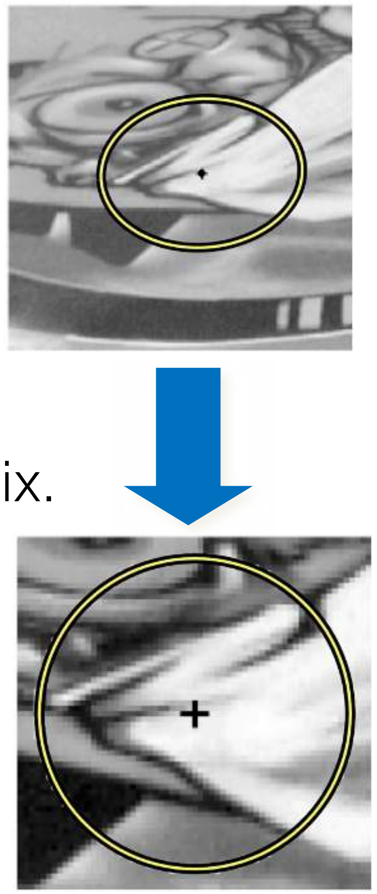
Affine Region Detectors: Covariant to Affine Transformations



Affine Transformation Estimation



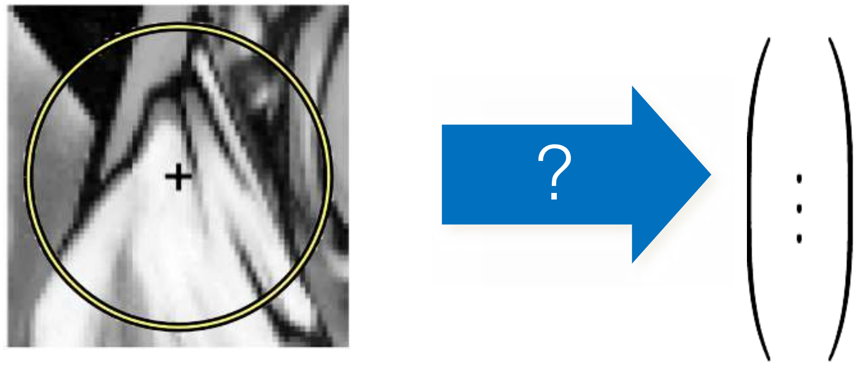
Warp by Affine Transformation $Q^{1/2}$, where Q is the auto-correlation matrix.

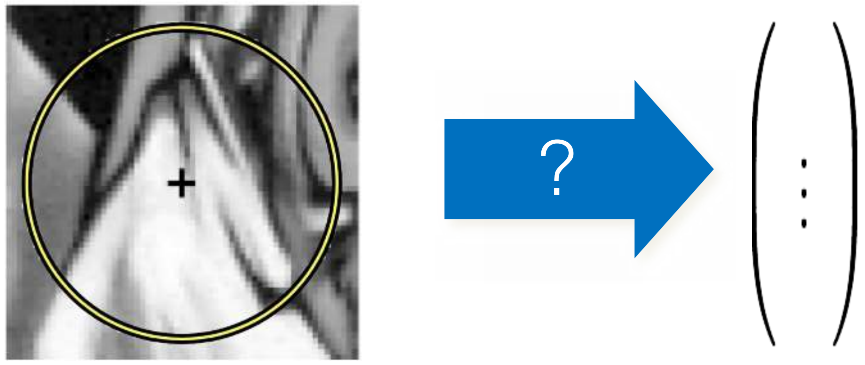


1 degree-of-freedom remains → use canonical orientation from SIFT

Invariance to Lighting & Small Transformations

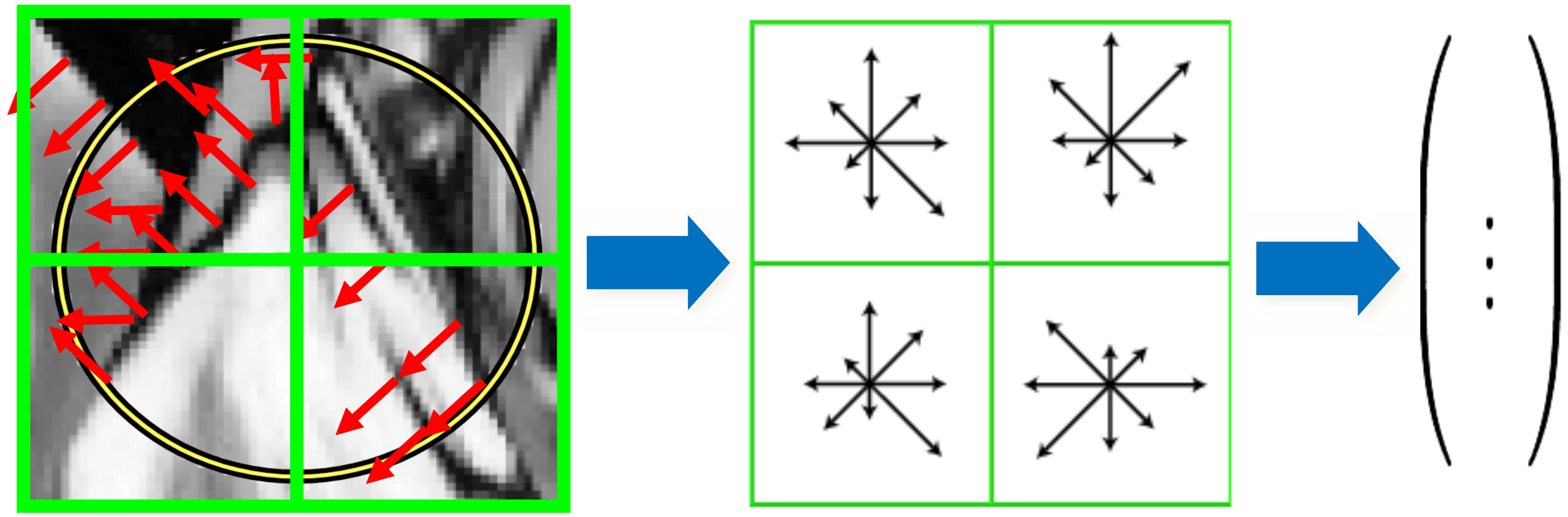
Descriptor





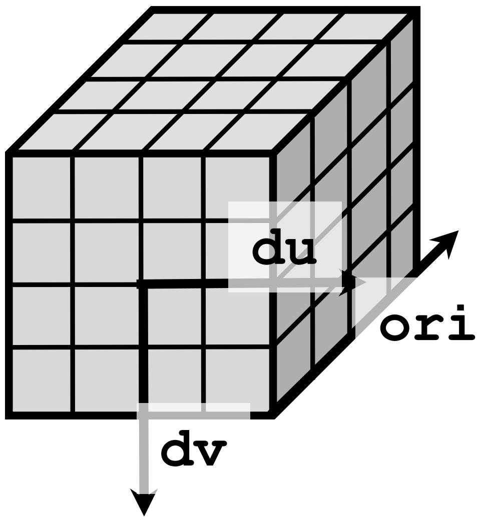
SIFT Descriptor

Made of local histograms of gradients:



In practice: 8 orientations x 4 x 4 histograms = 128 dimensions vector.

```
for du in [-d;+d]
  for dv in [-d:+d]
    weight <- exp(-(du²+dv²)/(2s²))
    mag <- weight * gradient_magnitude(u+du, v+dv)
    ori <- gradient_orientation(u+du, v+dv)
    u_bin,v_bin,ori_bin <- bin_index(du,dv,ori)
    u_frac,v_frac,ori_frac <- fractional_part(du,dv,ori)
    for r in {0,1}
      for c in {0,1}
        for o in {0,1}
          Histogram[u_bin+r][v_bin+c][ori_bin+o] +=
           mag * ((r == 0) ? (1-u_frac)   : u_frac) *
                 ((c == 0) ? (1-v_frac)   : v_frac) *
                 ((o == 0) ? (1-ori_frac) : ori_frac);
```



École des Ponts
ParisTech

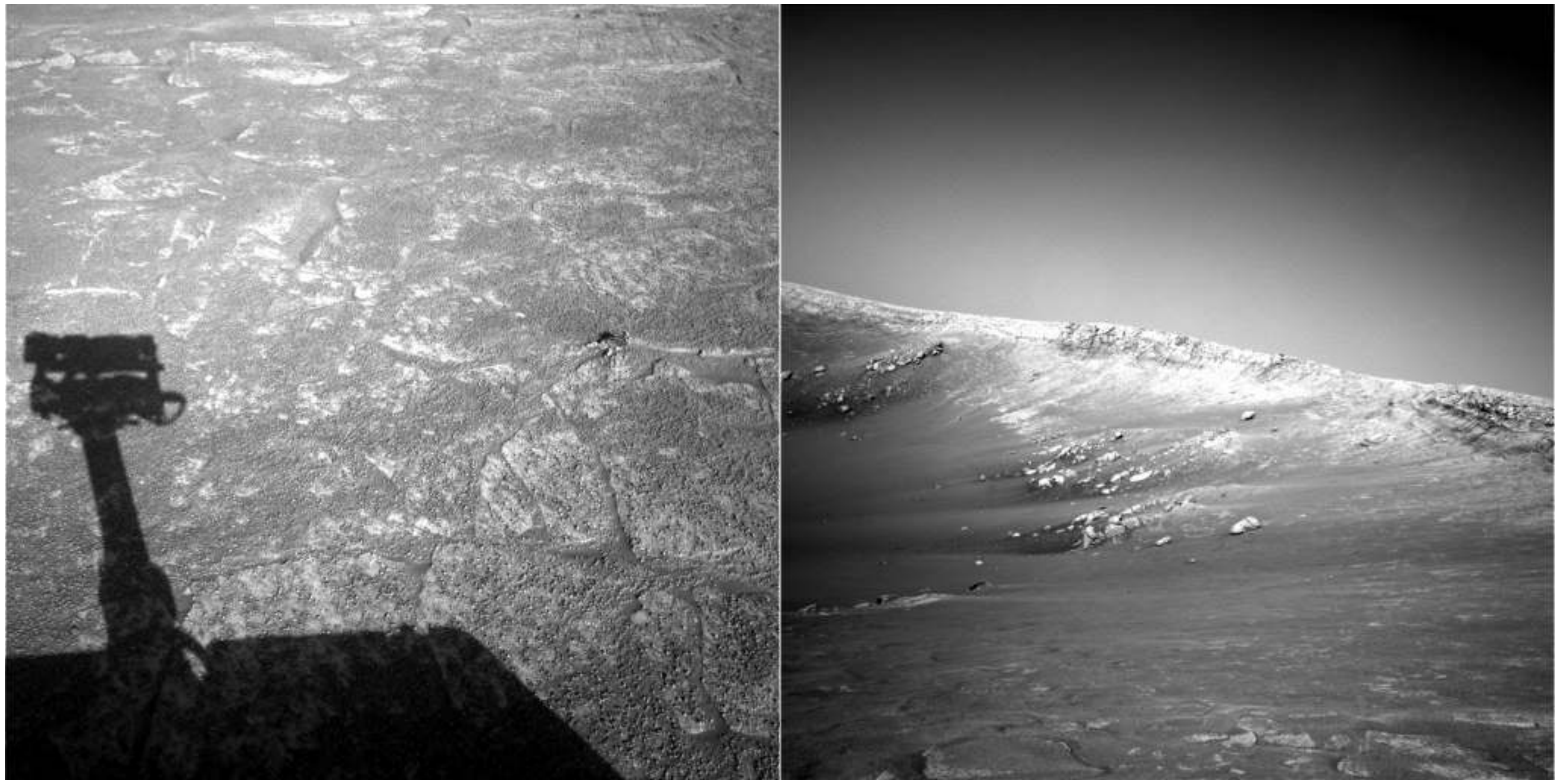
Handling Lighting Changes

- Gains do not affect gradients;
- Normalization to unit length removes contrast;
- Saturation affects gradient magnitudes much more than their orientations: magnitudes are thresholded.

Some Results

École des Ponts
ParisTech

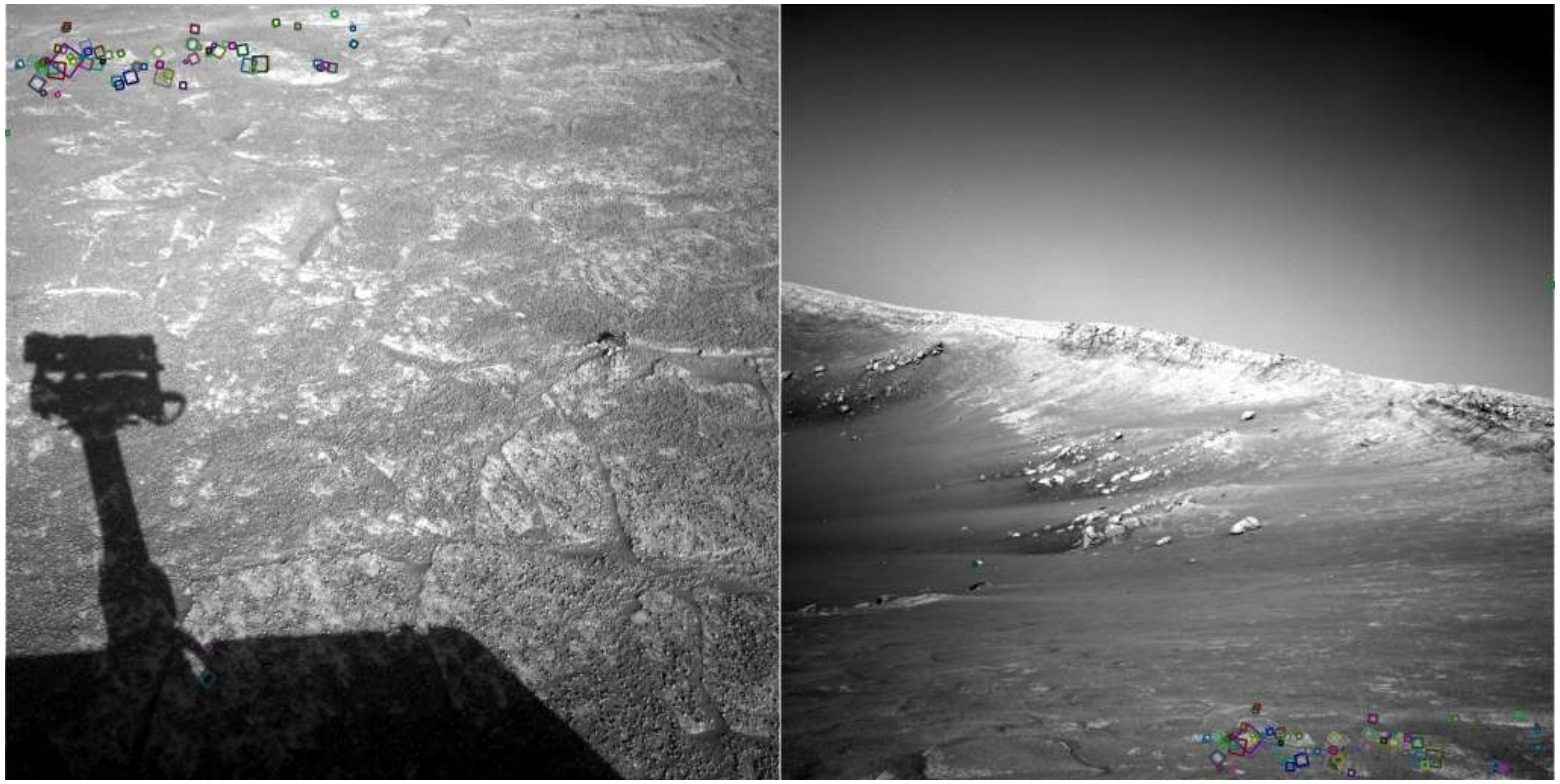
Other Example



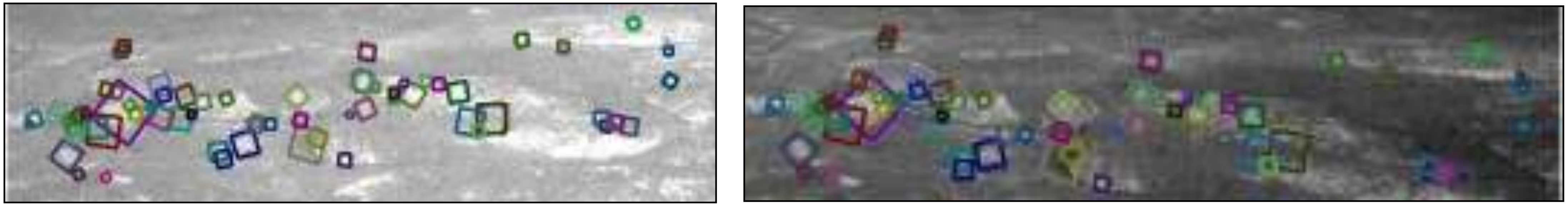
NASA Mars Rover images

École des Ponts ParisTech

Other Example


NASA Mars Rover images
with SIFT feature matches

Other Example



NASA Mars Rover images
with SIFT feature matches

Why Local Histograms of Gradients?

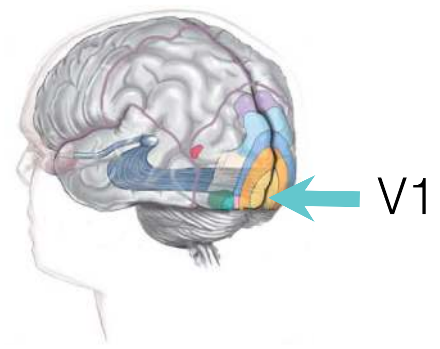
SIFT is motivated by the theory of Hubel and Wiesel (1962, Nobel Prize in 1981):

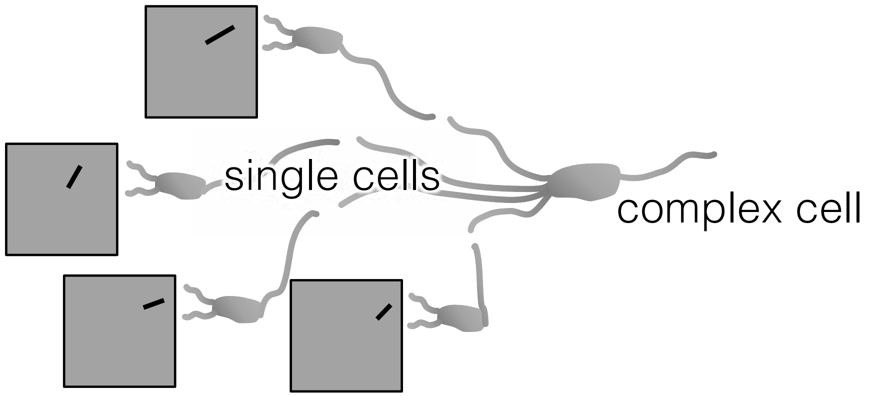
In V1:

- Some neurons ("simple cells") have a high response when presented with oriented gradients. Each simple cell is specialized for a specific location and gradient orientations.

- Some neurons ("complex cells") are connected to simple cells with similar gradient orientation and location specializations. A complex cell has a high response when at least one simple cell has a high response.

→ gradient is important.
→ it is also important to be tolerant to some orientation and position shift.



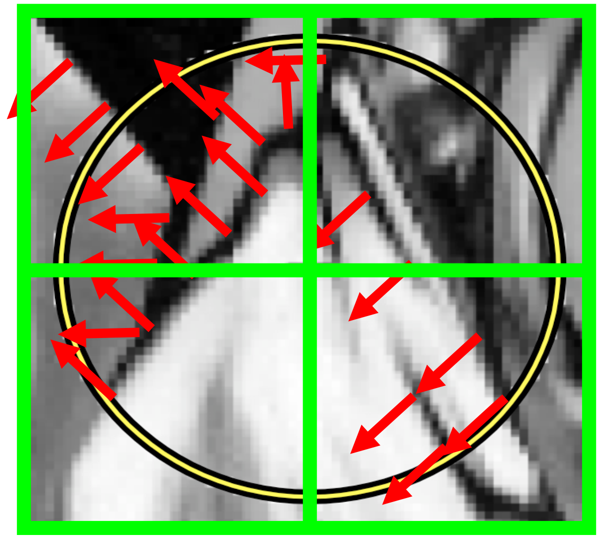


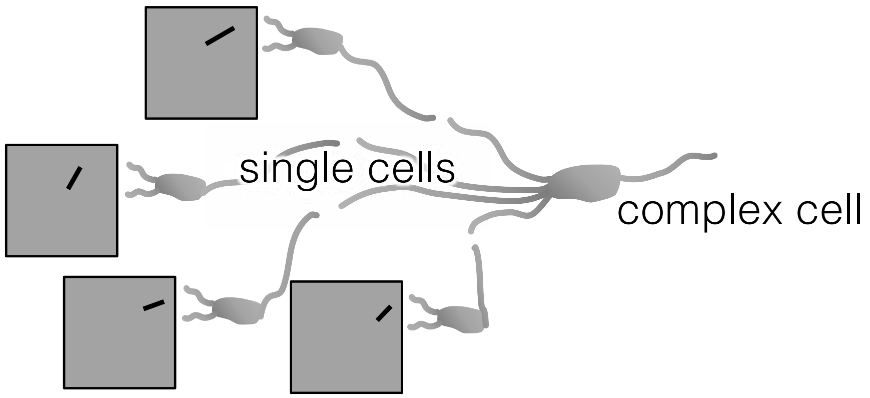
Why Local Histograms of Gradients?

SIFT is motivated by the theory of Hubel and Wiesel:

In SIFT:

- gradient detection plays the role of the single cells;
- local histograms play the role of the complex cells.



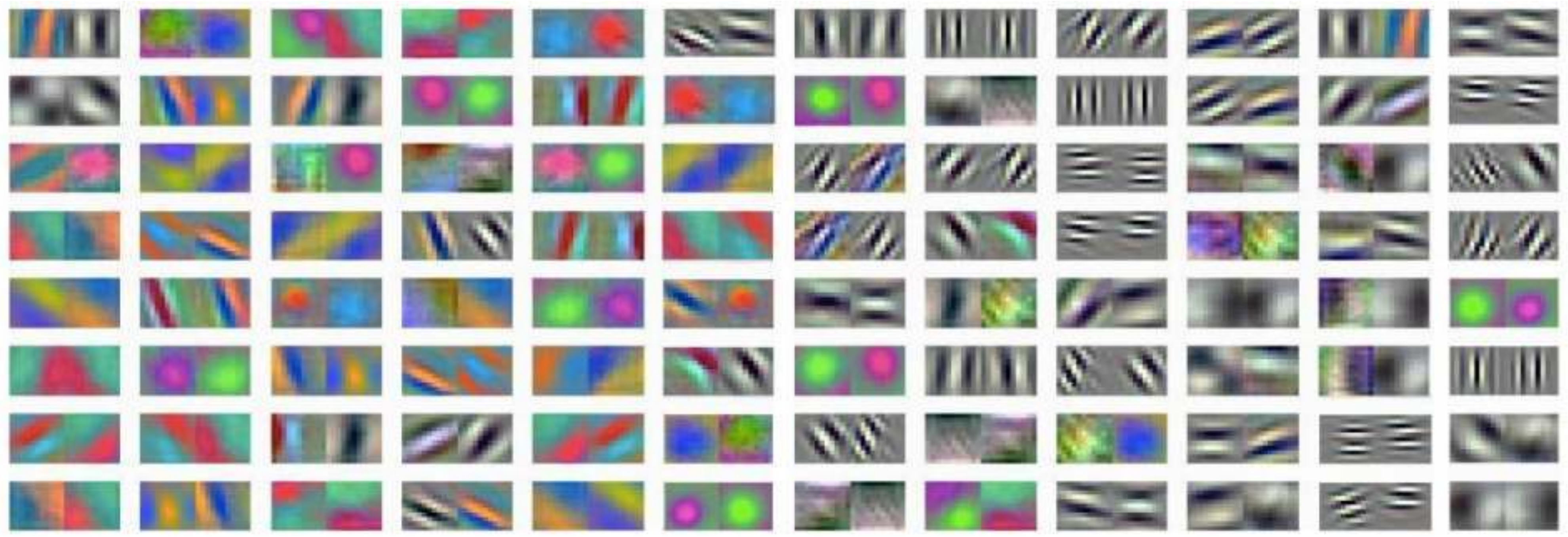


Convolutional Neural Networks (CNNs)

CNNs have also a mechanism inspired by complex cells: Pooling layers.

CNNs learn convolutional filters on which the pooling layers are applied.

On the first convolutional layer, CNNs tend to learn to detect oriented image gradients!



École des Ponts
ParisTech