Computer Vision:
Image feature descriptor

Siheng Chen 陈思衡

Image feature descriptor

Why do we need feature descriptors?

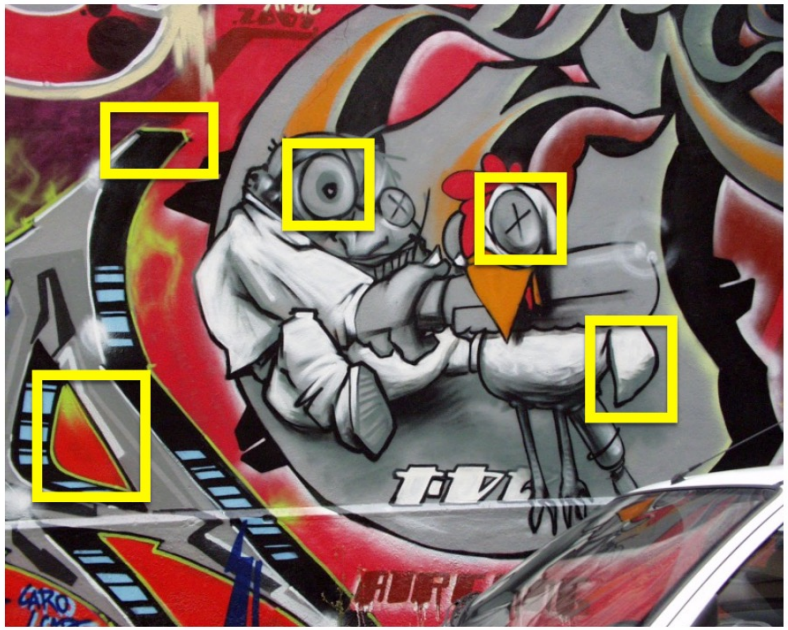
Designing feature descriptors

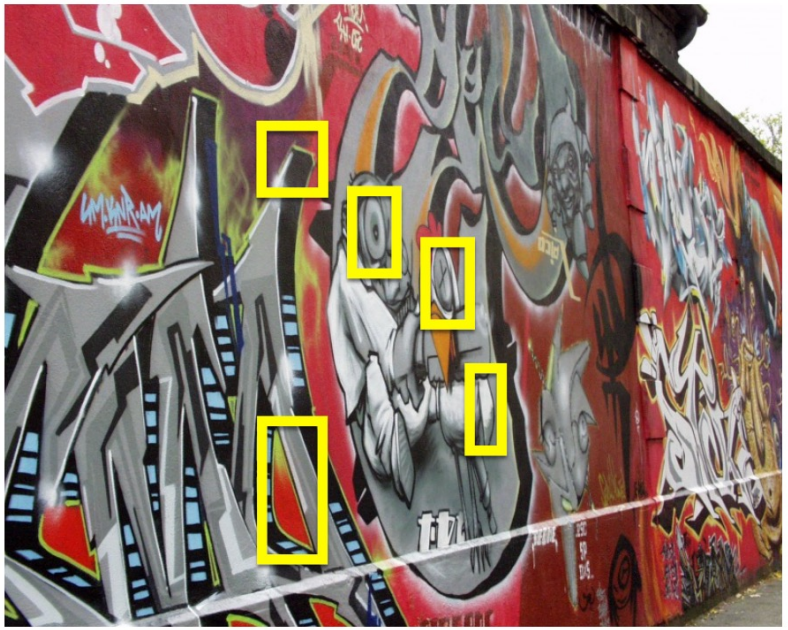
Scale invariant feature transform

Image matching

Local feature learning

Why do we need feature descriptors?





If we know where the good features are, how do we match them?

Why do we need feature descriptors?

Image feature descriptor

Why do we need feature descriptors?

Designing feature descriptors

Scale invariant feature transform

Image matching

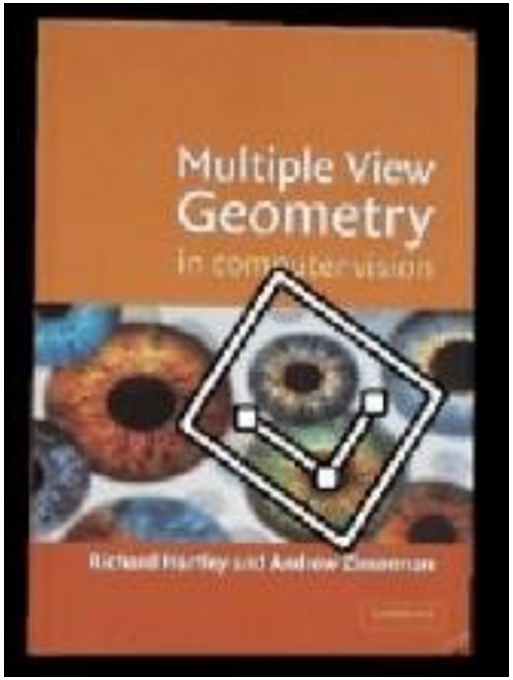
Local feature learning

Designing feature descriptors

Photometric transformations

Designing feature descriptors

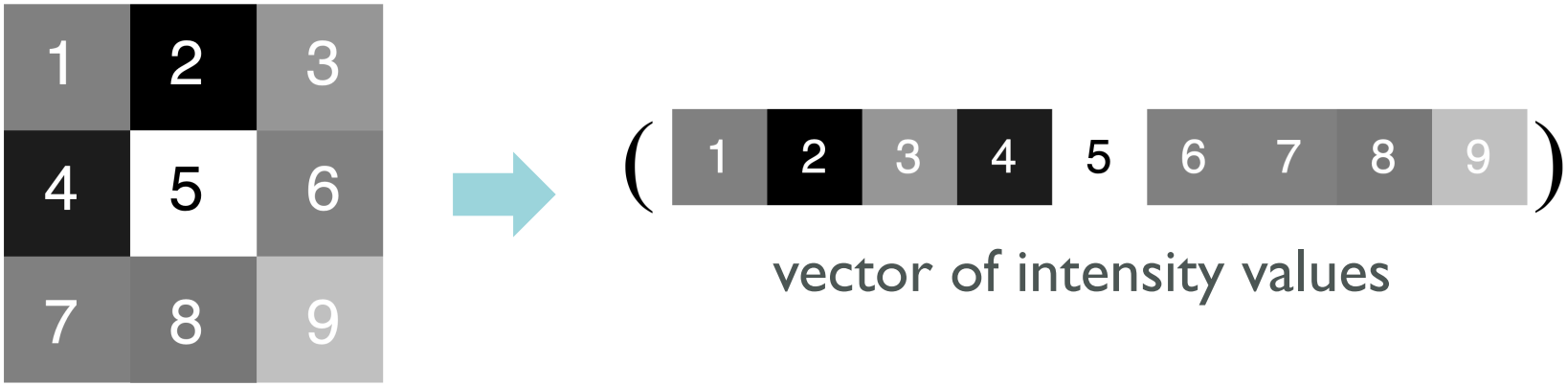
Geometric transformations



objects will appear at different scales, translation and rotation

Designing feature descriptors

Pixel values

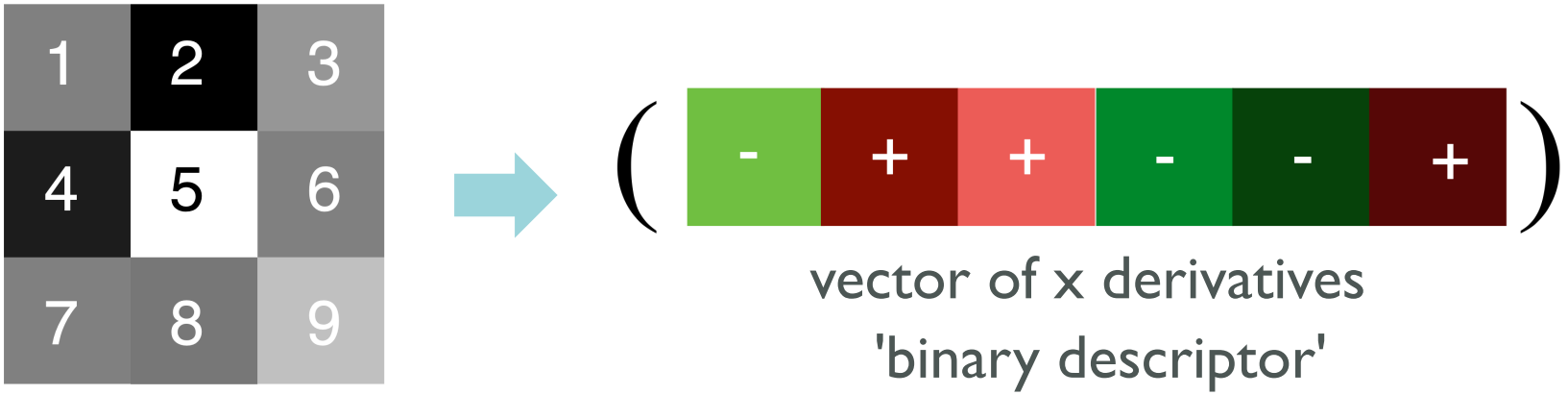


Perfectly fine if geometry and appearance is unchanged

How can you be less sensitive to absolute intensity values?

Designing feature descriptors

Gradient (pixel difference)

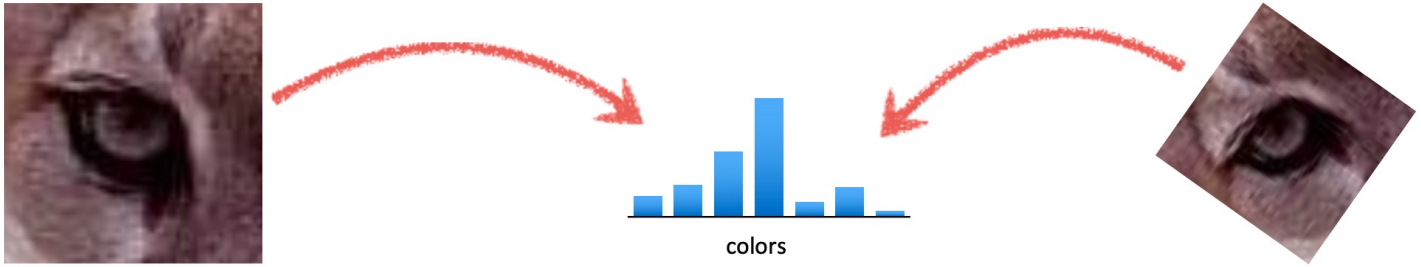


Feature is invariant to absolute intensity values

How can you be less sensitive to deformations?

Designing feature descriptors

Color histogram

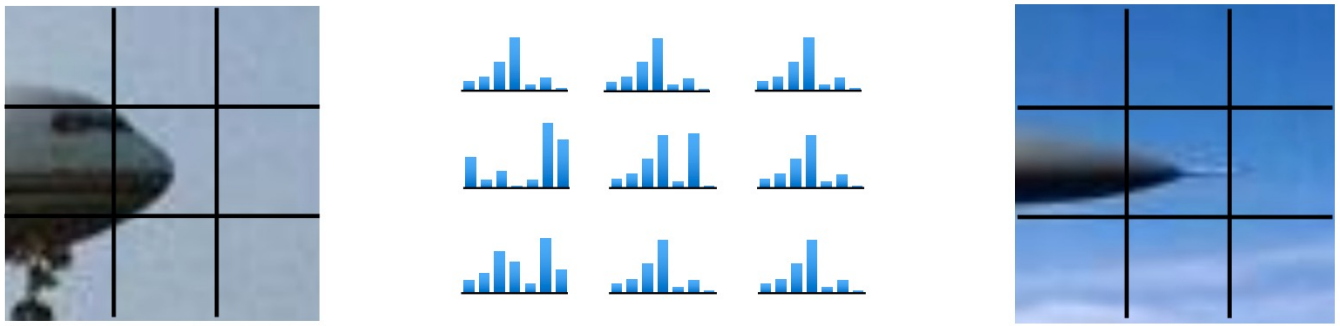


Invariant to changes in scale and rotation

How can you be more sensitive to spatial layout?

Designing feature descriptors

Spatial histogram



Retains rough spatial layout
Some invariance to deformations

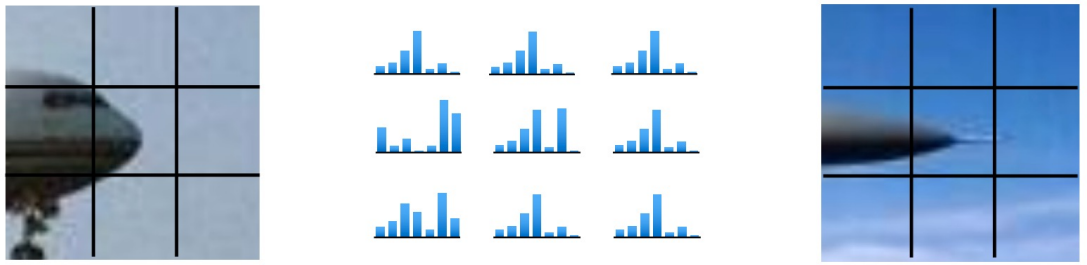
How can you be completely invariant to rotation?

Designing feature descriptors

Image patch descriptor

- lighting (absolute intensity values)
- deformations
- spatial layout
- rotation
- scale

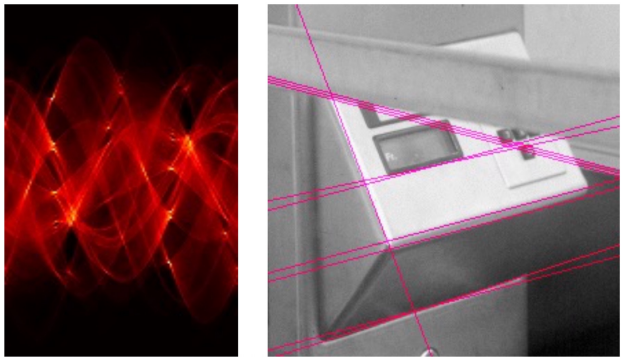
find key objectives!



spatial histogram: spatial resolution + texture

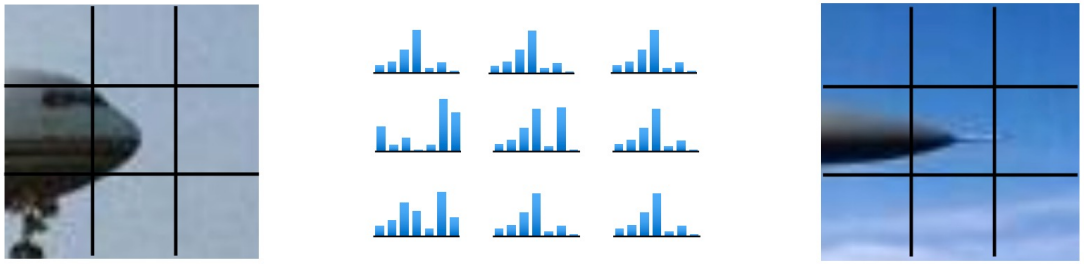
Designing feature descriptors

Strategy: **Solve problems from another perspective**



Hough transform

Fourier transform

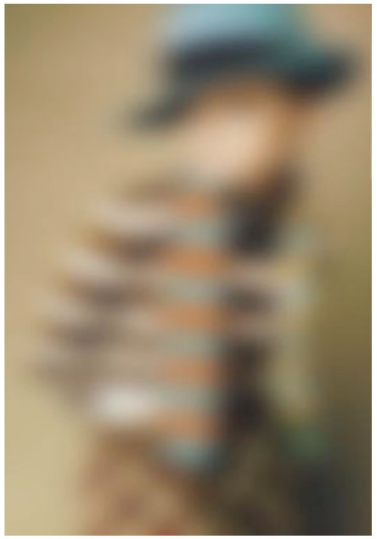


spatial histogram: spatial resolution + texture

Designing feature descriptors

Discriminative power

Raw pixels



Sampled

Locally orderless

Global histogram

Generalization power

Image feature descriptor

Why do we need feature descriptors?

Designing feature descriptors

Scale invariant feature transform

Image matching

Local feature learning

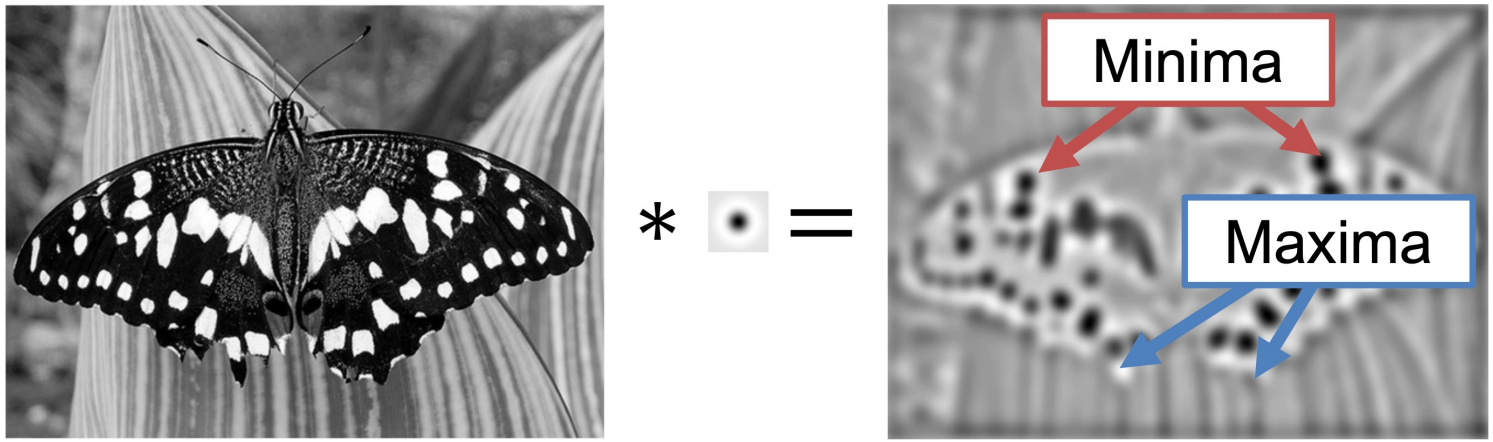
Scale invariant feature transform

1. Detect keypoints
2. Describe local features

David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004

Scale invariant feature transform

Keypoint detection



Find maxima and minima of filter response in scale and space

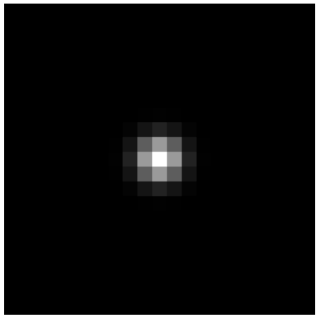
David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004

Scale invariant feature transform

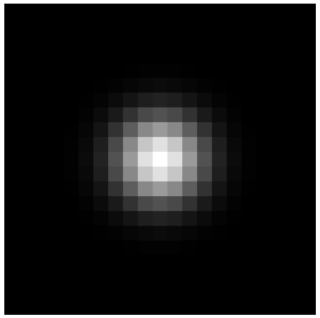
Keypoint detection

Gaussian filter $$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

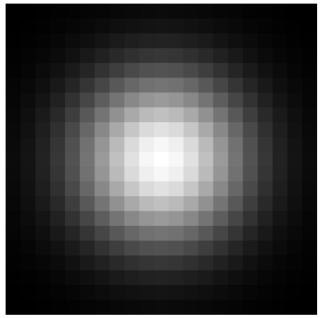
σ = 1
filter = 21x21



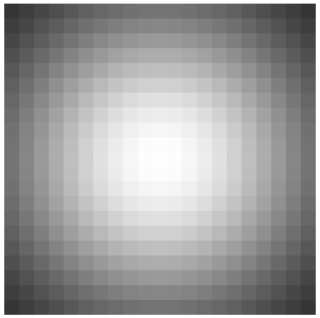
σ = 2
filter = 21x21



σ = 4
filter = 21x21



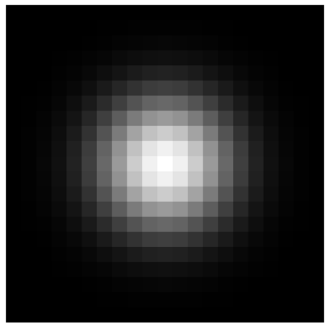
σ = 8
filter = 21x21



Scale invariant feature transform

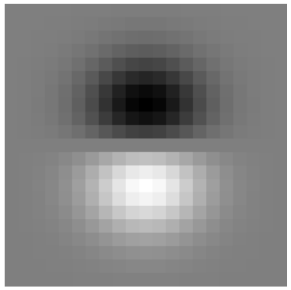
Keypoint detection

Gaussian filter

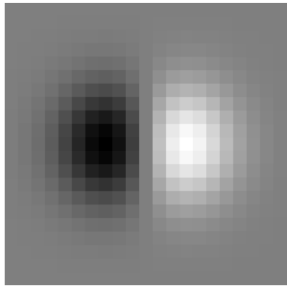


1st Derivative

$\frac{\partial}{\partial y} g$

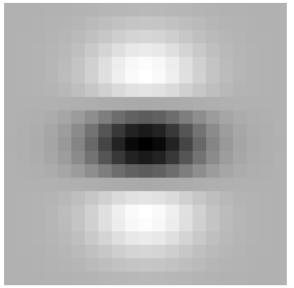


$\frac{\partial}{\partial x} g$

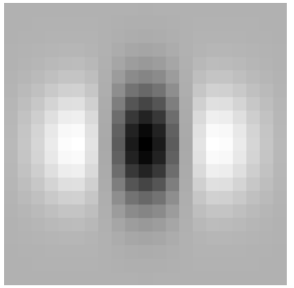


2nd Derivative

$\frac{\partial^2}{\partial^2 y} g$



$\frac{\partial^2}{\partial^2 x} g$

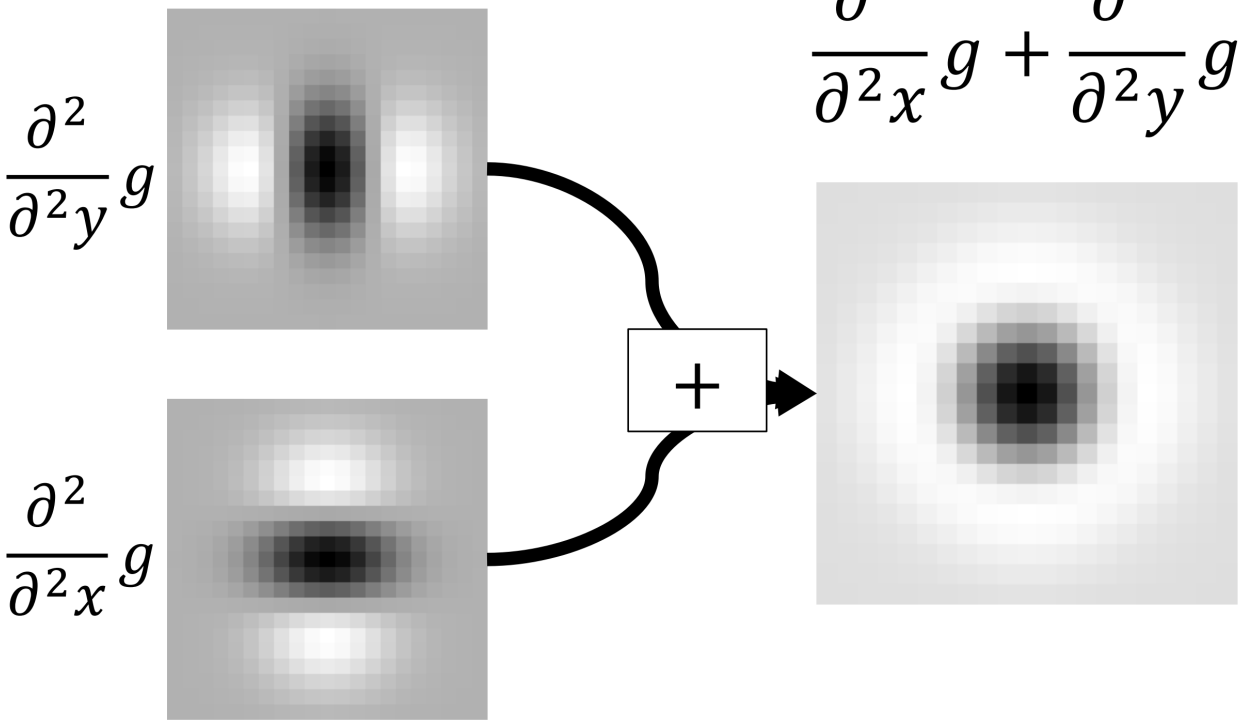


Scale invariant feature transform

Keypoint detection

Laplacian of Gaussian (LoG)

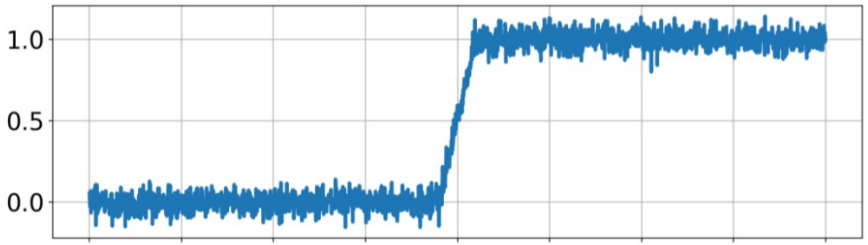
$$\frac{\partial^2}{\partial^2 x} g + \frac{\partial^2}{\partial^2 y} g$$



Scale invariant feature transform

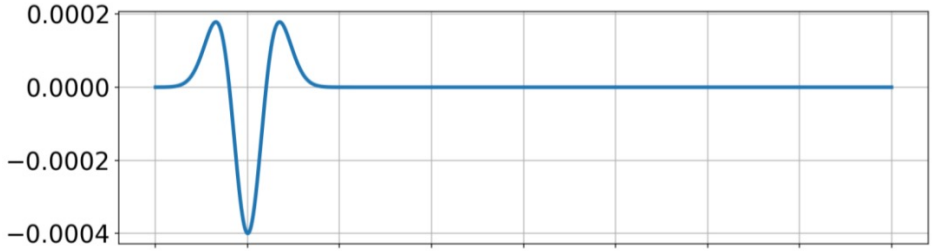
Keypoint detection

f



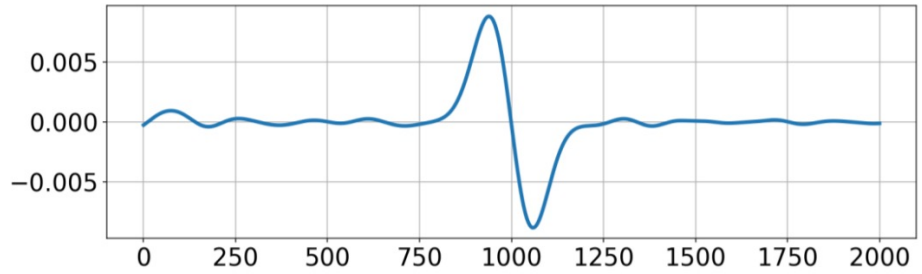
"edge" signal

$\frac{\partial^2}{\partial^2 x} g$



Laplacian of Gaussian (LoG)

$f * \frac{\partial^2}{\partial^2 x} g$

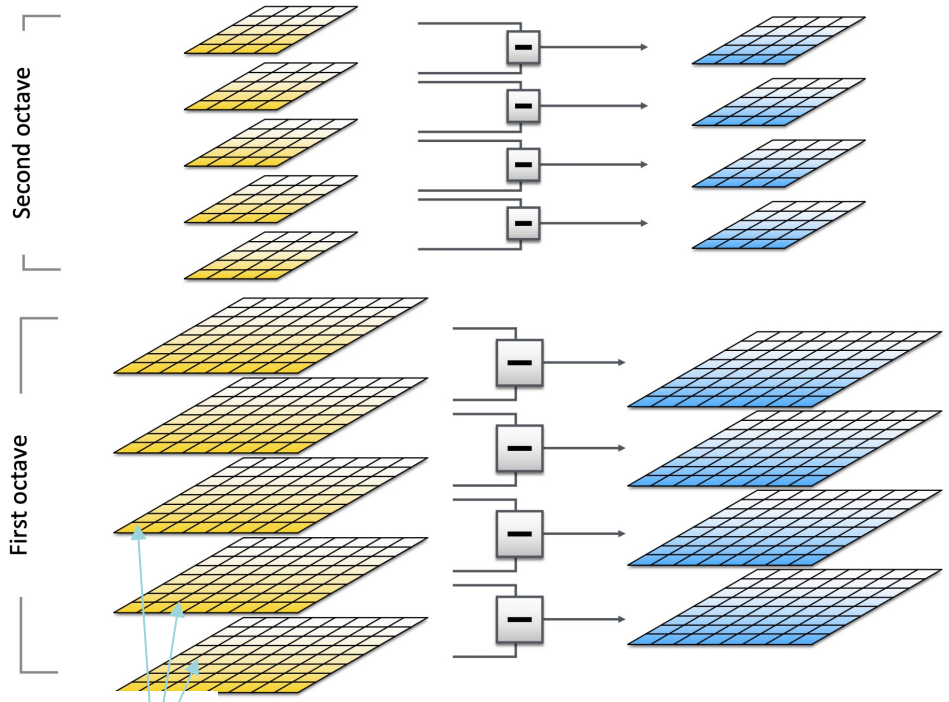


zero-crossing

Scale invariant feature transform

Keypoint detection

- Multi-scale extrema detection



$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

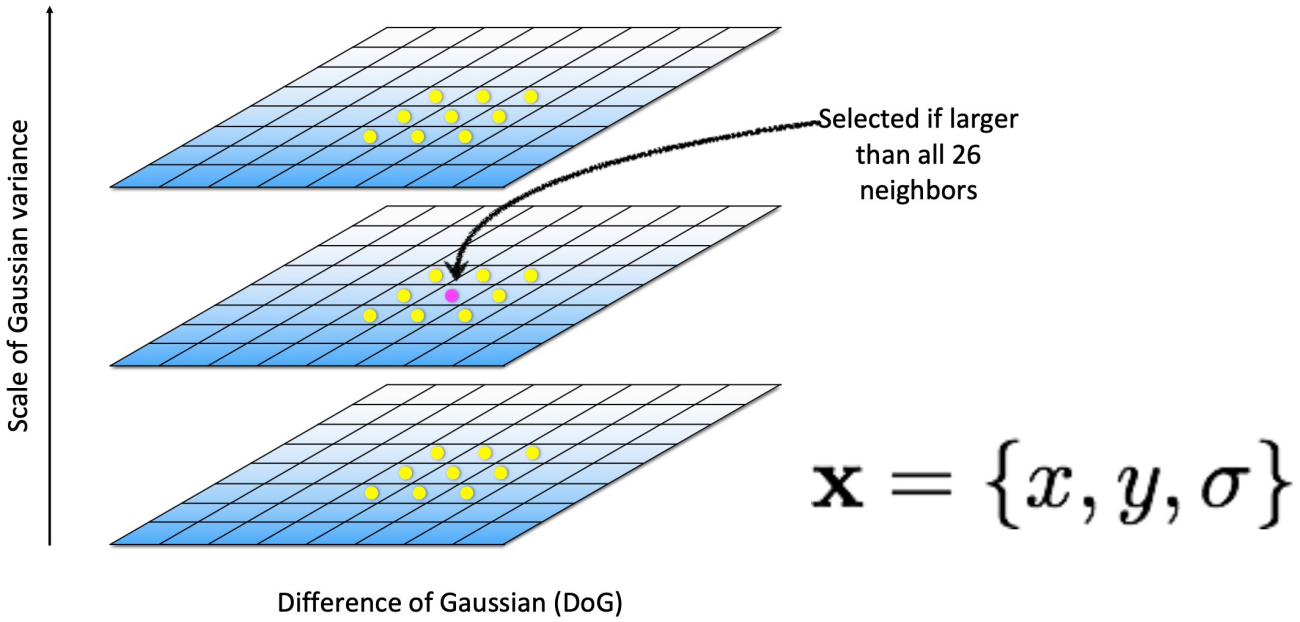
Approximation of Laplacian of Gaussian (LoG)

David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004

Scale invariant feature transform

Keypoint detection

- Multi-scale extrema detection



$$\mathbf{x} = \{x, y, \sigma\}$$

David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004

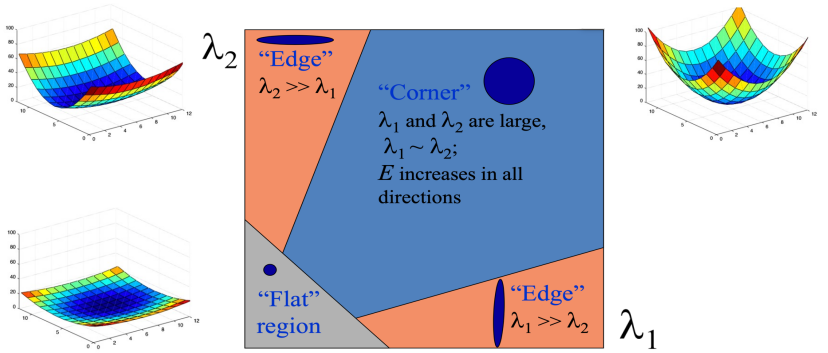
Scale invariant feature transform

Keypoint detection

- Multi-scale extrema detection
- **Localization**
  - low-contrast keypoints
  - edge keypoints

$$\mathbf{x}_m = -\frac{\partial^2 f}{\partial \mathbf{x}^2}^{-1} \frac{\partial f}{\partial \mathbf{x}}$$

$$\boldsymbol{M} = \begin{bmatrix} \sum_{x,y \in W} I_x^2 & \sum_{x,y \in W} I_x I_y \\ \sum_{x,y \in W} I_x I_y & \sum_{x,y \in W} I_y^2 \end{bmatrix} = \boldsymbol{R^{-1}} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \boldsymbol{R}$$



Scale invariant feature transform

Keypoint detection

- Multi-scale extrema detection
- Localization
- **Orientation assignment**

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

x-derivative y-derivative

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

$$\mathbf{x} = \{x, y, \sigma, \theta\}$$

Scale invariant feature transform

Keypoint detection

Keypoint descriptor

spatial histogram

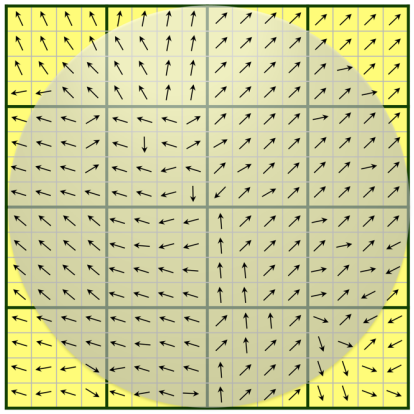
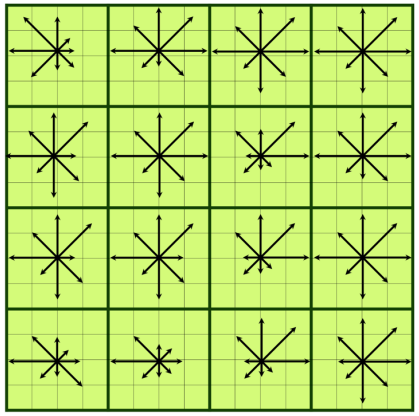


Image Gradients
(4 x 4 pixel per cell,
4 x 4 cells)



SIFT descriptor
(16 cells x 8 directions)

Scale invariant feature transform

Algorithm

- Multi-scale extrema detection
- Keypoint localization
- Orientation assignment
- **Keypoint descriptor**

Scale invariant feature transform

Think of feature as some non-linear filter that maps pixels to 128D feature

Scale invariant feature transform

Implementation

```
import numpy as np
import cv2 as cv

img = cv.imread('home.jpg')
gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)

sift = cv.SIFT_create()
kp = sift.detect(gray,None)

img=cv.drawKeypoints(gray,kp,img)

cv.imwrite('sift_keypoints.jpg',img)
```

Image feature descriptor

Why do we need feature descriptors?

Designing feature descriptors

Scale invariant feature transform

Image matching

Local feature learning

Image matching

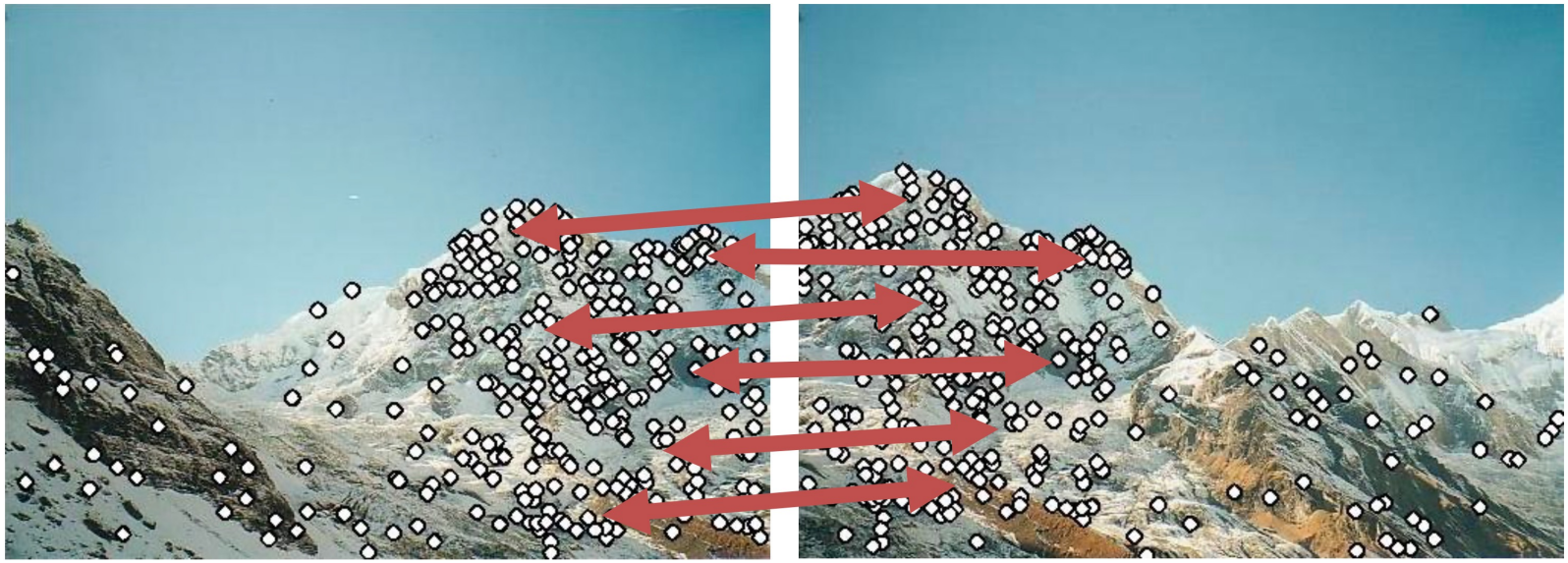


Image matching

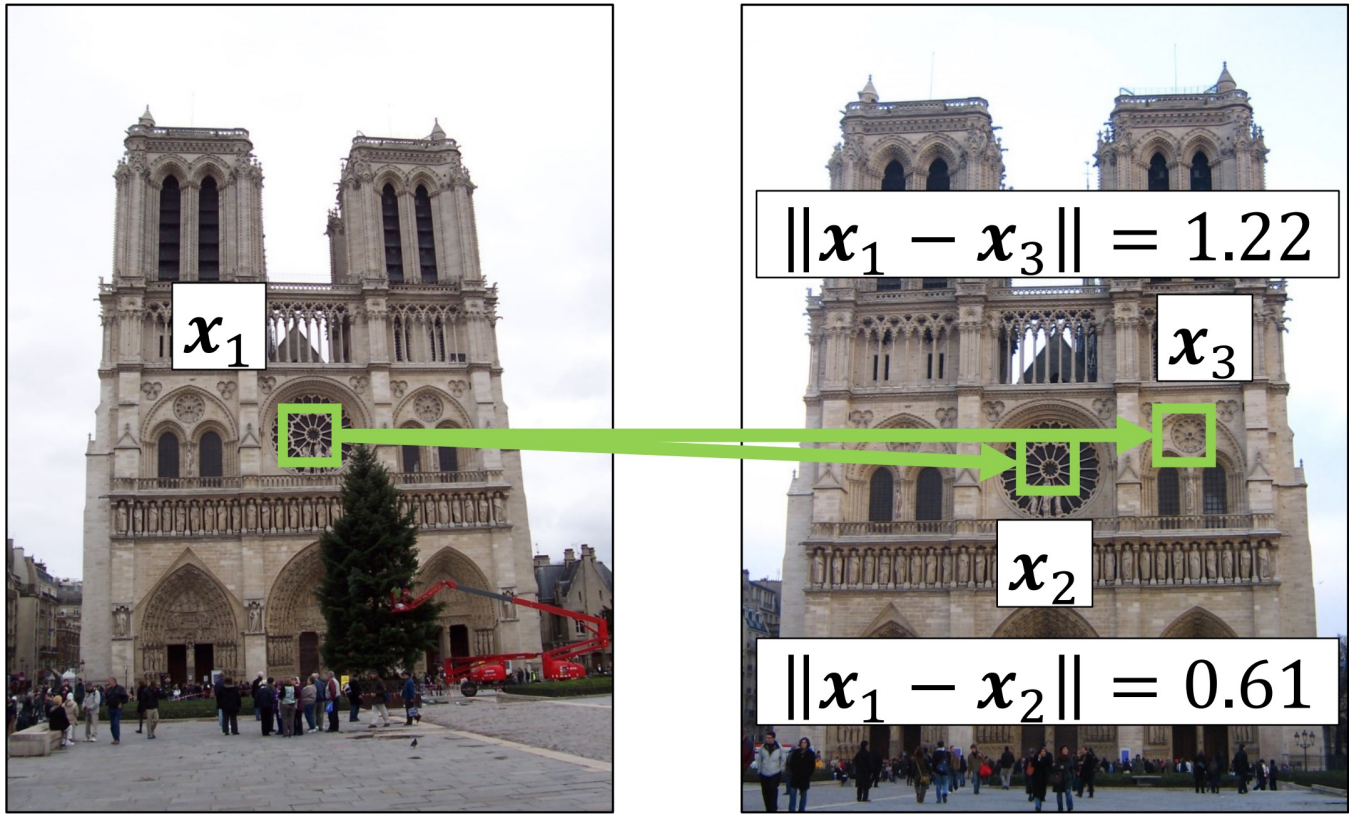


Image matching

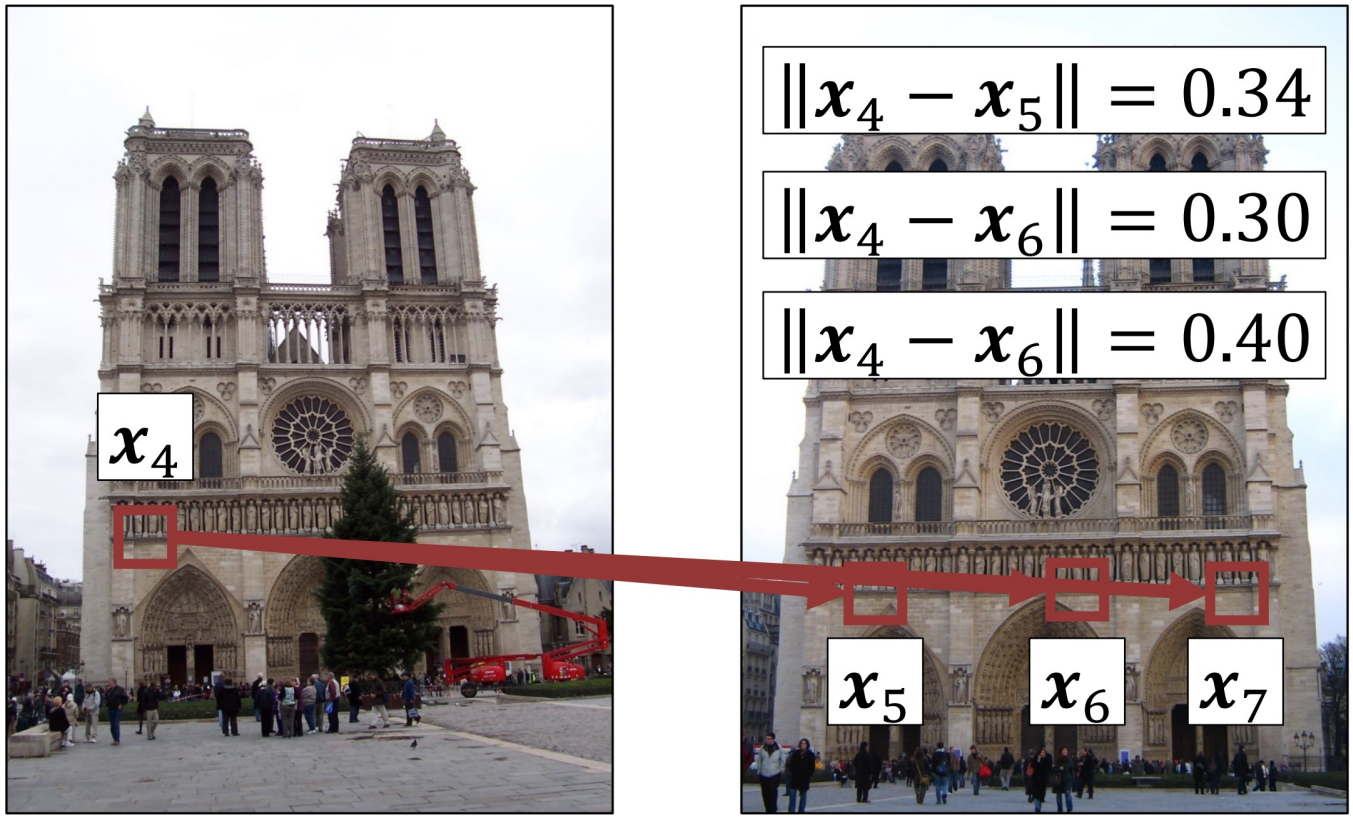


Image matching

2nd Nearest Neighbor Trick

- Given a feature $\boldsymbol{x}_q$, nearest neighbor to $\boldsymbol{x}$ is a good match, but distances can't be thresholded.
- Instead, find nearest neighbor $\boldsymbol{x}_{1NN}$ and second nearest neighbor $\boldsymbol{x}_{2NN}$. This ratio is a good test for matches:

$$r = \frac{\left\|\boldsymbol{x}_q - \boldsymbol{x}_{1NN}\right\|}{\left\|\boldsymbol{x}_q - \boldsymbol{x}_{2NN}\right\|}$$

Image matching

2nd Nearest Neighbor Trick

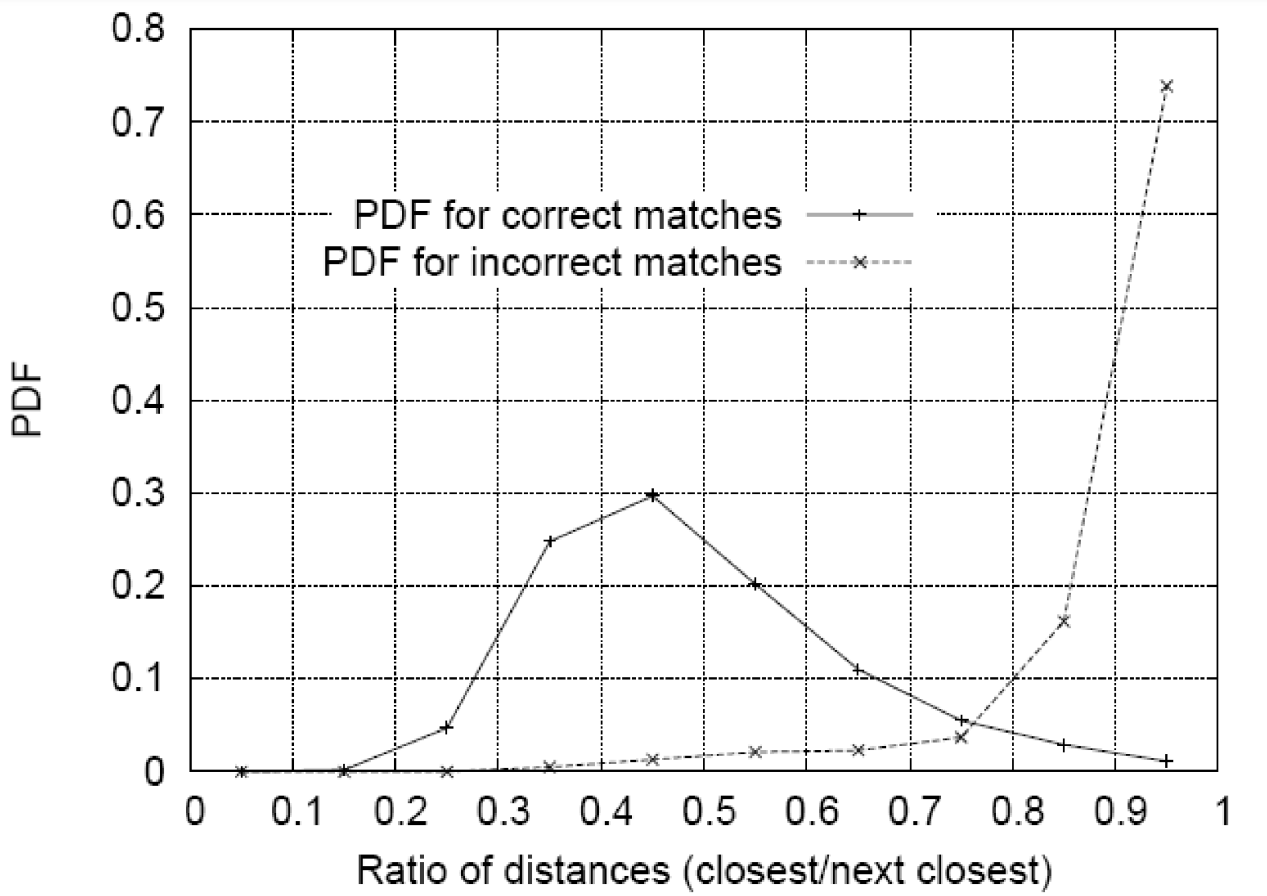


Image feature descriptor

Why do we need feature descriptors?

Designing feature descriptors

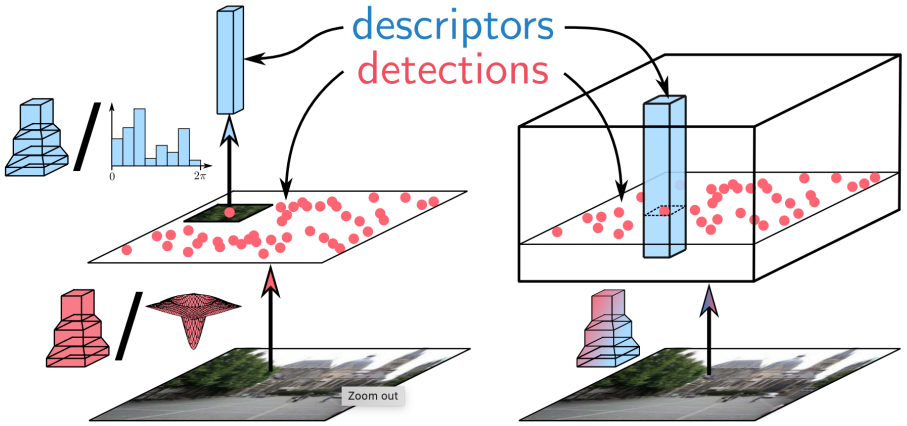
Scale invariant feature transform

Image matching

Local feature learning

Local feature learning

D2-Net: A Trainable CNN for Joint Description and Detection of Local Features



(a) detect-then-describe (b) detect-and-describe

Figure 2: **Comparison between different approaches for feature detection and description.** Pipeline (a) corresponds to different variants of the two-stage detect-then-describe approach. In contrast, our proposed pipeline (b) uses a single CNN which extracts dense features that serve as both descriptors and detectors.

Mihai Dusmanu, et al, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features", CVPR 2019

Local feature learning

D2-Net: A Trainable CNN for Joint Description and Detection of Local Features

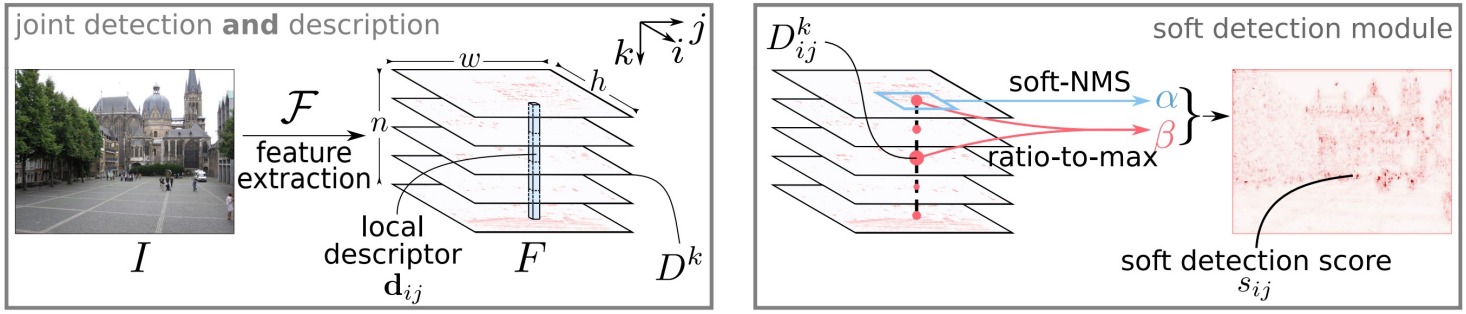


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Mihai Dusmanu, et al, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features", CVPR 2019

Local feature learning

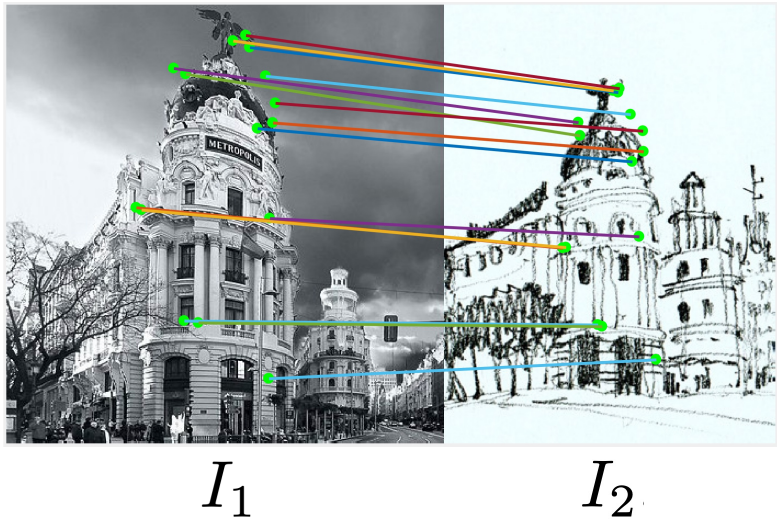
D2-Net: A Trainable CNN for Joint Description and Detection of Local Features

- Loss function

detection description

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c))$$

All correspondence — detection score



I_1 I_2

positive distance (corresponding descriptors)

$$p(c) = \|\hat{\mathbf{d}}_A^{(1)} - \hat{\mathbf{d}}_B^{(2)}\|_2$$

negative distance (the most confounding descriptor)

$$n(c) = \min\left(\|\hat{\mathbf{d}}_A^{(1)} - \hat{\mathbf{d}}_{N_2}^{(2)}\|_2, \|\hat{\mathbf{d}}_{N_1}^{(1)} - \hat{\mathbf{d}}_B^{(2)}\|_2\right)$$

triplet margin ranking loss

$$m(p(c), n(c)) = \max\left(0, M + p(c)^2 - n(c)^2\right)$$

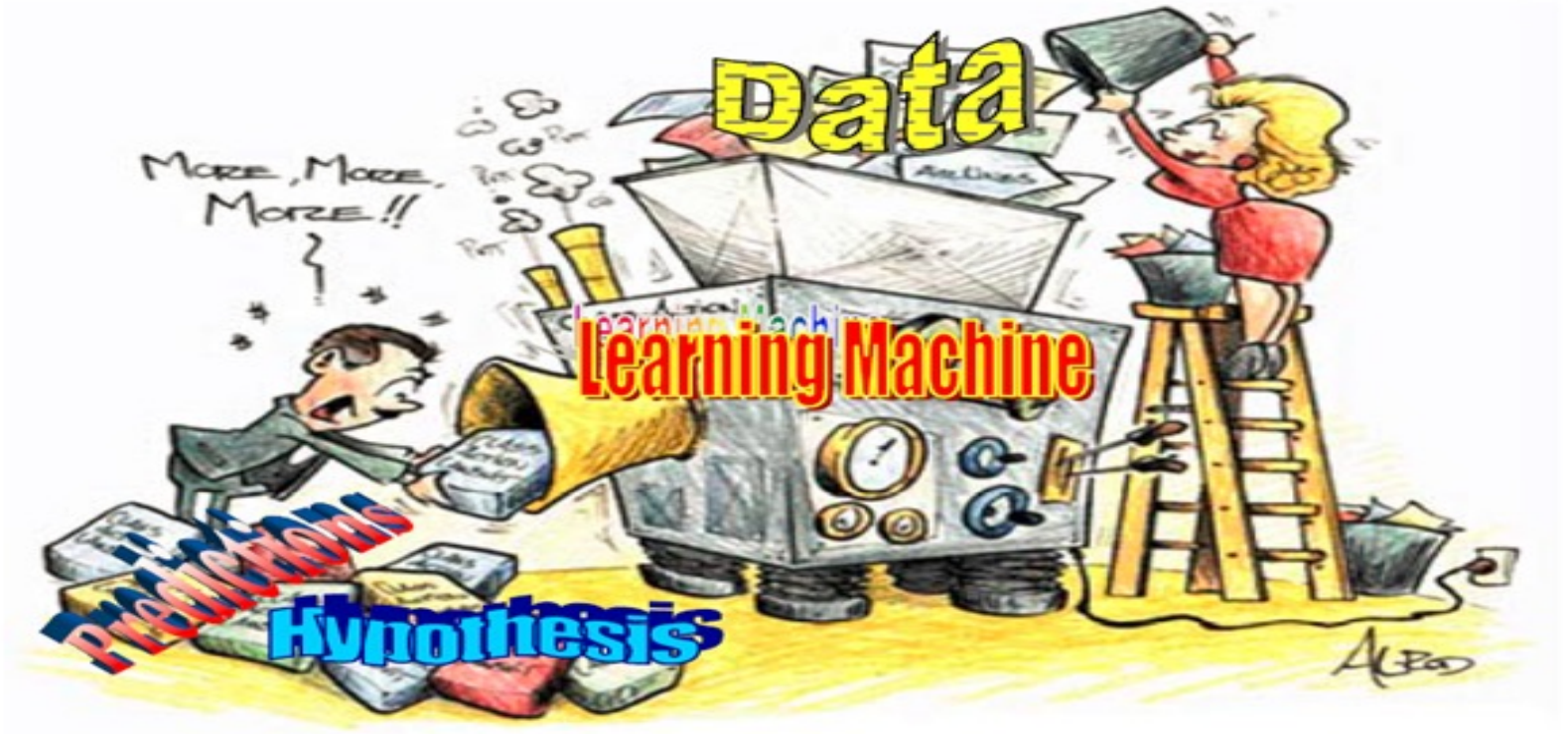
Mihai Dusmanu, et al, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features", CVPR 2019

Local feature learning

D2-Net: A Trainable CNN for Joint Description and Detection of Local Features

Mihai Dusmanu, et al, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features", CVPR 2019

Next lecture: Machine learning



Thank you very much!

sihengc@sjtu.edu.cn