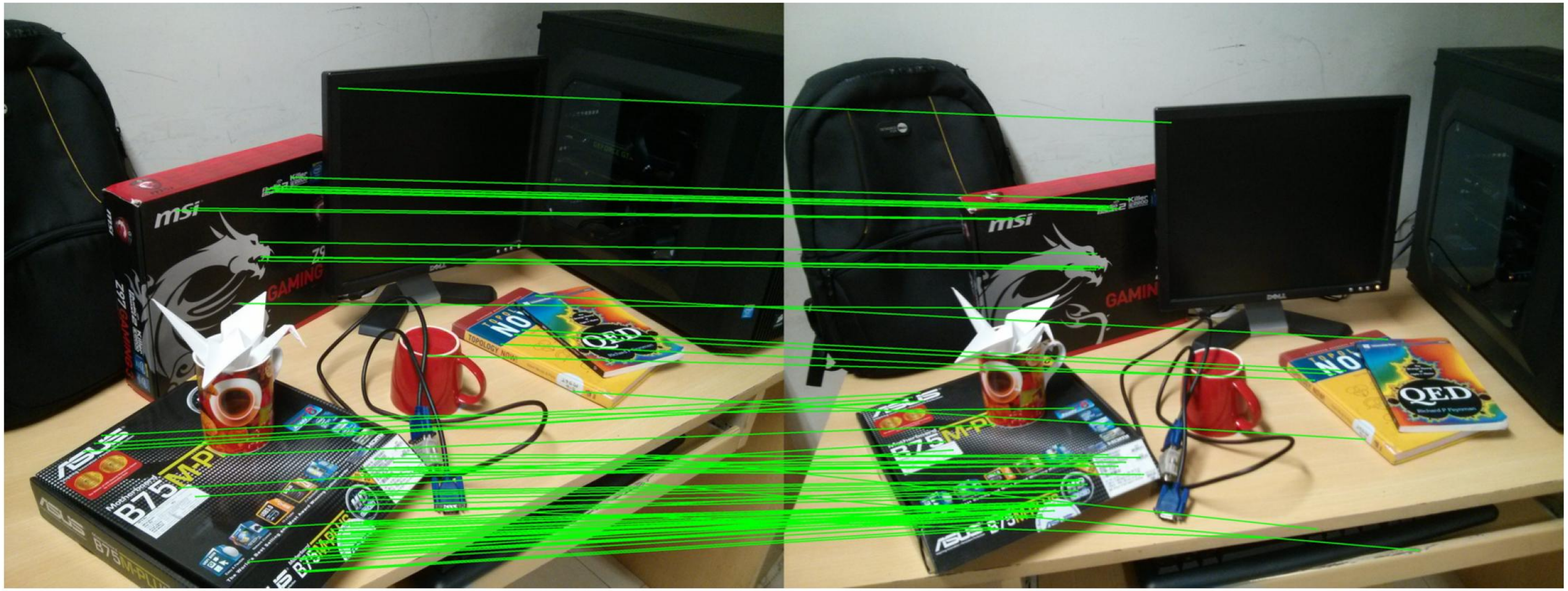
Scale Invariance Feature Transform

CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Feature Detection and Matching

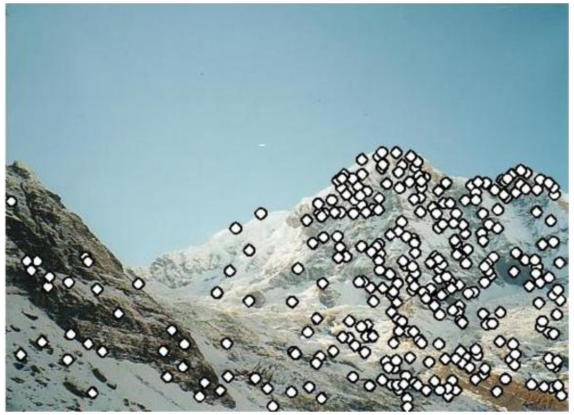


Geometry-aware Feature Matching for Structure from Motion Applications. Shah et al, WACV'15

Applications: stereo matching, image stitching, 3D reconstruction, camera pose estimation, object recognition

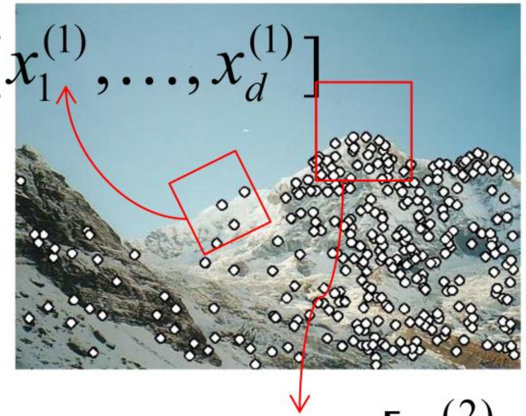
Scale Invariance Feature Transform (SIFT)

- Keypoint detection



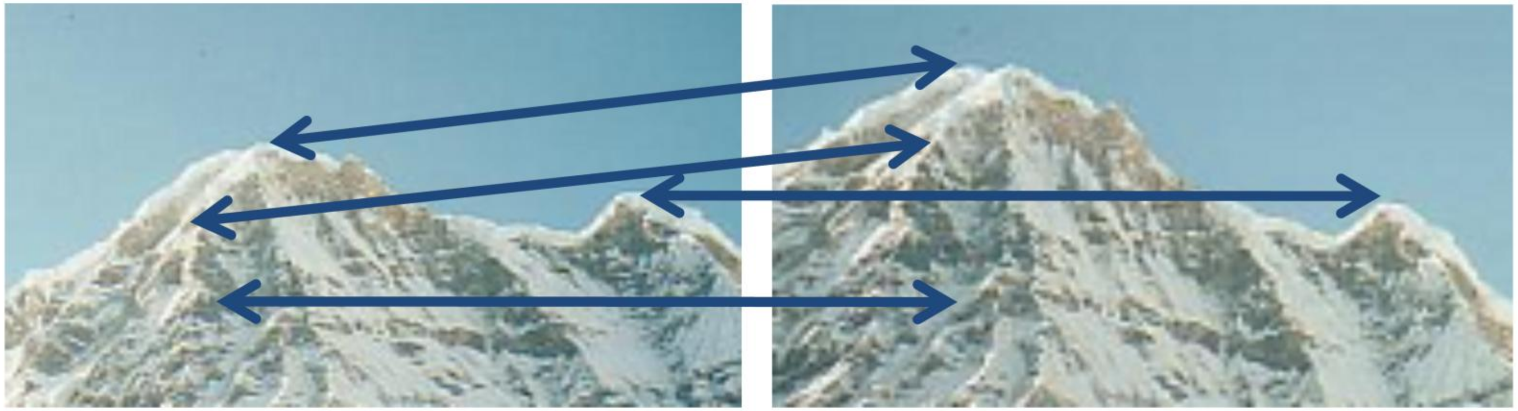
- Compute descriptors

$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$



$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$

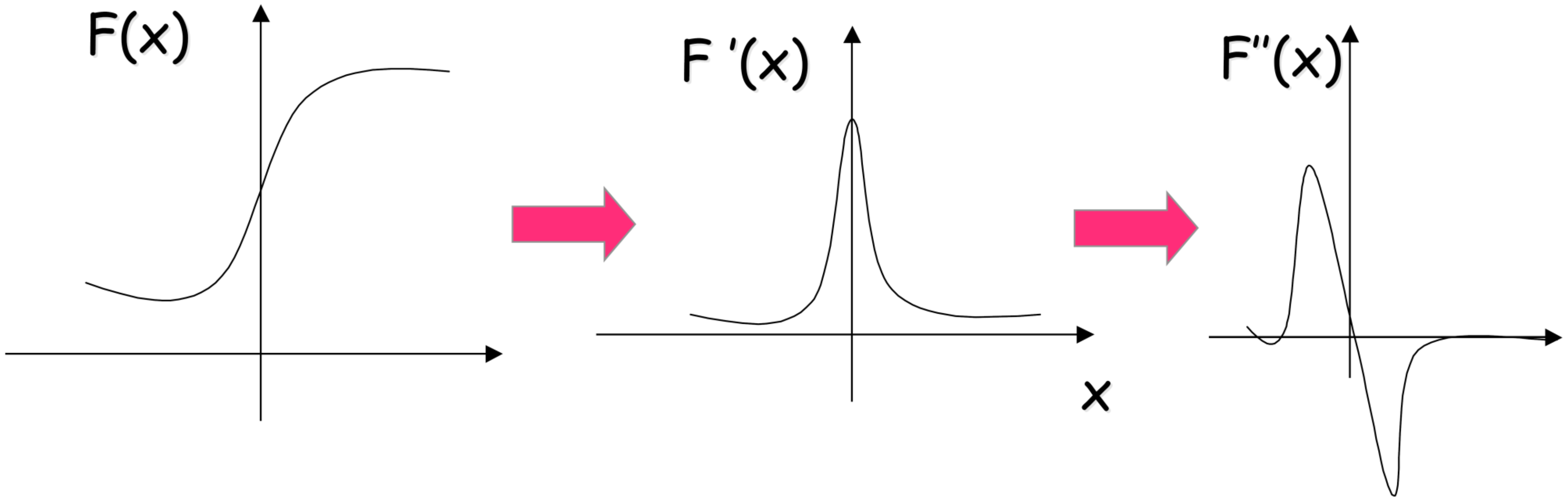
- Matching descriptors



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

Recall: Second Derivative Filters

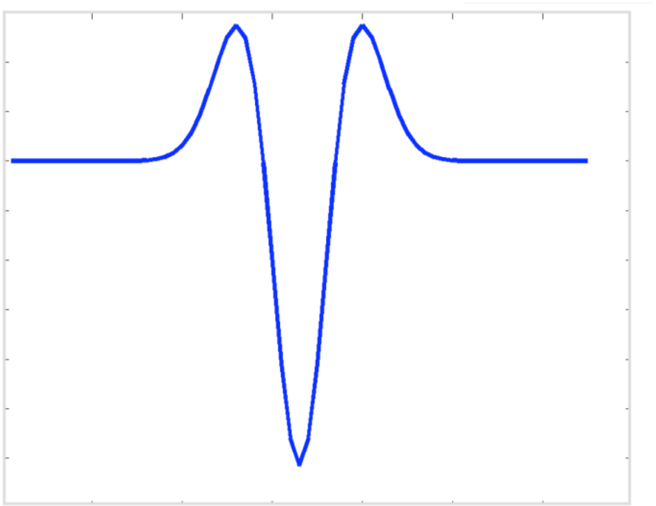
- Peaks or valleys of the first-derivative of the input signal, correspond to "zero-crossings" of the second-derivative of the input signal



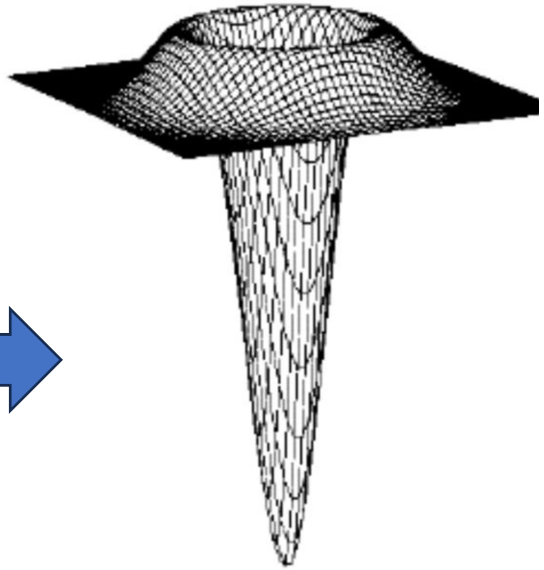
Recall: Second Derivate of Gaussian

$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}$$

$\nabla^2 h_\sigma(u, v)$



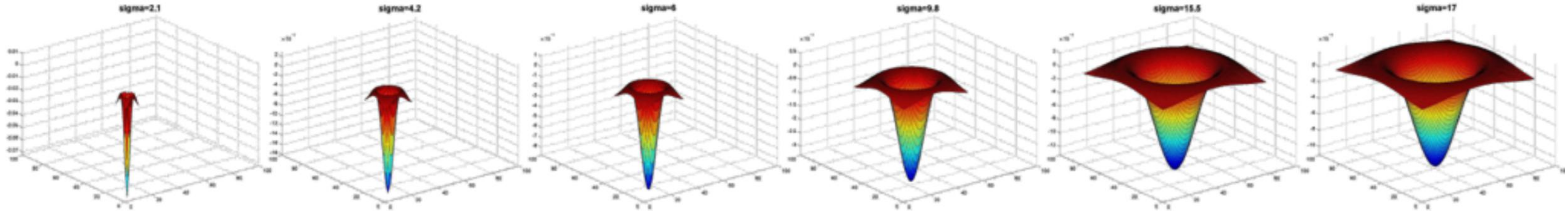
2D



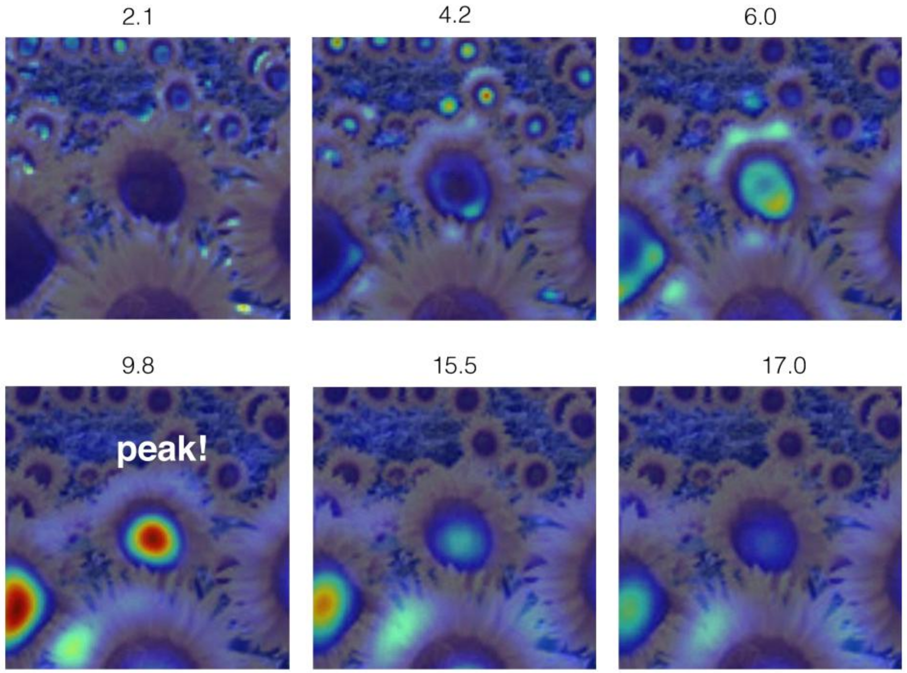
Laplacian of Gaussian

Mexican Hat Function

Laplacian of Gaussian for Scale Selection



Multi-scale
2D Blob detection



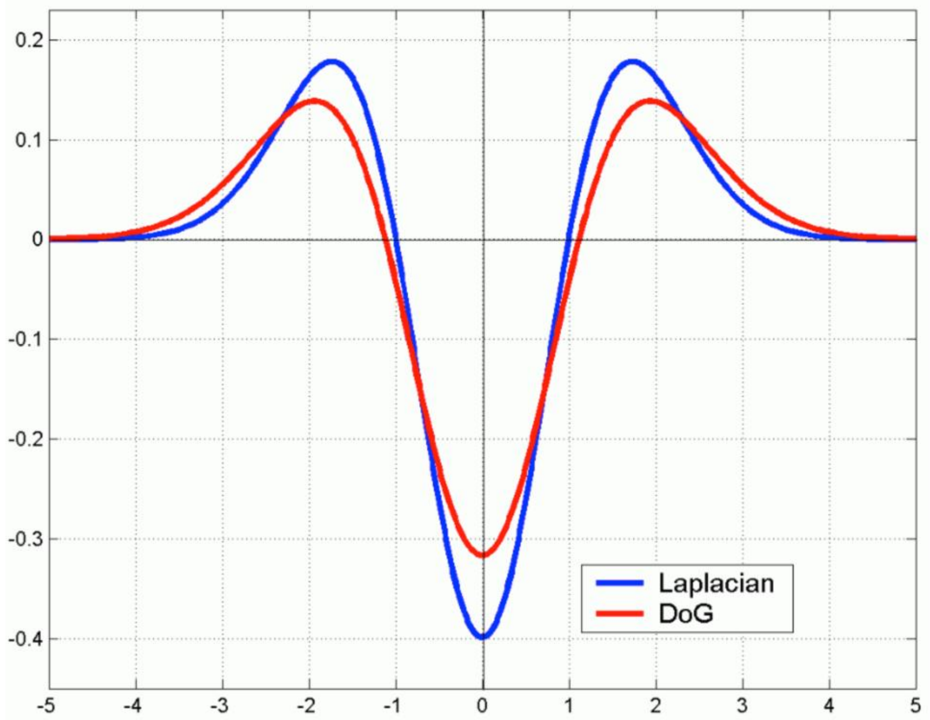
SIFT: Scale-space Extrema Detection

- Difference of Gaussian (DoG)

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

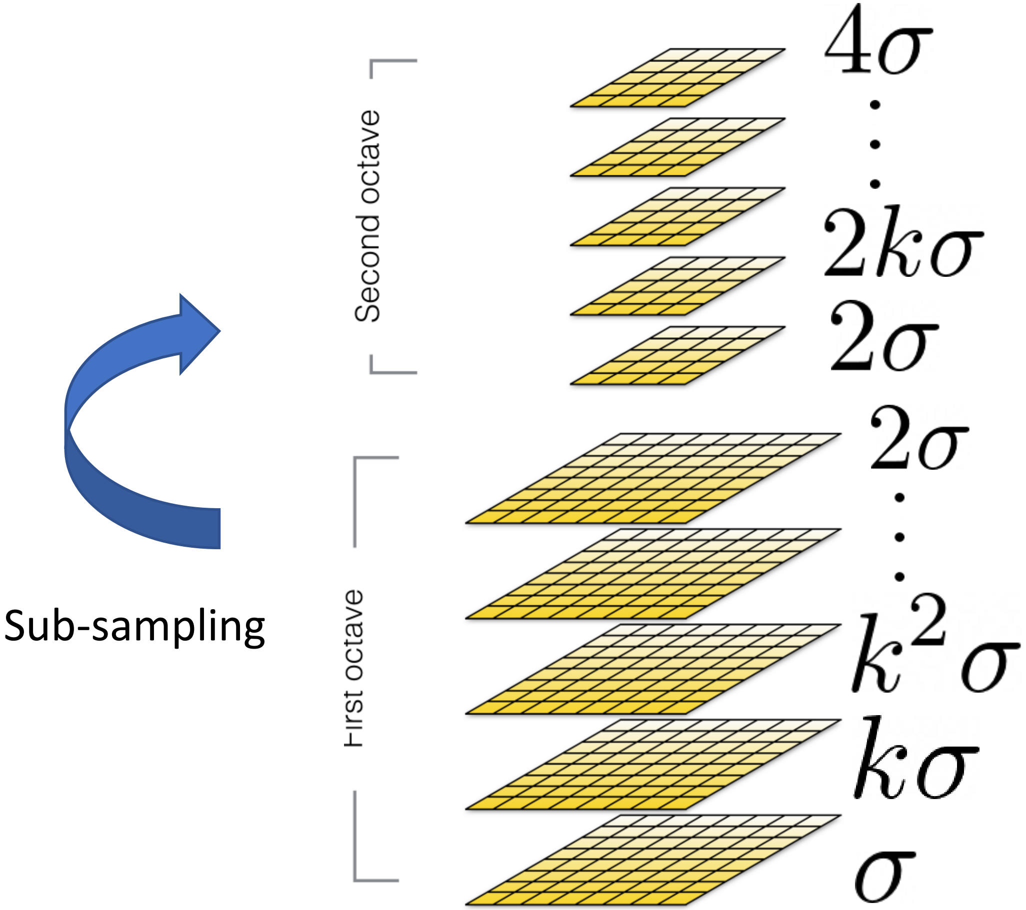


Approximate of Laplacian of Gaussian
(efficient to compute)

k is a constant

SIFT: Scale-space Extrema Detection

- Gaussian pyramid

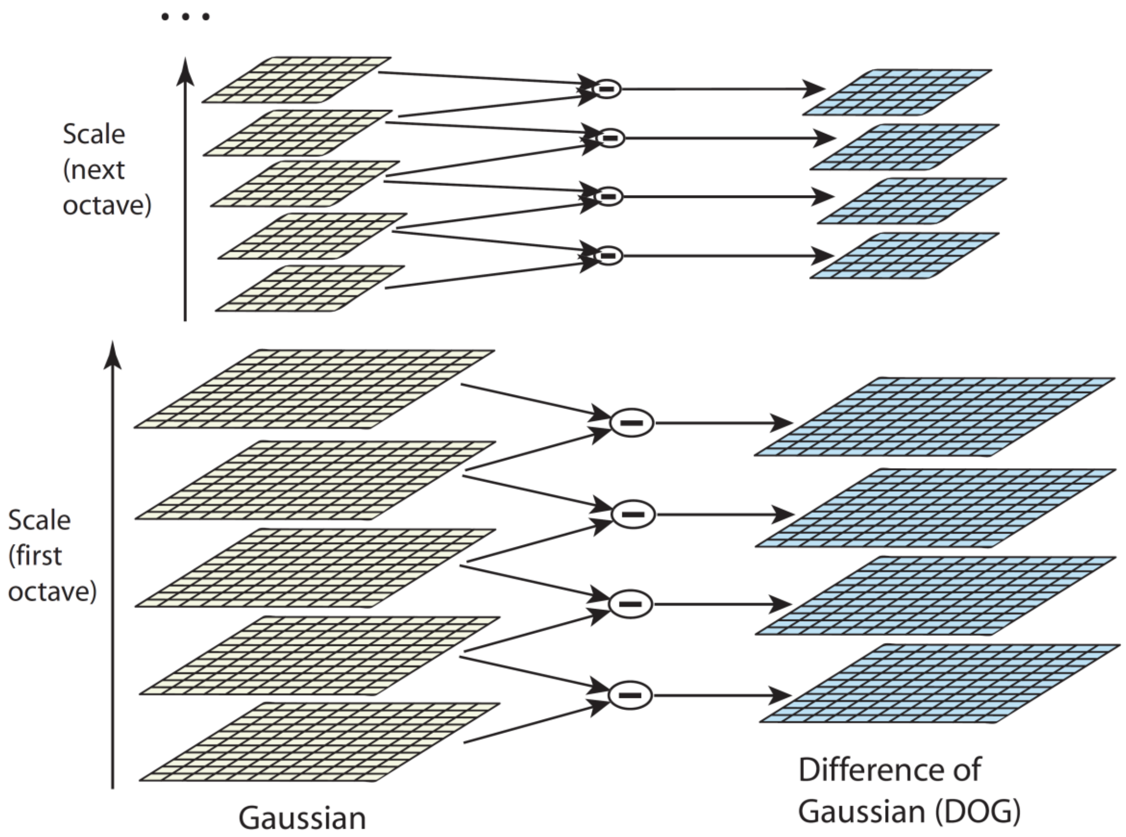


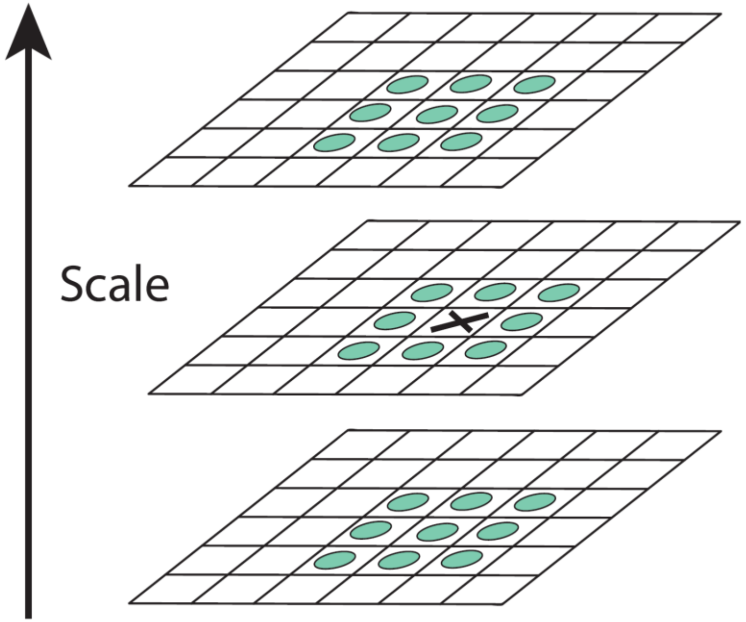
- Gaussian filters

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

- Sub-sampling by a factor of 2
  - Multiple the Gaussian kernel deviation by 2

SIFT: Scale-space Extrema Detection





Maxima and minima of DoG images

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

SIFT Descriptor

- Image gradient magnitude and orientation

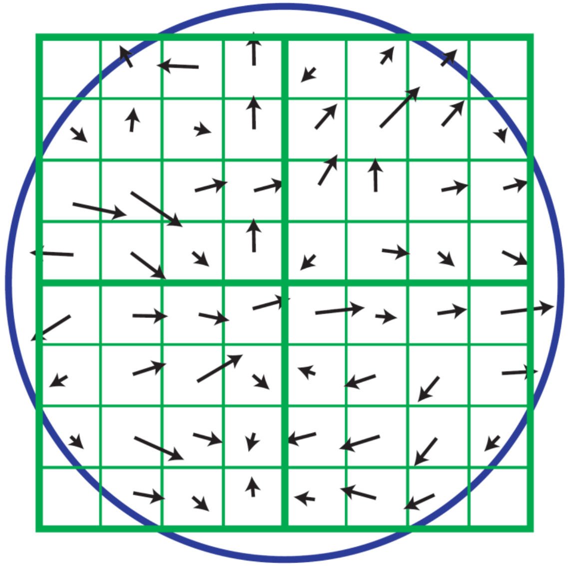


Image gradients

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

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

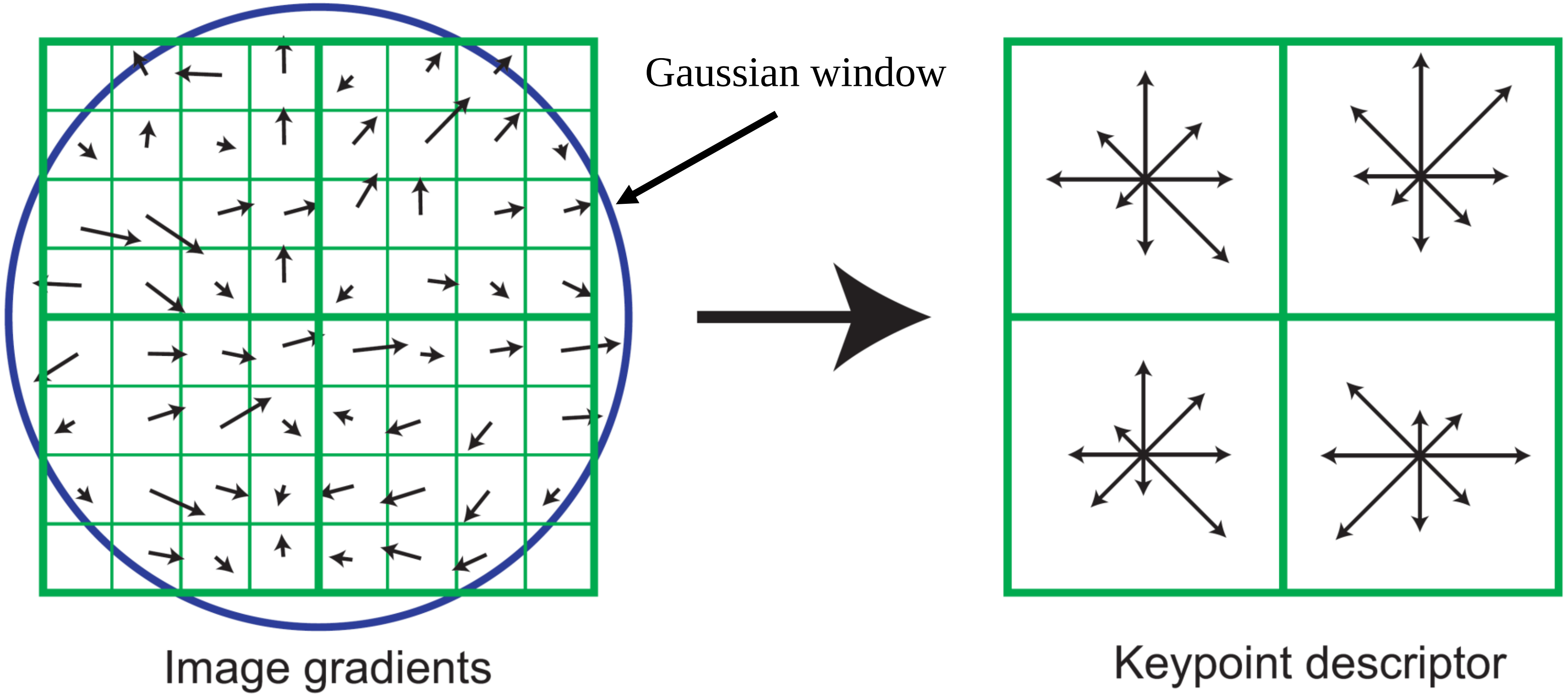
X-derivative Y-derivative

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

SIFT Descriptor

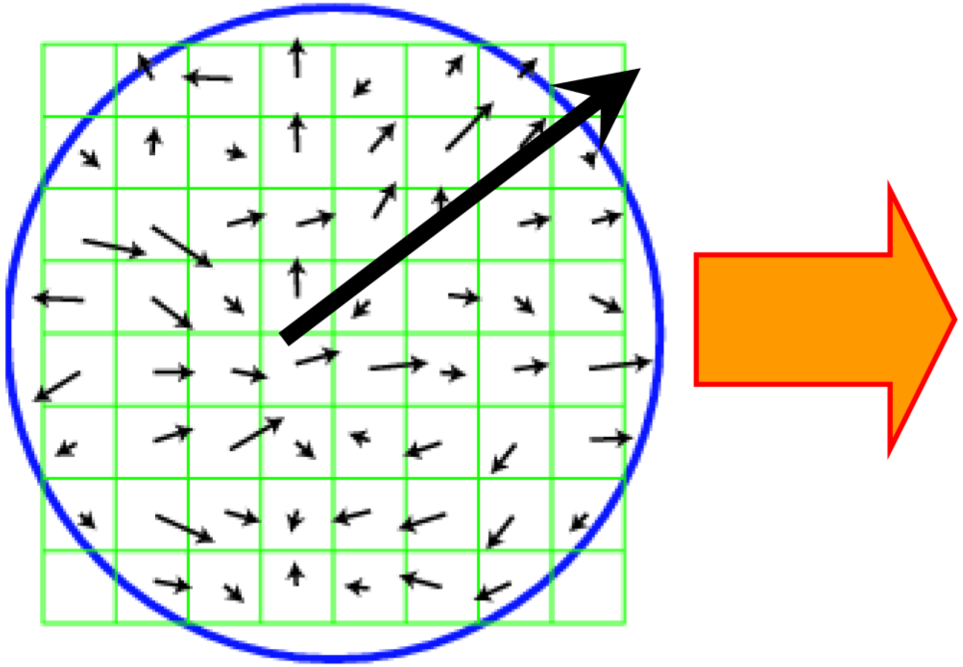
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

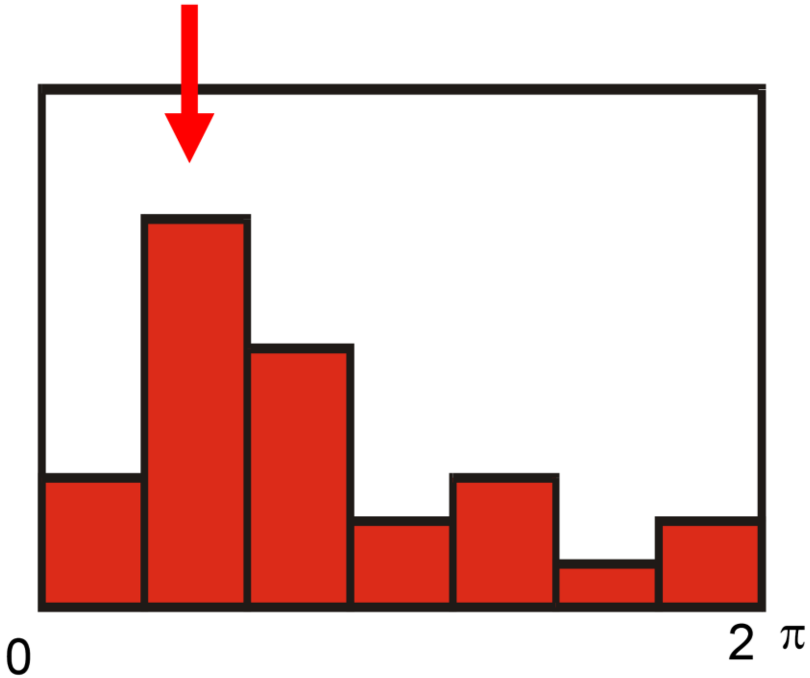
Using the scale of the keypoint to select the level of Gaussian blur for the image



SIFT: Rotation Invariance

- Rotate all orientations by the dominant orientation

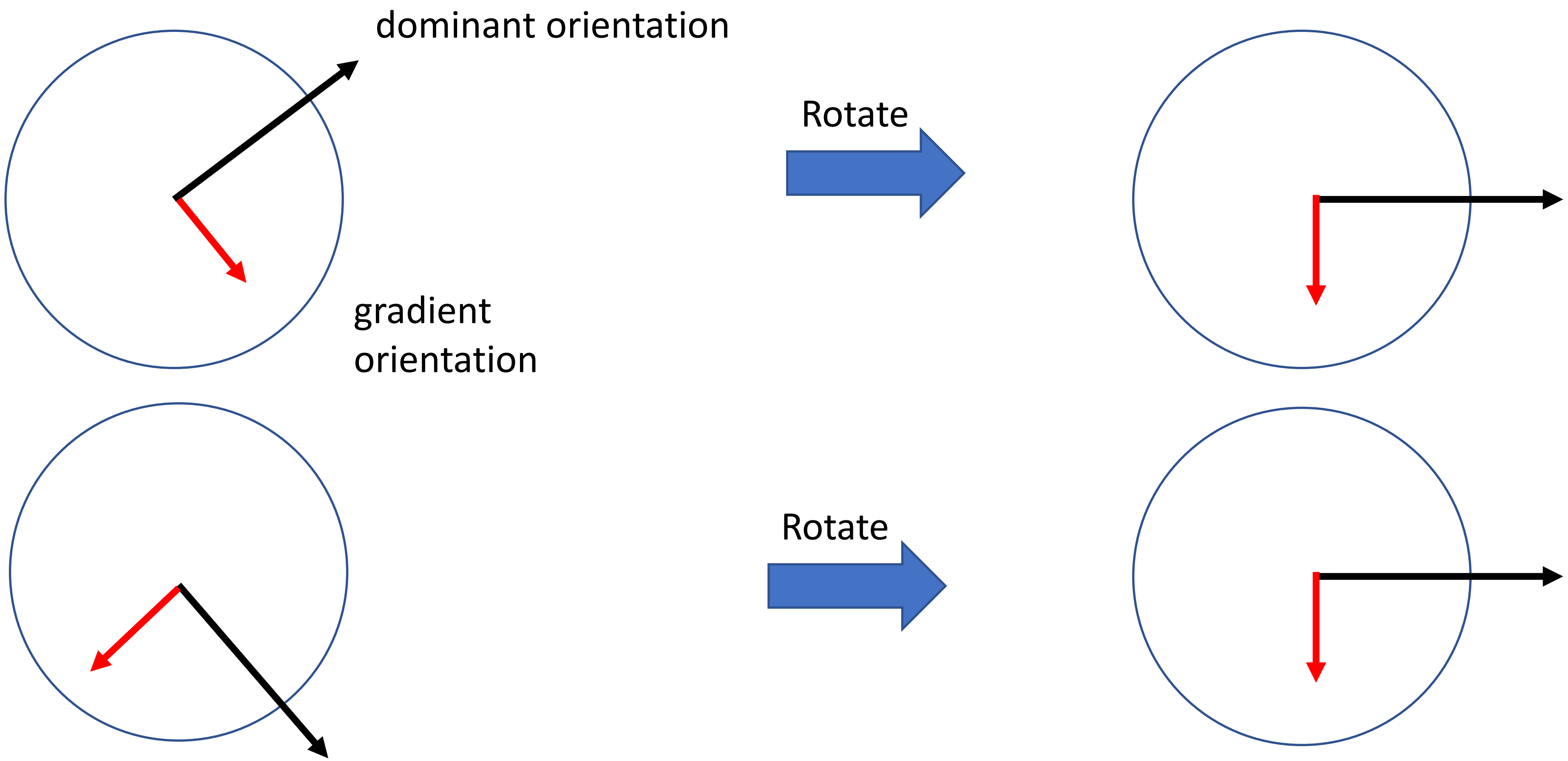




36 bins

SIFT: Rotation Invariance

- Rotate all orientations by the dominant orientation



SIFT Properties

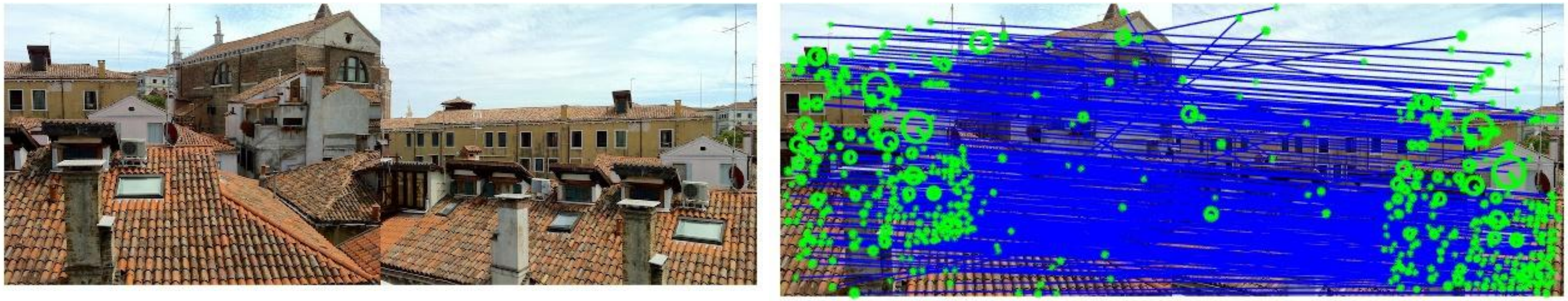
- Can handle change in viewpoint (up to about 60 degree out of plane rotation)

- Can handle significant change in illumination

- Relatively fast < 1s for moderate image sizes

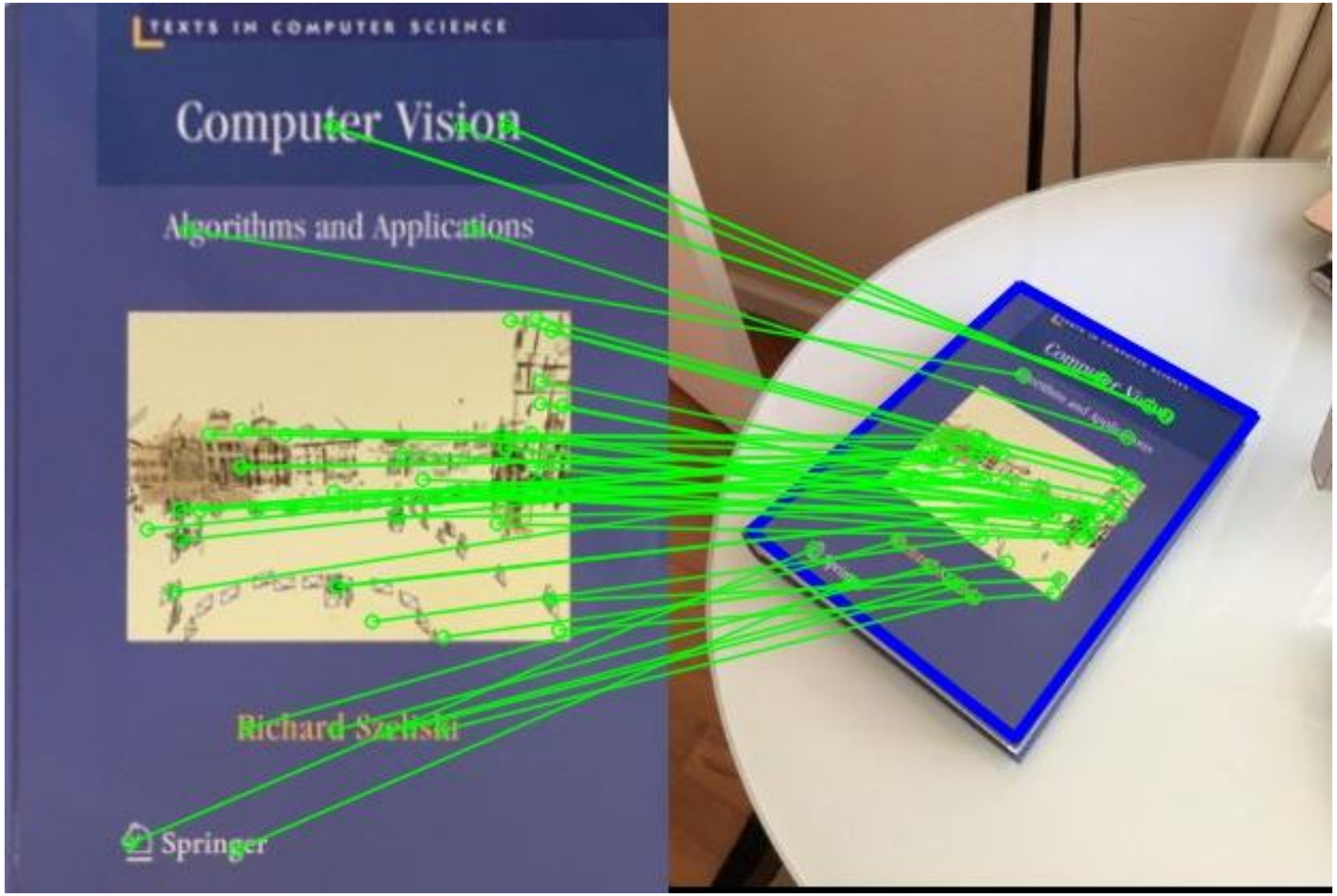
- Lots of code available
  - E.g., https://www.vlfeat.org/overview/sift.html

SIFT Matching Example

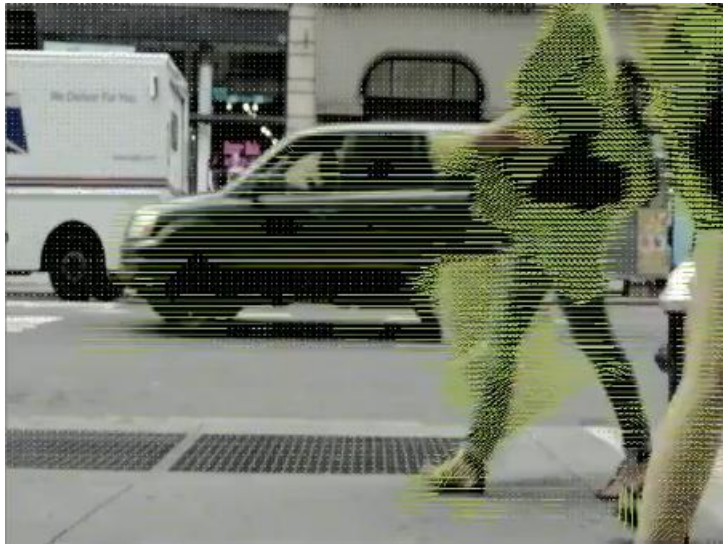


https://www.vlfeat.org/overview/sift.html

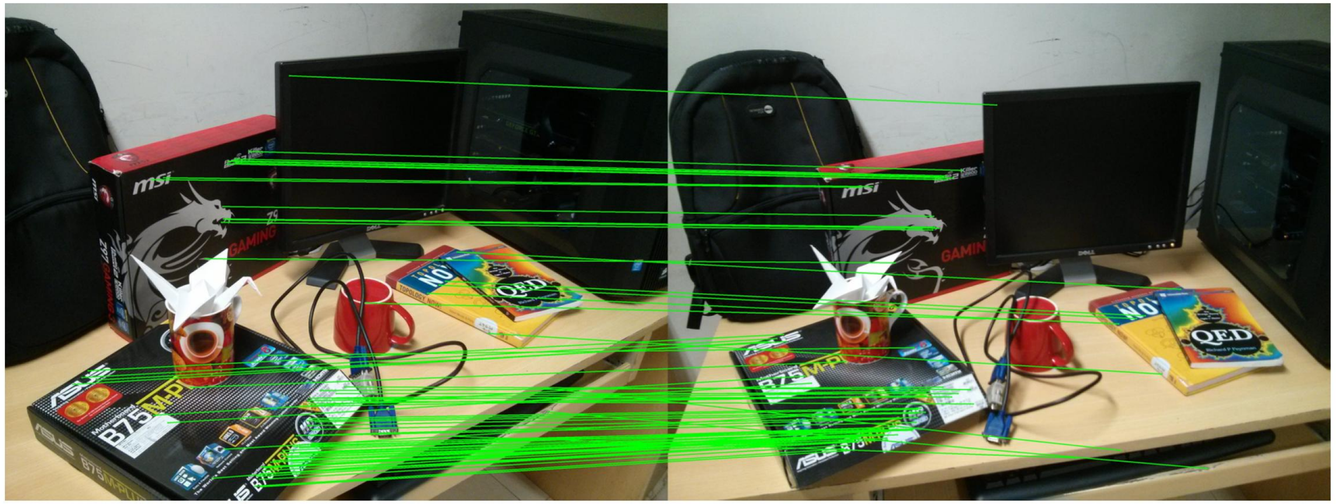
SIFT Matching Example



Correspondences



Optical flow

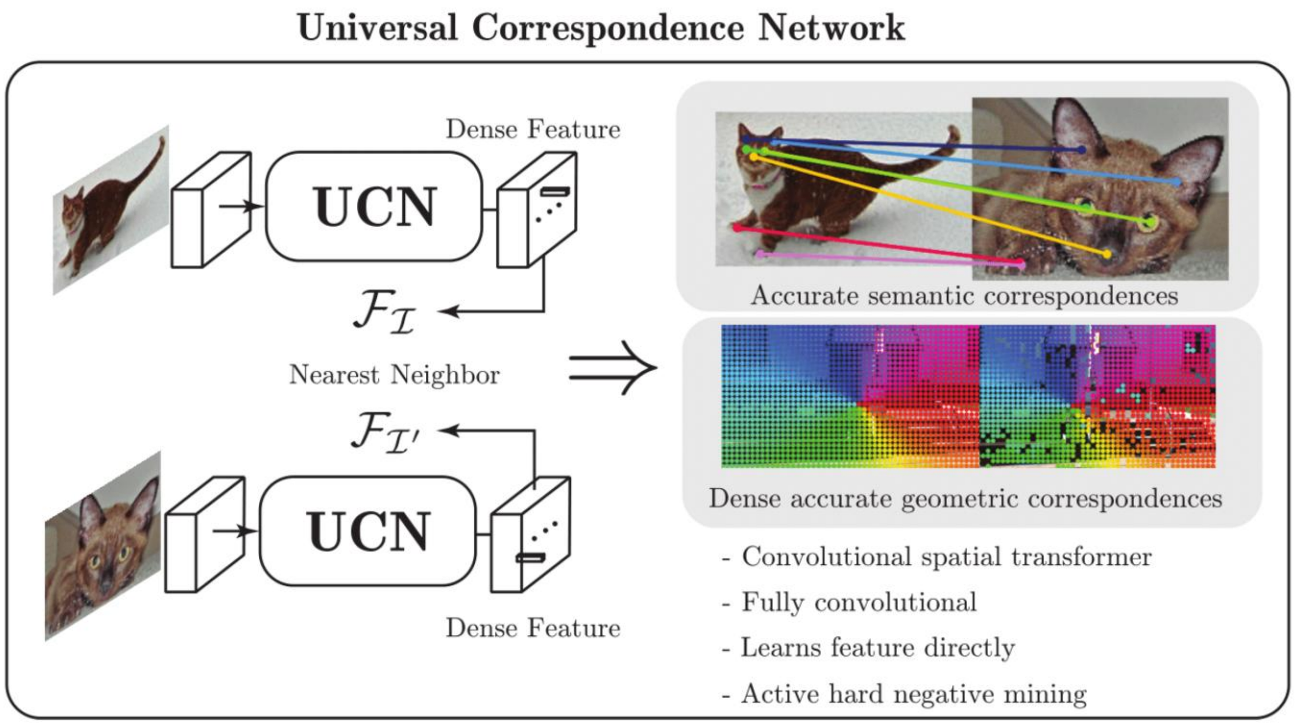


SIFT matching

Semantic keypoints

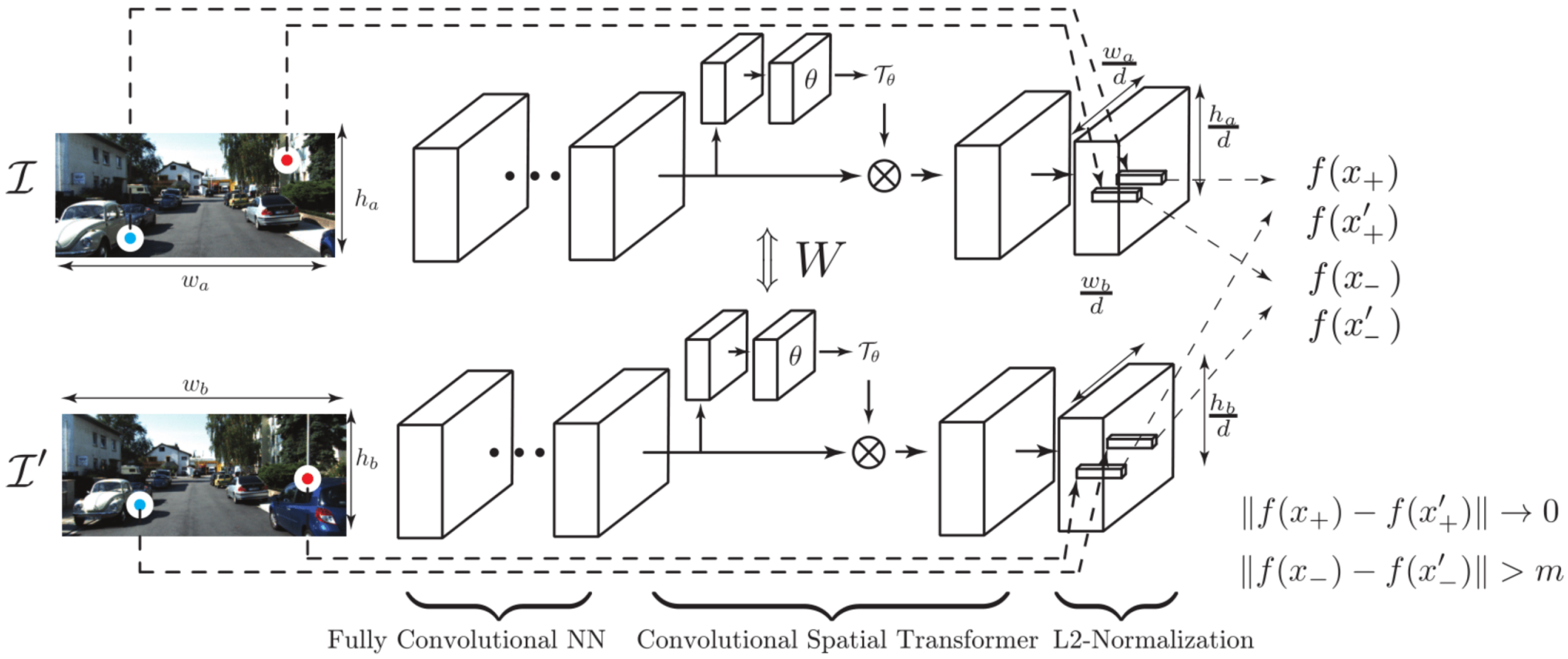
Advanced Topic: Universal Correspondences Network

- Learn pixel-wise features for matching
- Fully-convolutional network
- Contrastive loss function for feature learning
- Convolutional spatial transformer



Universal Correspondence Network. Choy et al., NuerIPS, 2016

Advanced Topic: Universal Correspondences Network



Universal Correspondence Network. Choy et al., NuerIPS, 2016

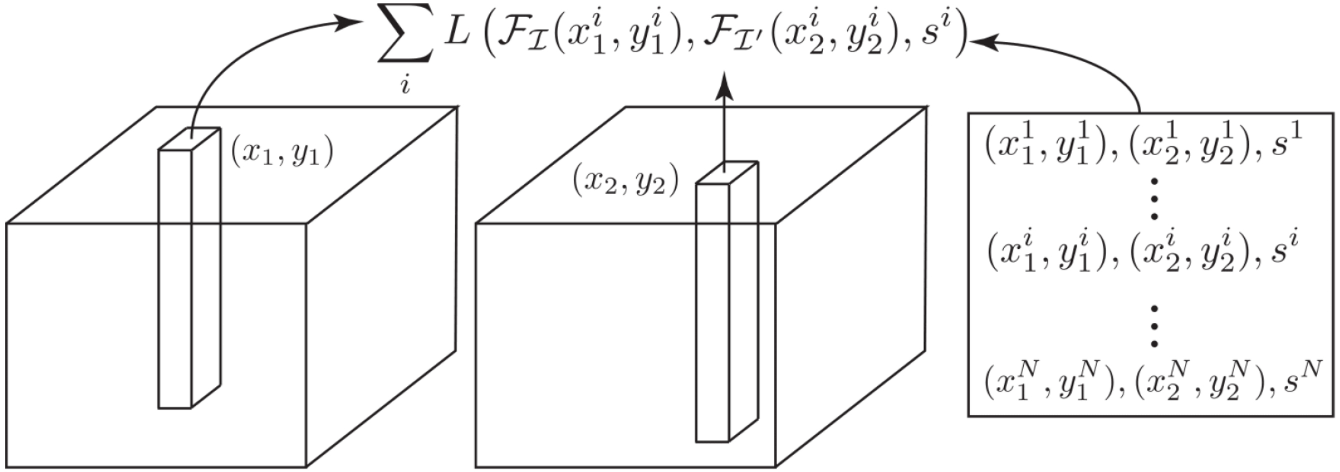
Advanced Topic: Universal Correspondences Network

- Correspondence contrastive loss

$$L = \frac{1}{2N}\sum_{i}^{N} s_i \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}_i) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|^2 + (1 - s_i)\max(0, m - \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|)^2$$

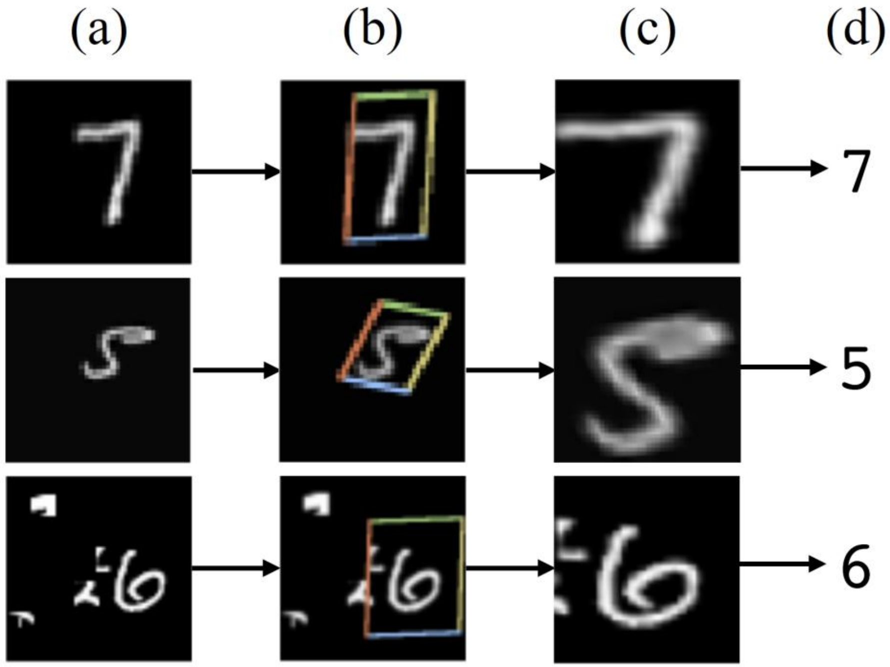
positive pair

negative pair

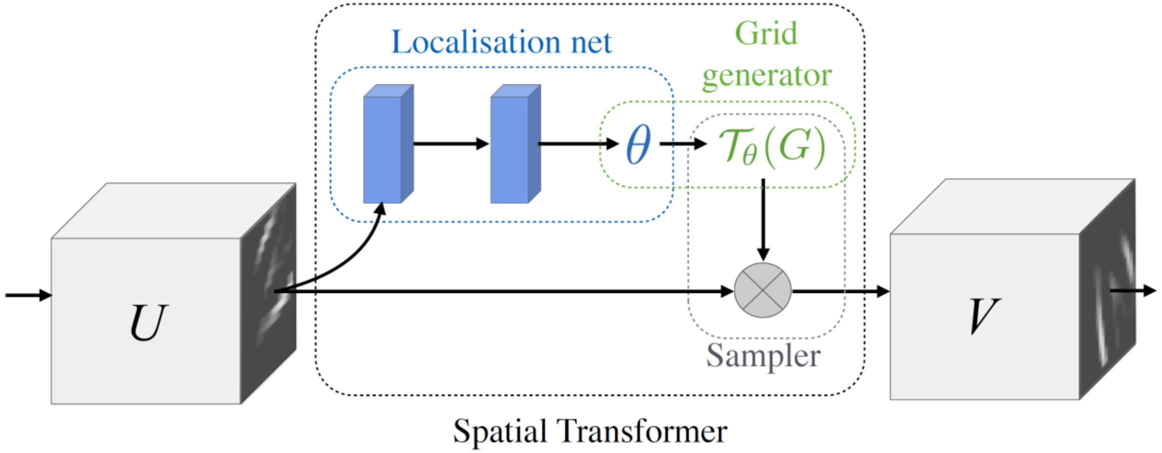


Universal Correspondence Network. Choy et al., NuerIPS, 2016

Spatial Transformer Network

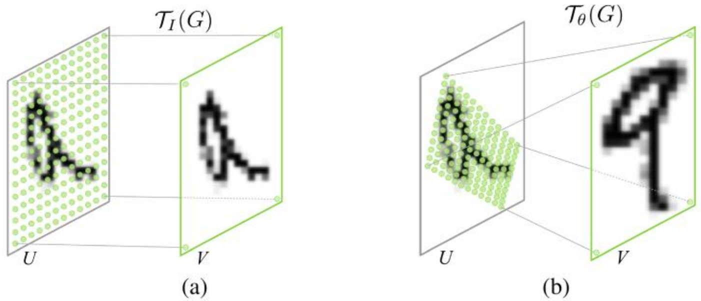


Input localization Transformation



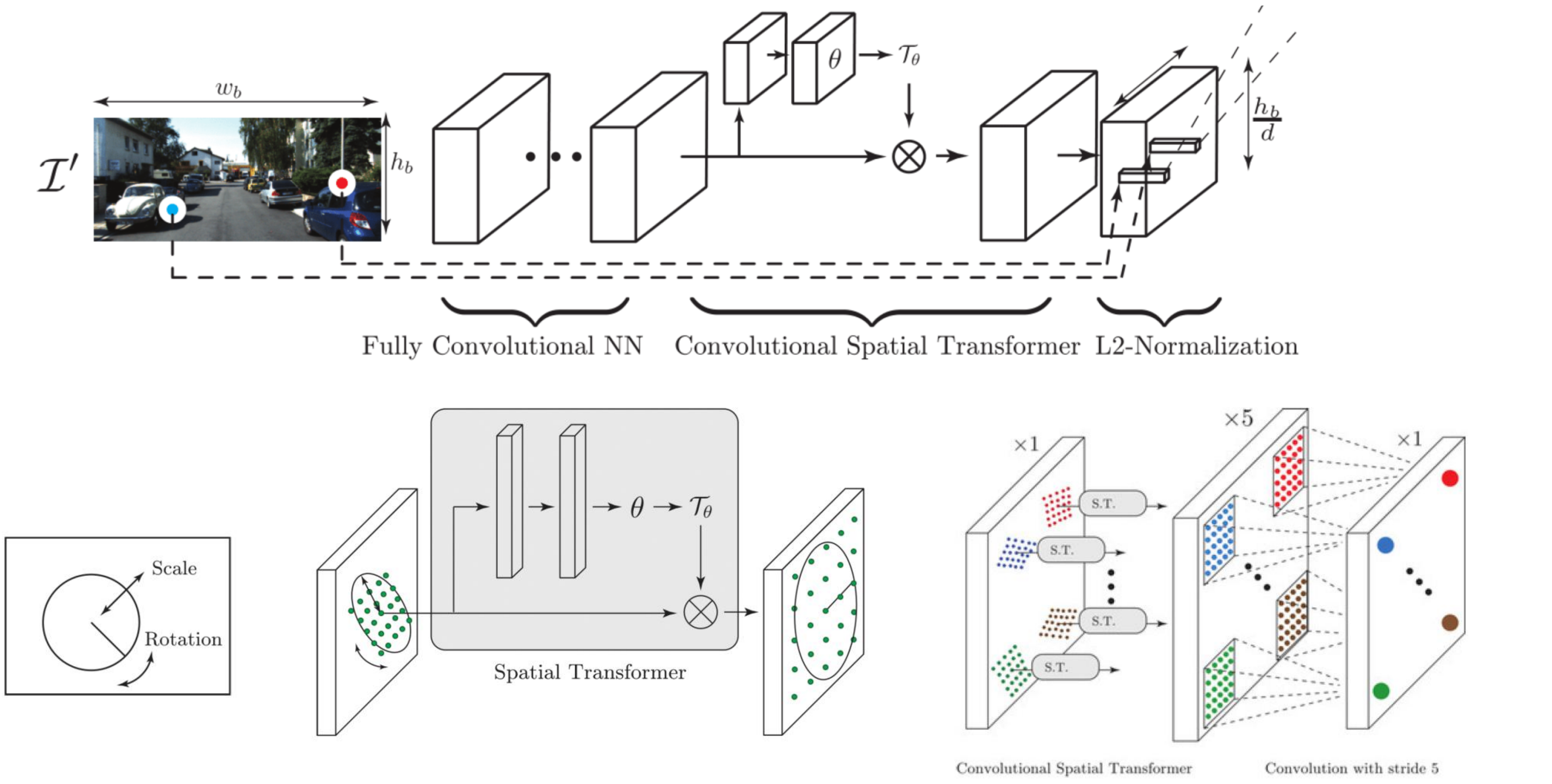
Affine transformation

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Spatial Transformer Networks. Jaderberg et al., NeurIPS, 2015

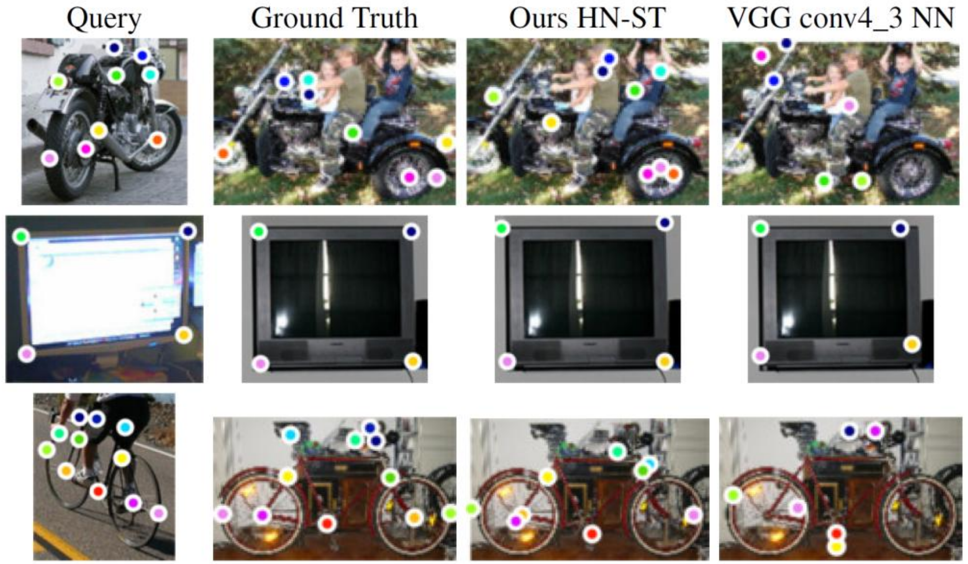
Advanced Topic: Universal Correspondences Network

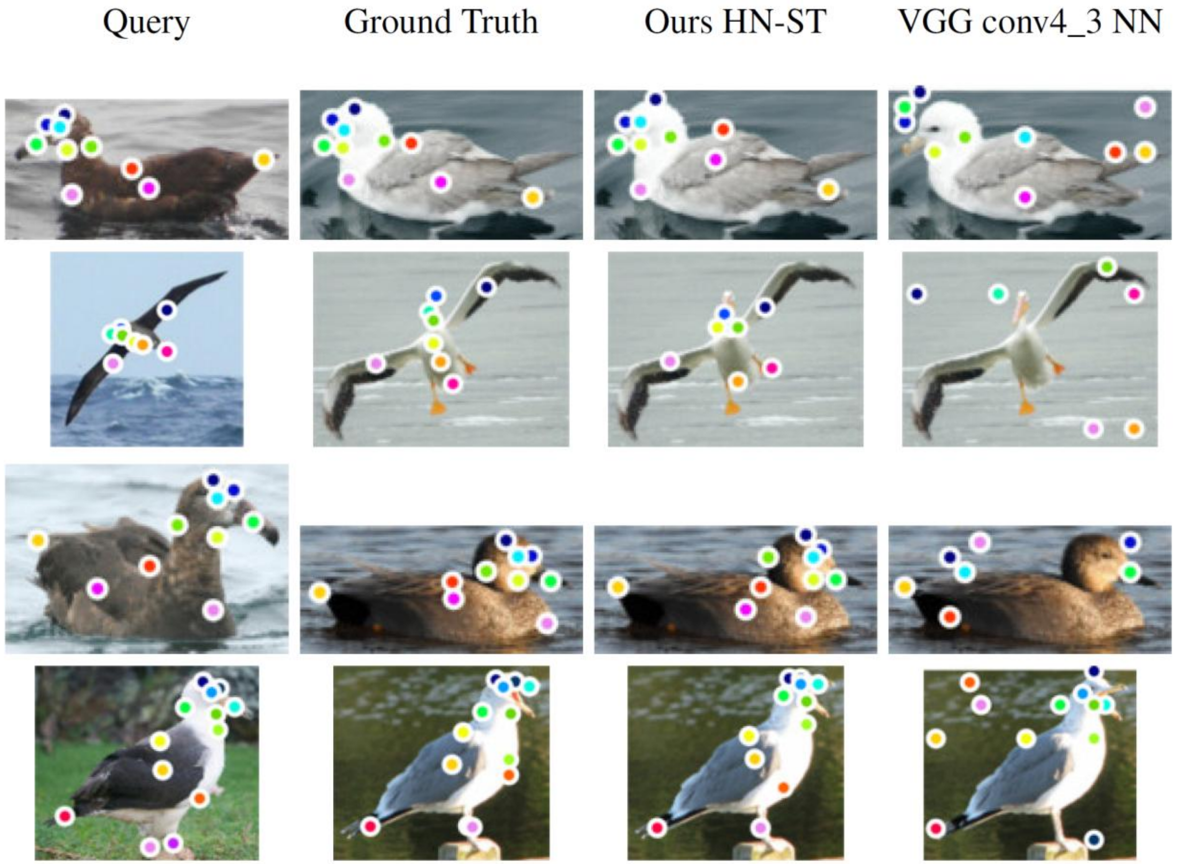


(a) SIFT (b) Spatial transformer (c) Convolutional spatial transformer

Universal Correspondence Network. Choy et al., NuerIPS, 2016

Advanced Topic: Universal Correspondences Network





Universal Correspondence Network. Choy et al., NuerIPS, 2016

Further Reading

- Section 7.1, Computer Vision, Richard Szeliski
- David Lowe, Distinctive Image Features from Scale-Invariant Keypoints. IJCV, 2004 https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf
- ORB: An efficient alternative to SIFT or SURF. Rublee et al., ICCV, 2011
- Universal Correspondence Network, 2016 https://arxiv.org/abs/1606.03558