Vision and Cognitive Systems

SCQ1097939 - LM CS,DS,CYB,PD

Foundations: local invariant features, descriptors

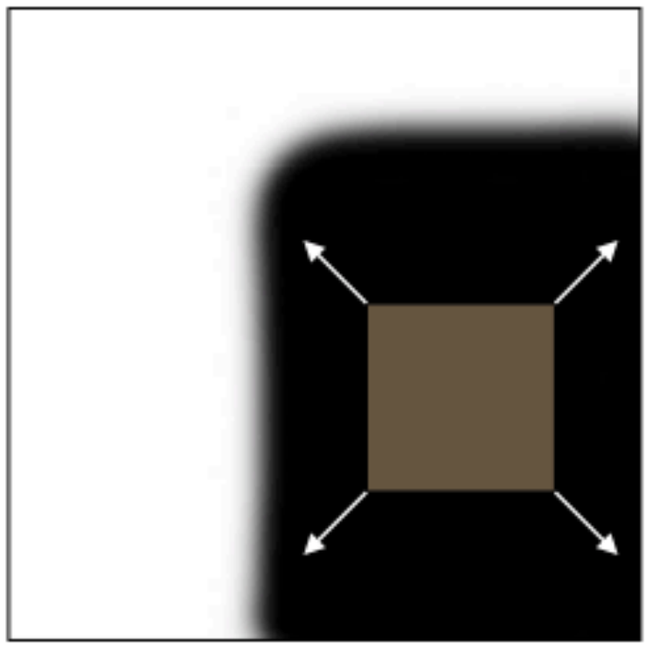
Prof. Lamberto Ballan

University of Padova, A.Y. 2023/24 November 7, 2023

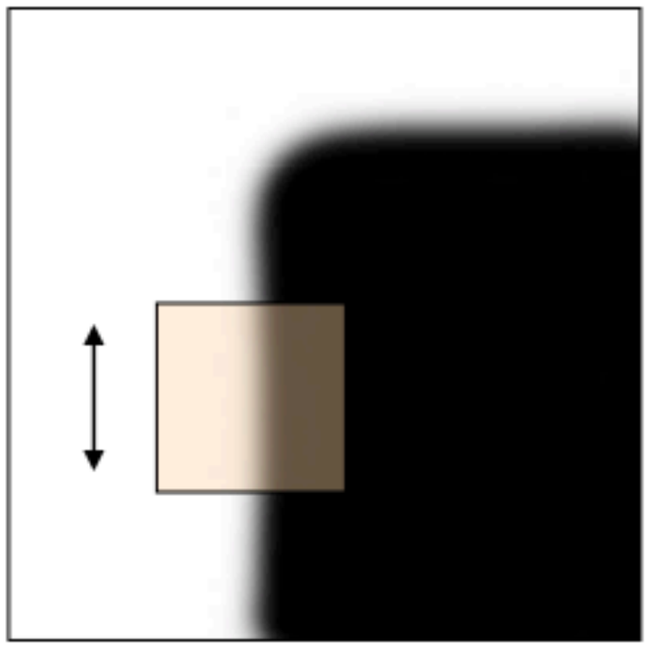
DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

Recap: edges, corners and blobs

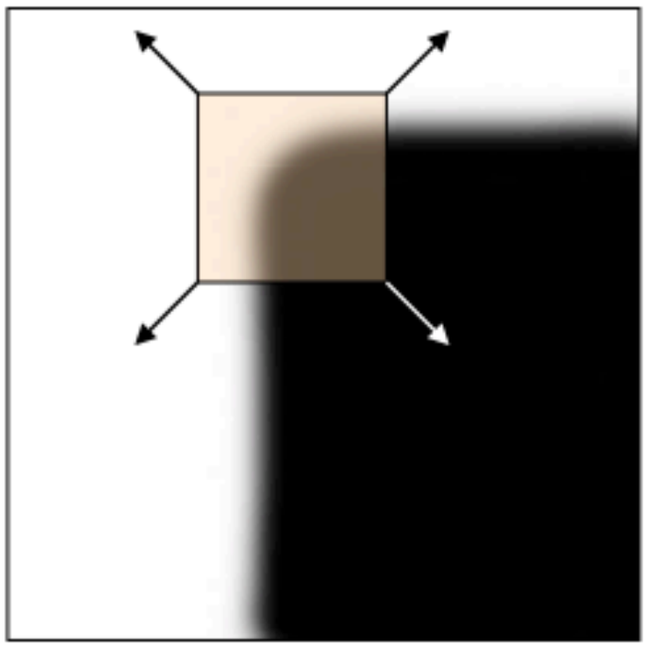
- Low-level image features: a basic classification



“flat” region:
no change in all directions



“edge”:
no change along the edge direction

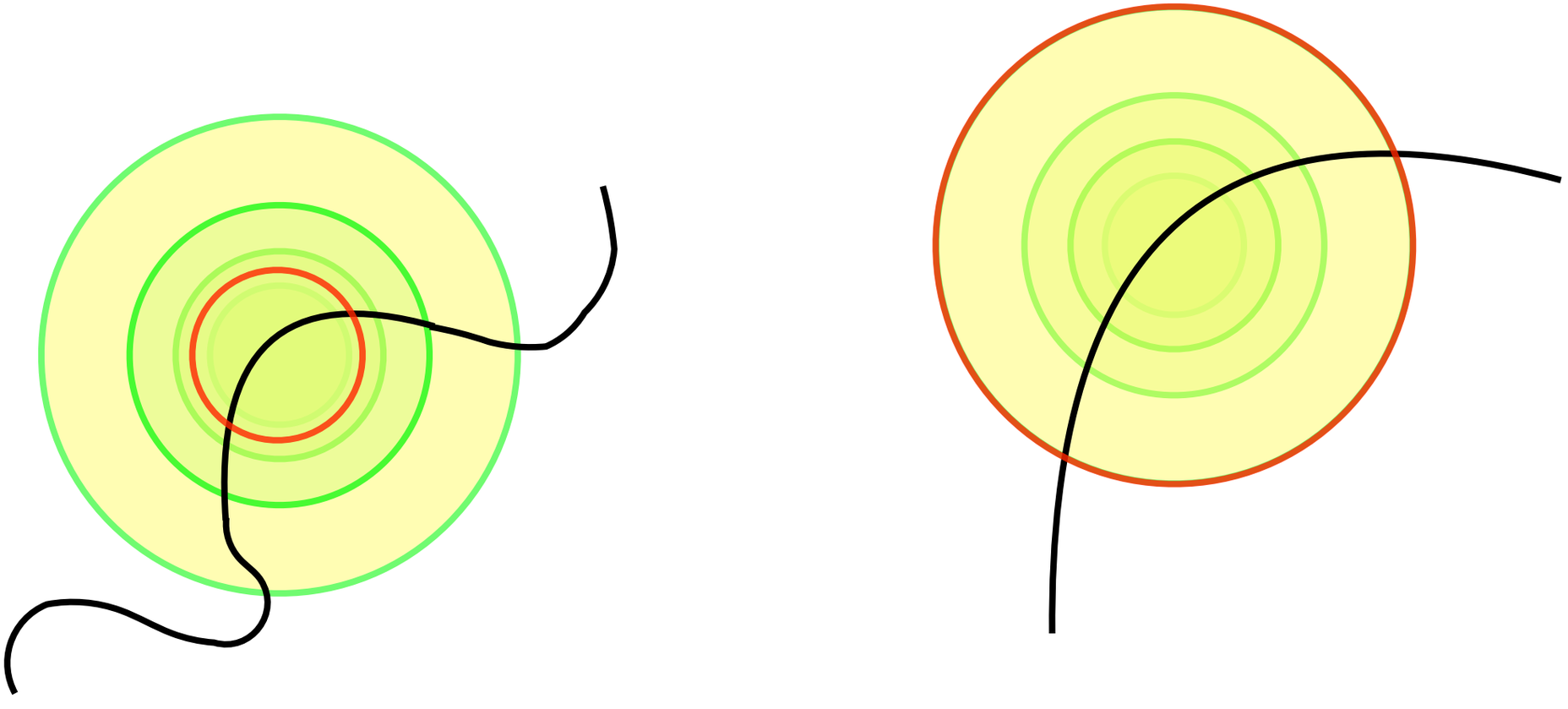


“corner”:
significant change in all directions

Slide credit: A. Efros

Recap: scale invariant detection

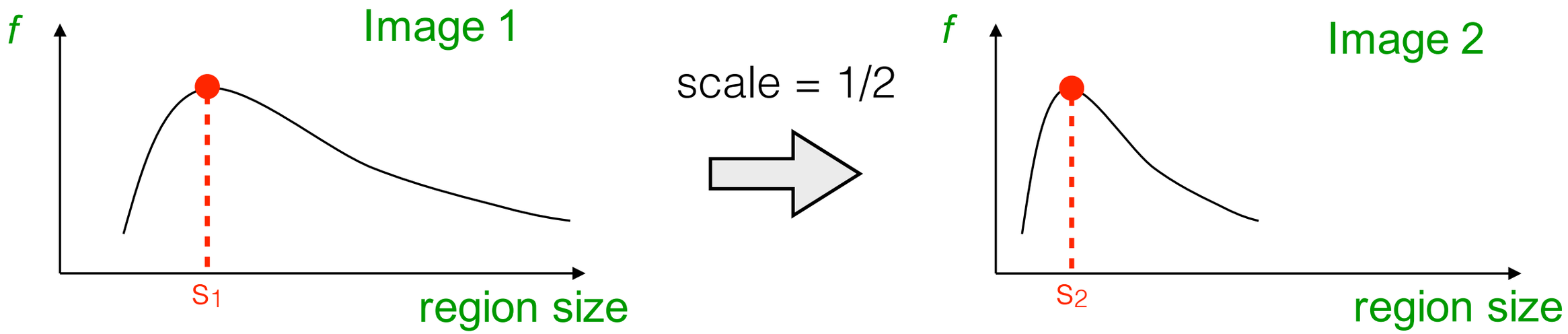
- The problem: how do we choose corresponding circles **independently** in each image?



Recap: scale invariant detection

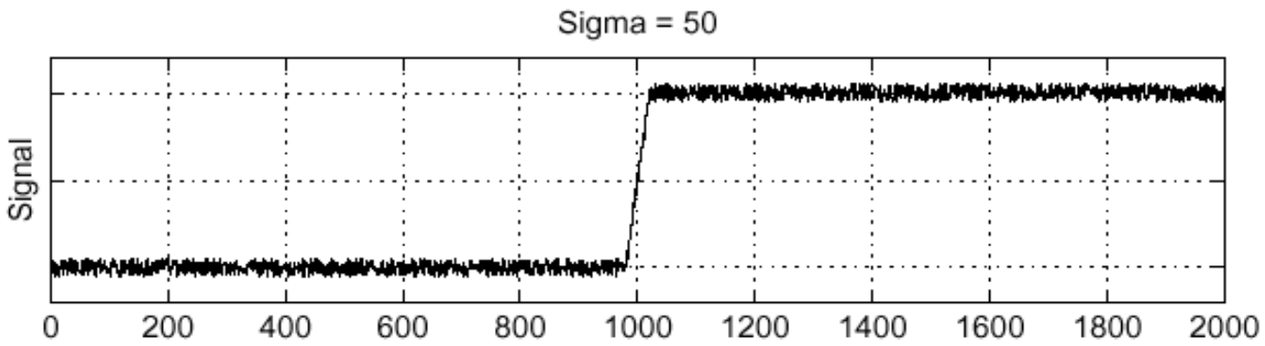
- Common approach: take a local max of this function
- Observation: region size for which the maximum is achieved should be *invariant* (*co-variant*) with scale

Important: this scale invariant region size is found in each image **independently**!



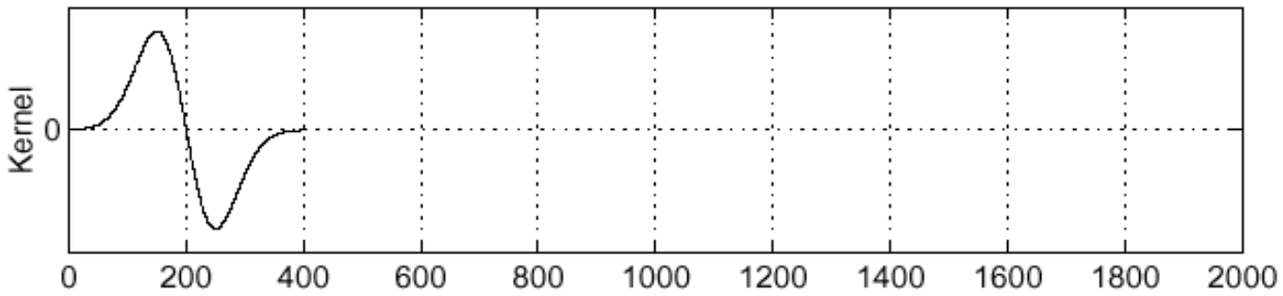
Recall: edge detection

f



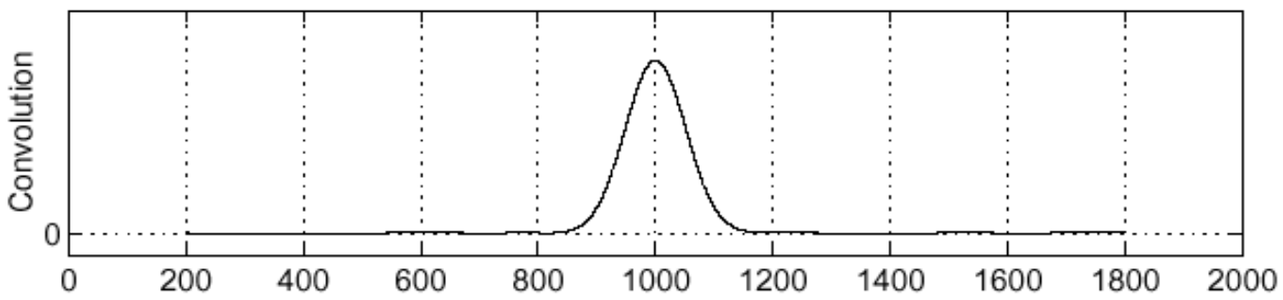
Edge

$\frac{d}{dx} g$



Derivative of Gaussian

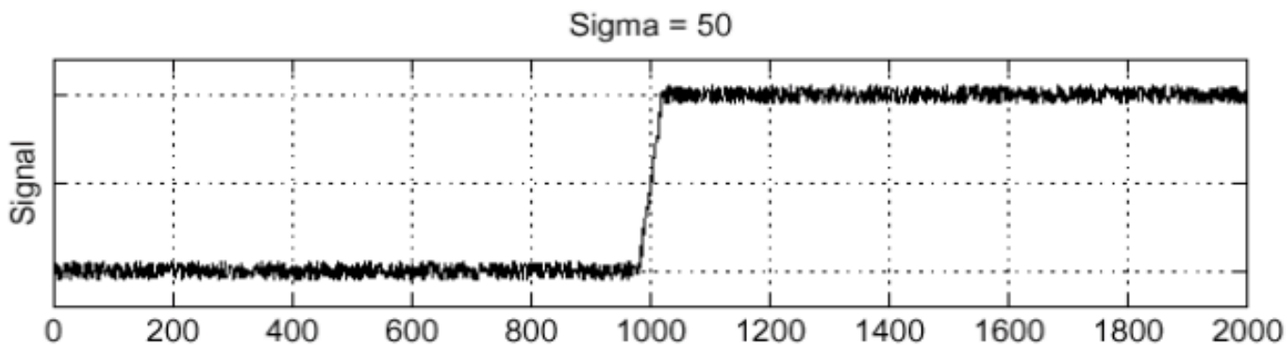
$f * \frac{d}{dx} g$



Edge = maximum of derivative

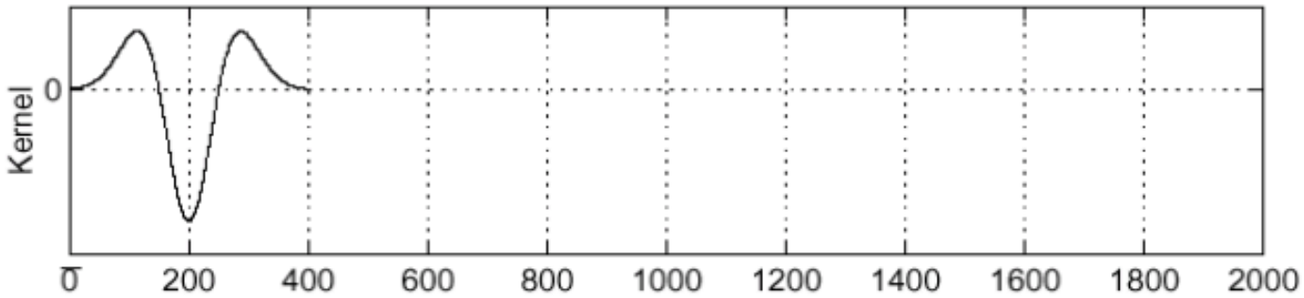
A bit more on edge detection

f



Edge

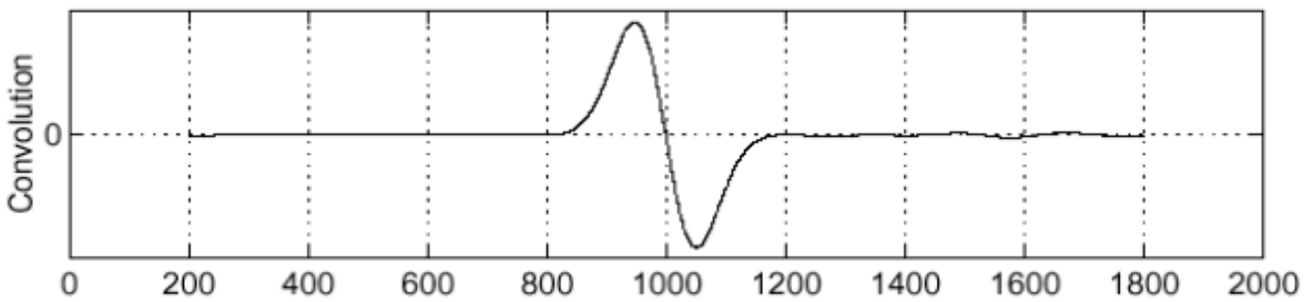
$\frac{d^2}{dx^2} g$



Second Derivative of Gaussian (Laplacian)

Mexican hat filter

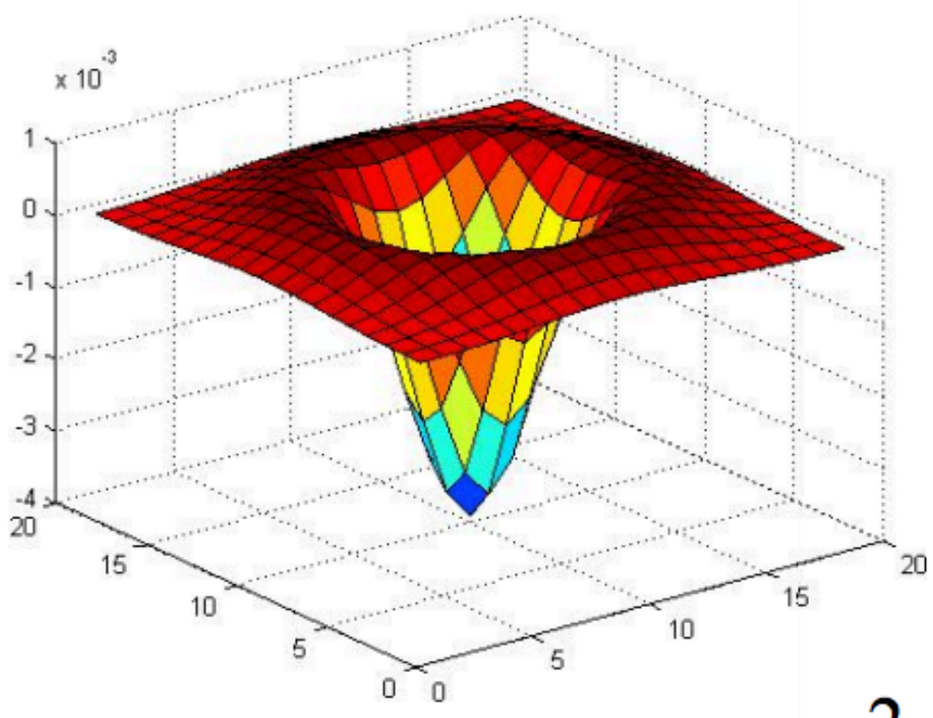
$f * \frac{d^2}{dx^2} g$

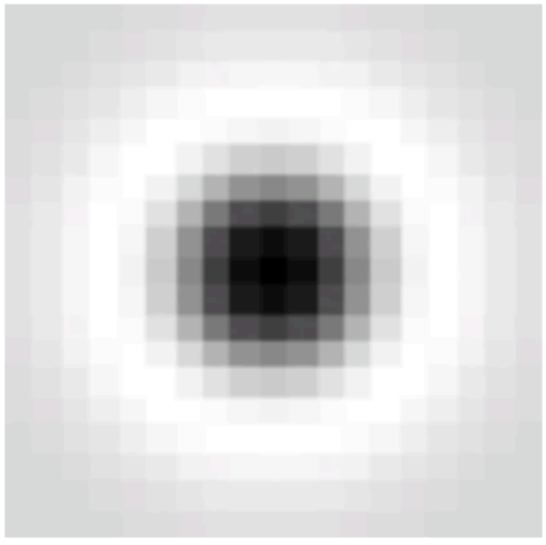


Edge = zero crossing of second derivative

Blob filter

- **Laplacian of Gaussian (LoG)**: circularly symmetric operator for blob detection in 2D

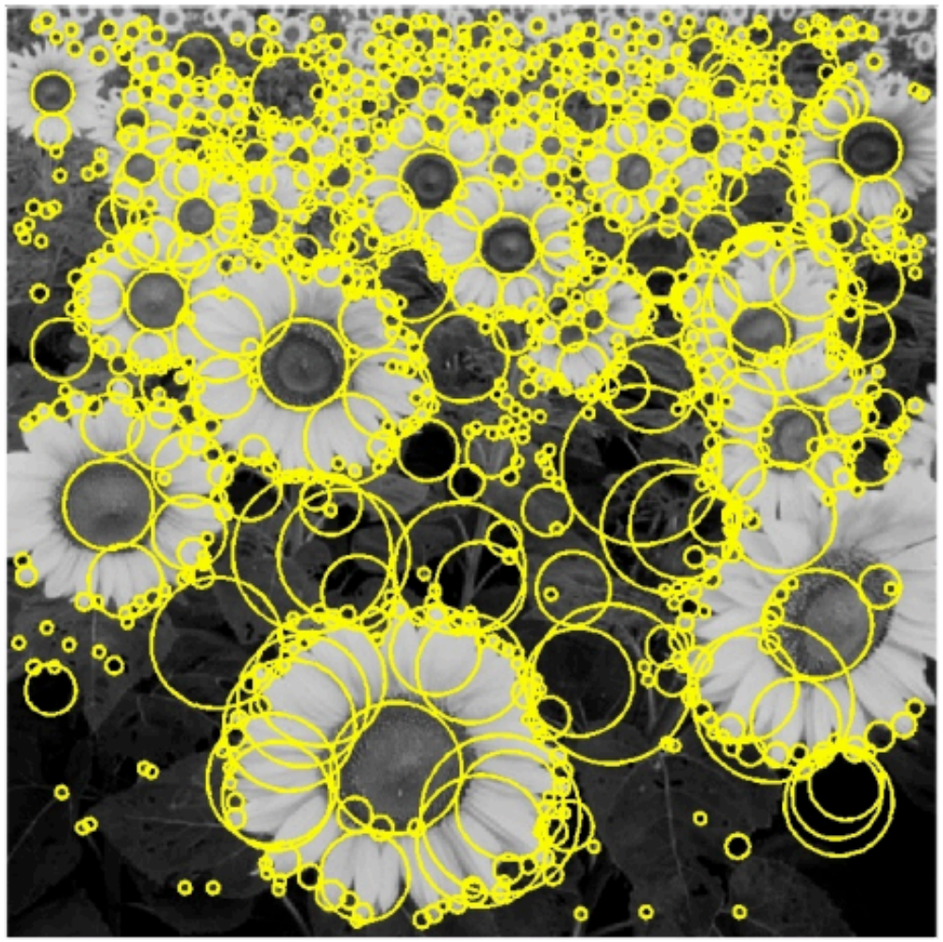


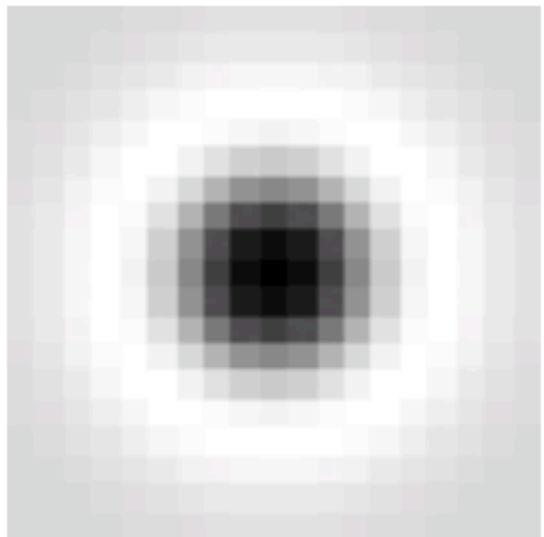


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection: key idea

- To detect “blobs”, convolve the image with a blob filter at multiple scales and look for extrema of filter response in the resulting *scale space*

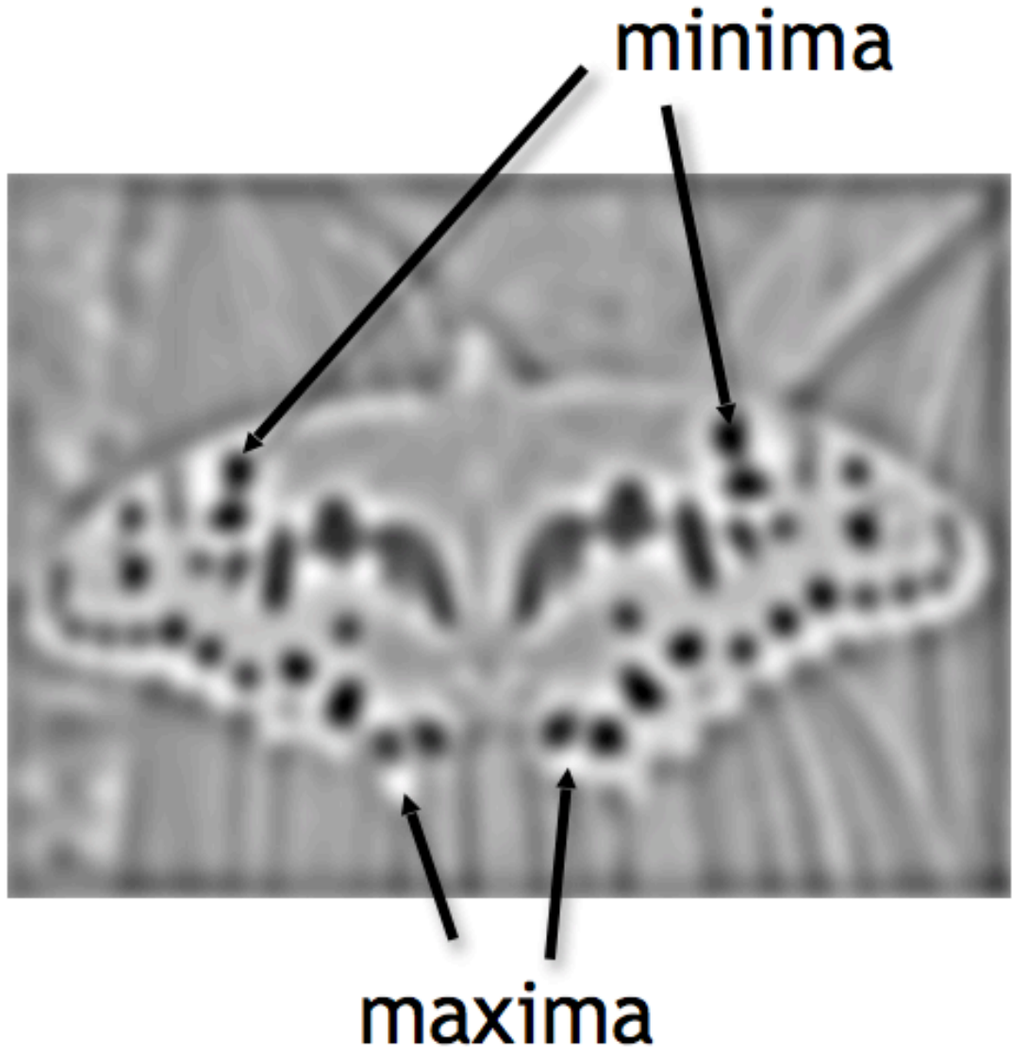




Blob filter

Blob detection: key idea

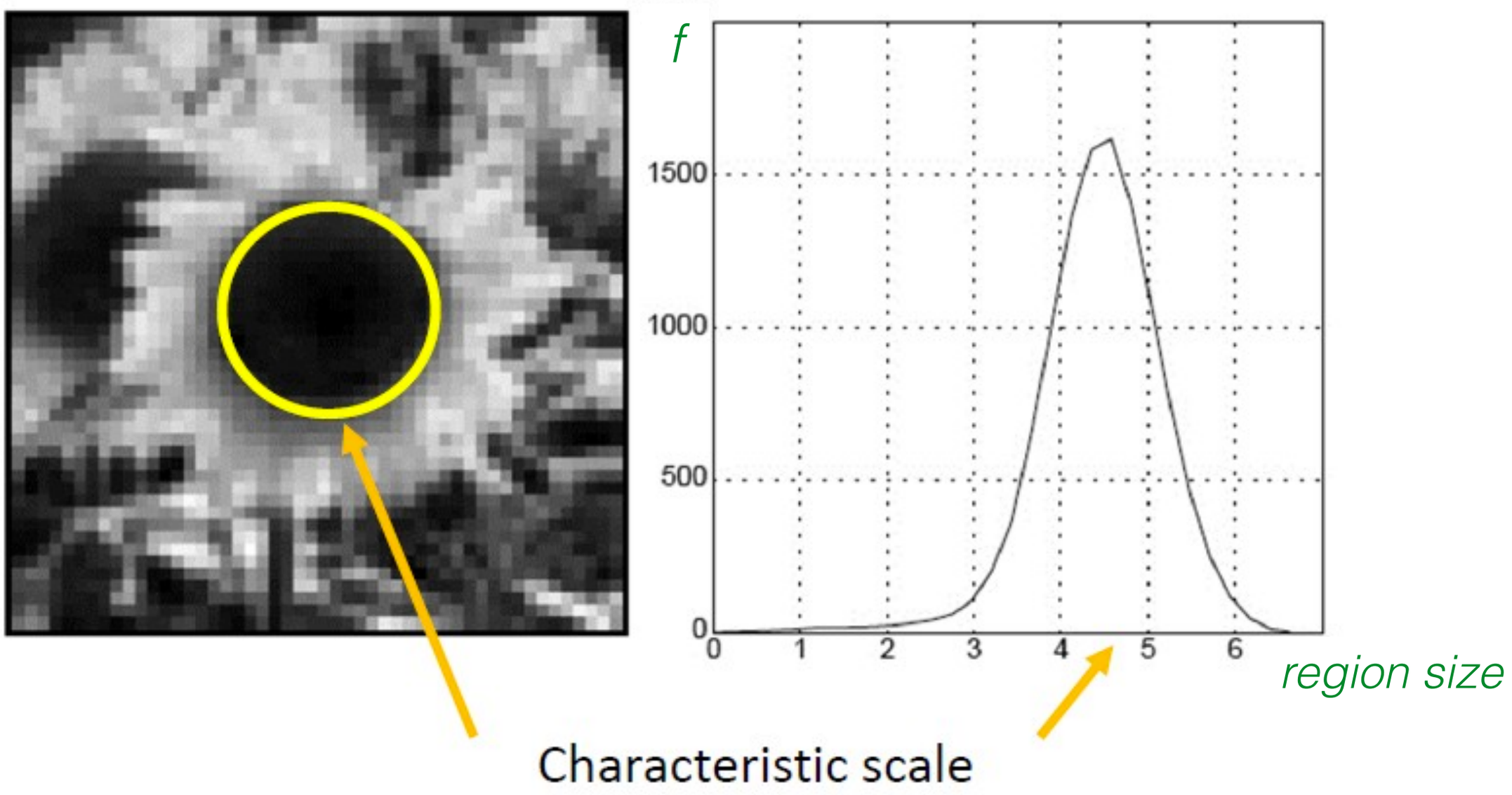
- Find maxima and minima of blob filter response in space and scale



Slide credit: N. Snavely

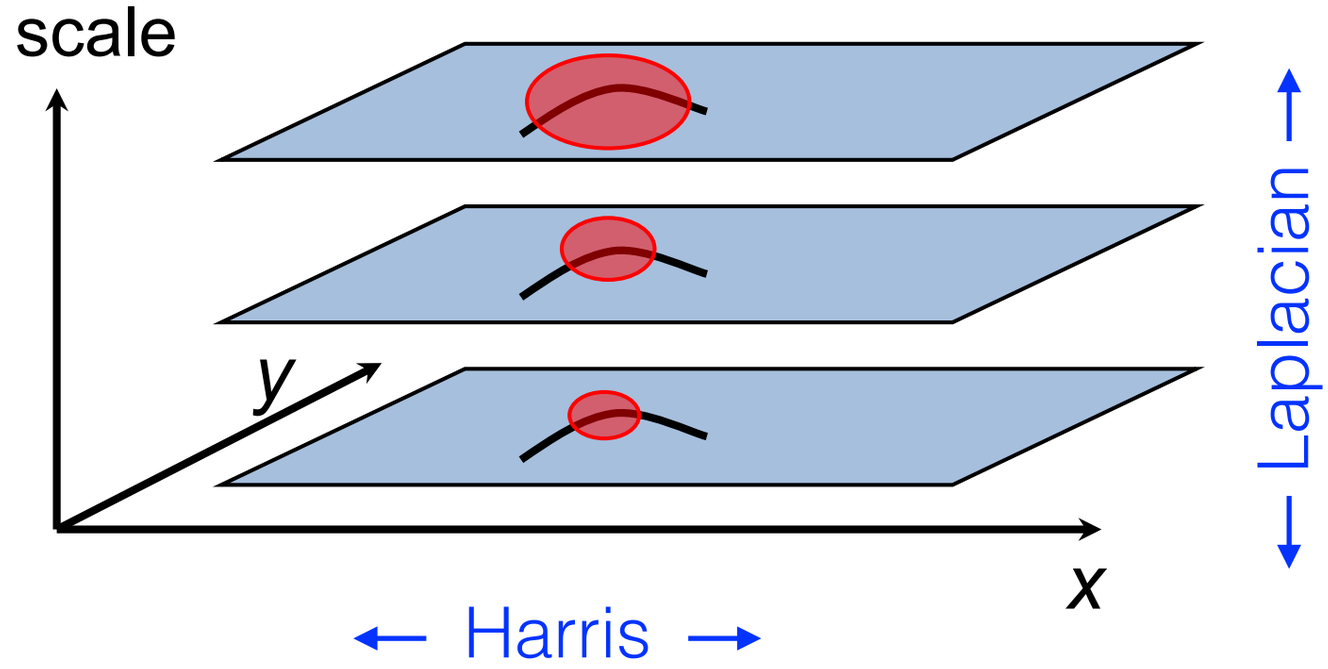
Scale invariant detection

- An example (Laplacian):



Scale invariant detection

- **Harris-Laplacian** detector: find local maximum of
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



K.Mikolajczyk, C.Schmid, “Indexing Based on Scale Invariant Interest Points”, ICCV 2001

Scale invariant detection

- Functions for determining scale: $f = \text{Kernel} * \text{Image}$

Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

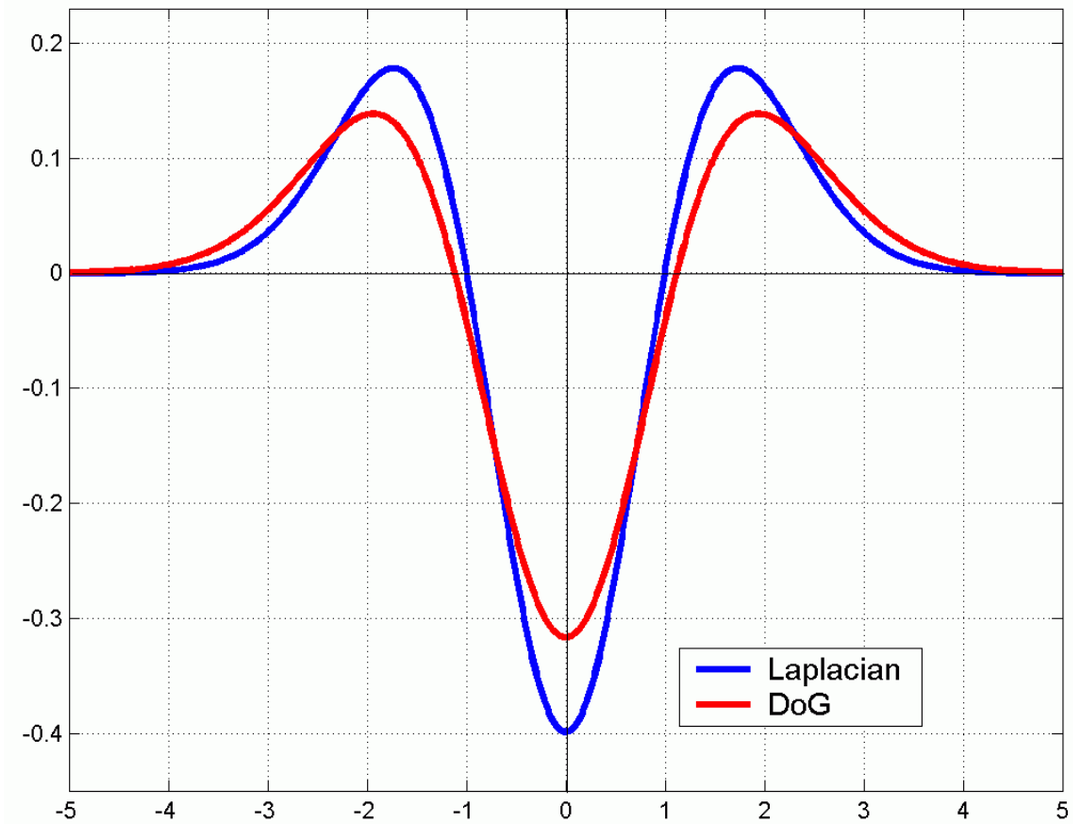
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian

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



Note: both kernels are invariant to scale and rotation

Scale invariant detection

- **SIFT** detector: find local maximum of
  - Difference of Gaussians in space
  - Difference of Gaussians in scale

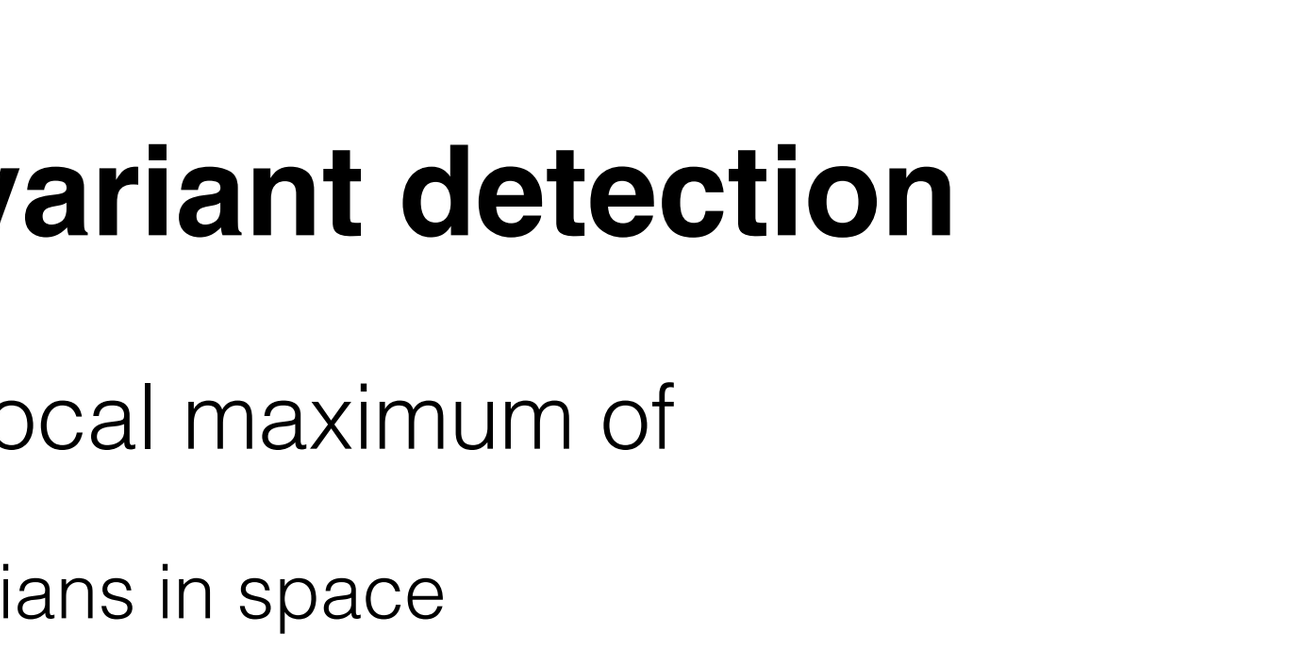
David Lowe

FOLLOW

Professor Emeritus, Computer Science Dept.,
University of British Columbia
Verified email at cs.ubc.ca - Homepage

Computer Vision Object Recognition

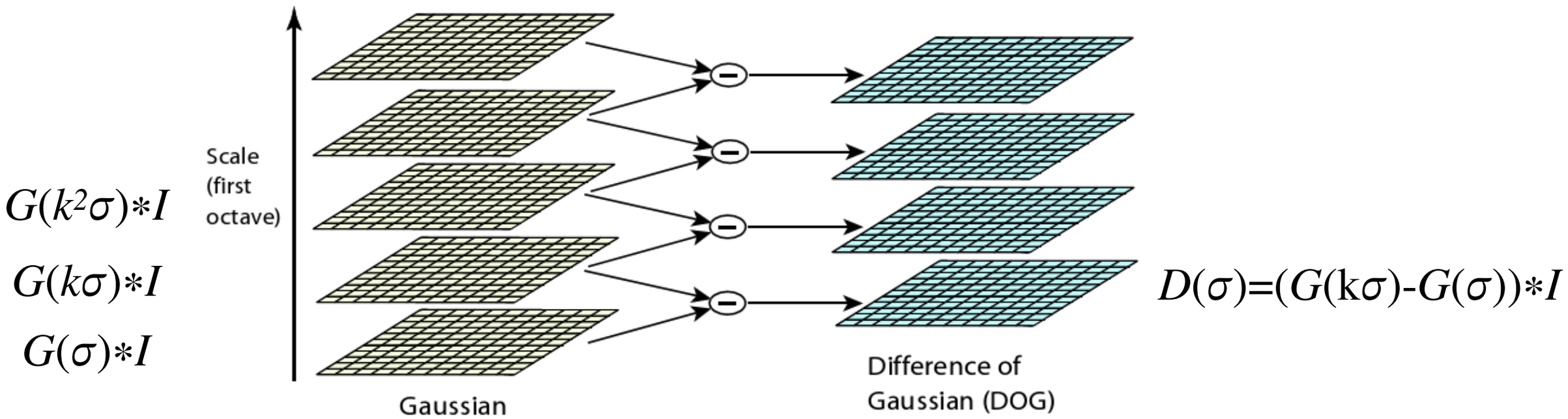
TITLE	CITED BY	YEAR
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	72551	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	24829	1999

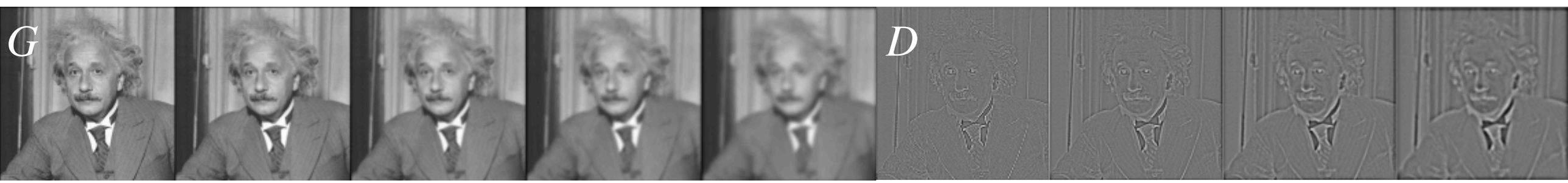


D. Lowe, “Distinctive image features from scale-invariant keypoints”, IJCV 2004

Difference of Gaussians (SIFT detector)

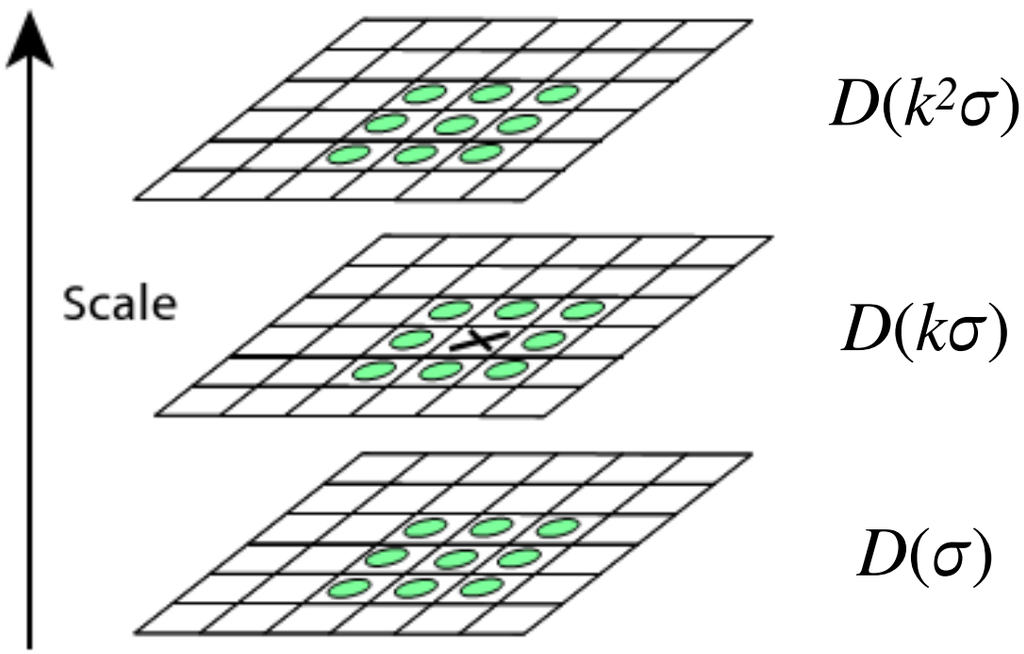
- Efficient approximation of LoG





Difference of Gaussians (SIFT detector)

- Keypoint are selected as scale-space extrema
  - Choose all extrema within 3x3x3 neighborhood



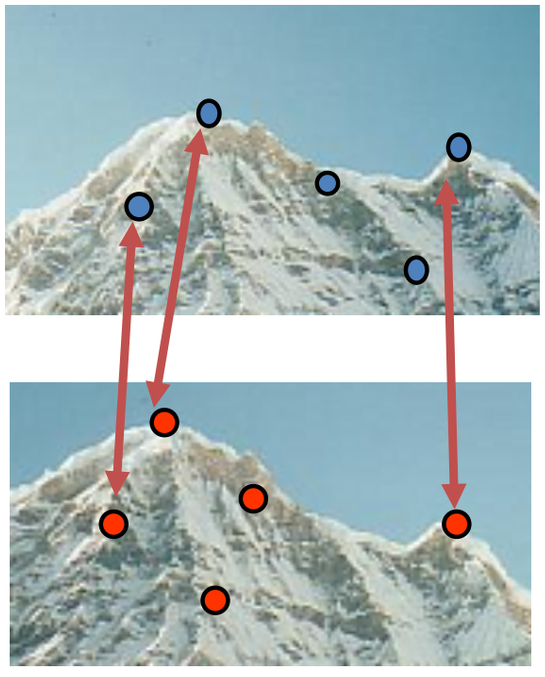
X is selected if it is larger or smaller than all 26 neighbors

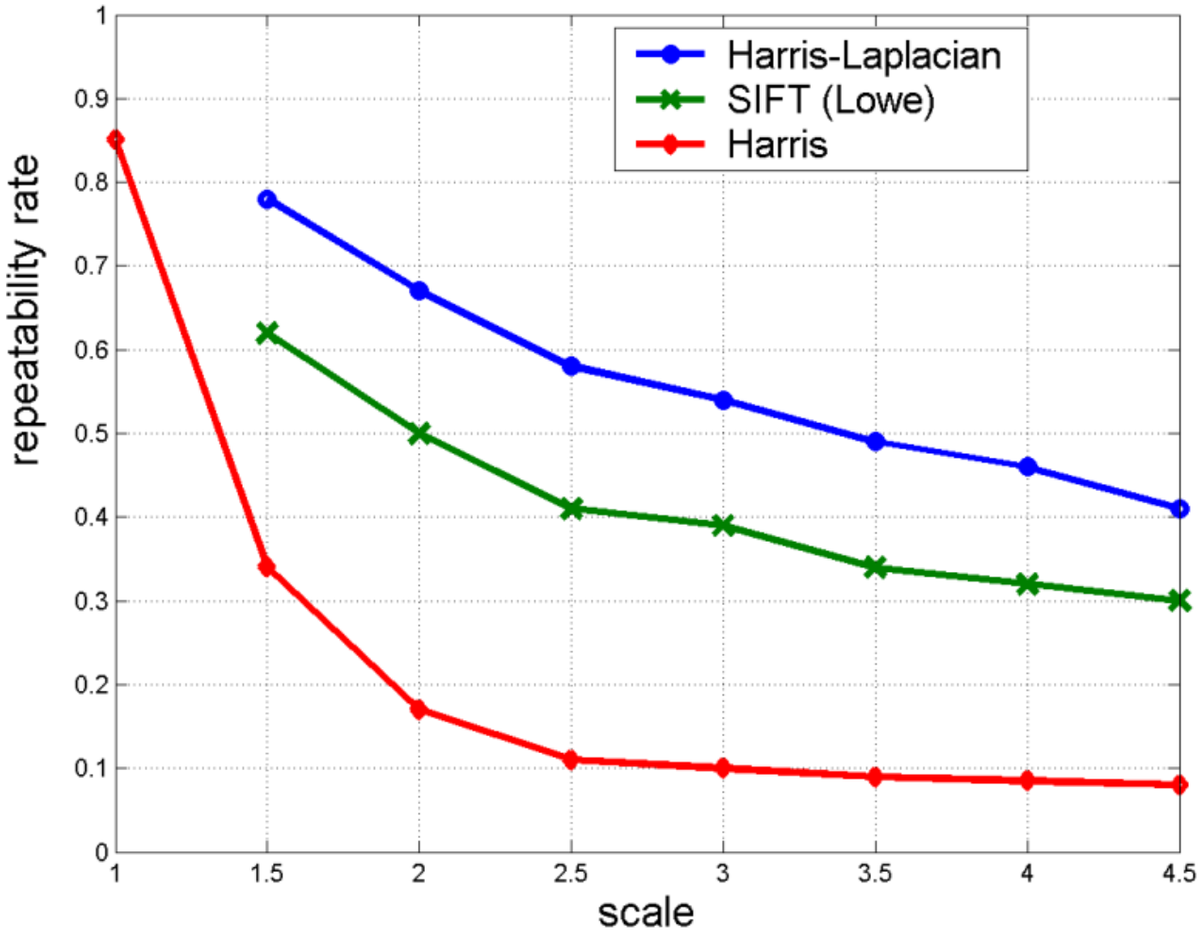
Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\text{ correspondences}}{\#\text{ possible correspondences}}$$





K. Mikolajczyk, C. Schmid, “Indexing Based on Scale Invariant Interest Points”, ICCV 2001

Scale Invariant Detectors

- Scale invariant detection summary:
  - **Given:** two images of the same scene with a large scale difference between them
  - **Goal:** find the same interest points independently in each image
  - **Solution:** search for maxima of suitable functions in scale and in space (over the image)

Popular methods:

1. Harris-Laplacian [Mikolajczyk, Schmid]: maximize Laplacian of Gaussian over scale, Harris' measure of corner response over the image
2. SIFT [Lowe]: maximize Difference of Gaussians over scale and space

Scale Invariant Detectors

- Actually there are many other keypoints/features…
- List/comparison of keypoint detectors:

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Key reference: http://www.robots.ox.ac.uk/~vgg/research/affine/

Source: T. Tuytelaars, K. Mikolajczyk

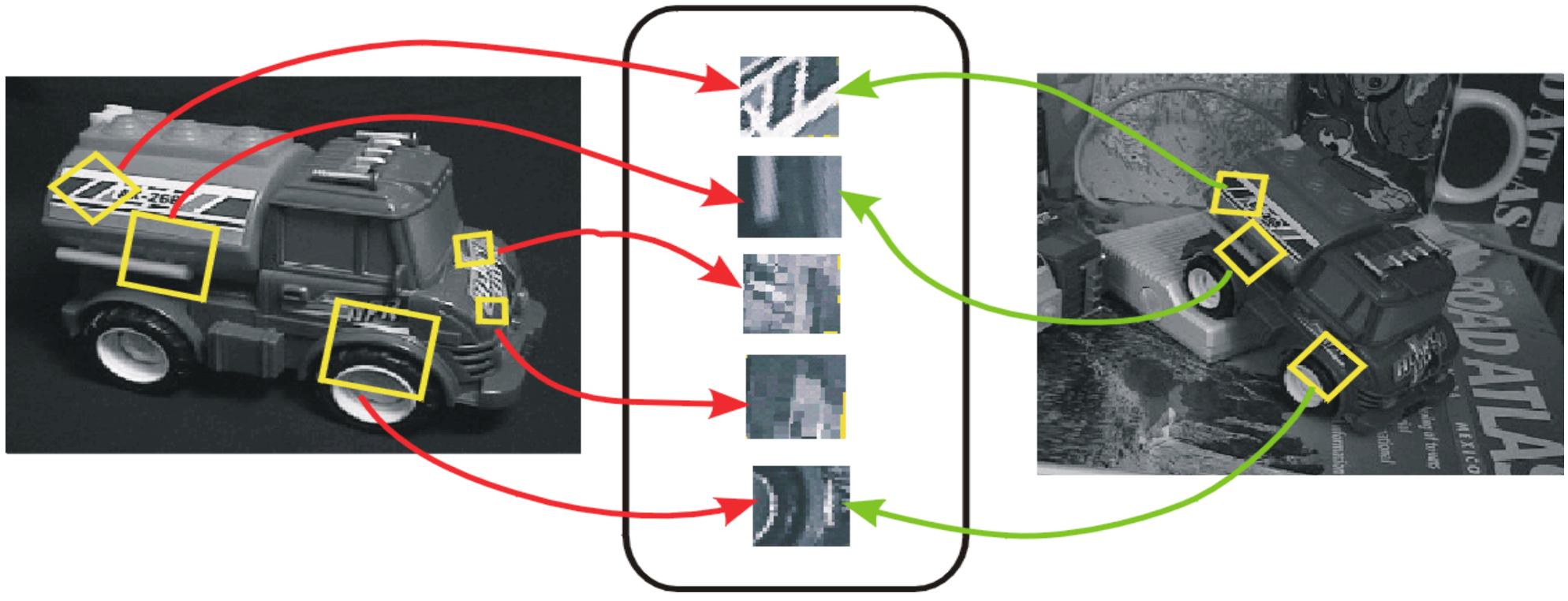
From feature detection to description

- Scaled & rotated versions of the same neighborhood will give rise to blobs that are related by the same transformation

- What to do if we want to compare the appearance of these image regions?
  - Normalization: transform the regions into same-size circles
  - Problem: **rotational ambiguity**

Invariant local features

- Image content is transformed into local feature that are invariant to *translation*, *rotation*, *scale*, and other imaging parameters



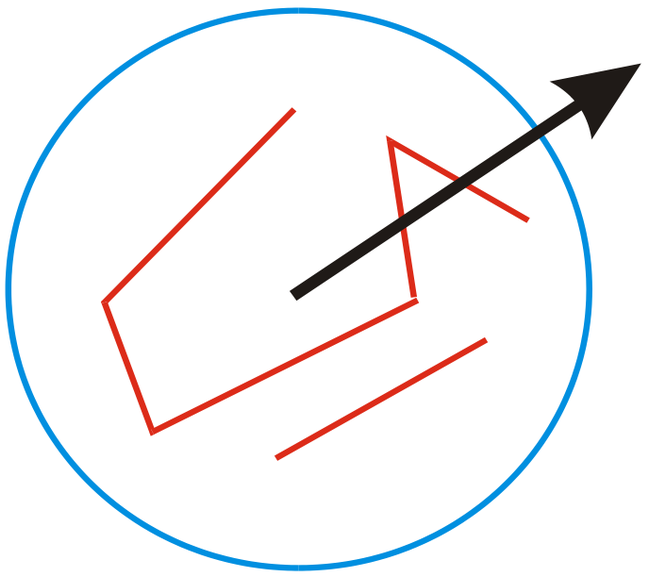
Reference: CVPR 2003 Tutorial on “Recognition and Matching Based on Local Invariant Features” by David Lowe

Desired properties of feature descriptors

- Easily compared (compact, fixed-dimensional)
- Invariant:
  - Translation
  - Rotation
  - Scale
  - Change in image brightness (illumination)
  - *Change in perspective?*

SIFT descriptors

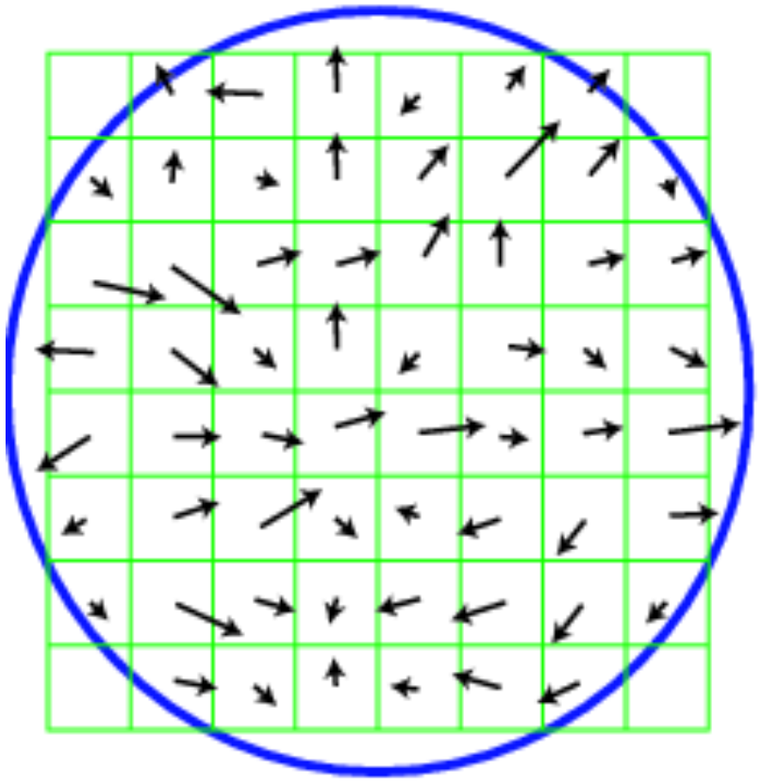
- We are given a keypoint and its scale from DoG
- To assign the orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Then we select a *characteristic orientation* for the keypoint based on the most prominent gradient (see next slides)
  - We will describe all features relative to this orientation

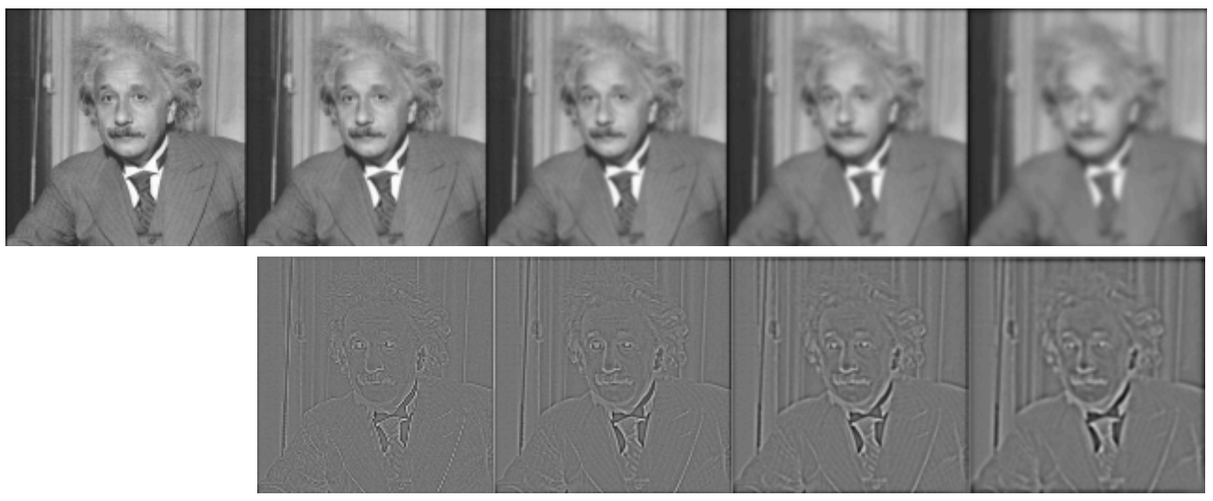


If the keypoint appears rotated in another image, the features will be the same, because they're relative to the characteristic orientation

SIFT descriptor formation

- Use the blurred image associated with keypoint's scale
- Take image gradients over the keypoint neighborhood
- To become rotation invariant, rotate the gradient directions AND locations by keypoint orientation



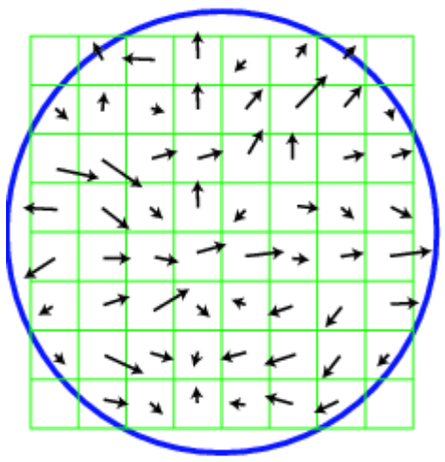


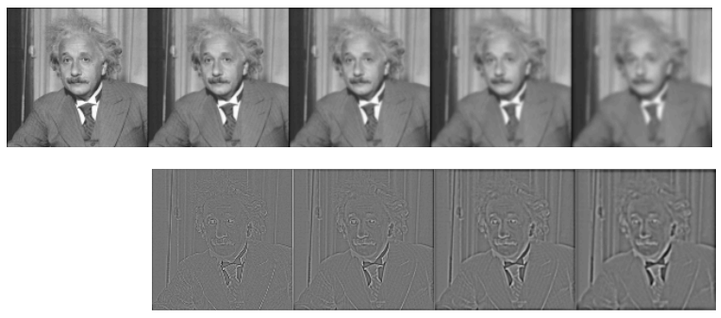
$L(x, y, \sigma)$

$D(x, y, \sigma)$

SIFT descriptor formation

- Use the blurred image associated with keypoint's scale
- Take image gradients over the keypoint neighborhood
- To become rotation invariant, rotate the gradient directions AND locations by keypoint orientation





$L(x, y, \sigma)$

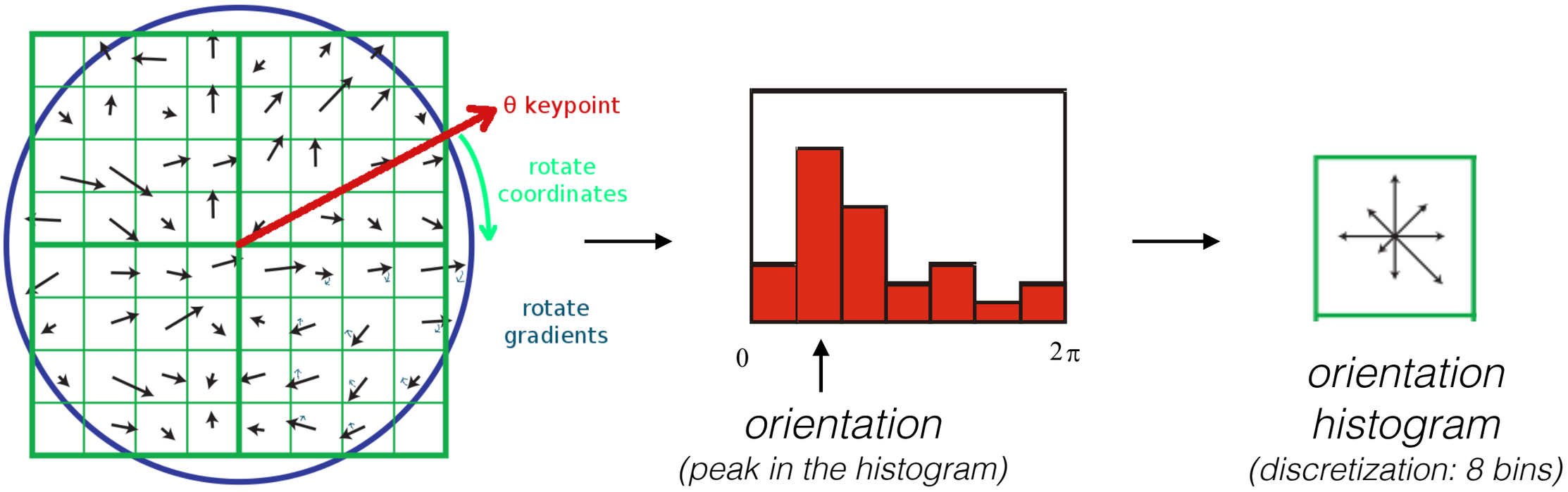
$D(x, y, \sigma)$

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

$$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y))$$

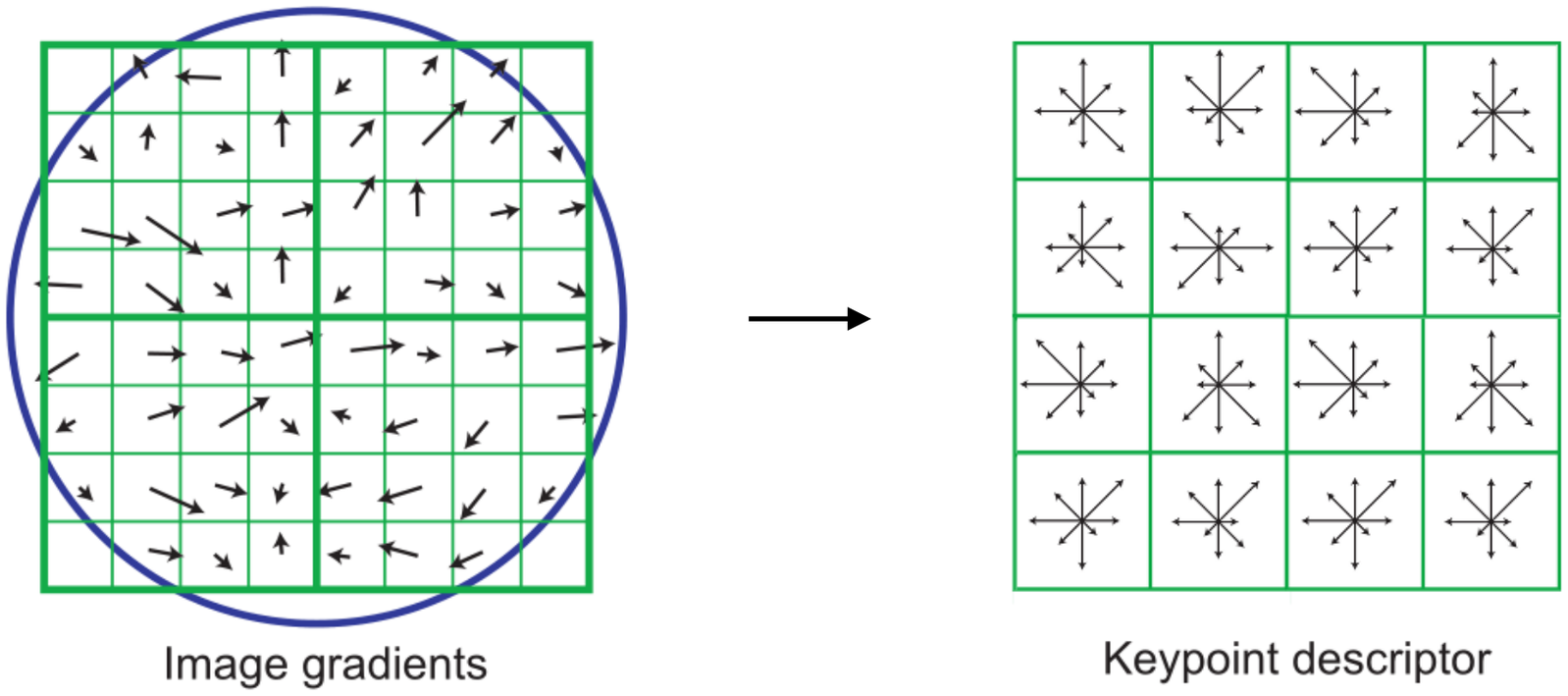
SIFT descriptor formation

- Use the blurred image associated with keypoint's scale
- Take image gradients over the keypoint neighborhood
- To become rotation invariant, rotate the gradient directions AND locations by keypoint orientation



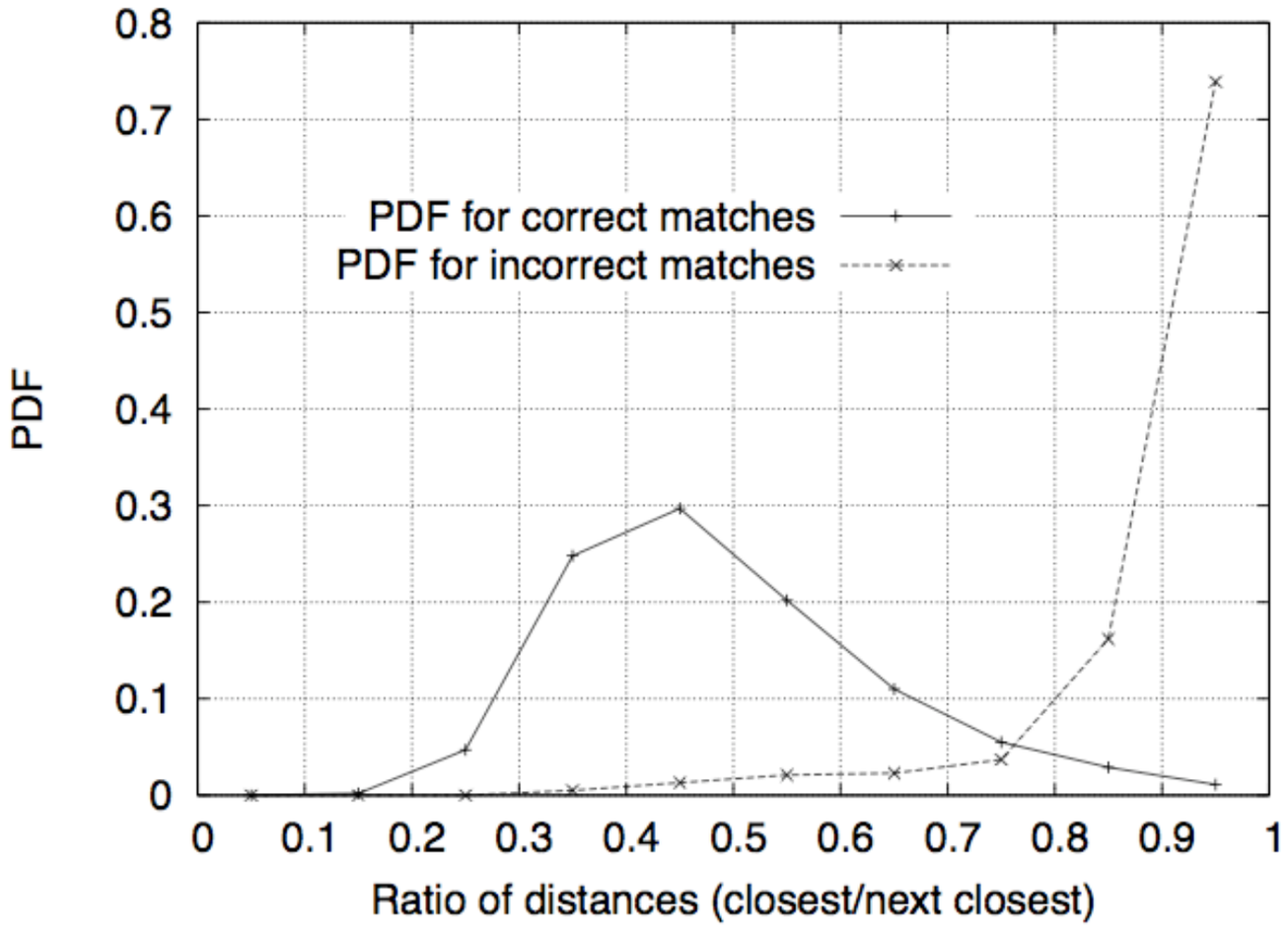
SIFT descriptor formation

- 8 orientation bins per histogram and a 4x4 histogram array yields 8x4x4=128 numbers
- So a SIFT descriptor is a 128-d vector invariant to rotation (because we rotated the descriptor) and scale (thanks to DoG)
- We can now compare each vector from image *A* to each vector from image *B* to find matching keypoints



SIFT matching

- Euclidean distance between SIFT descriptor gives a good measure of keypoint similarity
- Ratio of distances is more reliable for matching



The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest

(plot: PDF obtained using a database of approx. 40,000 keypoints)

Image credit: D. Lowe

SIFT matching

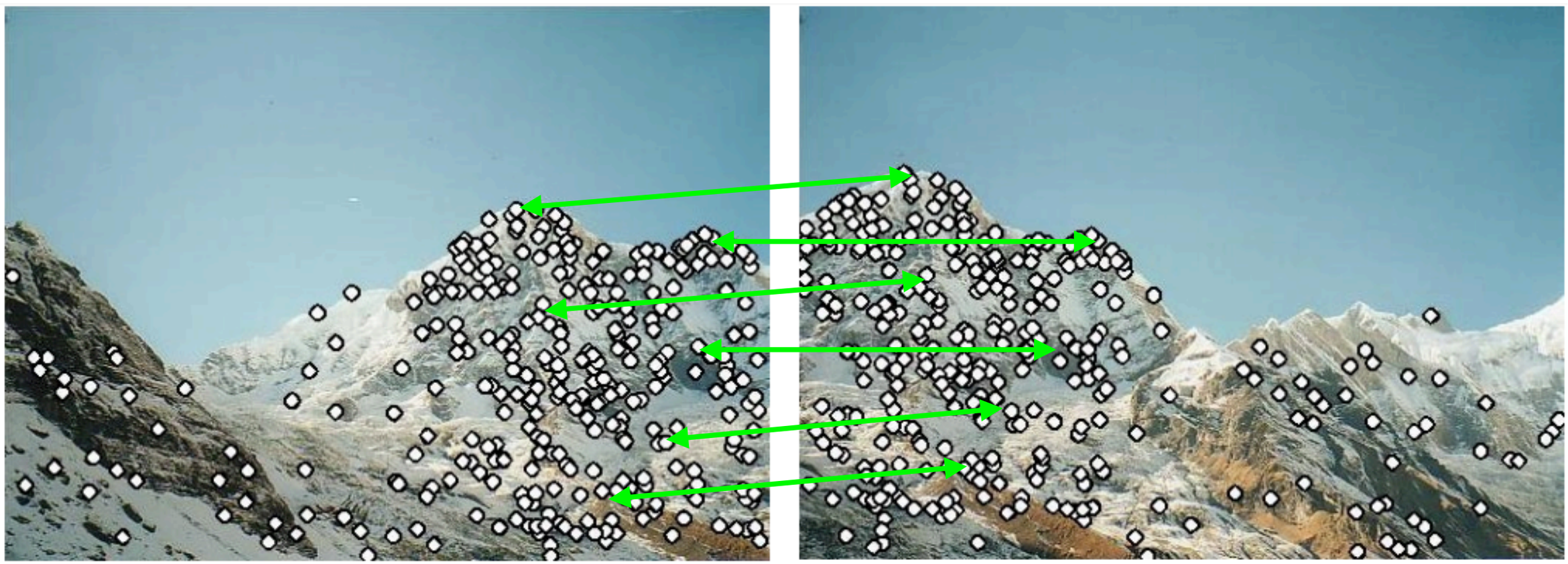
- An example:

Image matching: main flow

- Image matching with local visual features (SIFT)
  - Common pipeline: three (plus one) steps

Image matching: main flow

- Image matching with local visual features (SIFT)
  - Common pipeline: three (plus one) steps



Step 1: detect keypoints

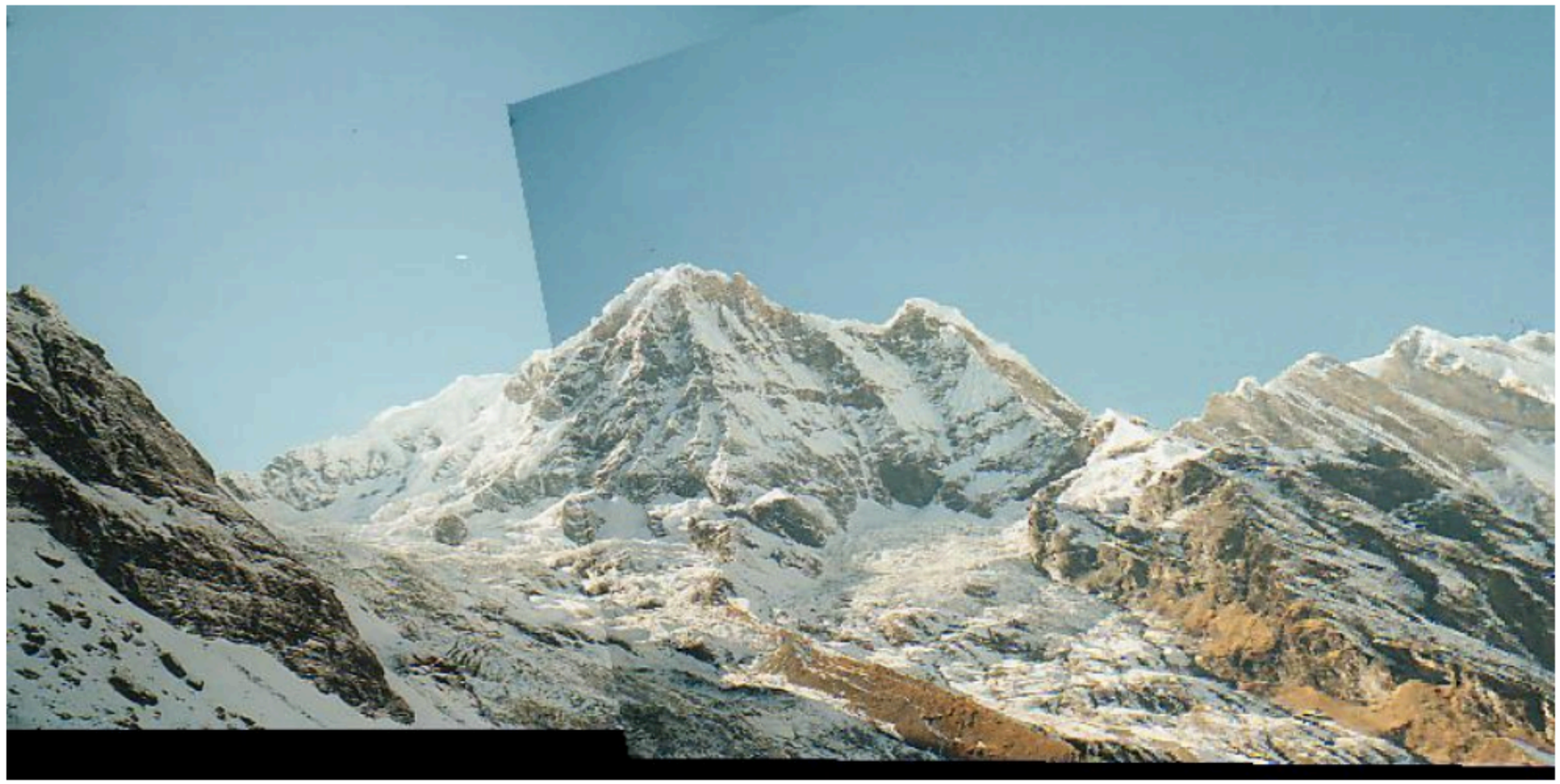
Step 2: build keypoints descriptors

Step 3: match keypoint features

Step 4: align images (i.e. fitting the transformation)

Image matching: main flow

- Image matching with local visual features (SIFT)
  - Common pipeline: three (plus one) steps



Step 1: detect keypoints

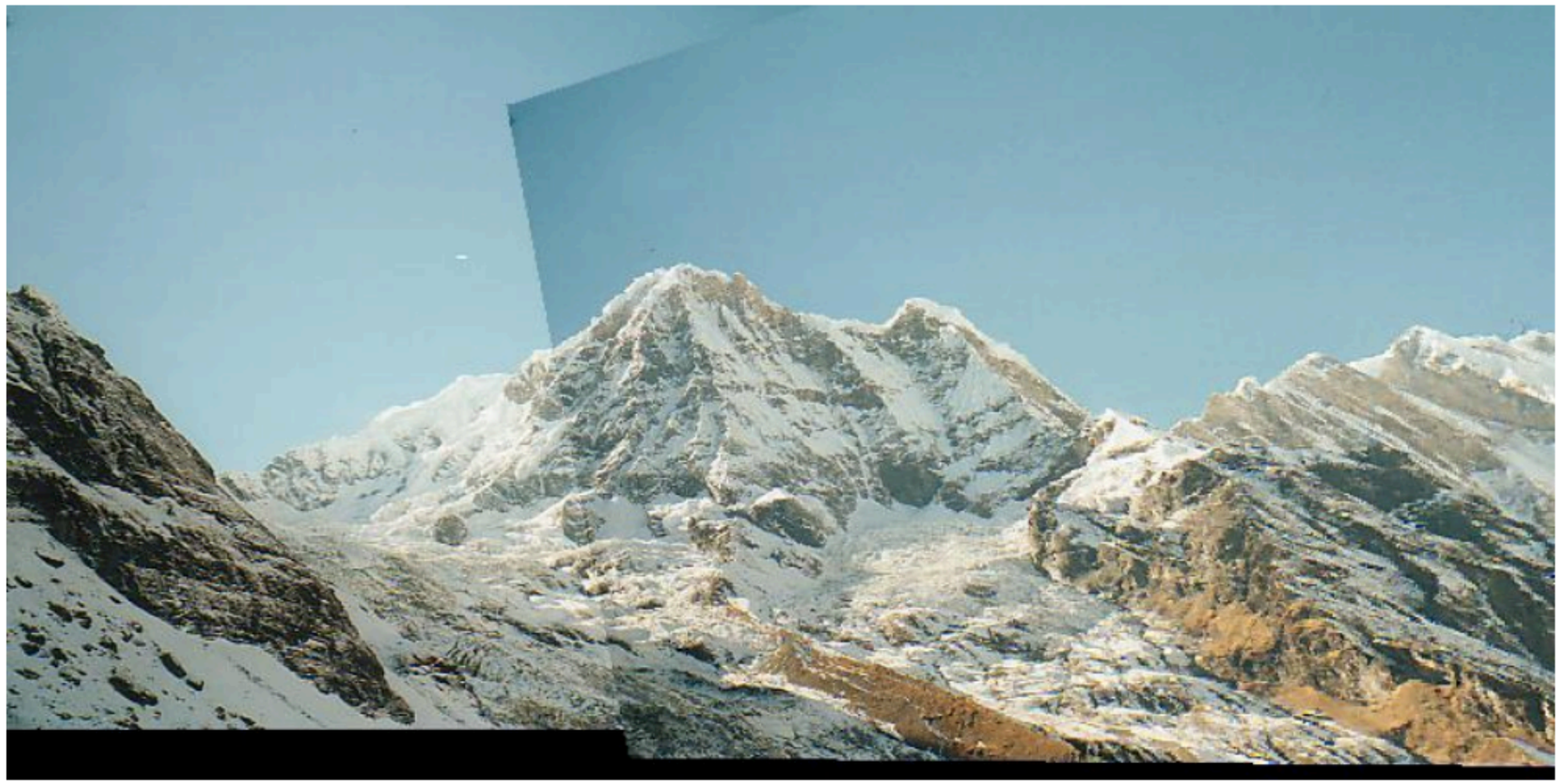
Step 2: build keypoints descriptors

Step 3: match keypoint features

Step 4: align images (i.e. fitting the transformation)

Image matching: main flow

- Image matching with local visual features (SIFT)
  - “Additional step”: image alignment



Step 1: detect keypoints

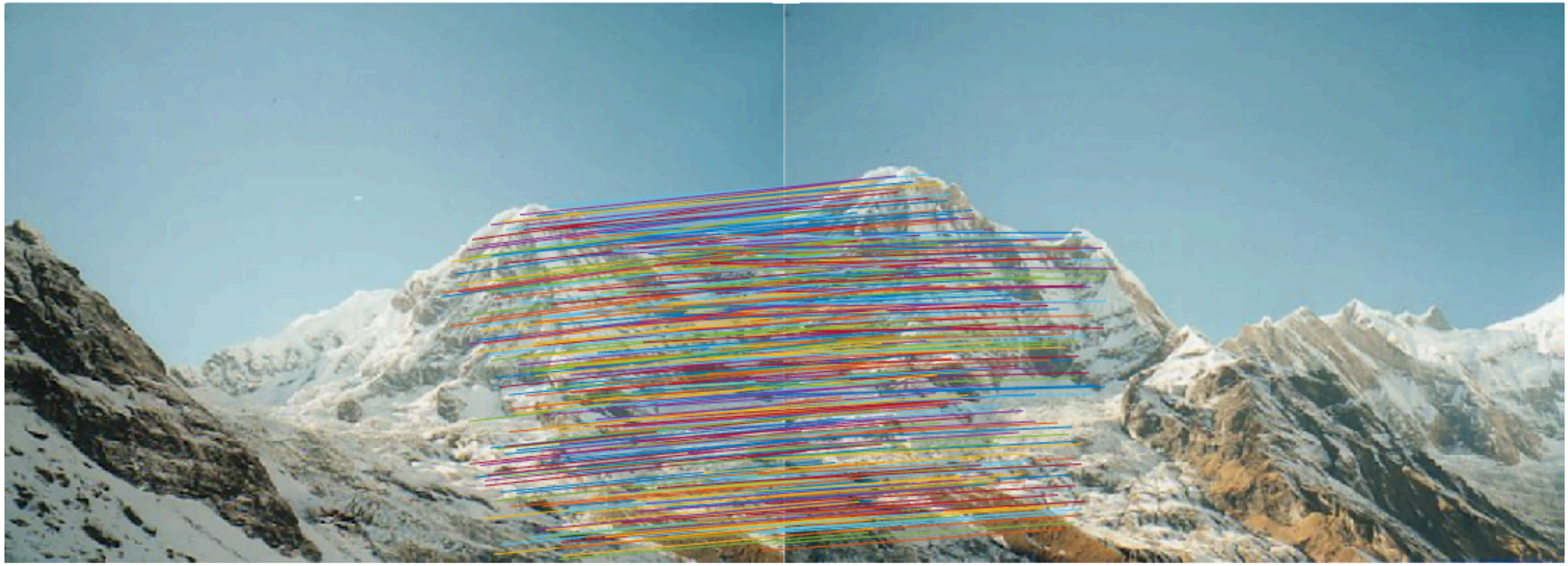
Step 2: build keypoints descriptors

Step 3: match keypoint features

Step 4: align images (i.e. fitting the transformation)

Geometric verification

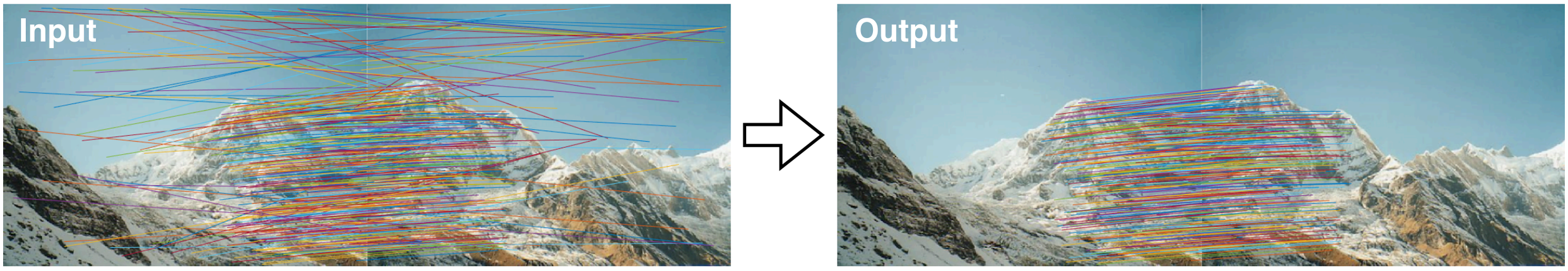
- A bit more on “step 4”, i.e. *geometric verification*
  - The idea is to test wether matches are consistent with an overall image transformation
  - Inconsistent matches are rejected



RANSAC: matching robust to outliers

(RANdom SAmple Consensus)

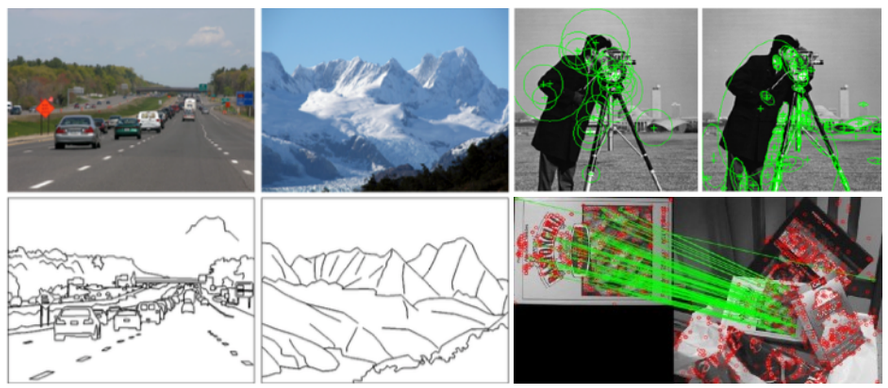
- **Input:** M tentative features matches (x_1, x_1'), …, (x_M, x_M')
- **Output:** (affine) transformation (A^*, T^*) with the largest number of inlier matches



1. Repeat a large number of times:
   - Randomly sample a **minimal subset** of matches sufficient to estimate (A, T)
   - Find **inliers**, i.e. other matches that are compatible with (A, T)
2. Return (A^*, T^*) as the pair (A, T) with the largest number of inliers

Summary

- Edge detection
- Image gradients
- Local invariant features
- DoG and SIFT descriptors

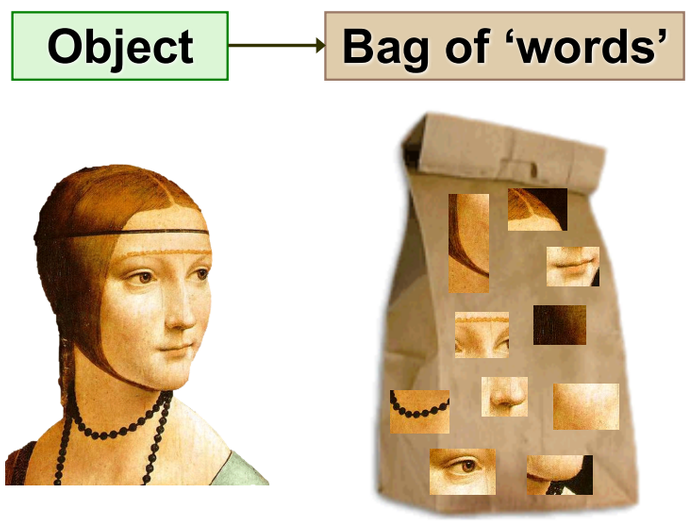


Background reading:
Forsyth and Ponce, Computer Vision, Chapter 8
D.G. Lowe, SIFT paper, IJCV 2014

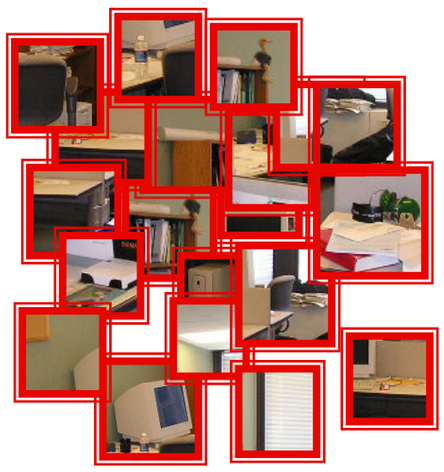
Coming up

- **Tuesday, November 14** (LabP140)**:**
  - Lab on visual features and image matching (+ contest!)

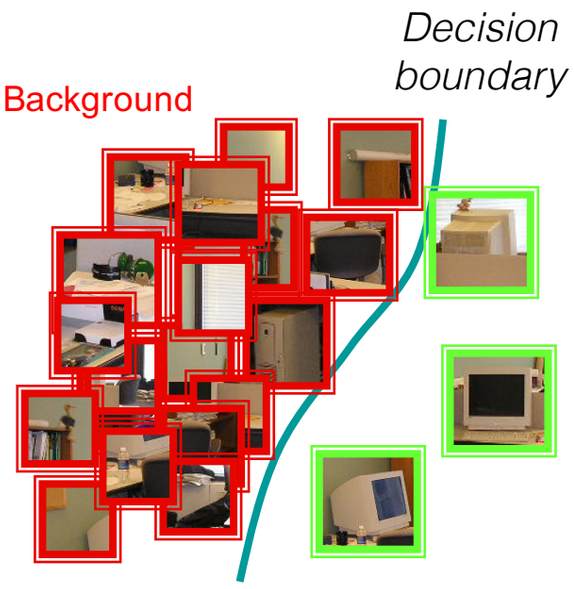
- **Monday, November 20:**
  - Introduction to visual recognition, image classification



Input image



Bag of image patches



Feature space