CS 554
Computer Vision

Feature Point Detection

Hamdi Dibeklioğlu

Slide Credits: L. van der Maaten

Image matching

What feature can we use to establish *correspondences* between images?

Feature point detection

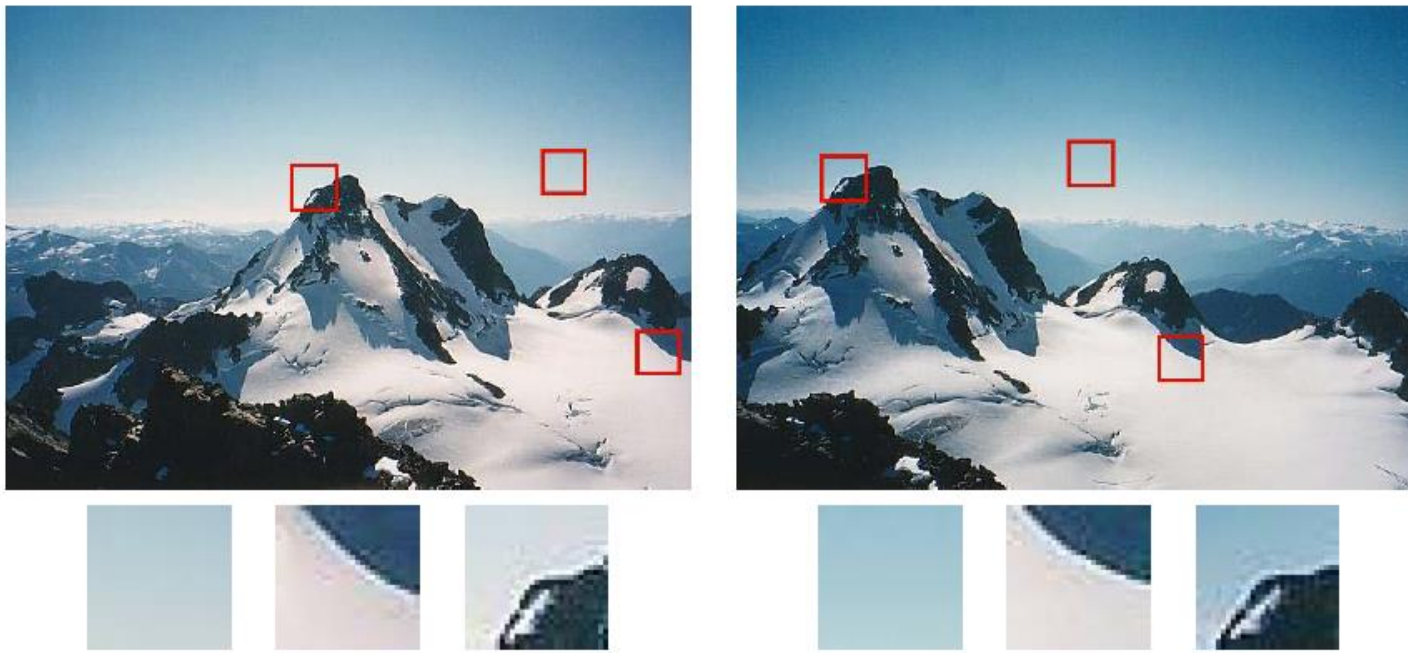
The goal is to find locations that are stable under *image transformations*

Feature points are frequently used in, among others:

- Stereo matching
- Image stitching
- Video stabilization
- Instance or object recognition

Feature point detection

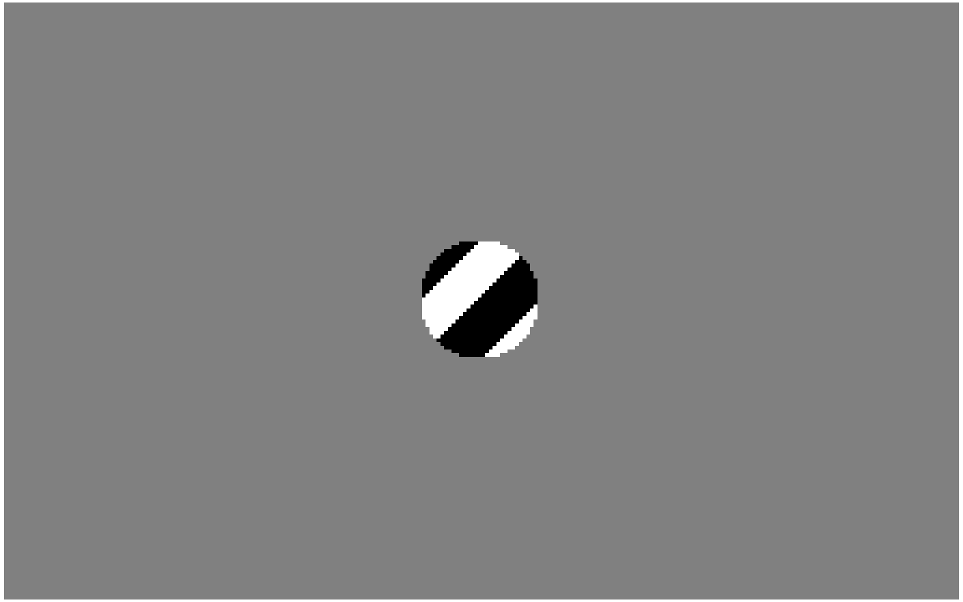
Some feature points can be matched more accurately than others:



Feature point detection

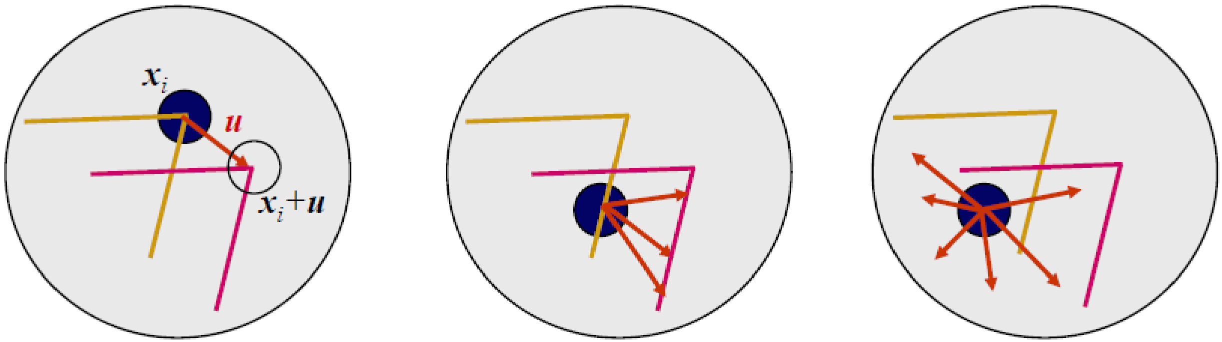
Patches with large contrast changes are easier to localize

However, straight lines with a single orientation suffer from *aperture problem*:



Feature point detection

The most reliable points for matching are *“corner”*-like points:

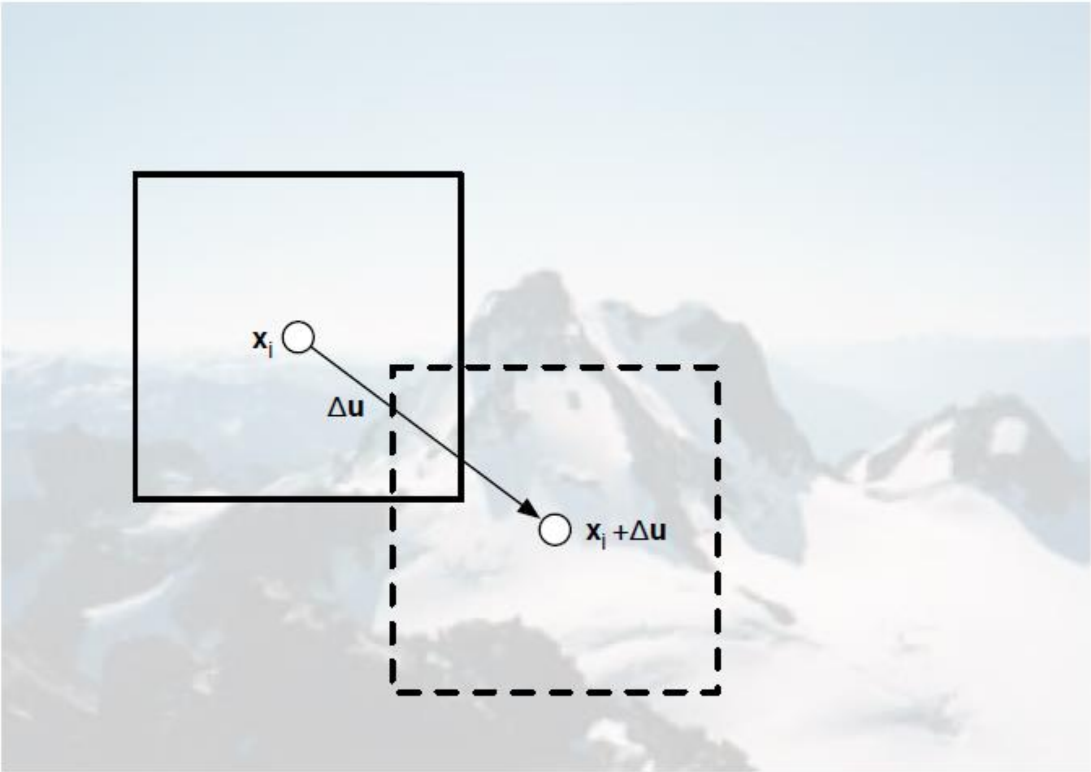


We can formalize these intuitions by looking at the *autocorrelation function*:

$$E_{AC}(\Delta \mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

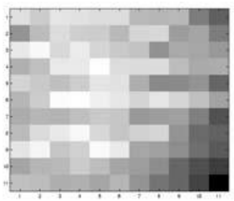
Local weighting function: the summation over the window

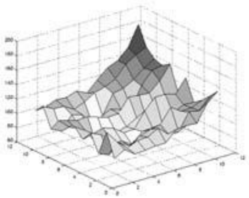
Autocorrelation function



$$E_{AC}(\Delta\mathbf{u}) = weighted\ SSE(\square, ⬚)$$

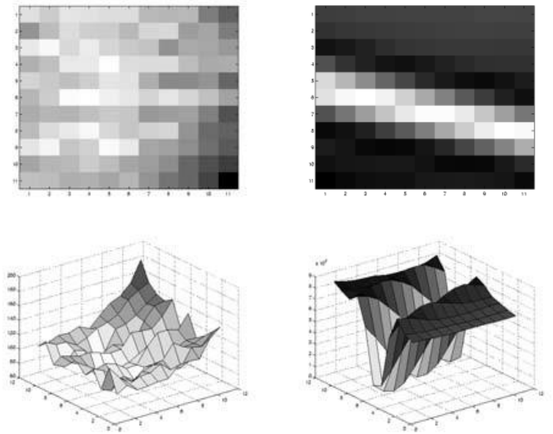
Autocorrelation surfaces





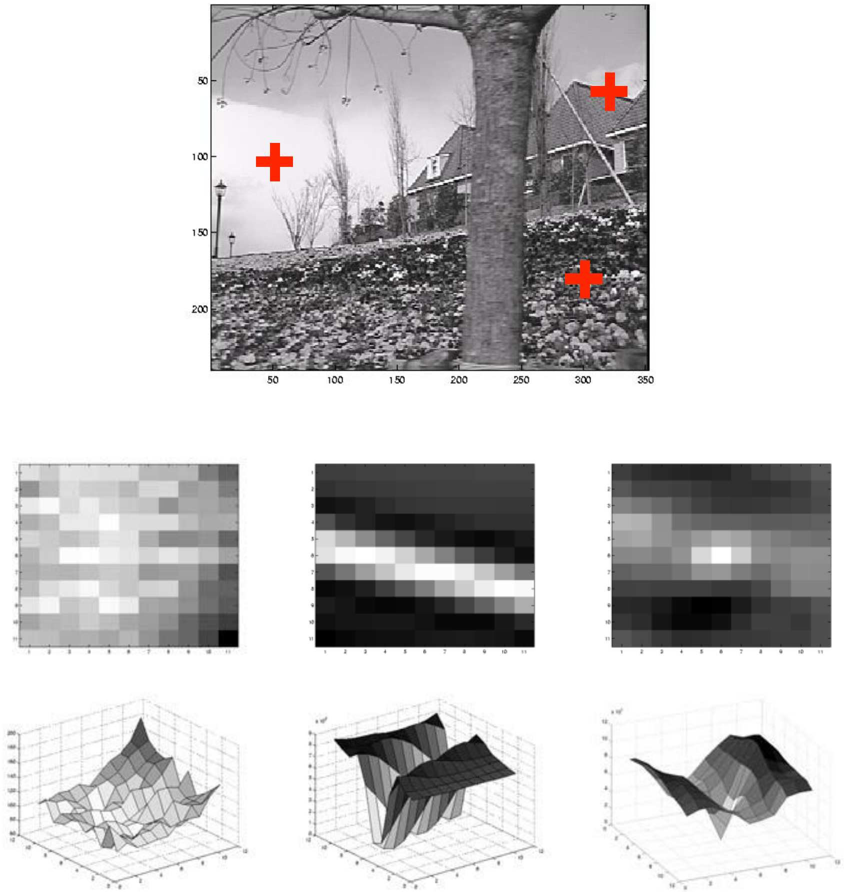
- For homogeneous regions, there is no clear minimum

Autocorrelation surfaces



- For homogeneous regions, there is no clear minimum
- For lines, there is an ambiguity

Autocorrelation surfaces



- For homogeneous regions, there is no clear minimum

- For lines, there is an ambiguity

- The flower does have a clear minimum
  (= easy to localize!)

Harris corner detector

We can approximate the autocorrelation via a Taylor expansion* of the image:

$$
\begin{aligned}
E_{AC}(\Delta\mathbf{u}) &= \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \\
&\approx \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2 \\
&= \sum_i w(\mathbf{x}_i)[\nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u}]^2 \\
&= \Delta\mathbf{u}^{\mathrm{T}}\mathbf{A}\Delta\mathbf{u}
\end{aligned}
$$

Here, we define the *autocorrelation matrix* (aka *second-moment matrix* or *structure tensor*):

$$
\mathbf{A} = w \times \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}
$$

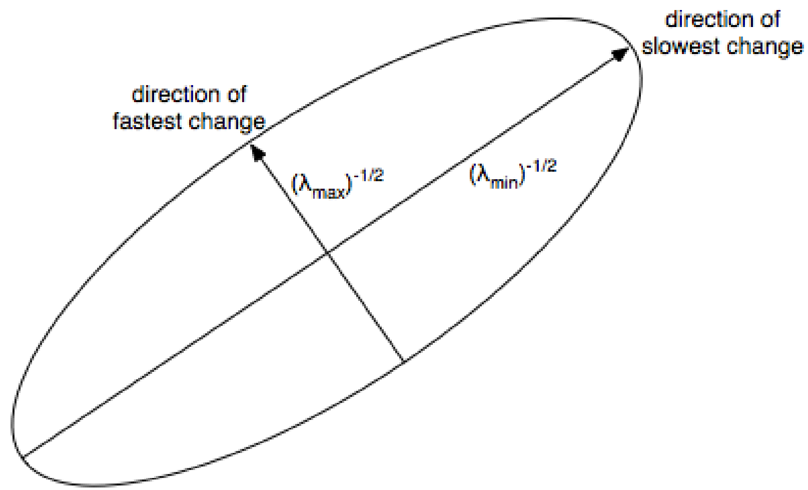
*A Taylor expansion of $f(x)$ around a is given by: $\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$

Harris corner detector

Corners correspond to large changes in E_{AC} in all directions

Eigen analysis of autocorrelation matrix reveals direction and speed of change:

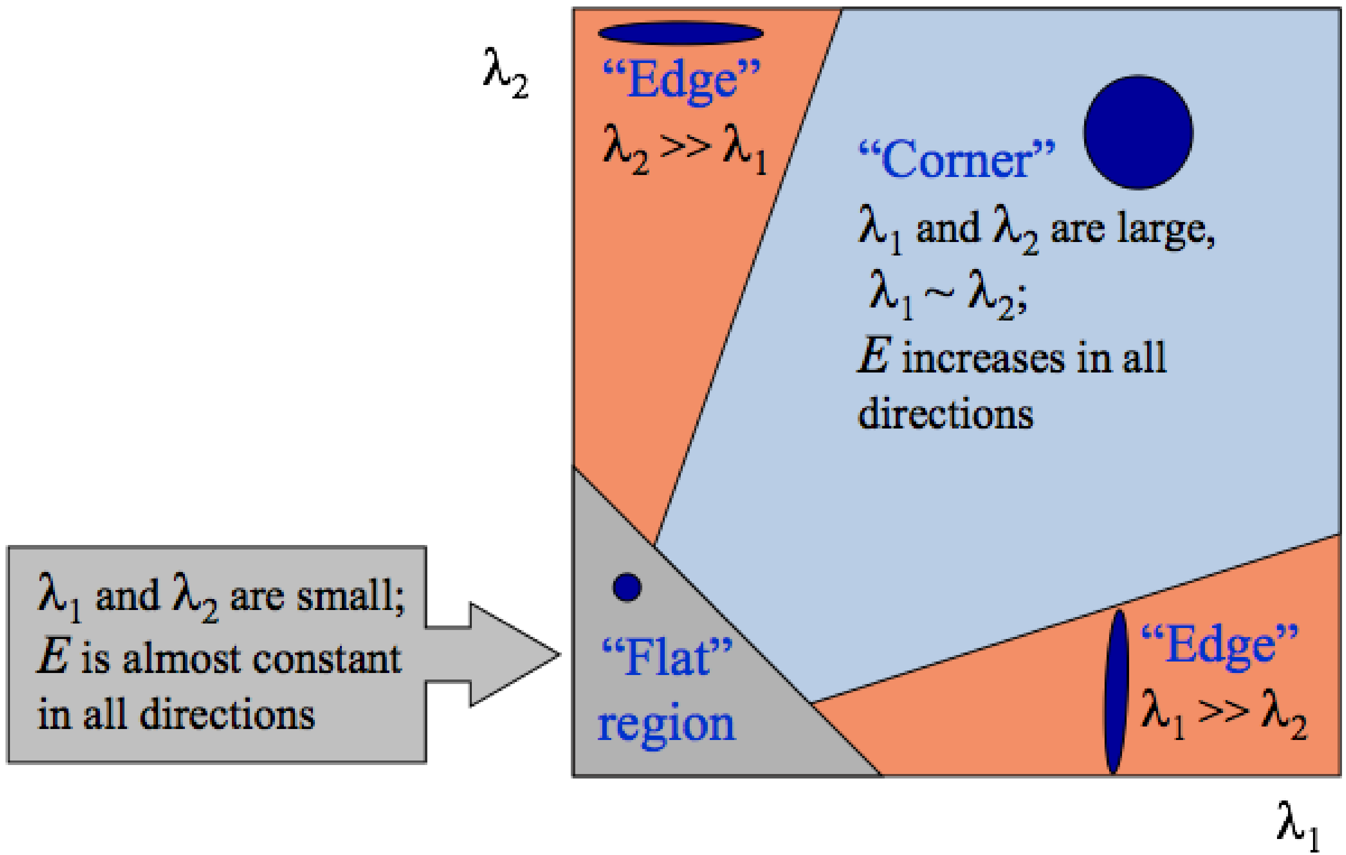
$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$



Harris detector finds local maxima of: $\det(\mathbf{A}) - \alpha\,\mathrm{trace}(\mathbf{A})^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$

Shi and Tomasi detector finds local maxima in the smallest eigenvalue λ_0

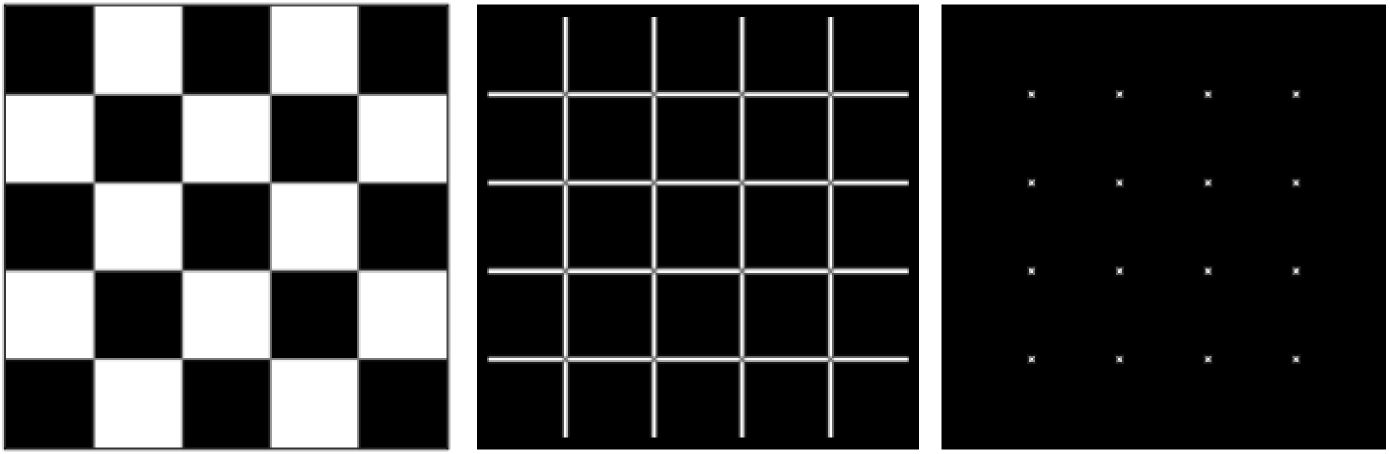
Interpreting the eigenvalues



Harris corner detector

Perform *eigen decomposition* of the second-moment matrix at each location

Find local maxima in the field of the smallest eigenvalues:



I $\lambda_{\max}$ $\lambda_{\min}$

Harris corner detector

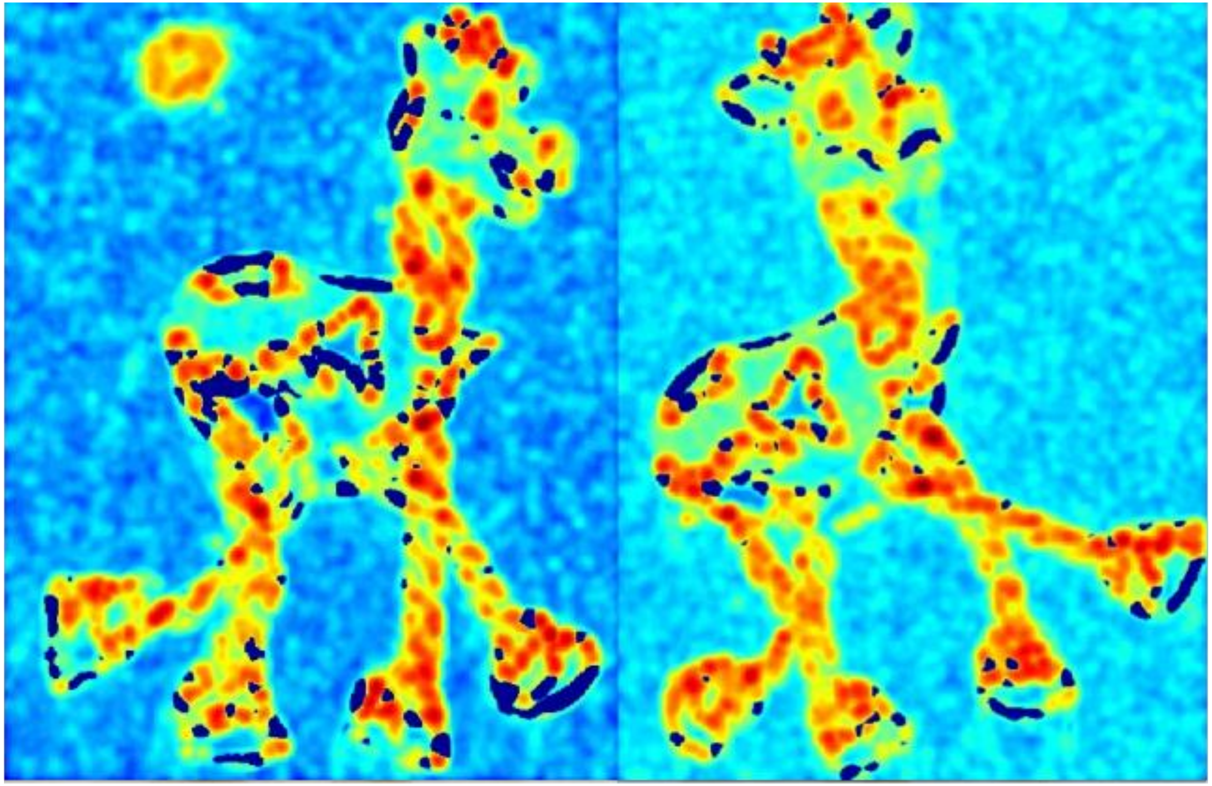
Example:

Harris corner detector

Example:

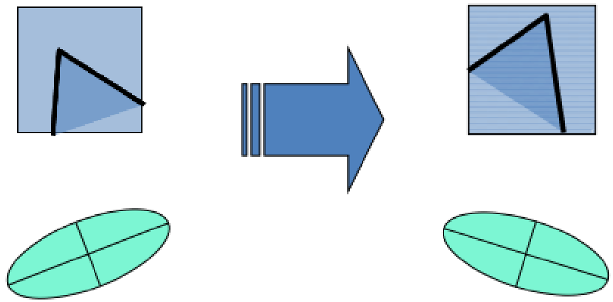
Harris corner detector

Example:

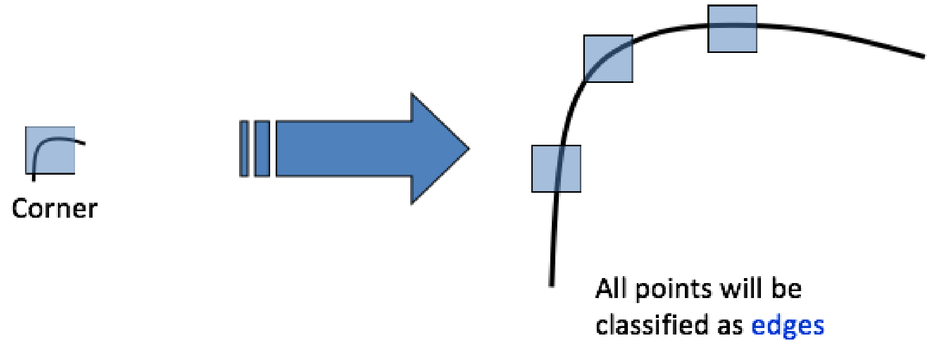


Invariances of Harris detector

The Harris detector is invariant to *rotations*:



The Harris detector is not invariant to *scale changes*:



Scale-Invariant Feature Transform

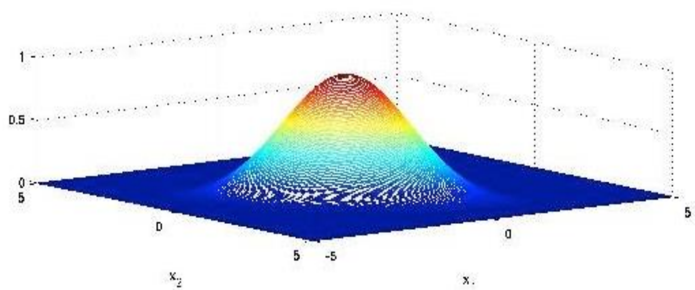
In contrast to Harris corners, SIFT features are stable in location and *scale*

Overview of the SIFT feature point detector:

- Perform *band-pass filtering* on a wide range of image scales
- *Non-maxima suppression* to find candidate keypoints in location <u>and</u> scale
- Remove candidate keypoints in low-contrast regions and keypoints on edges

Scale-Invariant Feature Transform

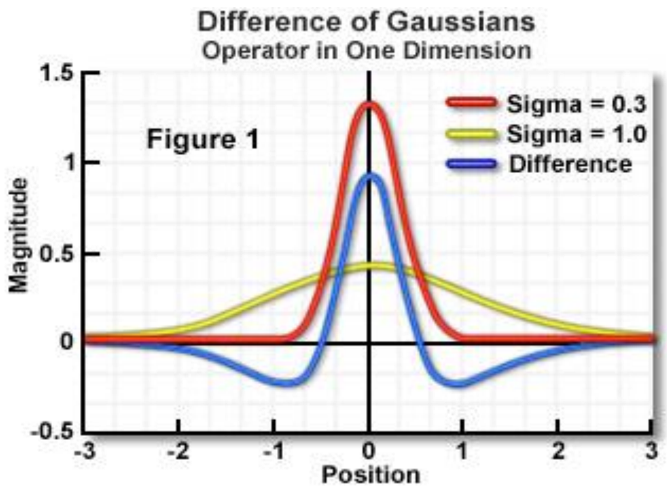
Gaussian scale space removes features at fine scales via repeated blurring:

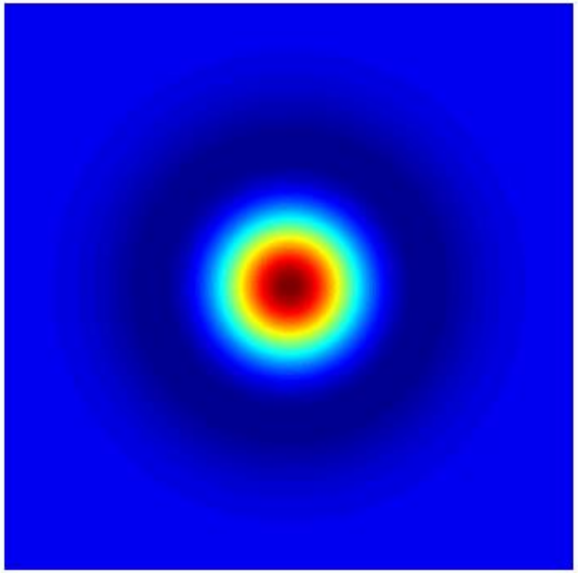


larger sigma / coarser scale →

Scale-Invariant Feature Transform

SIFT performs filtering using *difference-of-Gaussian* (DoG) filters:

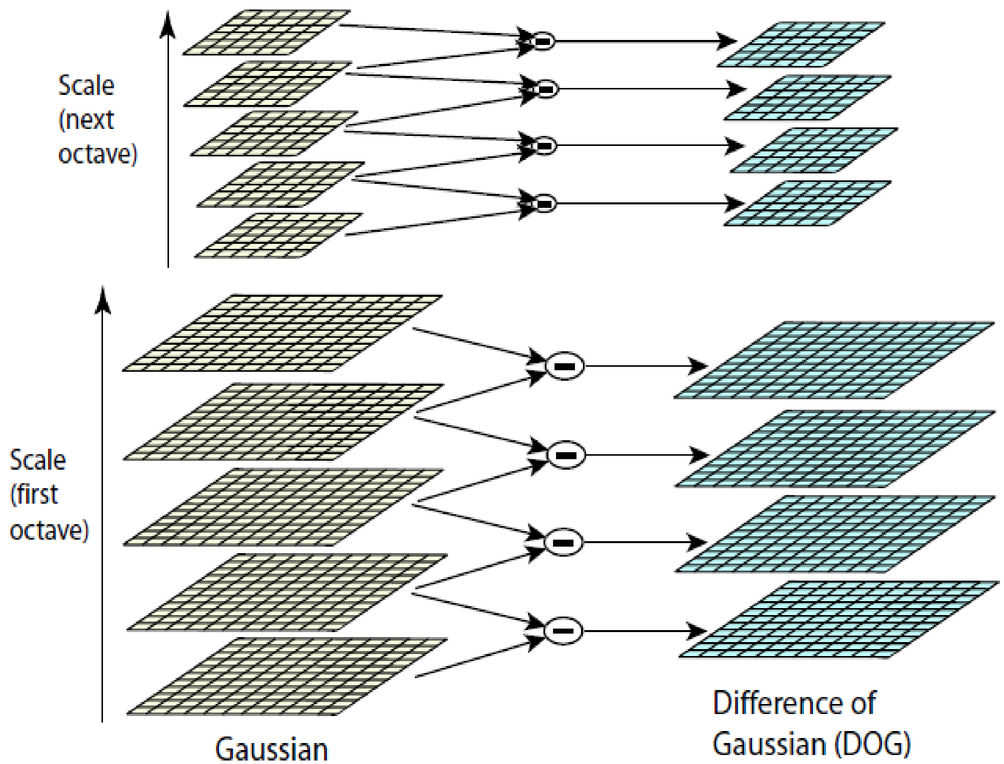


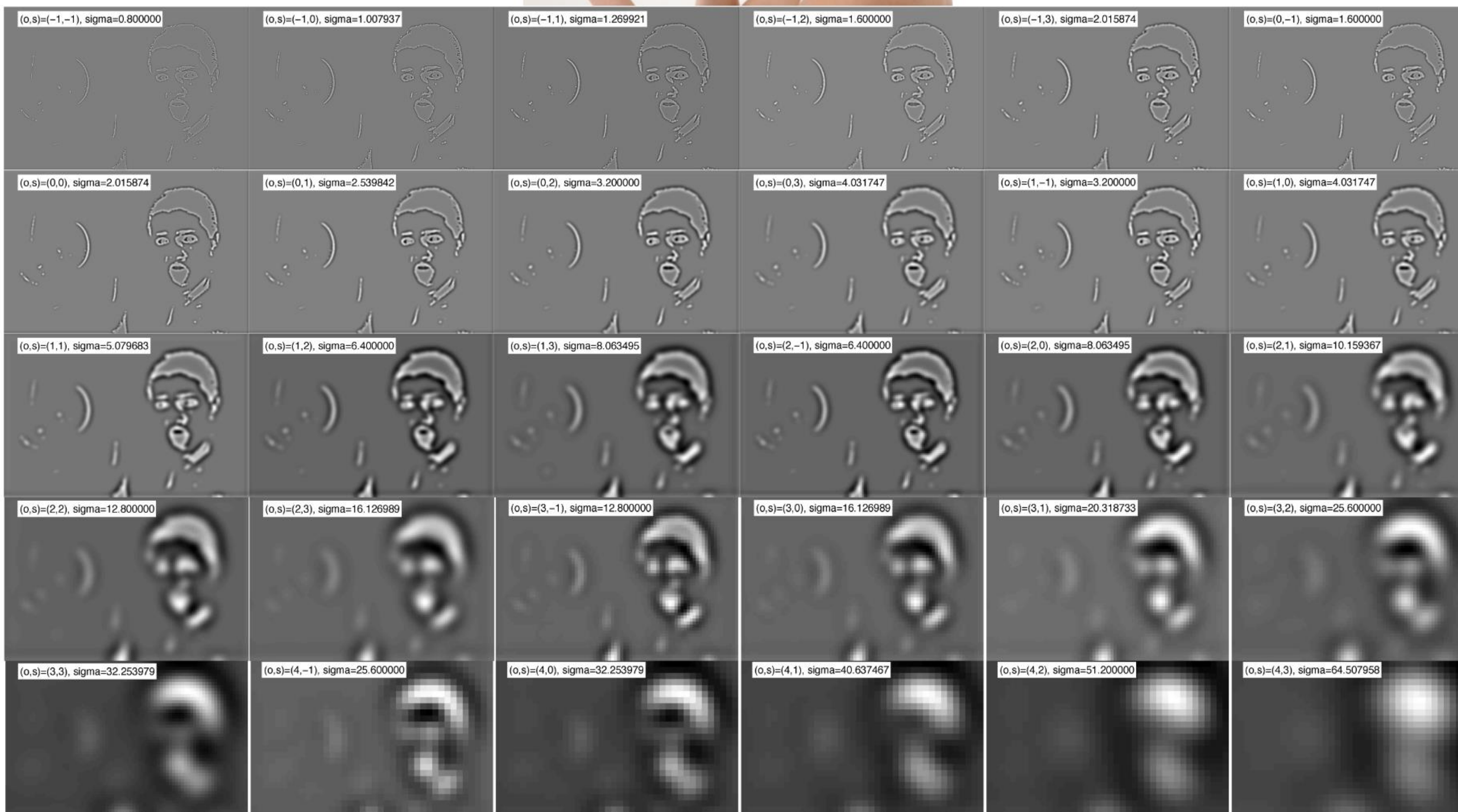


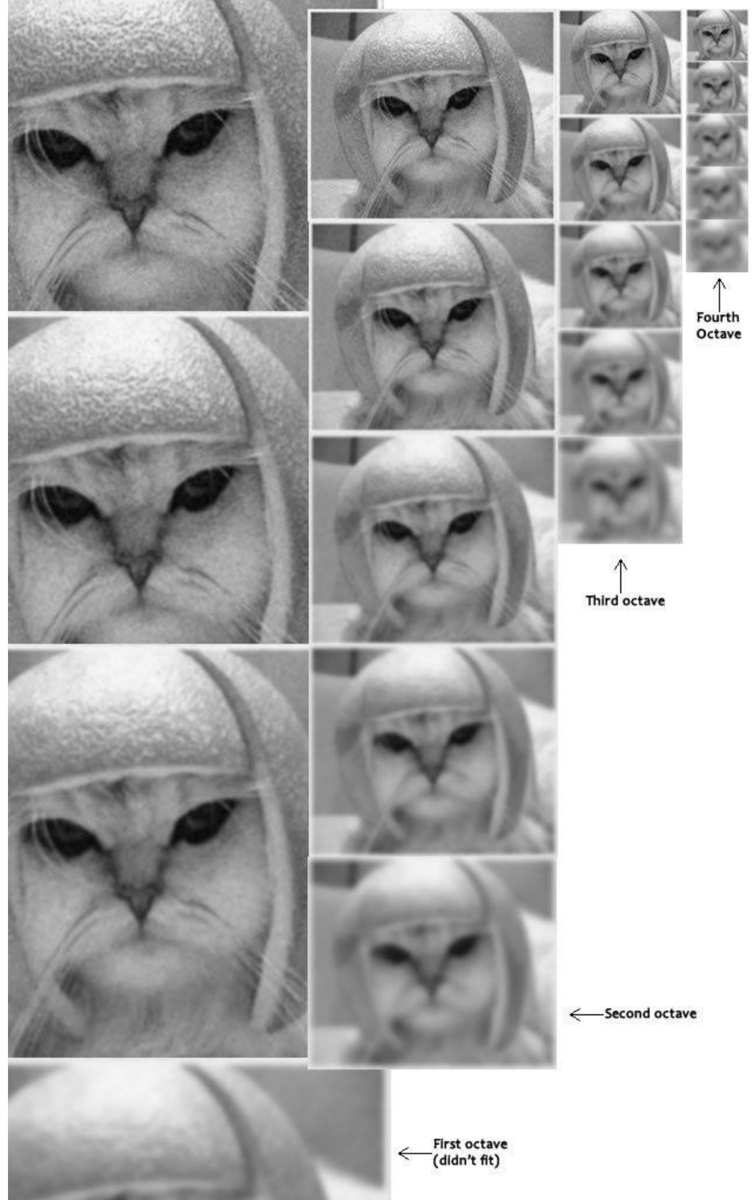
This is a *band-pass filter* that only retains particular spatial frequencies

Scale-Invariant Feature Transform

Efficiently computing difference-of-Gaussian response images:



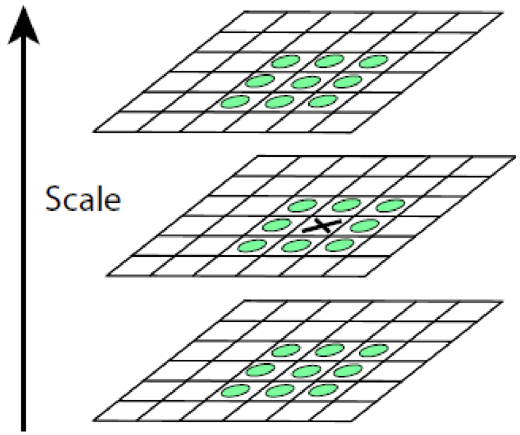

(o,s)=(−1,−1), sigma=0.800000
(o,s)=(−1,0), sigma=1.007937
(o,s)=(−1,1), sigma=1.269921
(o,s)=(−1,2), sigma=1.600000
(o,s)=(−1,3), sigma=2.015874
(o,s)=(0,−1), sigma=1.600000
(o,s)=(0,0), sigma=2.015874
(o,s)=(0,1), sigma=2.539842
(o,s)=(0,2), sigma=3.200000
(o,s)=(0,3), sigma=4.031747
(o,s)=(1,−1), sigma=3.200000
(o,s)=(1,0), sigma=4.031747
(o,s)=(1,1), sigma=5.079683
(o,s)=(1,2), sigma=6.400000
(o,s)=(1,3), sigma=8.063495
(o,s)=(2,−1), sigma=6.400000
(o,s)=(2,0), sigma=8.063495
(o,s)=(2,1), sigma=10.159367
(o,s)=(2,2), sigma=12.800000
(o,s)=(2,3), sigma=16.126989
(o,s)=(3,−1), sigma=12.800000
(o,s)=(3,0), sigma=16.126989
(o,s)=(3,1), sigma=20.318733
(o,s)=(3,2), sigma=25.600000
(o,s)=(3,3), sigma=32.253979
(o,s)=(4,−1), sigma=25.600000
(o,s)=(4,0), sigma=32.253979
(o,s)=(4,1), sigma=40.637467
(o,s)=(4,2), sigma=51.200000
(o,s)=(4,3), sigma=64.507958


Fourth
Octave
Third octave
Second octave
First octave
(didn't fit)

Scale-Invariant Feature Transform

Keypoints are found as local maxima and minima of the DoG responses

Search for maxima in eight-pixel spatial neighborhood across three scales:



Scale-Invariant Feature Transform

Candidate keypoints with low contrast or that lie on edges are removed

This is done based on the *local Hessian* of the response $D(x, y, \sigma)$:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Eigenvalues of the local Hessian are proportional to the *principal curvature*

SIFT looks at the ratio of the two eigenvalues $r = \frac{\lambda_1}{\lambda_2}$; it rejects whenever:

$$\frac{(r+1)^2}{r} = \frac{\mathrm{trace}(\mathbf{H})^2}{\det(\mathbf{H})} > 10$$

Scale-Invariant Feature Transform

Example of removing unstable candidate feature points:

Scale-Invariant Feature Transform

Feature points in low-contrast regions are removed:

Scale-Invariant Feature Transform

Feature points that lie on edges are removed:

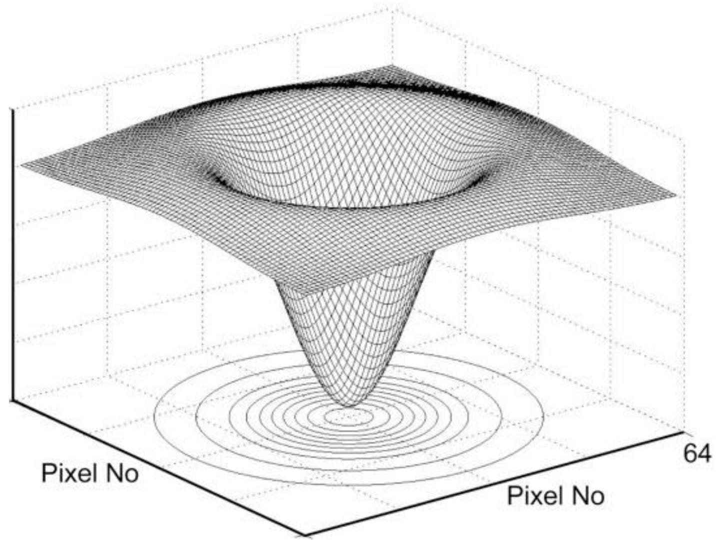
Harris-Laplace detectors

Harris-Laplace combines Harris corner detector with Gaussian scale space:

- Find *characteristic scale* for location using *Laplacian-of-Gaussian* (LoG):

$$\det(LoG(\mathbf{x},\sigma)) = \sigma^2(L_{xx}(\mathbf{x},\sigma) + L_{yy}(\mathbf{x},\sigma))$$

- The maximum value of this quantity is used as a basis for the *scale selection*
- Note LoG and DoG look alike!
- Apply the Harris corner detector at the characteristic scale



*L is Gaussian-blurred image

Hessian-Laplace detectors

Hessian-Laplace uses the same *scale selection* criterion as *Harris-Laplace*

Do not look at structure tensor, but at Hessian of Gaussian-filtered image:

$$\mathbf{H} = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

Specifically, find points for which *trace* and *determinant* of Hessian is highest:

$$\det(\mathbf{H}) = \sigma^2 (L_{xx} L_{yy} - L_{xy}^2)$$

$$\mathrm{trace}(\mathbf{H}) = \sigma (L_{xx} + L_{yy})$$

What have we learned in this lecture?

- Feature points are stable in location, rotation, and preferably also scale
- Harris performs spectral analysis of second-moment matrix / structure tensor
- By contrast, SIFT inspects the local Hessian of difference images

Reading material: Section 3 and 4.1.1 of Szeliski