CSCI 4830 / 5722
Computer Vision

University of Colorado **Boulder**

CSCI 4830 / 5722
Computer Vision

Dr. Ioana Fleming
Spring 2019
Lecture 16

Reminders

Submissions:

- Homework 3: Sat 3/2 at 11 pm

Readings:

- Szeliski:
  - chapter 4.1 (Feature detection – Points and patches)
- P&F:
  - chapter 5 (Local features – corners, SIFT features)

Today

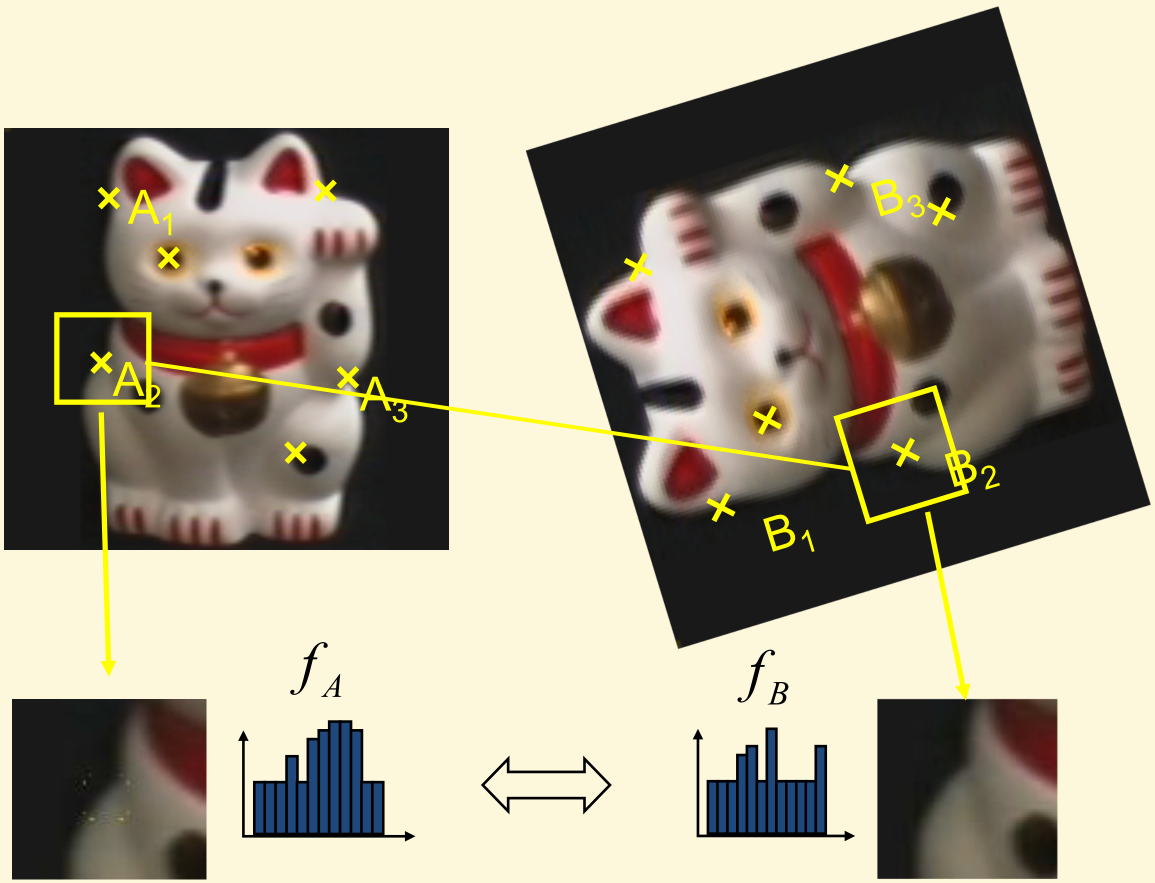
- Multi-scale oriented patches
- Laplacian and LOG
- SIFT features

Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

K. Grauman, B. Leibe

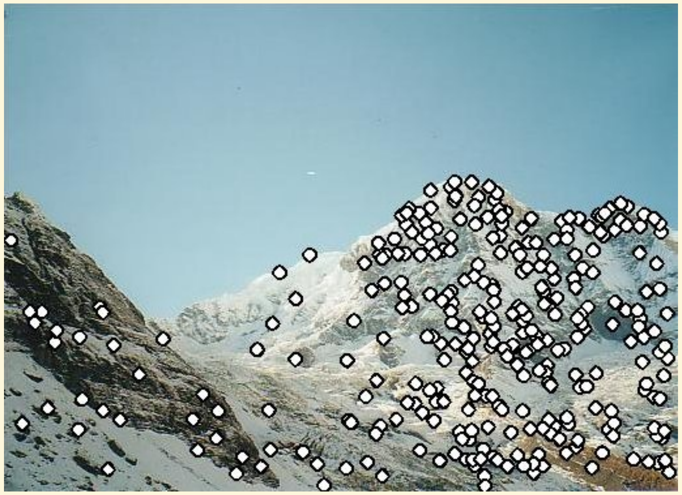
Overview of Keypoint Matching



$$d(f_A, f_B) < T$$

1. **Find a set of distinctive key-points**
2. **Define a region around each keypoint**
3. **Extract and normalize the region content**
4. **Compute a local descriptor from the normalized region**
5. **Match local descriptors**

Feature descriptors?



Intuitively:

- Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical "fingerprint" that can be used to differentiate one feature from another. Ideally this information would be invariant under image transformation, so we can find the feature again even if the image is transformed in some way.

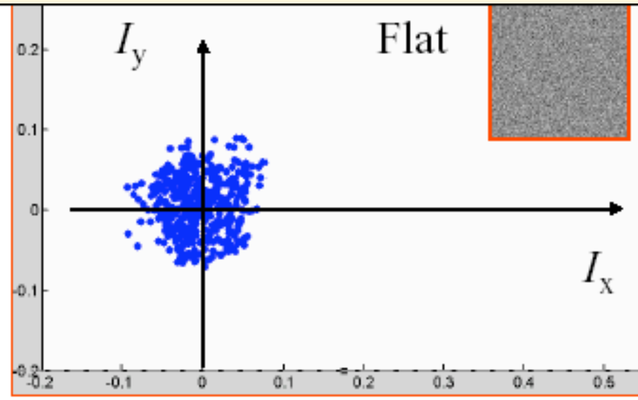
K. Grauman, B. Leibe

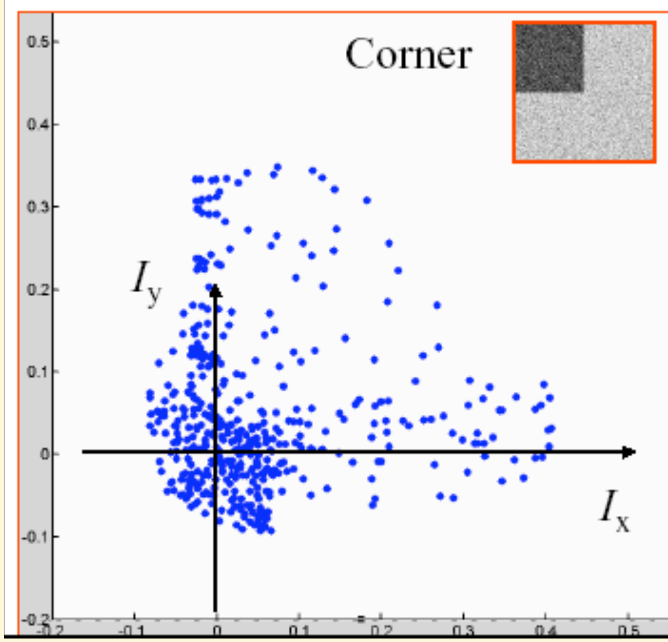
Many Existing Detectors Available

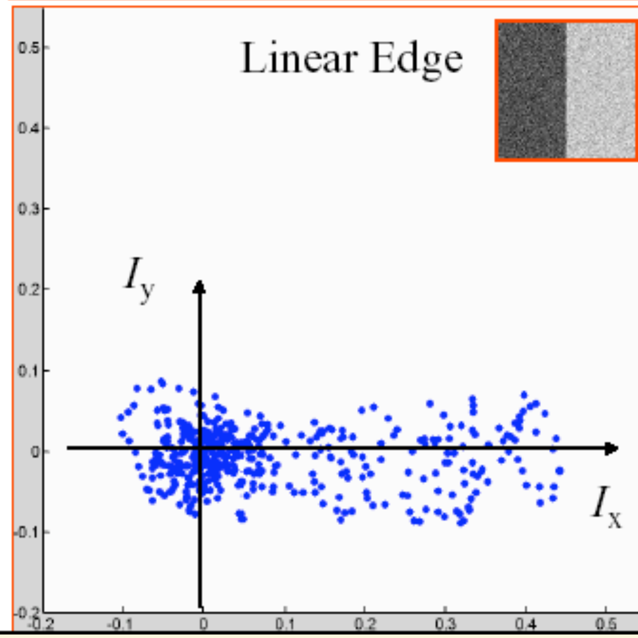
Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
Laplacian, DoG	[Lindeberg ‘98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid ‘01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid ‘04]
EBR and IBR	[Tuytelaars & Van Gool ‘04]
MSER	[Matas ‘02]
Salient Regions	[Kadir & Brady ‘01]
SIFT	[Lowe 2004]
Others…	

Plotting derivatives

The distribution of the x and y derivatives is very different for all three types of patches

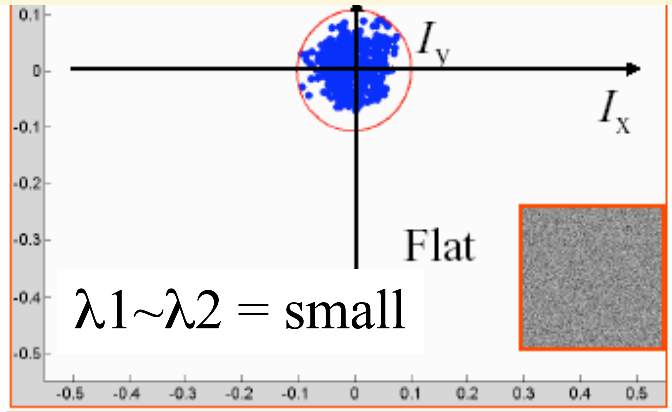


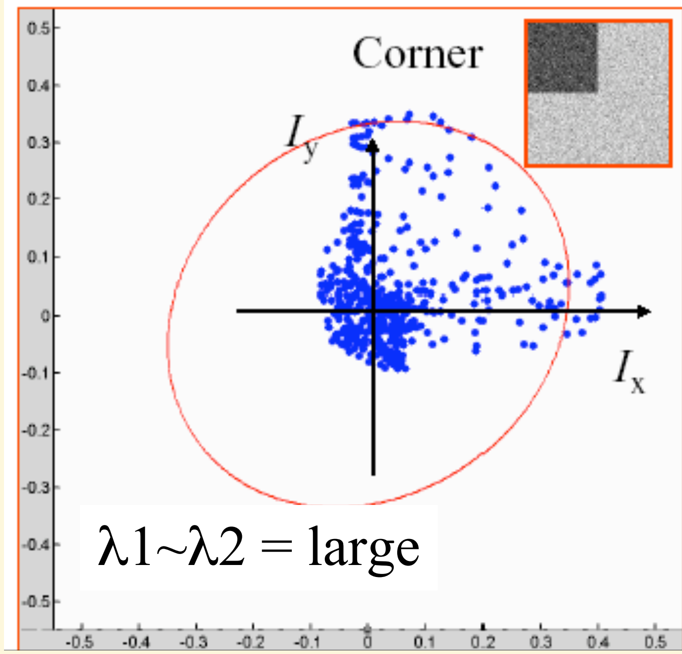


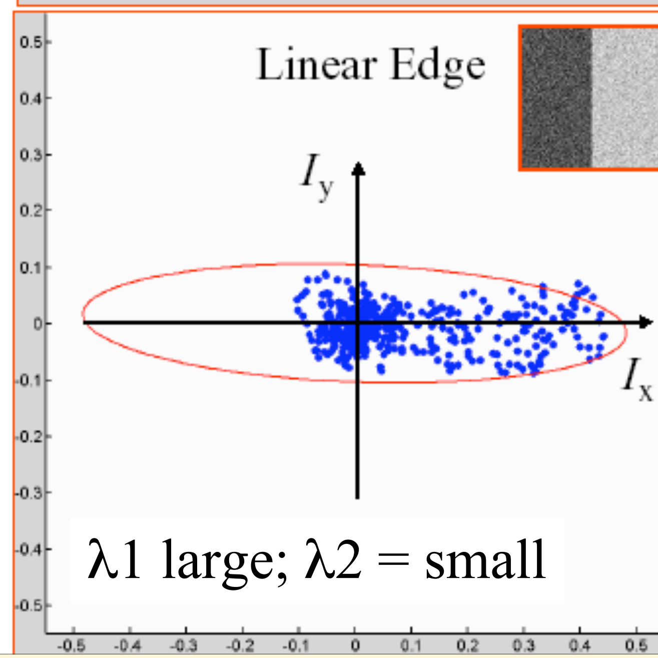


Fitting ellipses

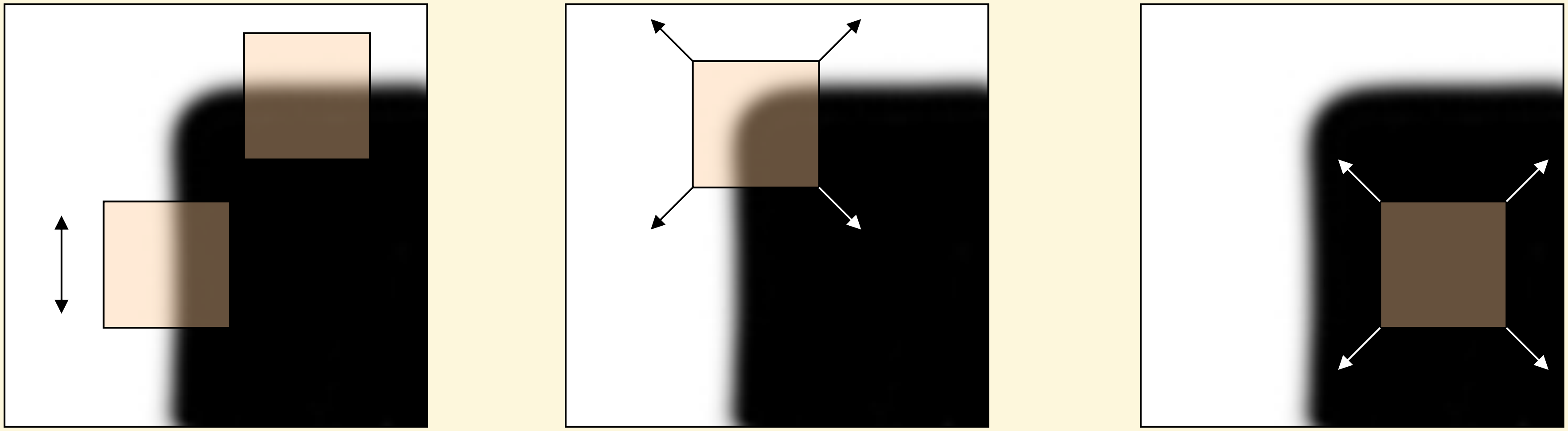
The distribution of x and y derivatives can be characterized by the shape and size of the principal component ellipse







Corner response function



"edge":

$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$

"corner":

λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$;

"flat" region

λ_1 and λ_2 are small;

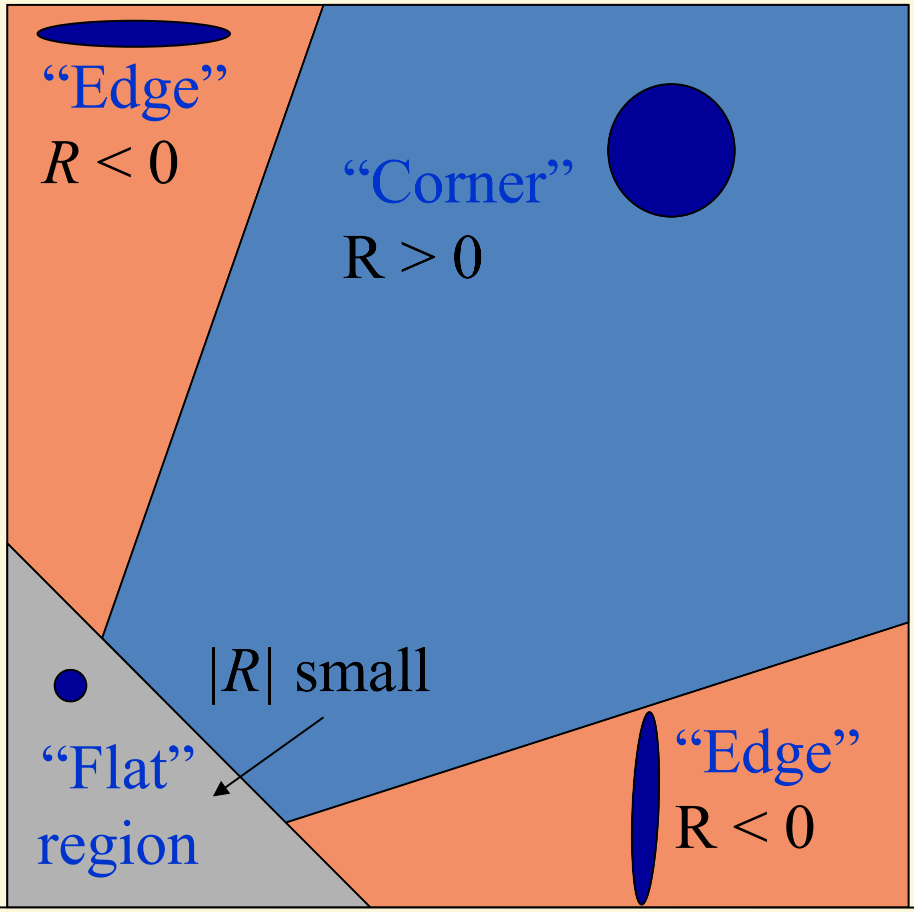
$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

Corner response function

$$R = \det(M) - \alpha\,\mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

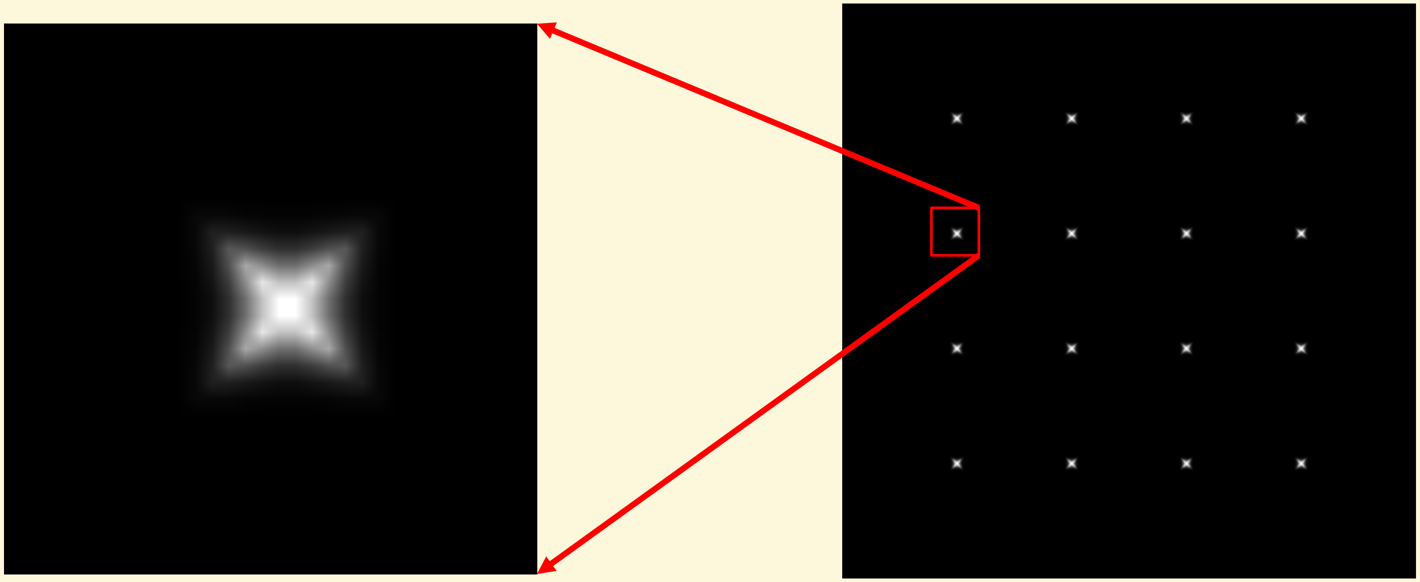
R = “cornerness”



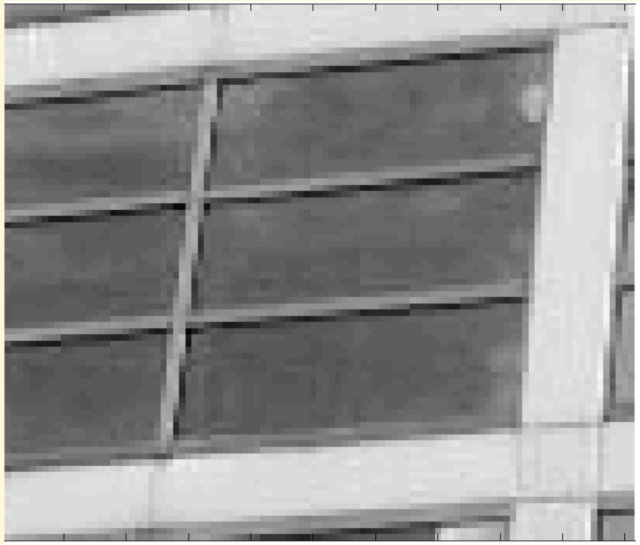
Harris corner detector

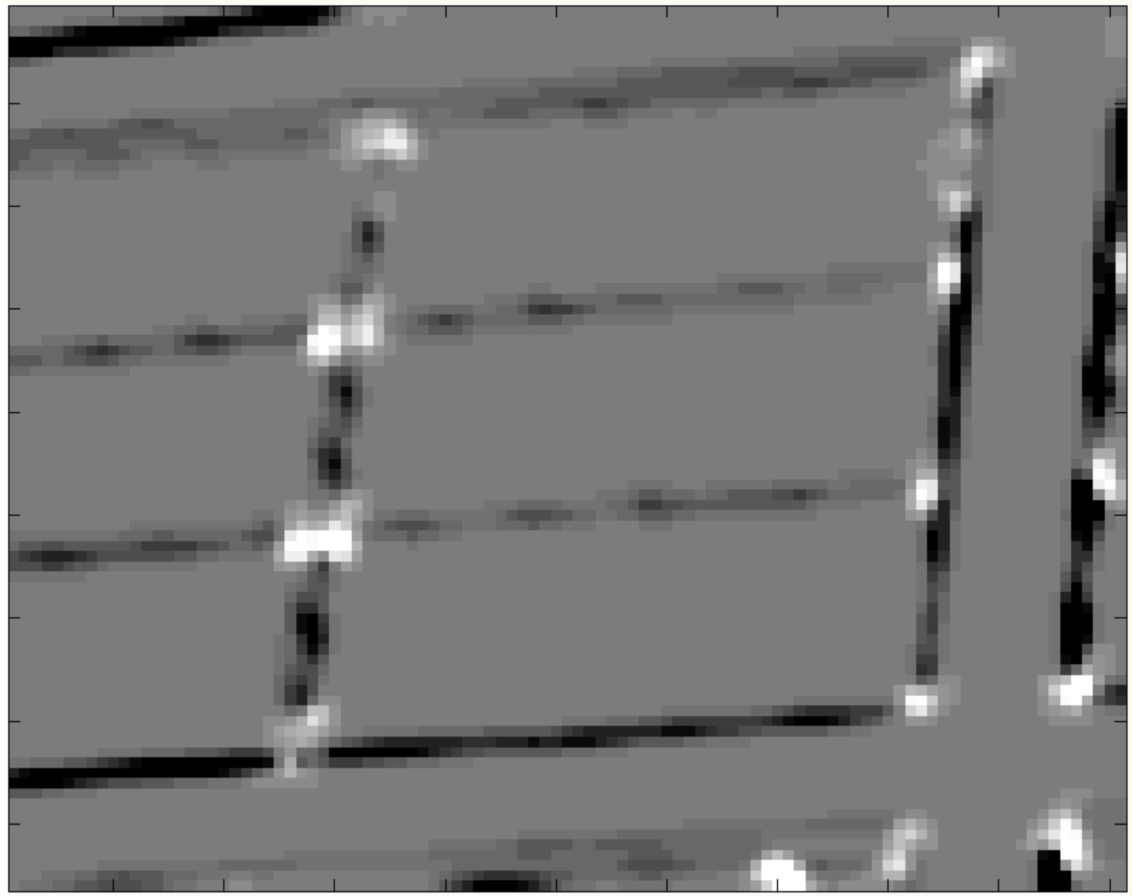
1) Filter image with Gaussian window W.
2) Compute magnitude of the gradient everywhere.
3) Compute *M* matrix for each image window, using the same W Gaussian window function
4) Computer the *cornerness (R)* scores for each image window.
5) Find points whose surrounding window gave large corner response (R > threshold)
6) Take the points of local maxima (*non-maximum suppression*)

Non-Maximum Supression



Corner Response Example



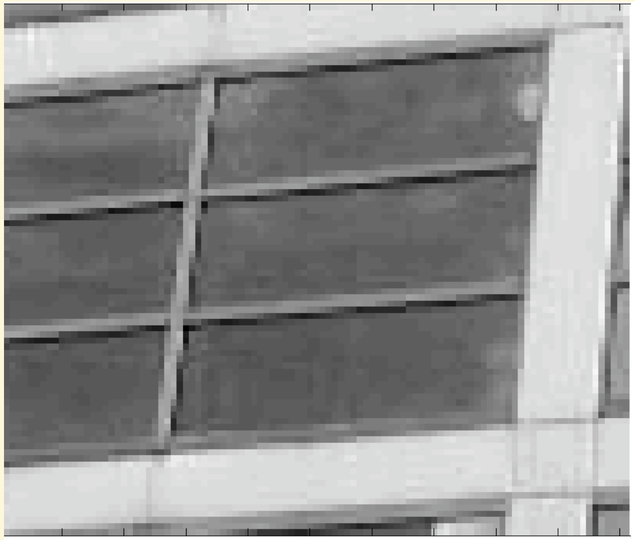


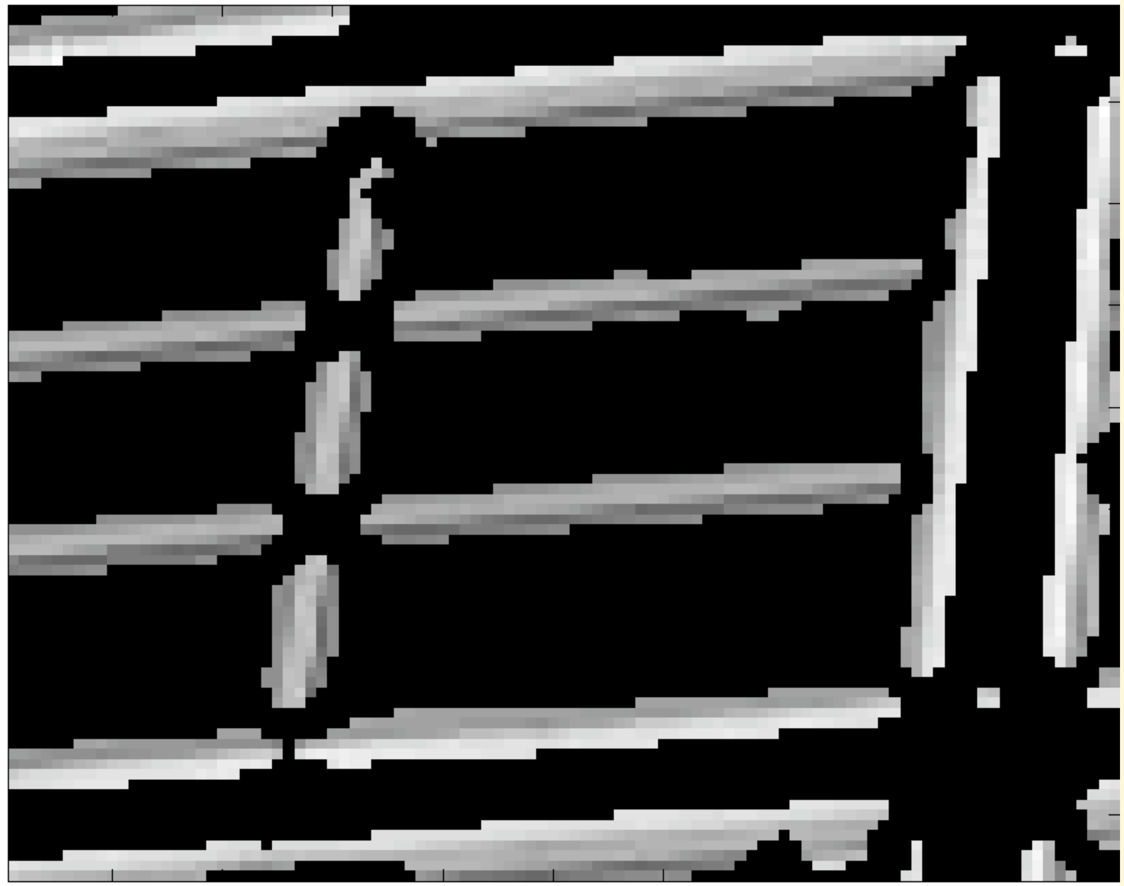
Harris R score.

Ix, Iy computed using Sobel operator

Windowing function w = Gaussian, sigma=1

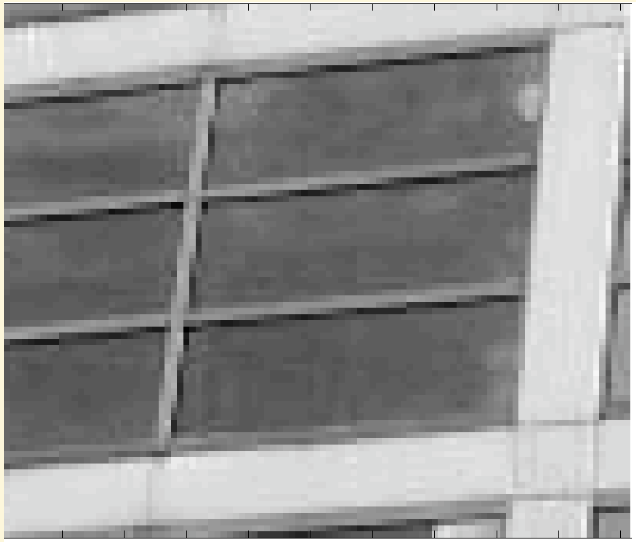
Corner Response Example

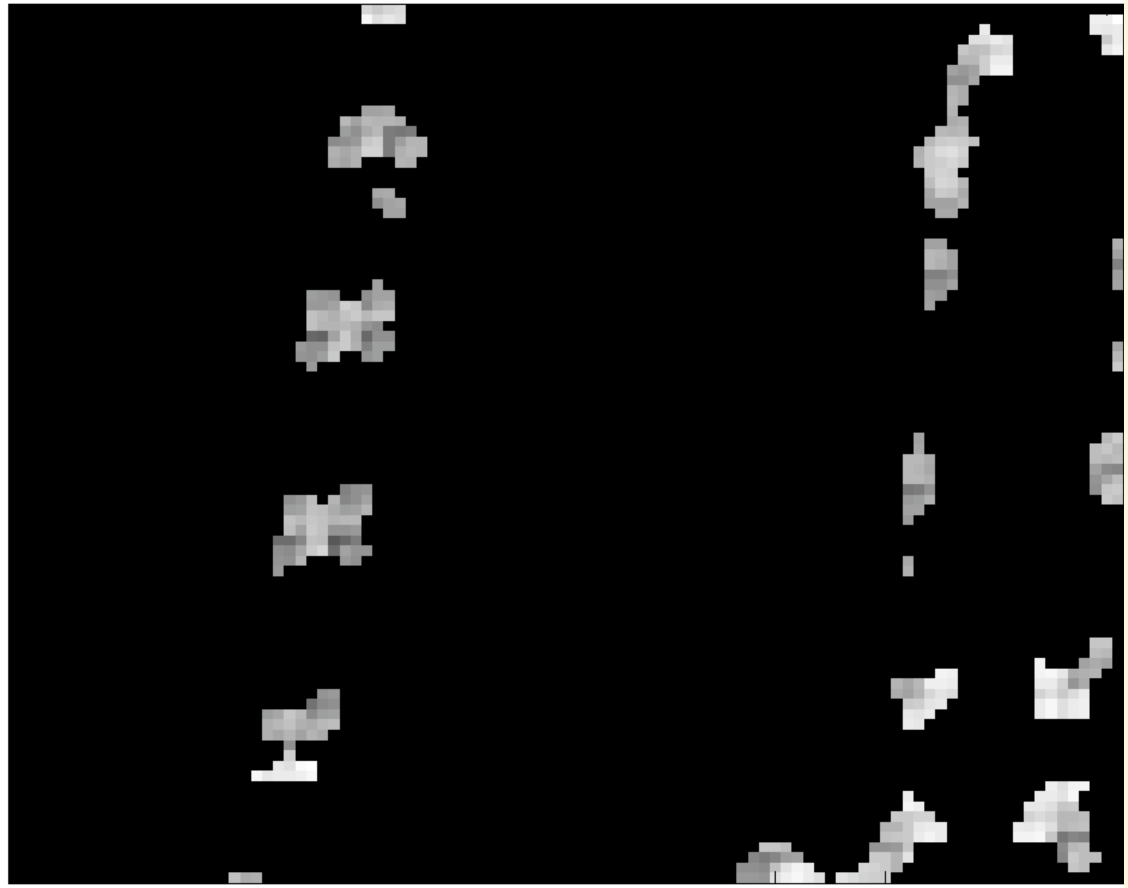




Threshold: R < -10000
(edges)

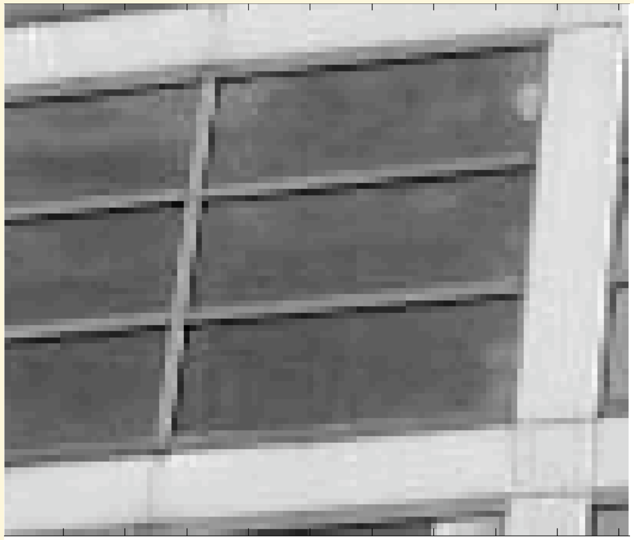
Corner Response Example

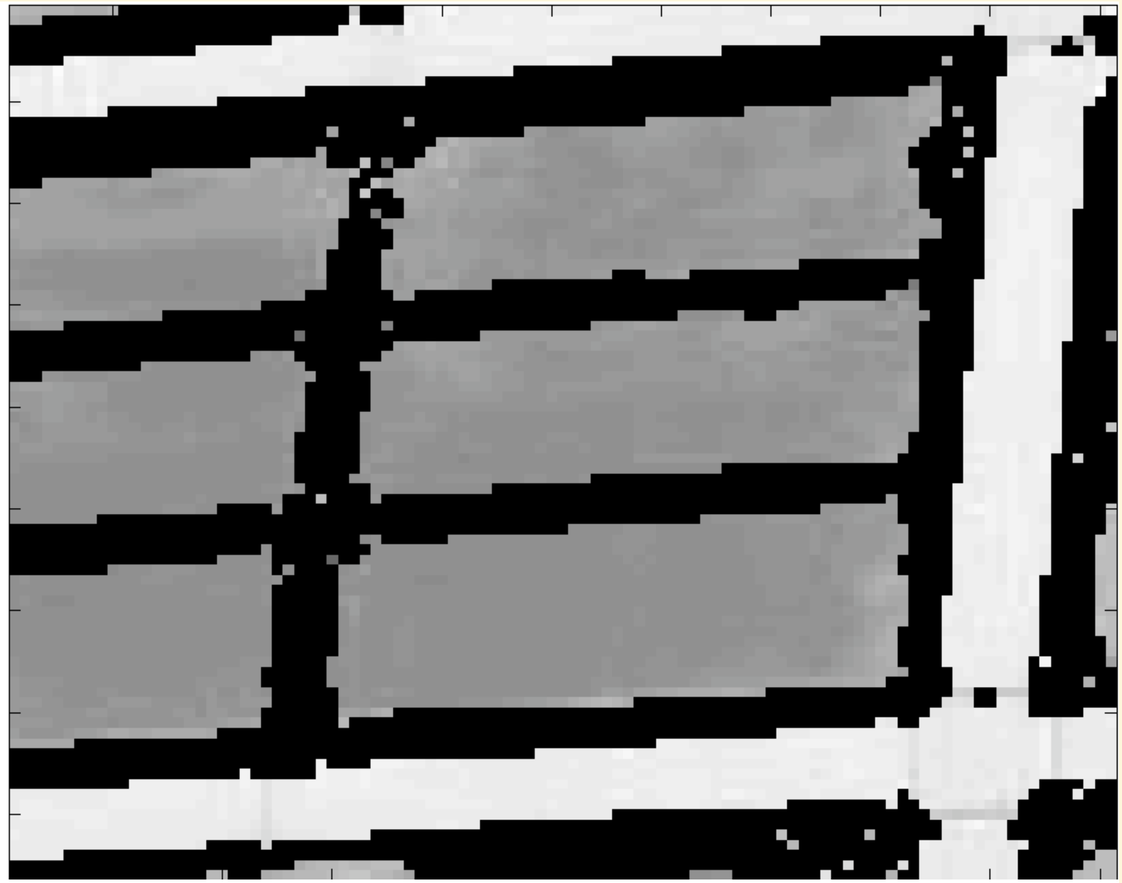




Threshold: > 10000
(corners)

Corner Response Example





Threshold: $-10000 < R < 10000$
(neither edges nor corners)

Properties of the Harris corner detector

- Rotation invariant?

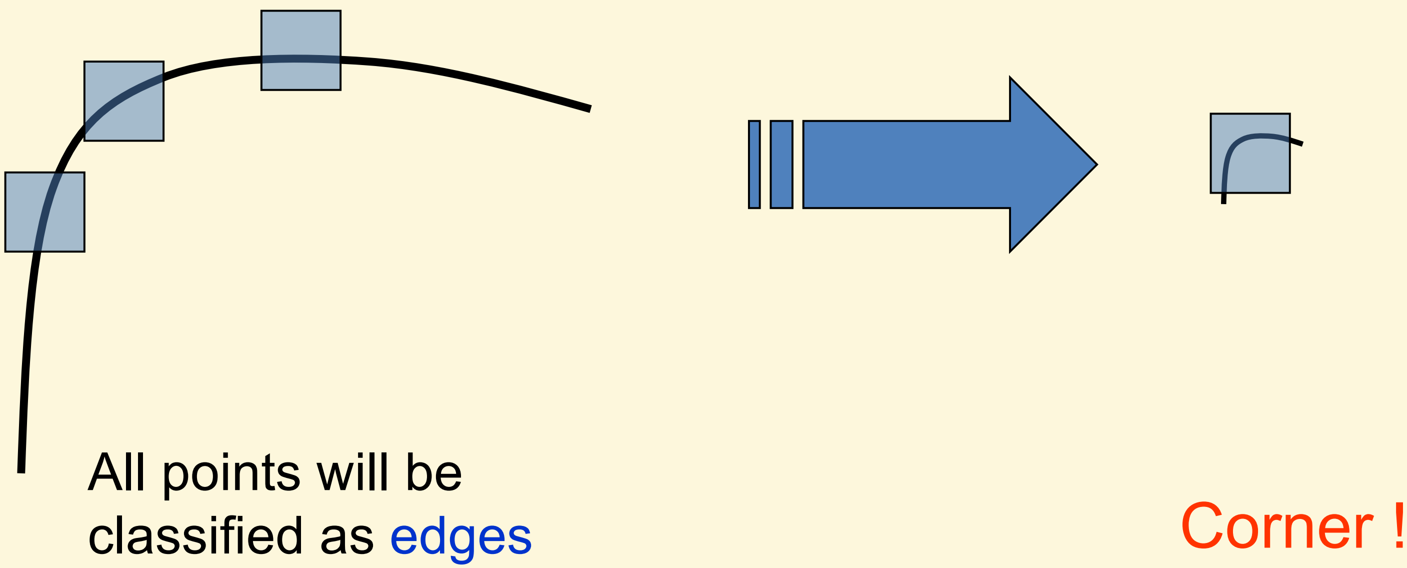
Yes $$M = X\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}X^T$$

- Scale invariant?

Properties of the Harris corner detector

- Rotation invariant? Yes

- Scale invariant? No



Scale variance solution:

- Exhaustive multi-scale approach

Scale variance solution:

- Exhaustive multi-scale approach

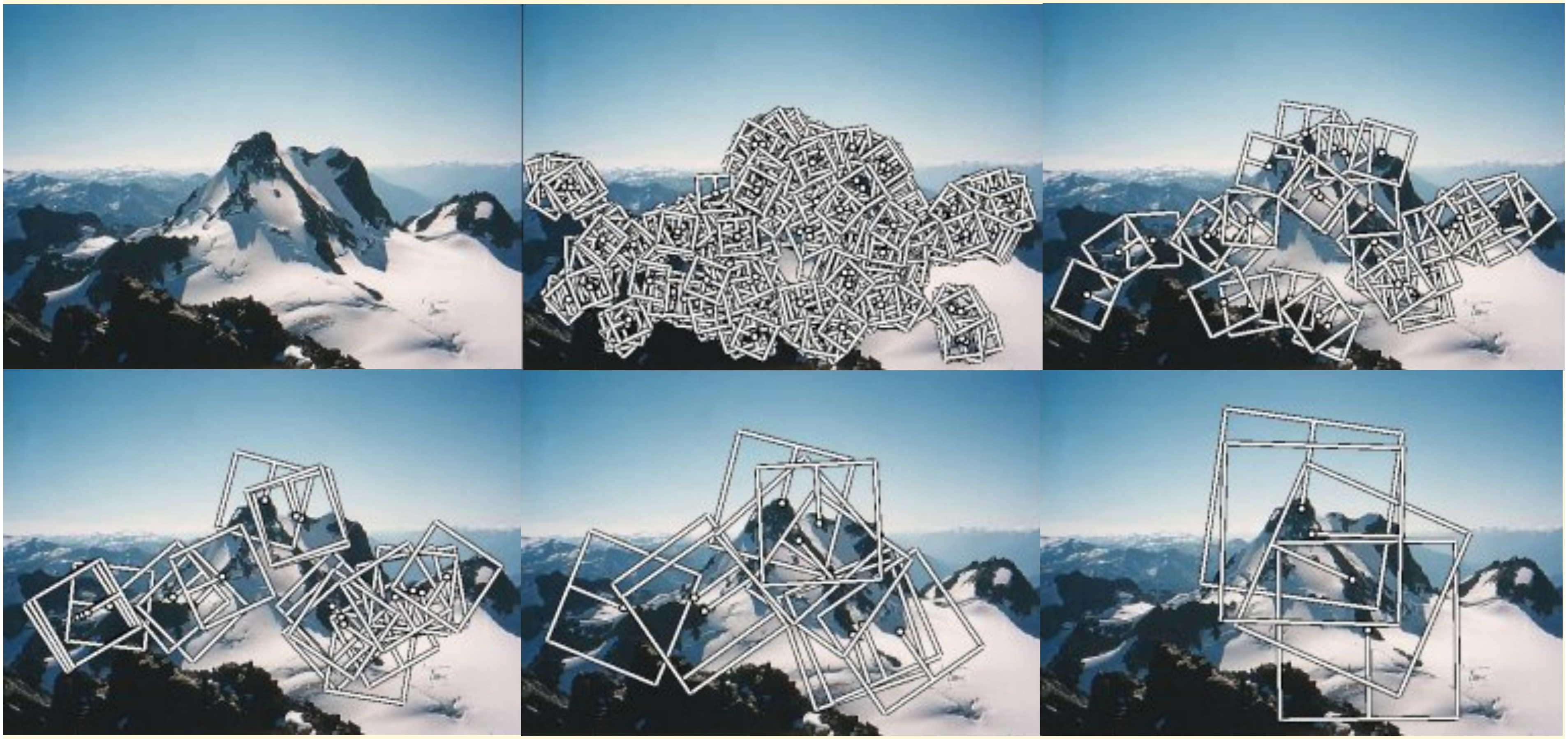
Scale variance solution:

- Exhaustive multi-scale approach

Scale variance solution:

- Exhaustive multi-scale approach

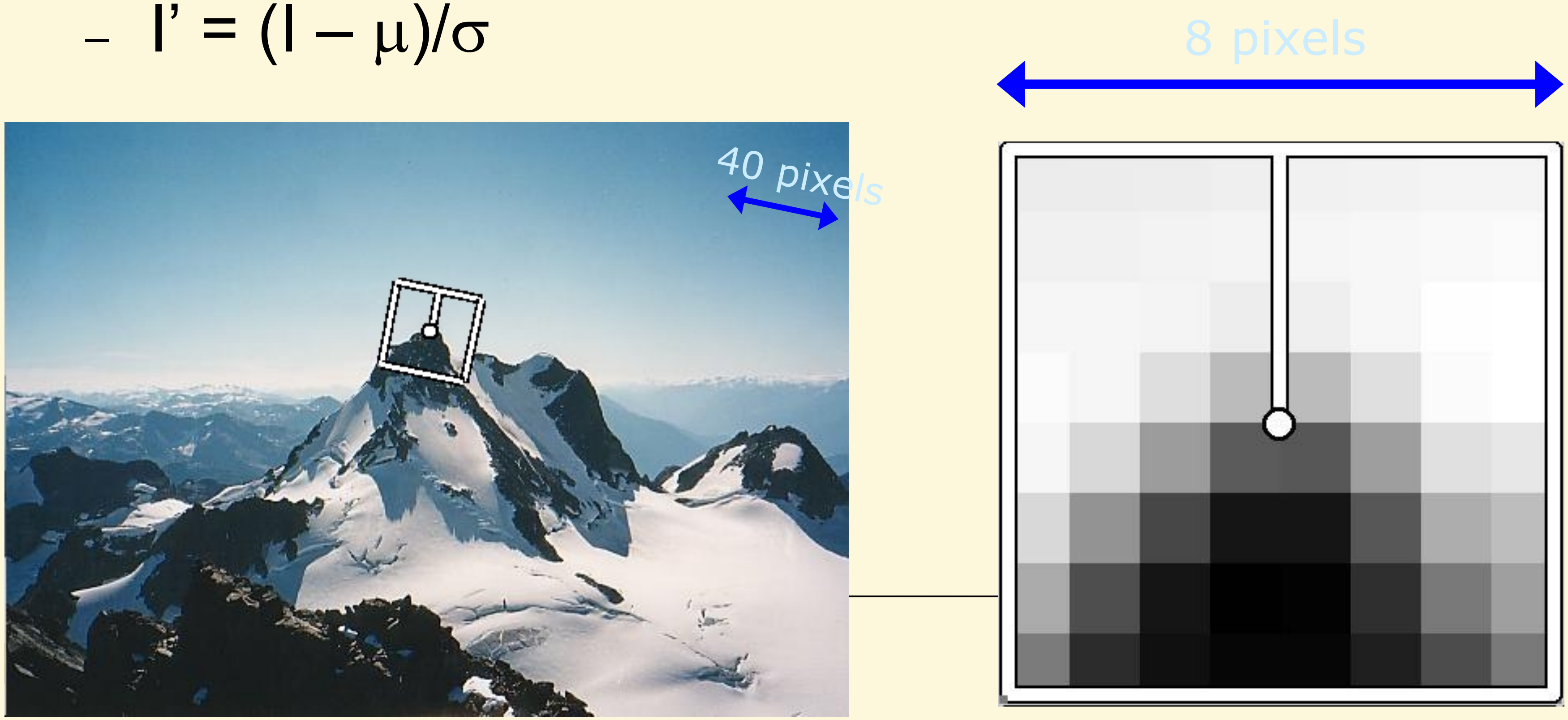
Multi-Scale Oriented Patches



- Extract oriented patches at multiple scales using dominant orientation

Multi-Scale Oriented Patches

- Sample scaled, oriented patch
  - 8x8 patch, sampled at 5 x scale
- Bias/gain normalized (subtract the mean of a patch and divide by the variance to normalize)
  - $I' = (I - \mu)/\sigma$



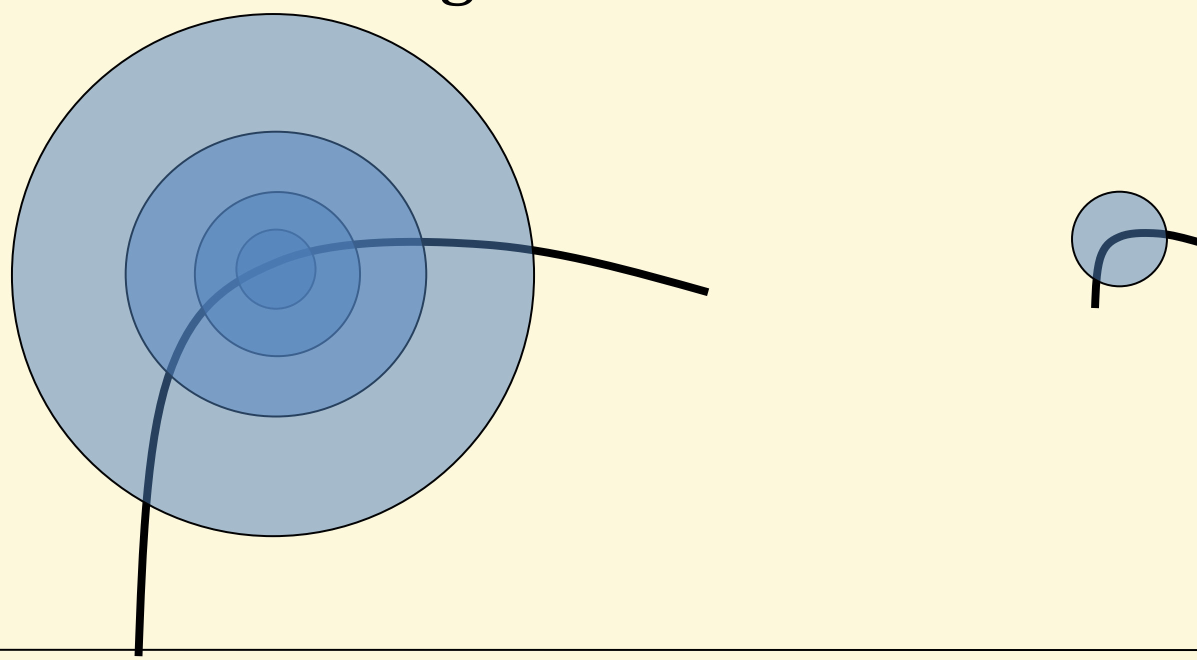
Matching Interest Points: so far

- Harris corners / correlation
  - Extract and match repeatable image features
  - Robust to clutter and occlusion
  - BUT not invariant to scale and rotation
- Multi-Scale Oriented Patches
  - Corners detected at multiple scales
  - Descriptors oriented using local gradient
  - Invariant to scale and rotation

Leads to: **SIFT** – state of the art image features

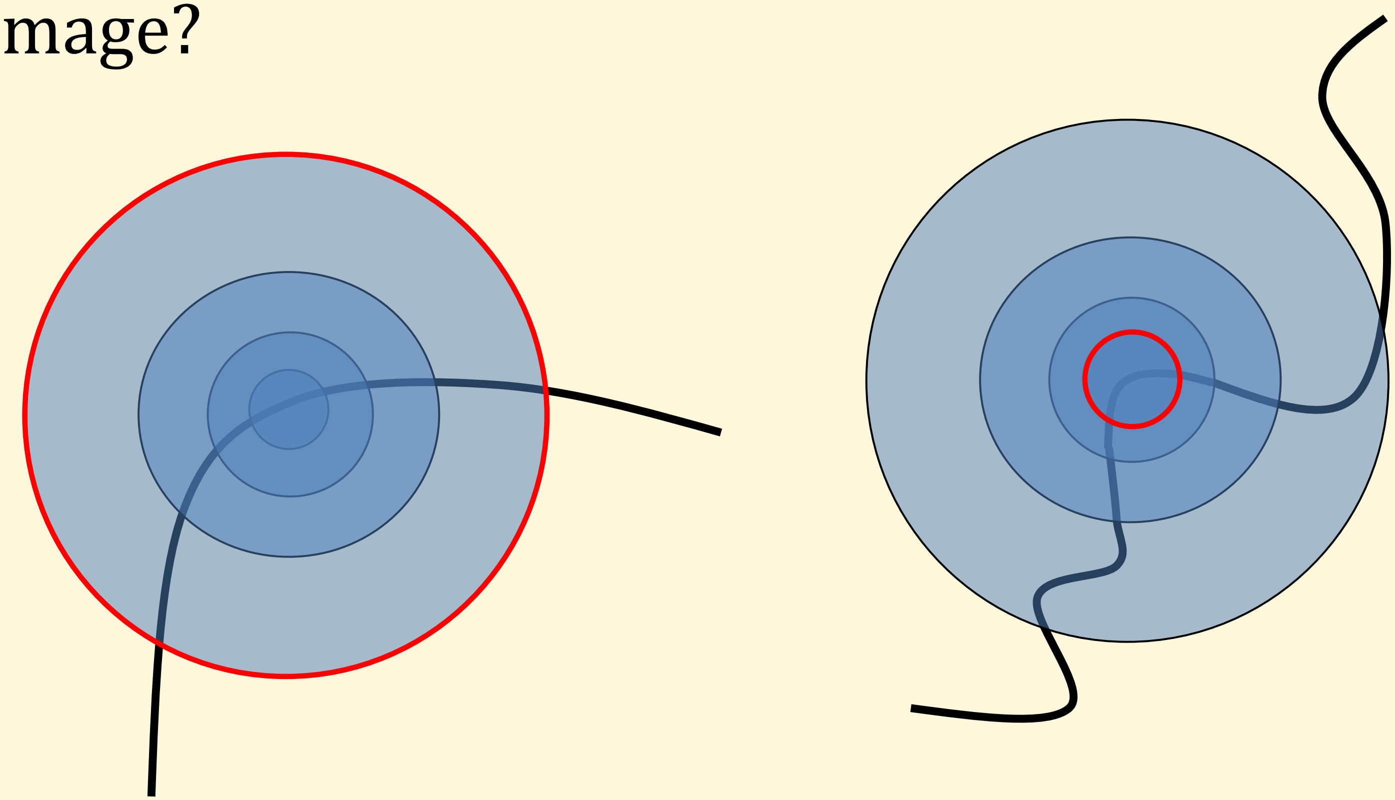
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

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

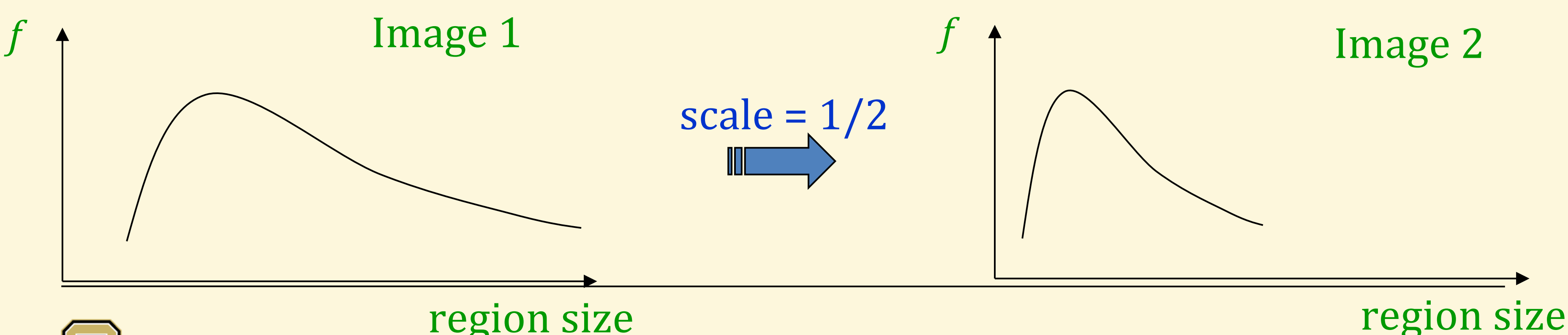


Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (circle radius)



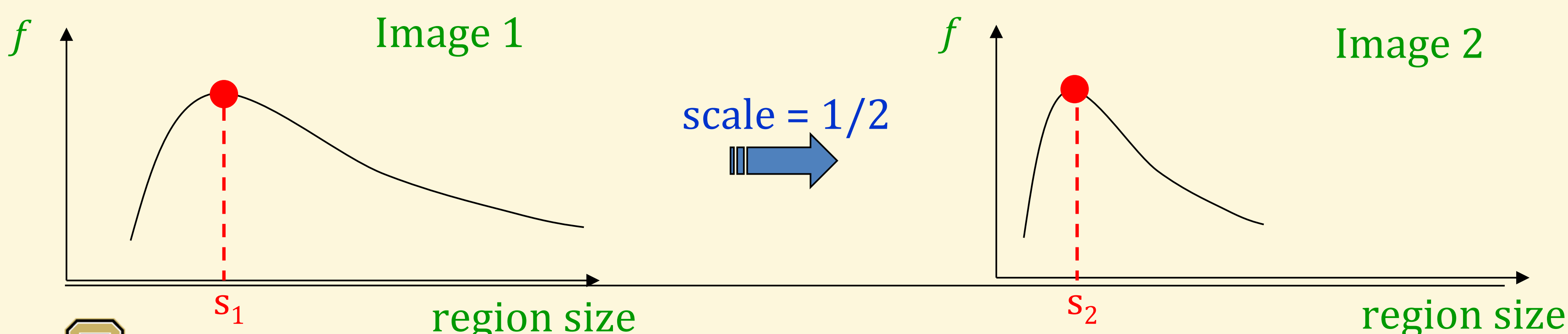
Scale Invariant Detection

- Common approach:

  Take a local maximum of this function

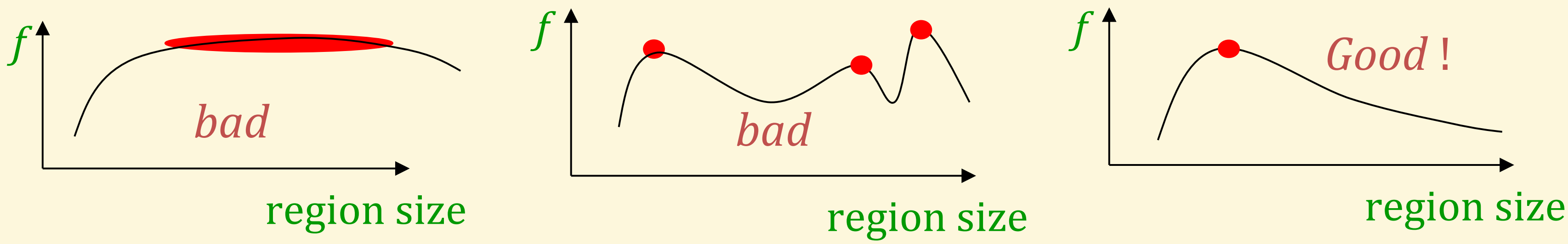
  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

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



Scale Invariant Detection

- A "good" function for scale detection:
  has one stable sharp peak



- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)

Scale Invariant Detection

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

Kernels:

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

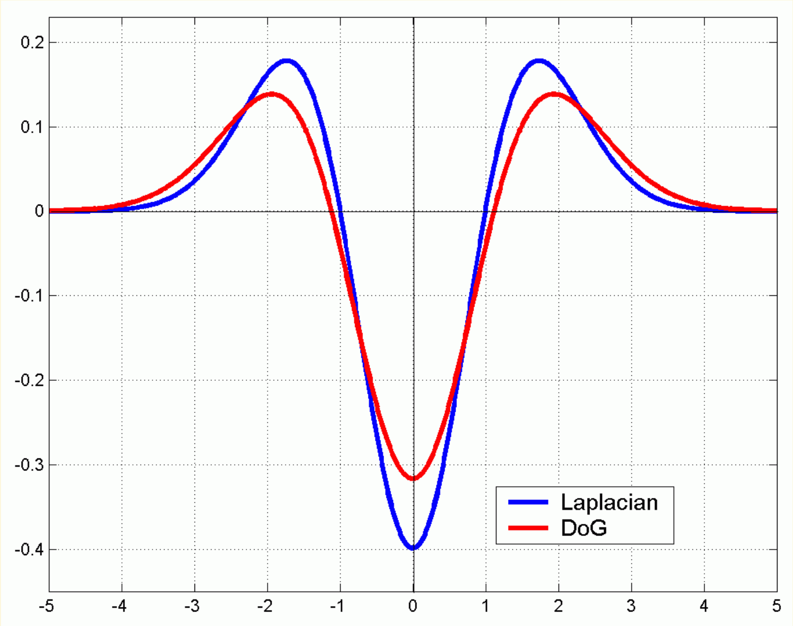
(Laplacian)

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

(Difference of Gaussians)

where Gaussian

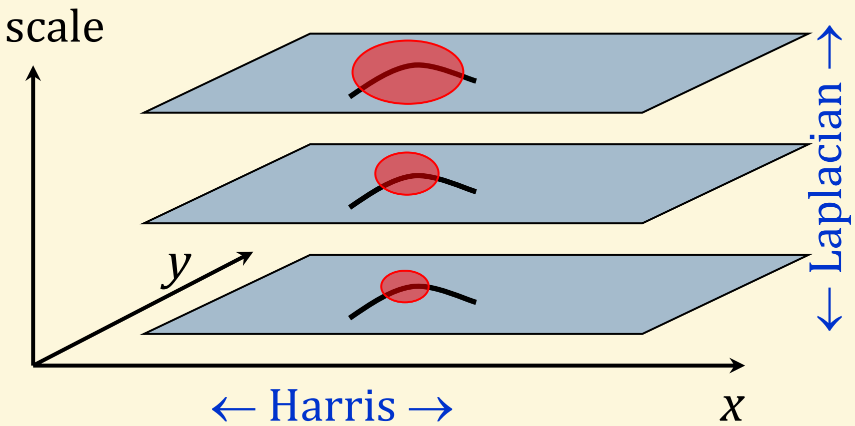
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



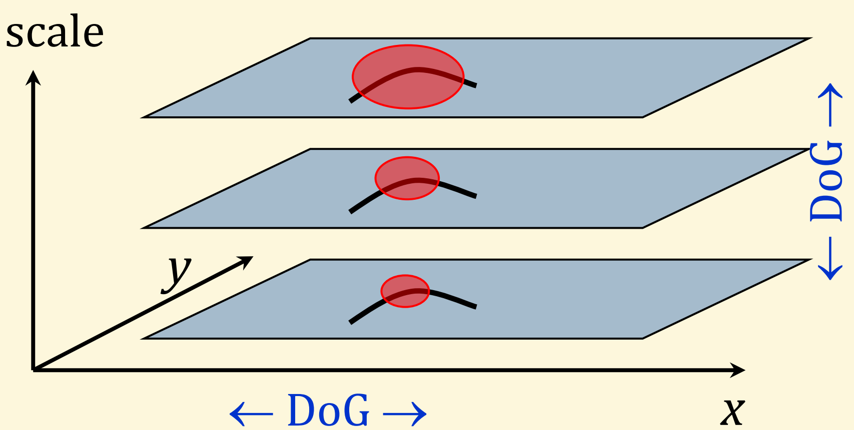
Note: both kernels are invariant to *scale* and *rotation*

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



- SIFT (Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

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)

Methods:

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

SIFT: major stages of computation

1. Scale-space extrema detection.
2. Keypoint localization.
3. Orientation assignment.
4. Keypoint descriptor.

Typical image of size 500x500 pixels produces about 2000 stable keypoints

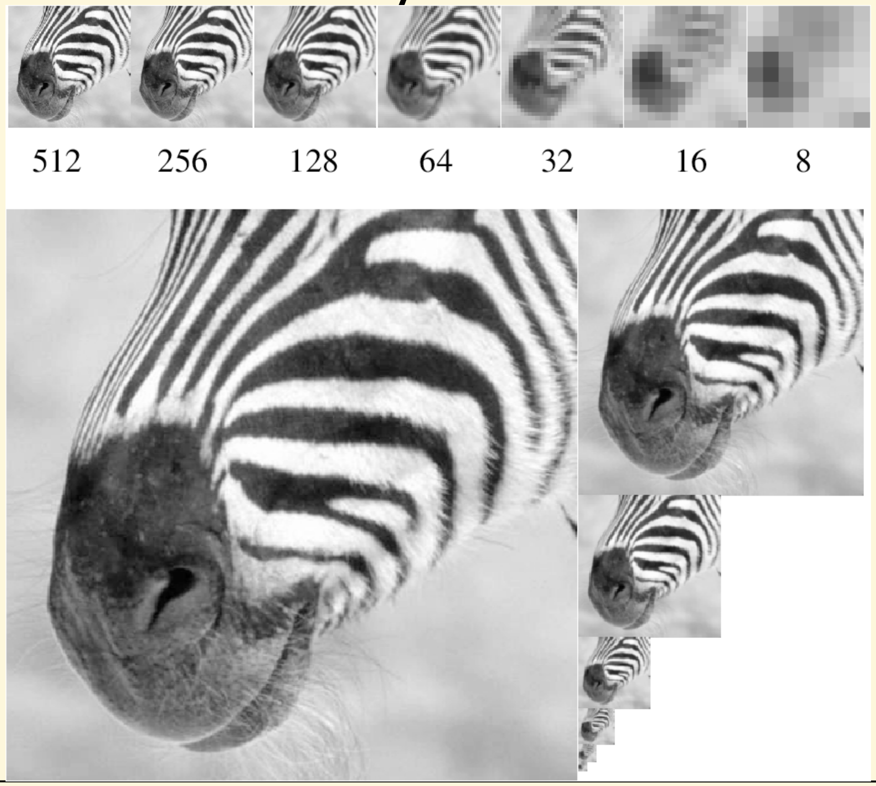
- Near Real-time performance:

***Cascade approach* –**

keep heavy operations only to keypoints that "survive".

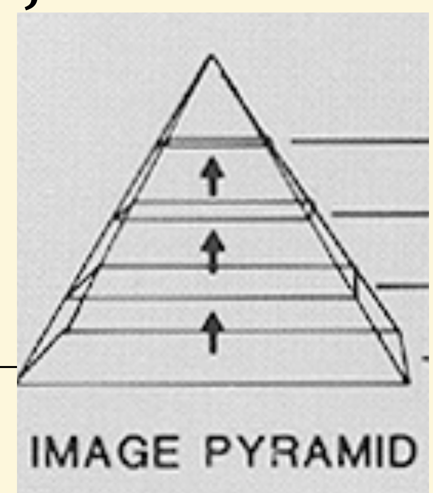
1. Scale Space

- Different scales are appropriate for describing different objects in the image, and we may not know the correct scale/size ahead of time.

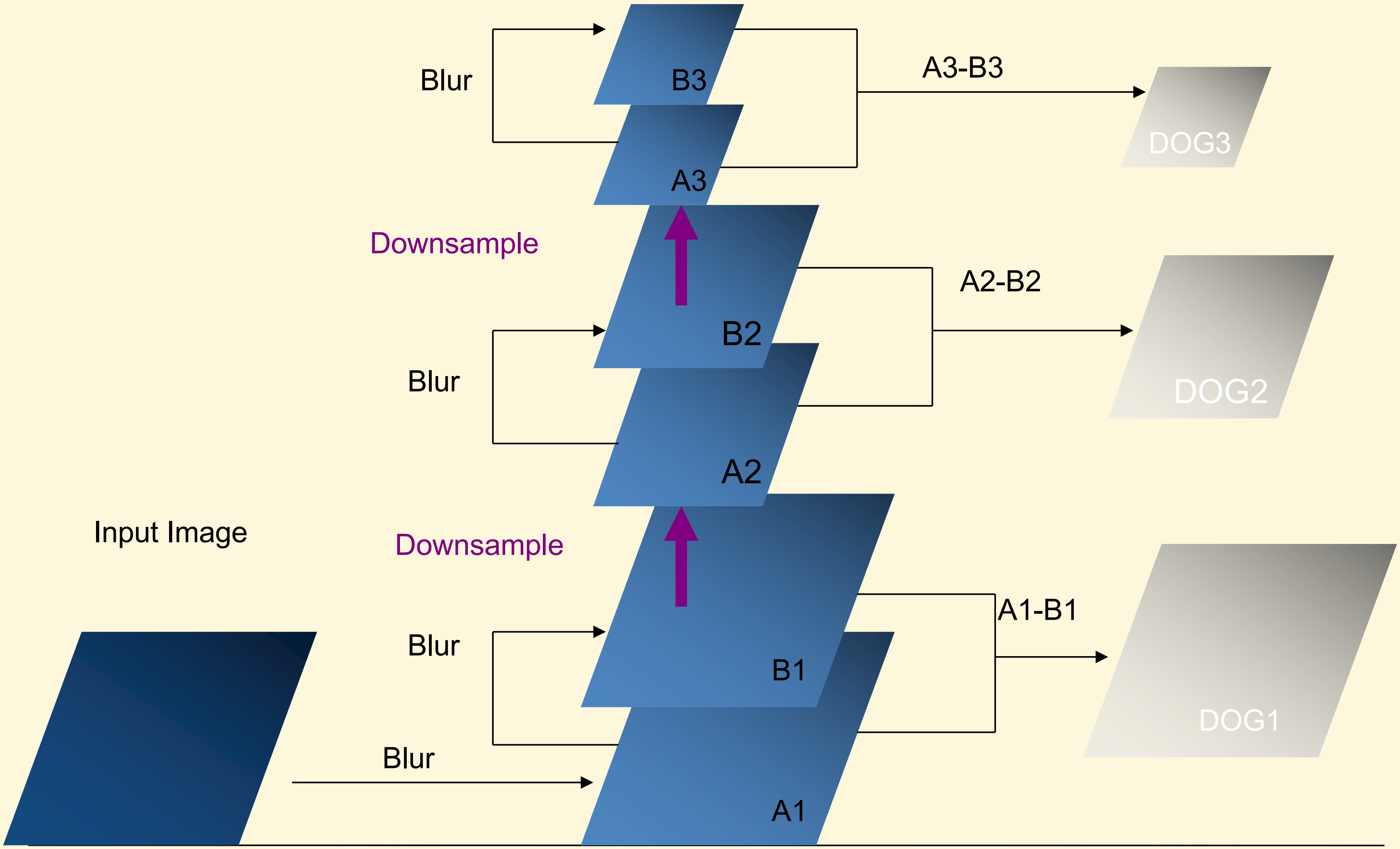


Difference of Gaussian

1. A = Convolve image with vertical and horizontal 1D Gaussians, σ=sqrt(2)
2. B = Convolve A with vertical and horizontal 1D Gaussians, σ=sqrt(2)
3. DOG (Difference of Gaussians) = A – B
4. Down-sample B with bilinear interpolation by a factor of 2 (linear combination of 4 adjacent pixels)



Difference of Gaussian Pyramid

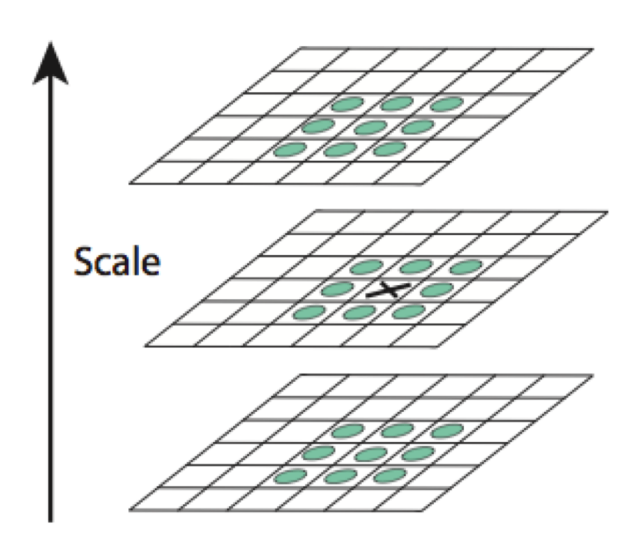


University of Colorado **Boulder**

Pyramid Example

Feature detection

- Find maxima and minima of scale space
- For each point on a DOG level:
  - Compare to its 8 neighbors at same level
  - Compare with its 9 neighbors in the scale above and below.
  - It is selected only if it is larger than all of these neighbors or smaller than all of them. If not, discard.
- Repeat for each DOG level
- Those that remain are key points



2. Keypoint Localization
a) Refining Key List: Illumination

- For all levels, use the A smoothed image to compute
  - Gradient Magnitude

$$M_{ij} = \sqrt{(A_{ij} - A_{i+1,j})^2 + (A_{ij} - A_{i,j+1})^2}$$

- Threshold gradient magnitudes:
  - Remove all key points with M_{ij} less than 0.1 times the max gradient value
- Motivation: Low contrast is generally less reliable than high contrast for feature points

2. Keypoint Localization
b) Refining Key List: reject edges

- Reject points with strong edge response in one direction only
- Like Harris - using Trace and Determinant of Hessian

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

2. Keypoint Localization
b) Refining Key List: reject edges

- Let α and β be the eigenvalue with the largest and the smallest magnitude

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

- Ratio of principal curvatures *r* is the ratio between eigenvalues: $\alpha = r\,\beta$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

2. Keypoint Localization
b) Refining Key List: reject edges

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

The quantity $(r+1)^2/r$ is at a minimum when the two eigenvalues are equal and it increases with *r.*

Therefore, to check that the ratio of principal curvatures is below some threshold, *r*, we only need to check

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

2. Keypoint Localization
b) Refining Key List: reject edges

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

- It is very efficient to compute: < 20 f.p. operations required to test each keypoint.
- SIFT paper uses ***r= 10***
  - *initially 832 keypoints*
  - *then 729 keypoints after minimum contrast*
  - *then 536 keypoints after threshold on **r***

(a) (b) (c) (d)

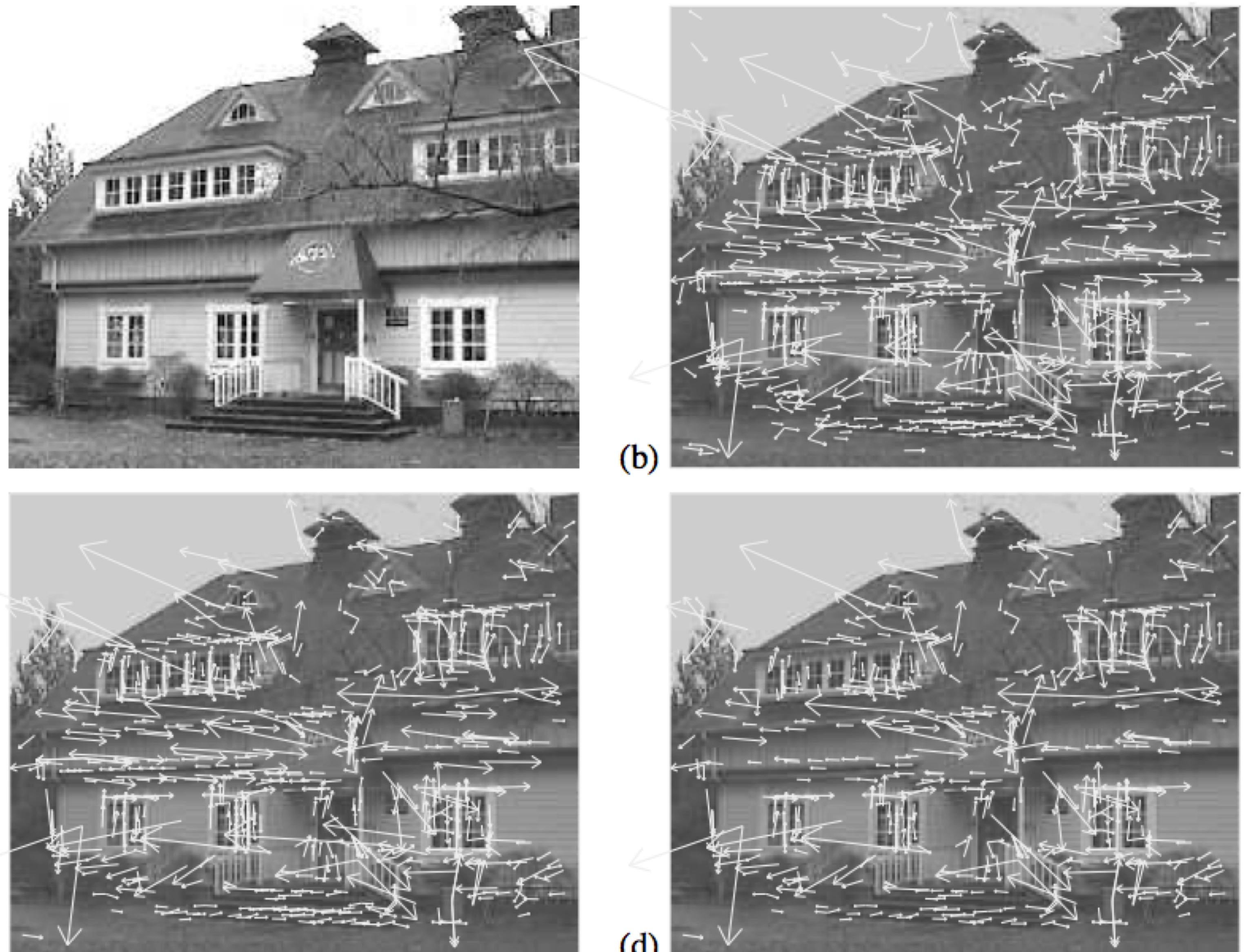
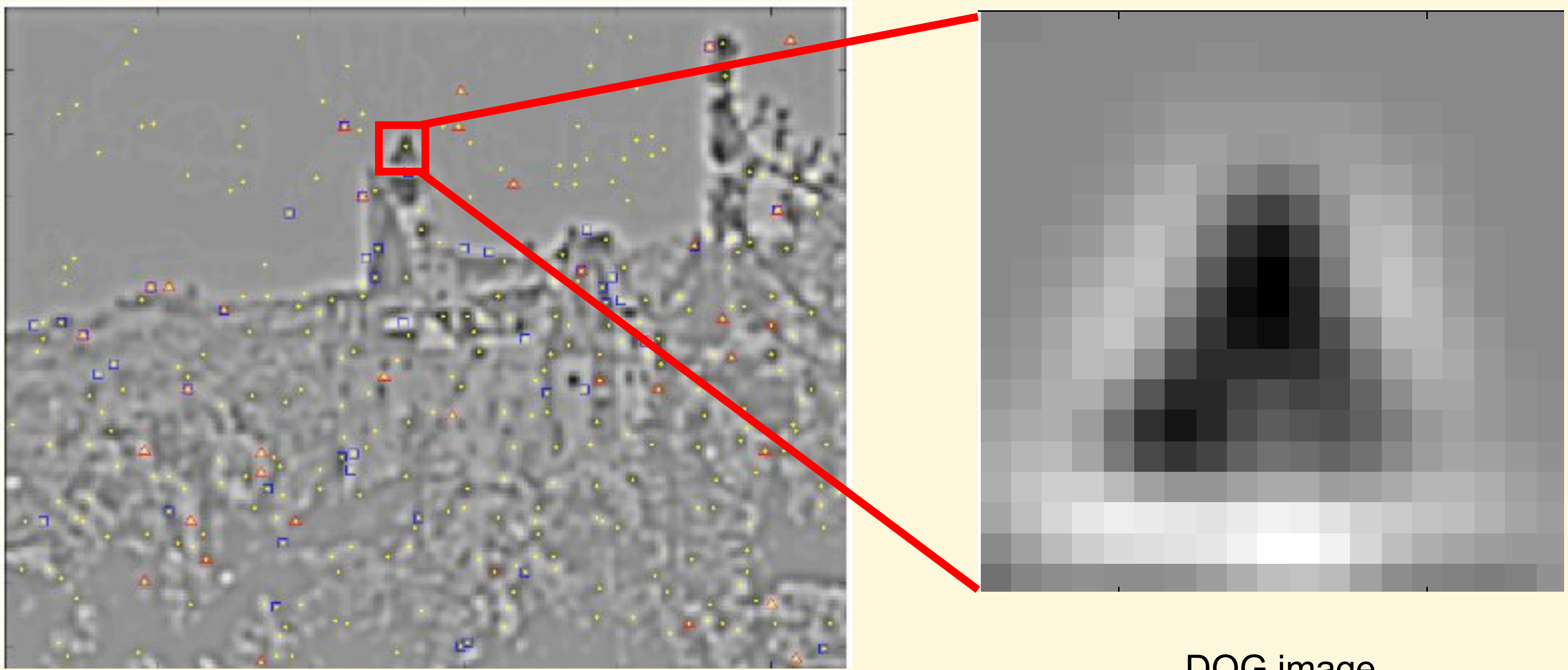


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

3. Orientation assignment

- For each remaining key point:
  - Choose surrounding N x N window at DOG scale level it was detected

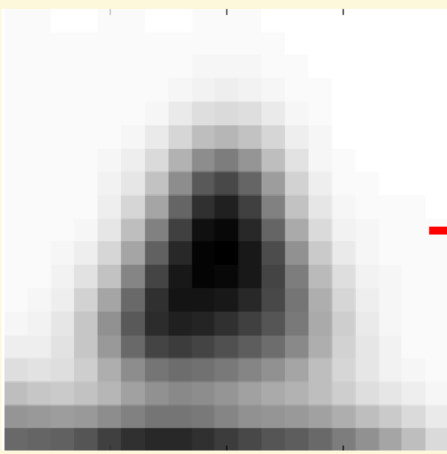


DOG image

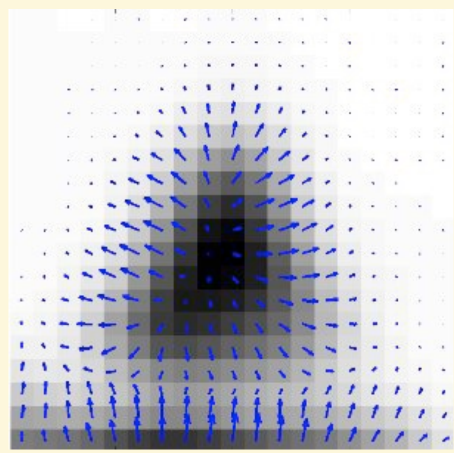
3. Orientation assignment

- For all levels, use the A smoothed image to compute
  - Gradient Orientation

$$R_{ij} = \text{atan2}(A_{ij} - A_{i-1,j}, A_{i,j+1} - A_{ij})$$

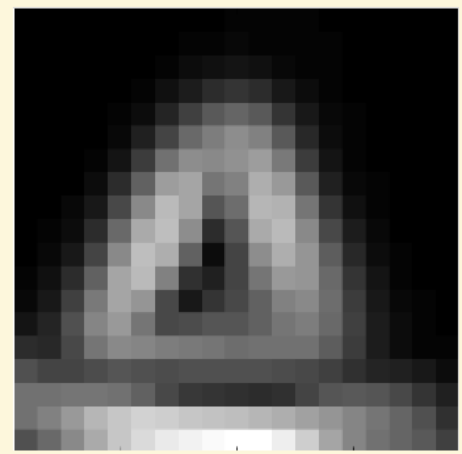


Gaussian Smoothed Image



Gradient Orientation

+

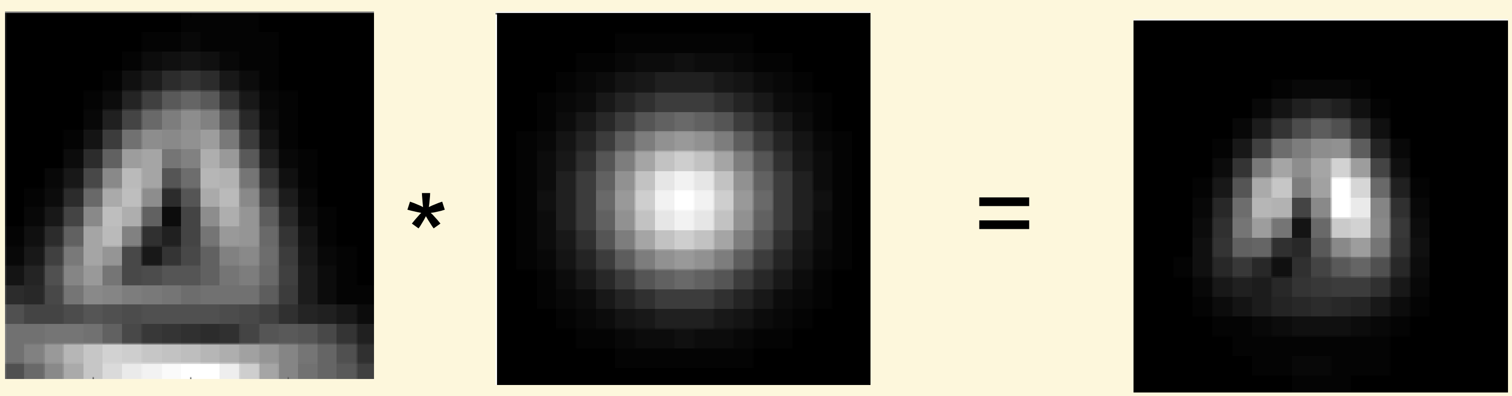


Gradient Magnitude

3. Orientation assignment

- Gradient magnitude weighted by 2D Gaussian

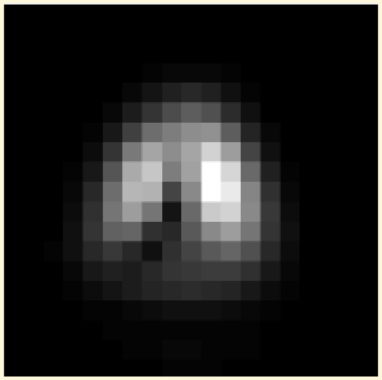
Note: Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint.



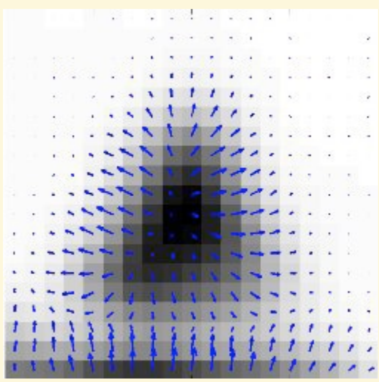
Gradient Magnitude * 2D Gaussian = Weighted Magnitude

3. Orientation assignment

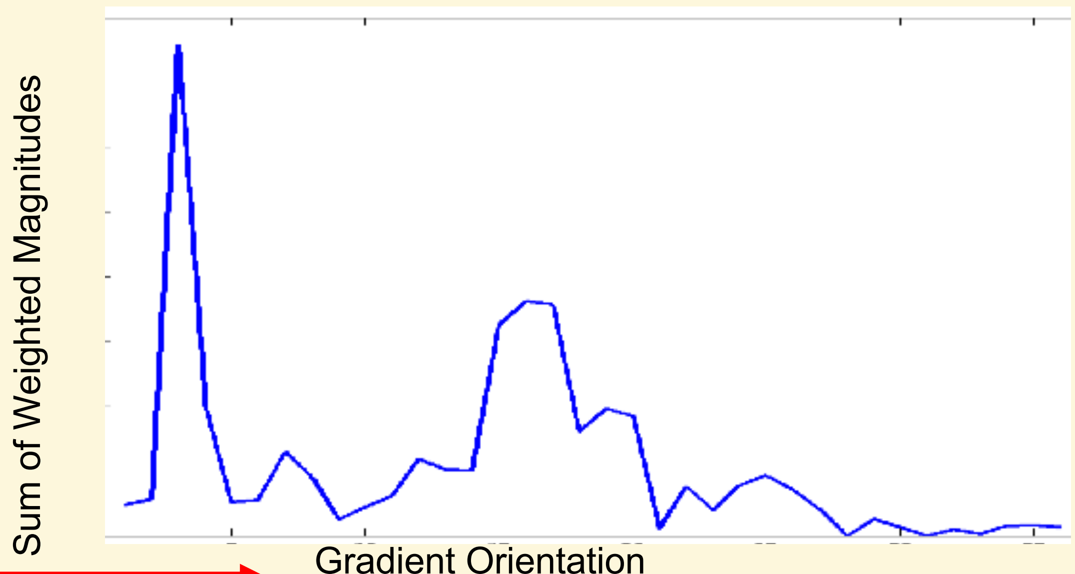
- Accumulate in histogram based on orientation
- Histogram has 36 bins with 10° increments



Weighted Magnitude

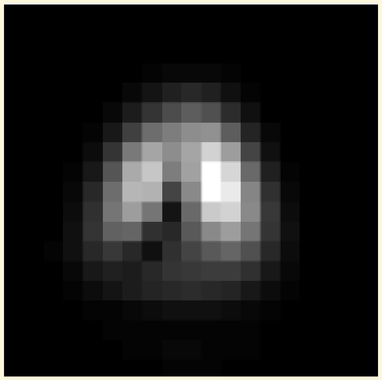


Gradient Orientation

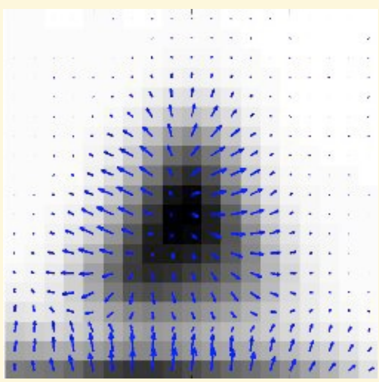


3. Orientation assignment

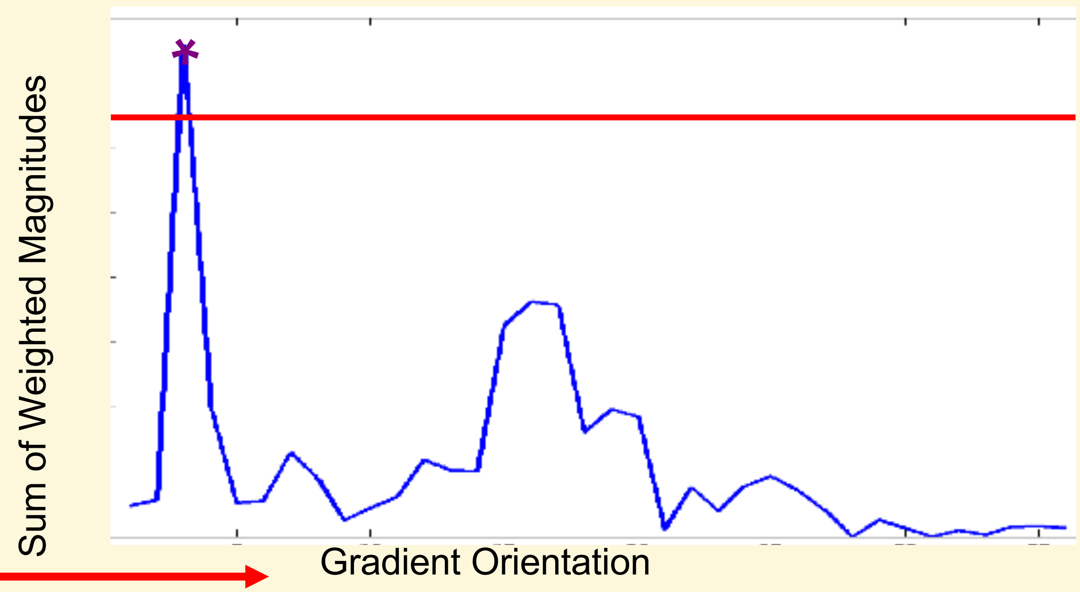
- Identify peak and assign orientation and sum of magnitude to key point



Weighted Magnitude



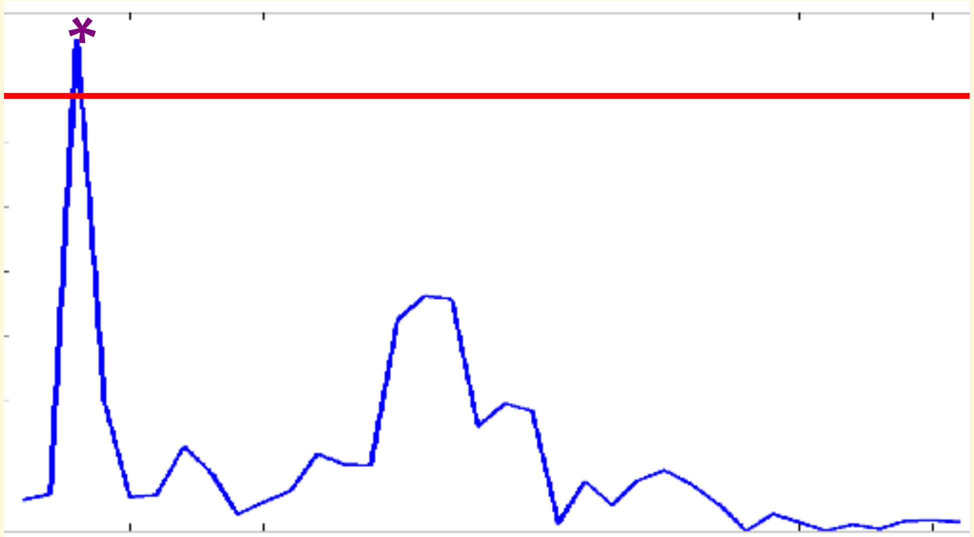
Gradient Orientation



3. Orientation assignment

- The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation.
- Multiple peaks of similar magnitude = multiple keypoints created at the same location and scale but different orientations.

Only about 15% of points are assigned multiple orientations, but these contribute significantly to the stability of matching.



So far we have:

- SIFT keys each assigned:
  - Location
  - Scale (analogous to level it was detected)
  - Orientation (assigned in previous canonical orientation steps)