TDS3651
Visual Information Processing

Local Invariant Features
Lecture 9

Faculty of Computing and Informatics
Multimedia University
prepared by Lai-Kuan, Wong
modified by Yuen Peng, Loh

Some graphics and materials are taken from UTexas at Austin CS376 & CS381V notes, Stanford Uni's CS131 notes and Gonzalez & Wood's DIP textbook

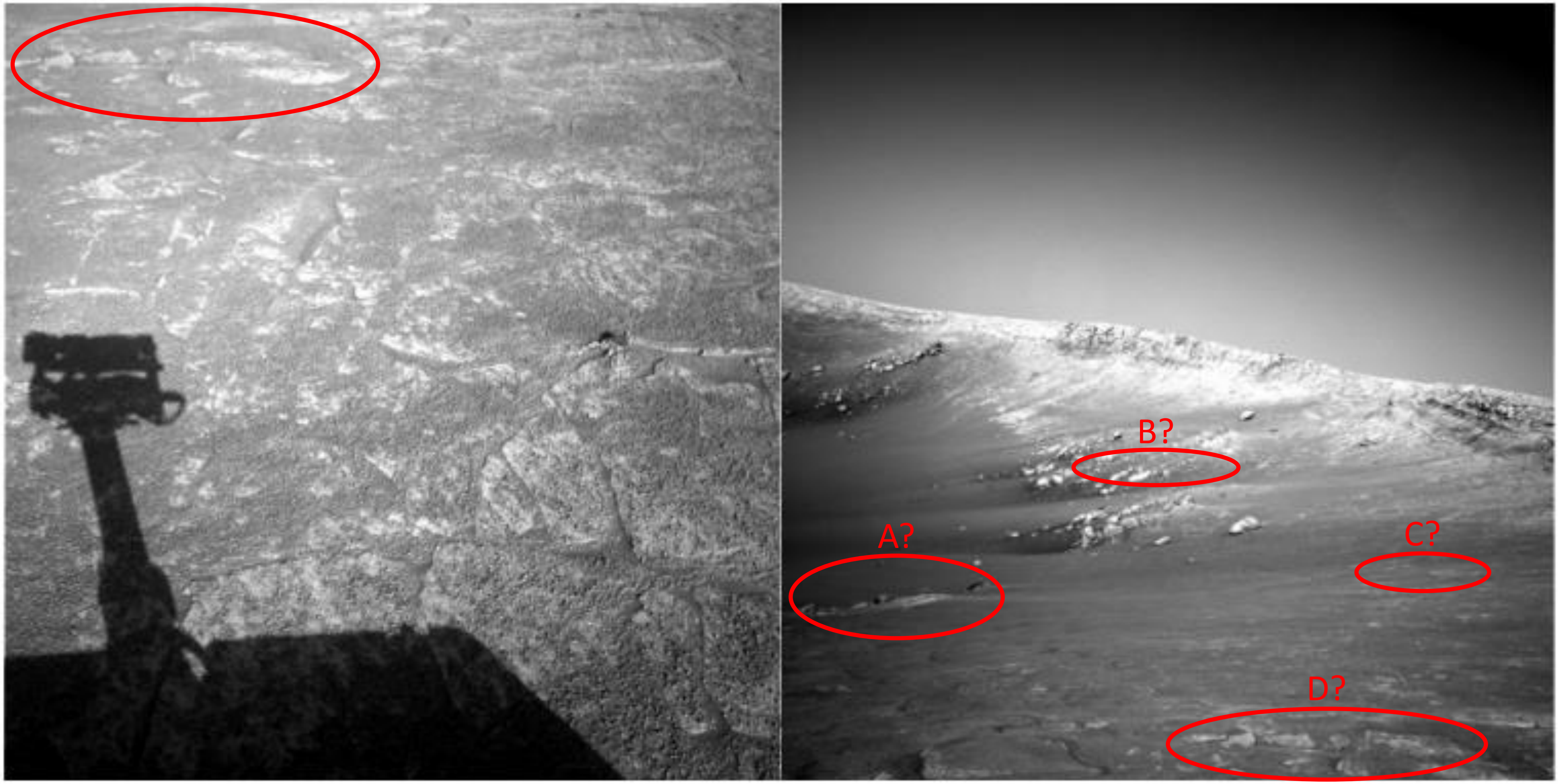
Lecture Outline

- Local invariant features
  - Motivation
  - Requirements, Invariances
- Feature detection – keypoint localization
  - Harris corner detector
  - Scale-space blob detector
- Feature description
  - Scale Invariant Feature Transform (SIFT)
- Applications

Image Matching: Challenging!

Image Matching: Challenging!

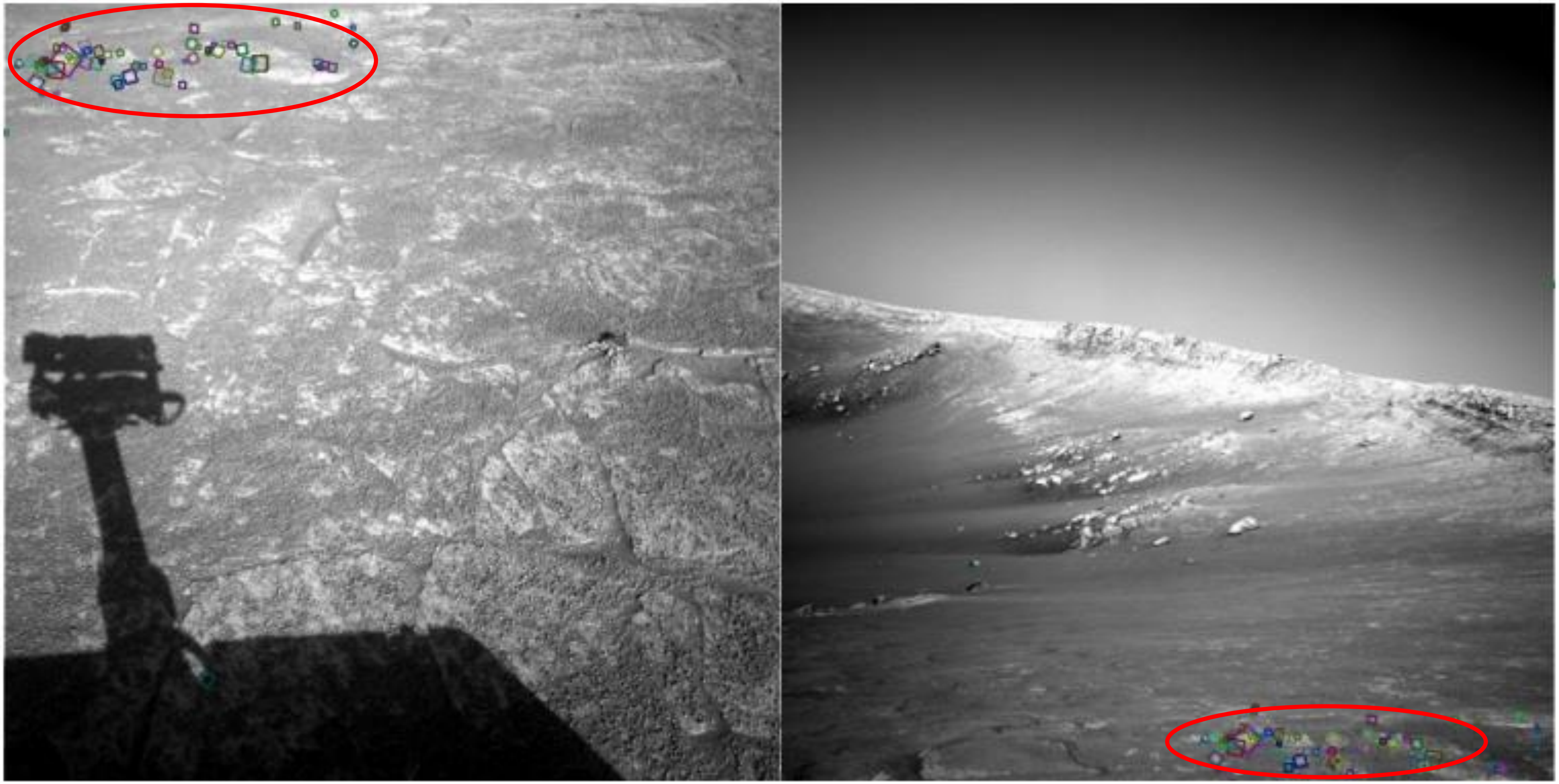
Image Matching: Even harder!



NASA Mars Rover images

Image Matching: Even harder!

Here's the answer

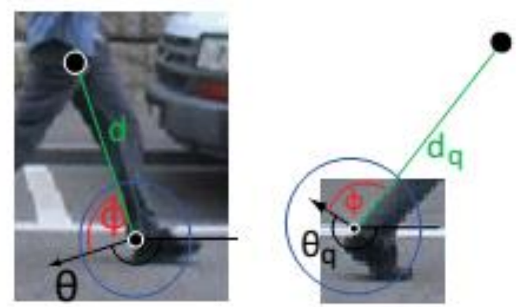


NASA Mars Rover images with SIFT feature matches

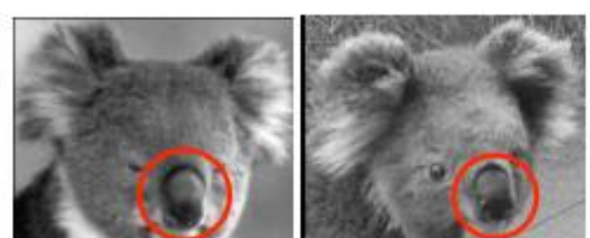
Motivation of using local features

- Global representations have major limitations
- How about…we describe and match only **local regions**
- Increased robustness to
  - Occlusions

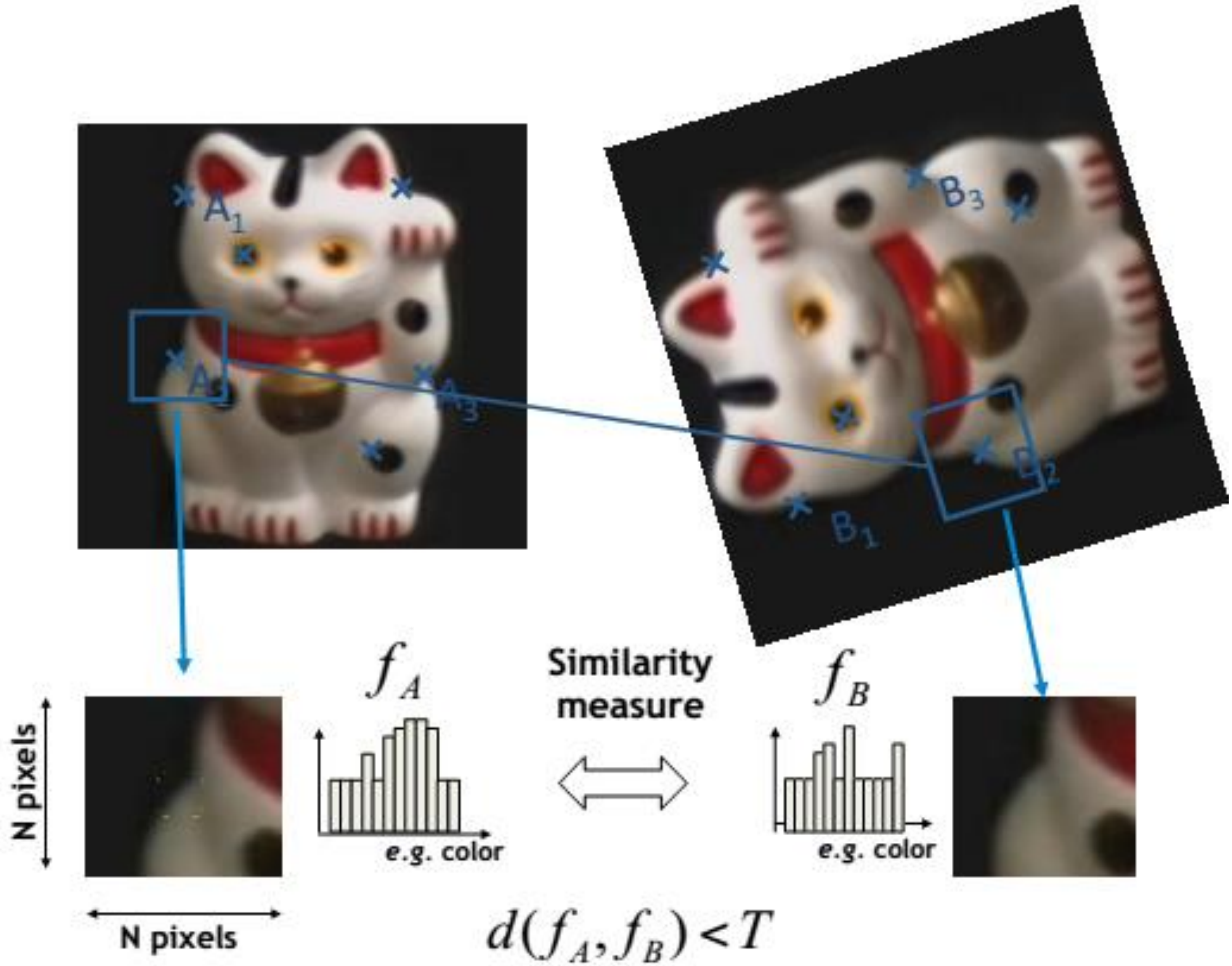
  - Articulation



  - Intra-category variations



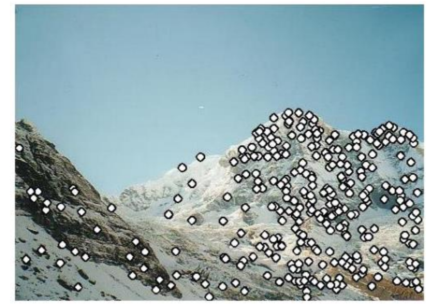
General Approach to Matching



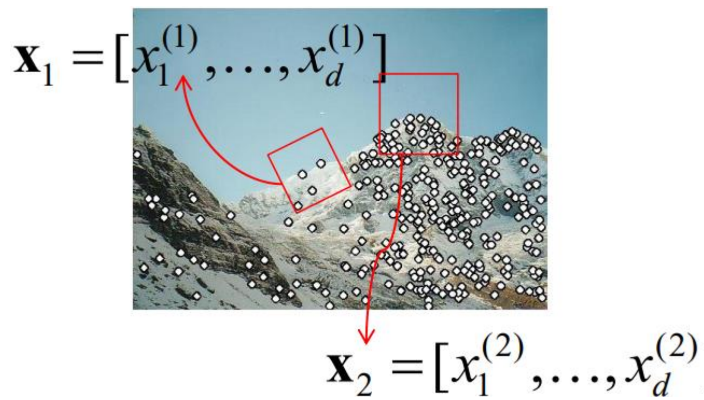
1. Find a set of distinctive key points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute local descriptor from the region
5. Match local descriptors

Local Features: Main Components

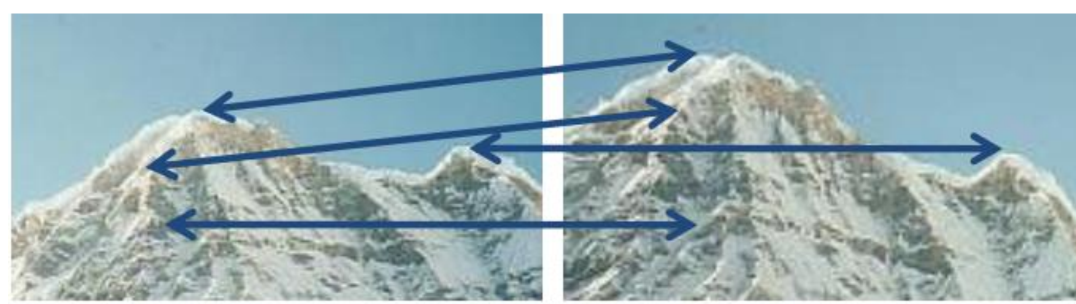
1. **Detection**: Identify the interest points



2. **Description**: Extract vector feature descriptor surrounding each interest point

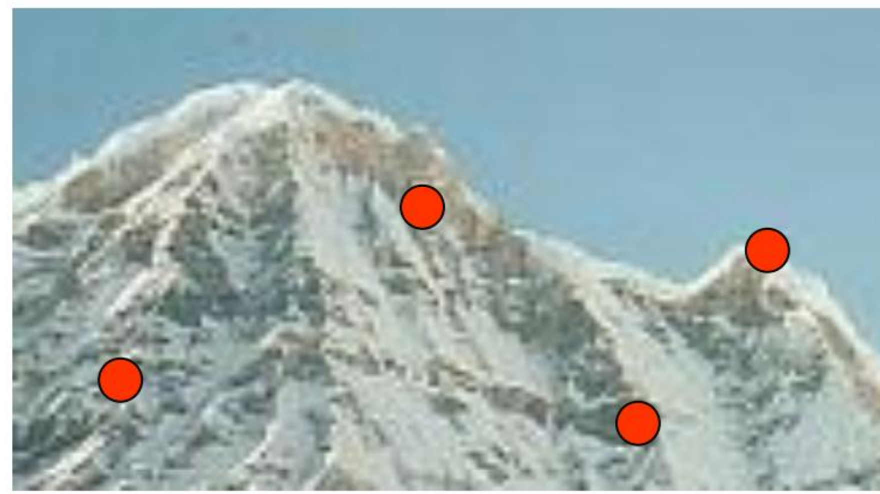


3. **Matching**: Determine correspondence between descriptors in two views



Local Features: Requirements

- Problem #1:
  - Detect (at least some of) the same points independently in both images

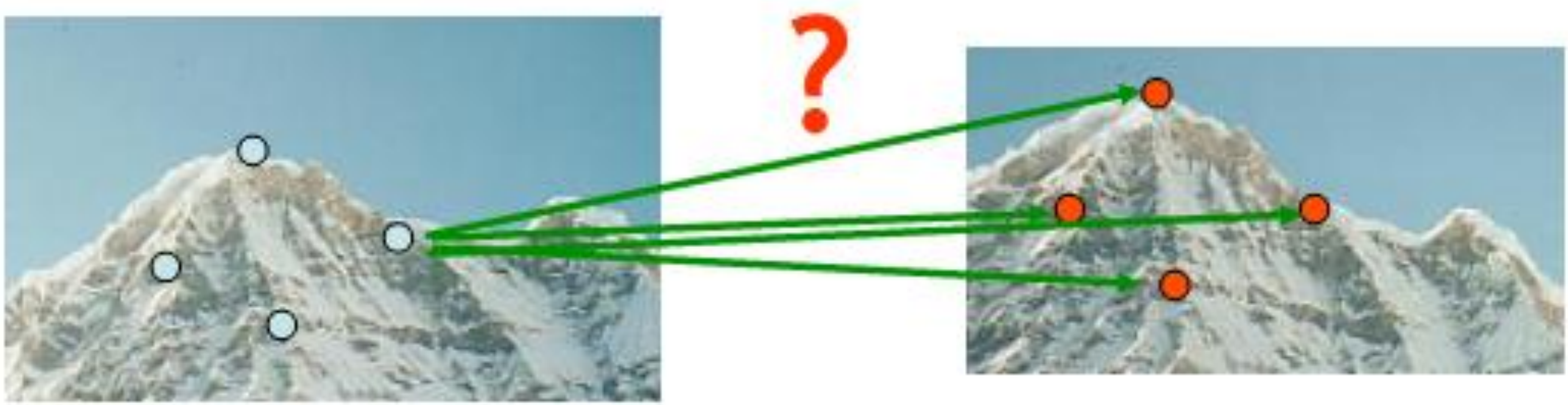



No chance to find true matches

We need a repeatable detector

Local Features: Requirements

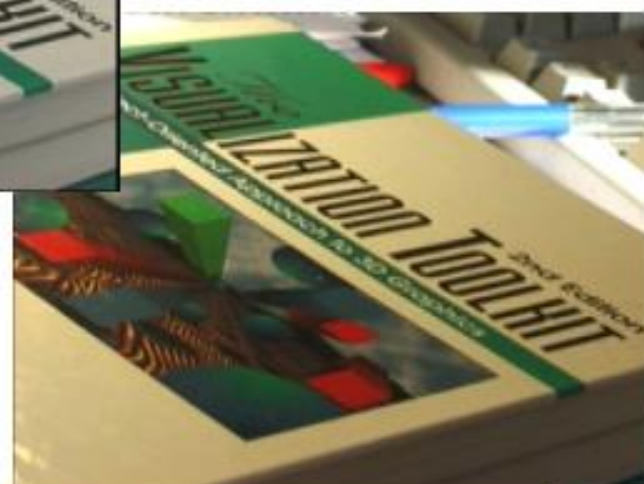
- Problem #1:
  - Detect the same point independently in both images
- Problem #2:
  - For each point correctly determine which point goes with which corresponding one



We need a reliable and distinctive descriptor

Invariance: Geometric Transformations

Invariance: Photometric Transformations



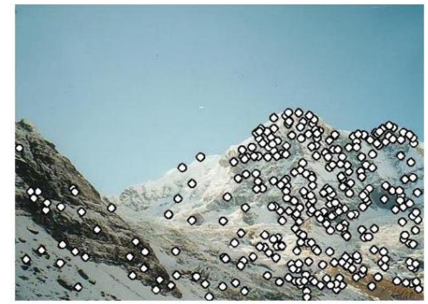
- Often modeled as a linear transformation:
  - Scaling + Offset

Local Features: Desired Properties

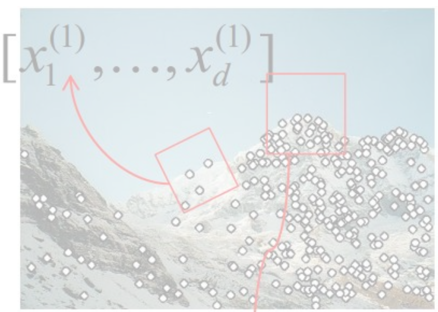
- Repeatability (Invariance)
  - The same feature can be found in images despite geometric and photometric transformations
- Distinctiveness (Saliency)
  - Each feature has a distinctive description, or with “interesting” structure
- Compactness and efficiency
  - Much fewer features than image pixels, but sufficient to cover
  - Fast
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Local Features: Main Components

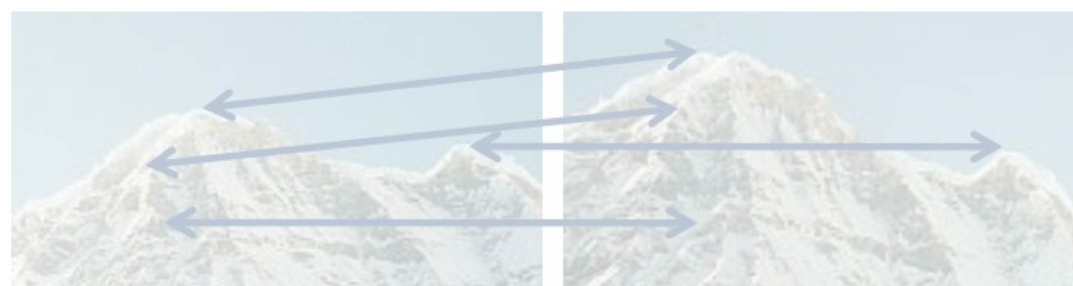
1. **Detection**: Identify the interest points



2. Description: Extract vector feature descriptor surrounding each interest point



3. Matching: Determine correspondence between descriptors in two views



Interest Point Detection

Detection: The first task

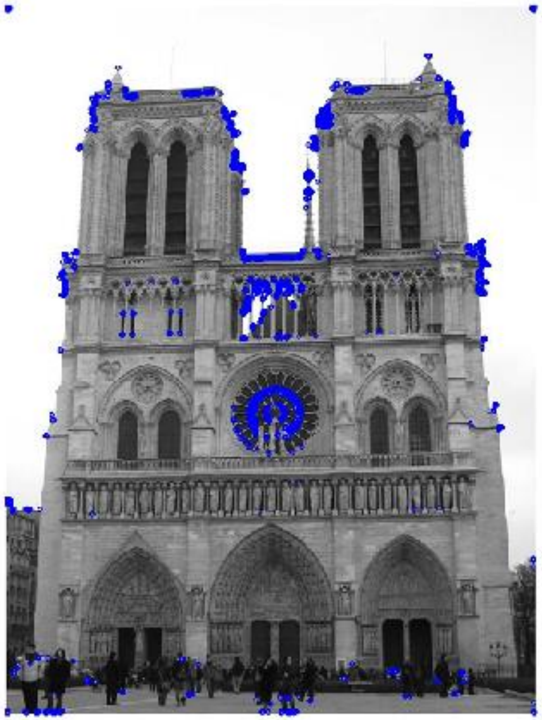
- Many existing detectors available
  - Hessian & Harris — [Beaudet '78], [Harris '88]
  - Laplacian, DoG — [Lindeberg '98], [Lowe '99]
  - Harris-/Hessian-Laplace — [Mikolajczyk & Schmid '01]
  - Harris-/Hessian-Affine — [Mikolajczyk & Schmid '04]
  - EBR and IBR — [Tuytelaars & Van Gool '04]
  - MSER — [Matas '02]
  - Salient Regions — [Kadir & Brady '01]
  - Others …

✓ These detectors have become a basic building block for many applications in CV

Which points would you choose?

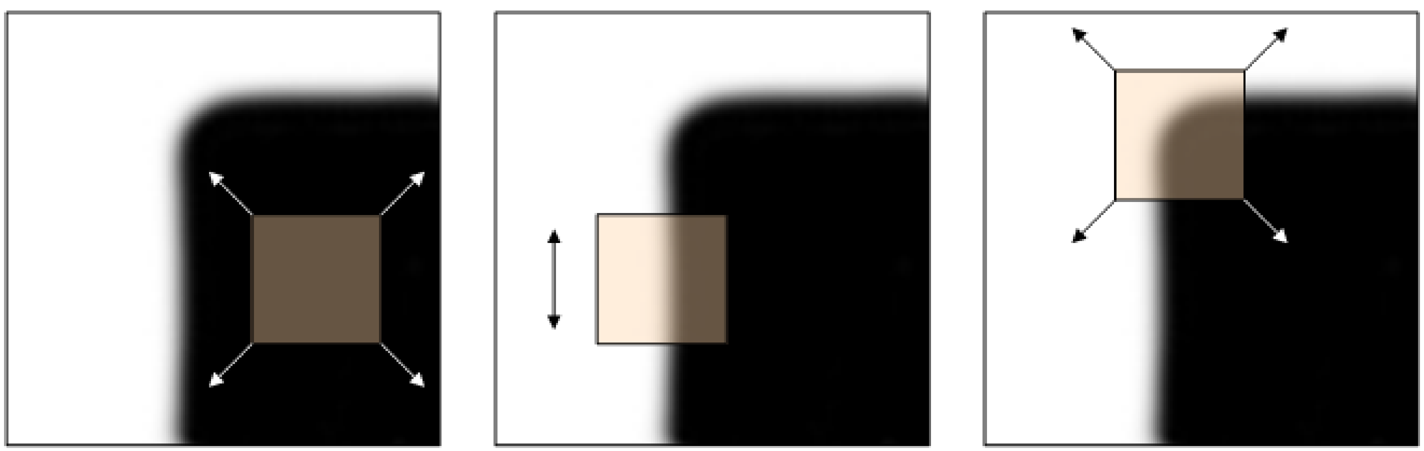
Finding Corners

- Key property:
  - In a region around a corner, image gradient has two or more dominant directions
  - Corners are repeatable and distinctive



Corners as Distinctive Interest Points

- **Design Criteria**
  - Easy to recognize the point by looking through a small window (locality)
  - Shifting the window in any direction should give a large change in intensity (good localization)



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Corners as Distinctive Interest Points

- 2x2 matrix of image derivatives (averaged in neighbourhood of a point)

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Gradient with respect to x, times gradient with respect to y

Sum over image region – the area we are checking for corner

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

Derivation of matric M: http://aishack.in/tutorials/harris-corner-detector/

Corners as Distinctive Interest Points

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

- 2x2 matrix of image derivatives (averaged in neighbourhood of a point)

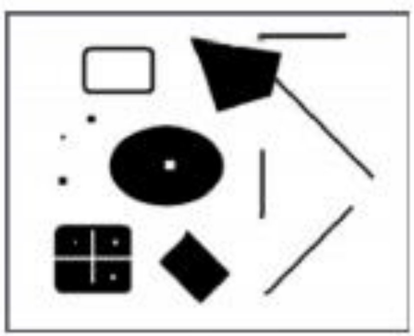


Image I

I_x

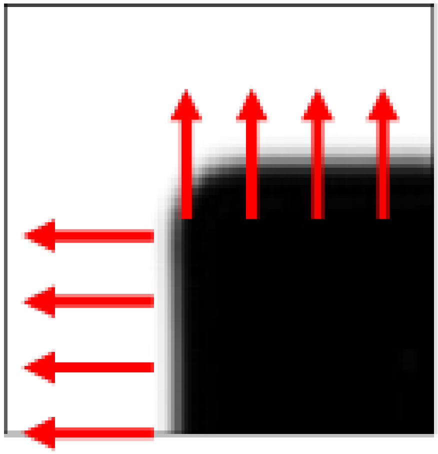
I_y

$I_x I_y$

What does this matrix reveal?

$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- First, consider an axis-aligned corner:



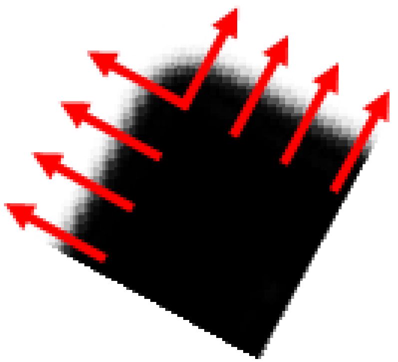
Look for locations where **both** λ's are large

If either λ is close to 0, then this is **not** corner-like

What does this matrix reveal?

- What about a corner that is not aligned with the image axes?
- Since *M* is symmetric, we have

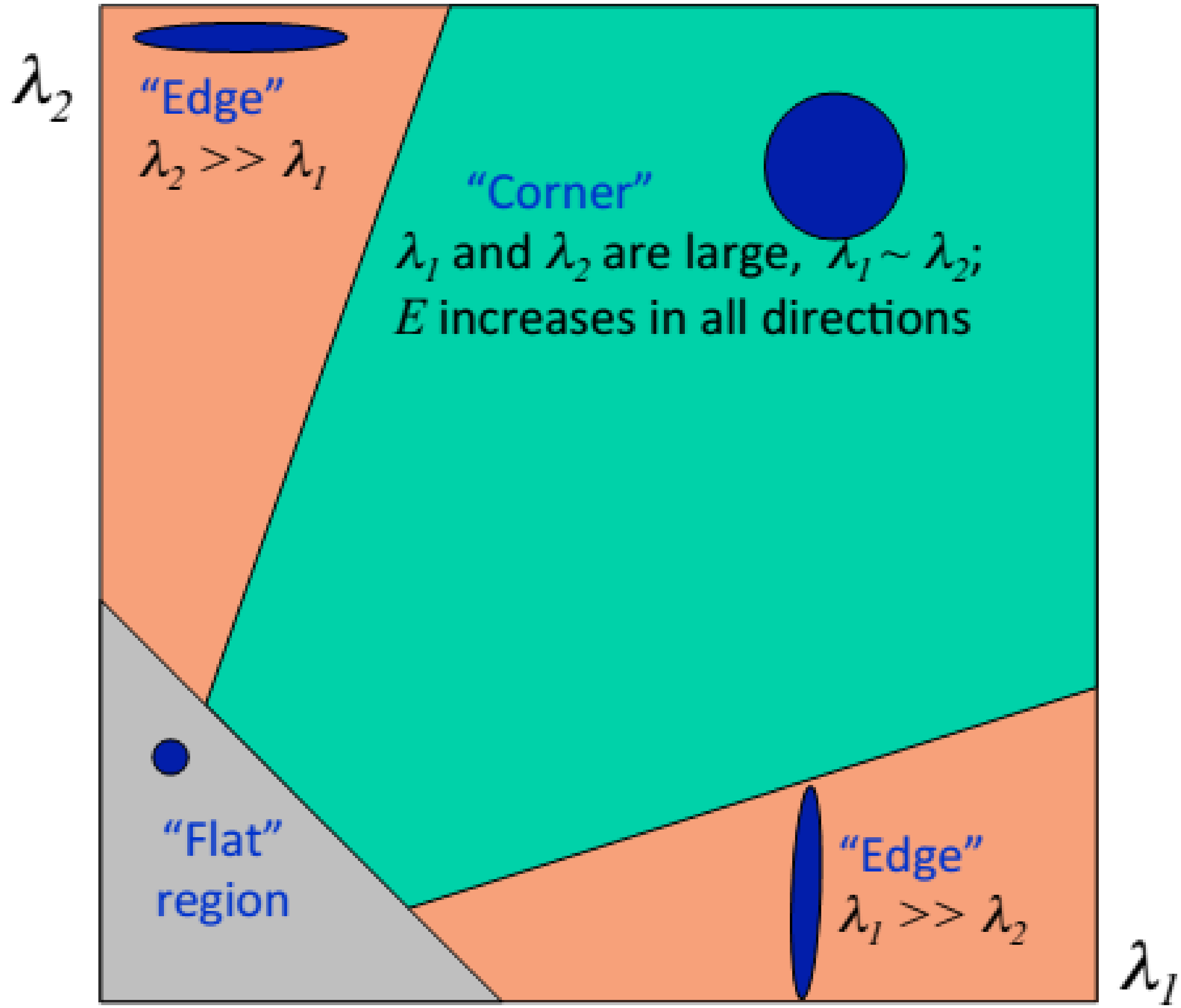
$$M = R^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}R$$



"Eigenvalues" of the matrix $(Mr_i = \lambda_i r_i)$

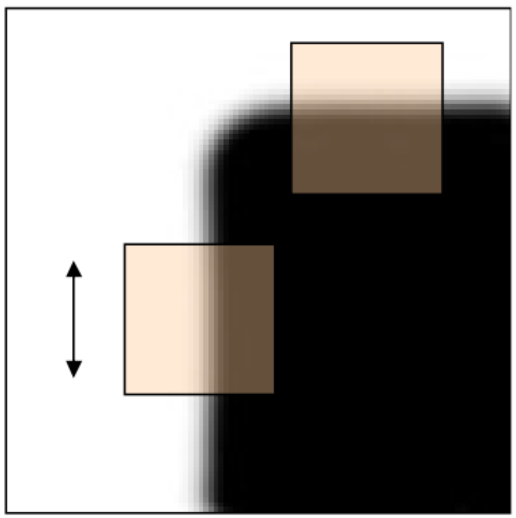
Eigenvalues of *M* reveal amount of intensity change in the two principal orthogonal gradient directions in the window

Interpreting the eigenvalues



Harris "corner-ness" score $\theta = \det(M) - \alpha \operatorname{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$

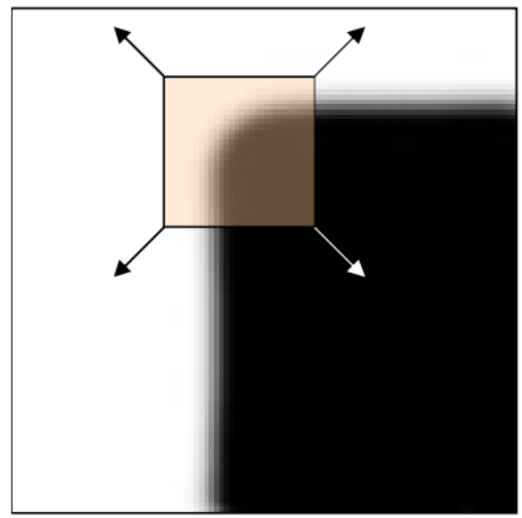
Corner response function



"edge":

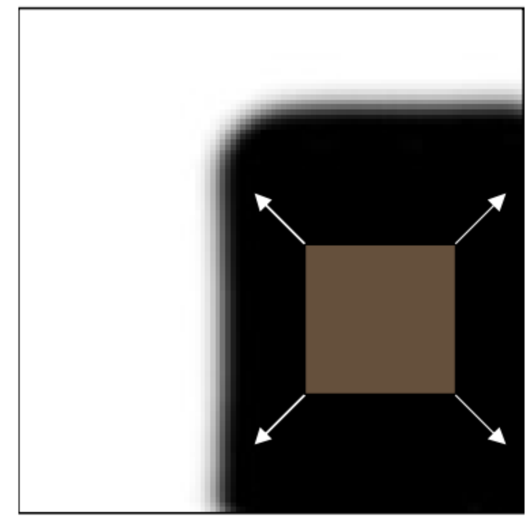
$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$



"corner":

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



"flat" region

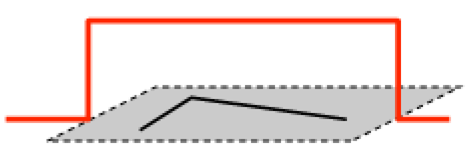
λ_1 and λ_2 are small;

Window function $w(x,y)$

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

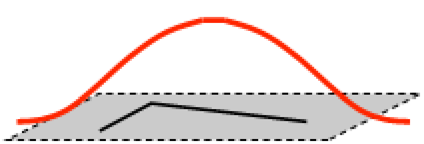
- To enable some form of rotation invariance, use Gaussian function to perform weighted sum

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



1 in window, 0 outside

$$M = g(\sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Gaussian

Harris corner detector

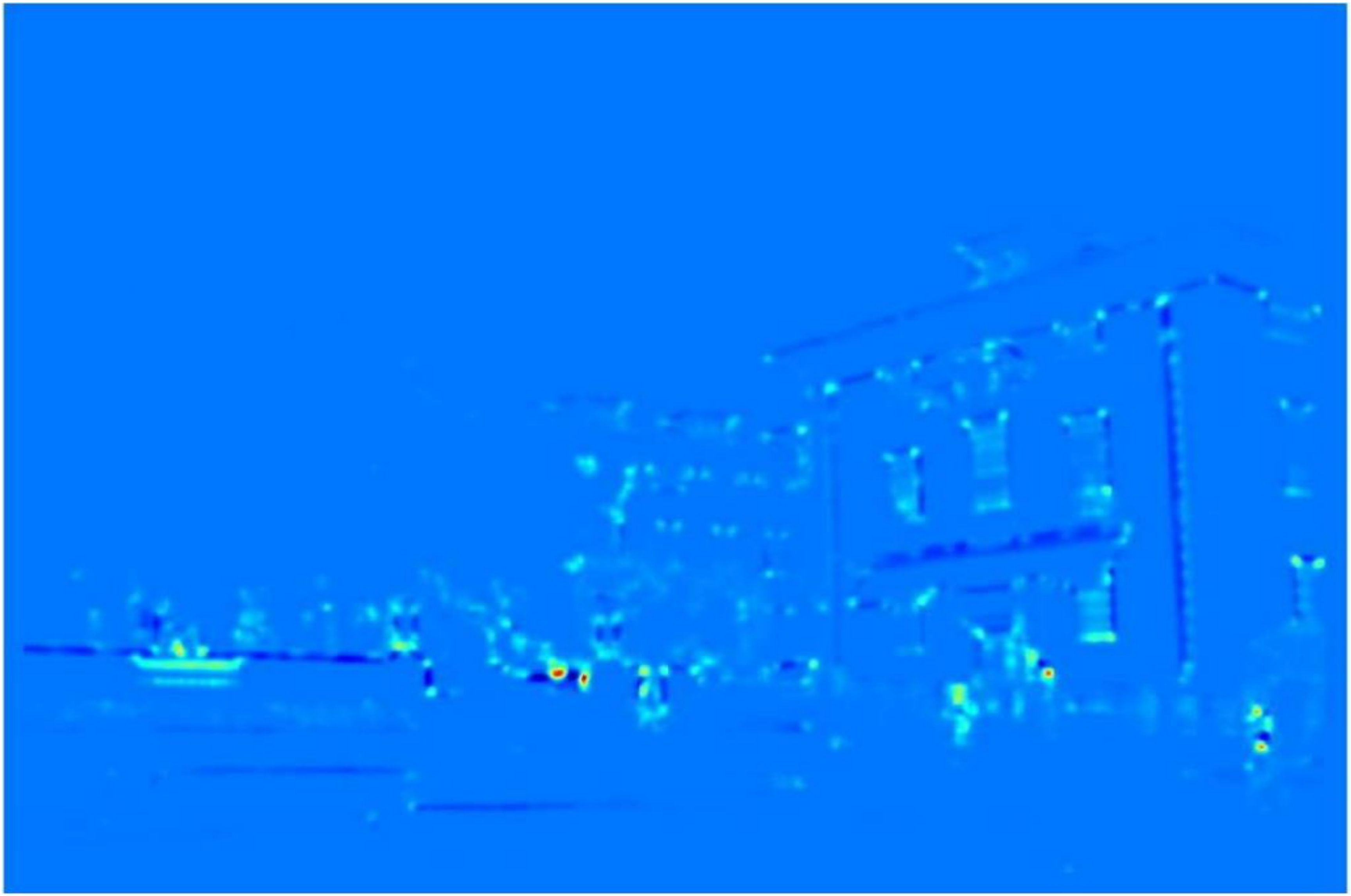
1. Compute M matrix for each image window to get their "*corner-ness*" score, or corner response θ
2. Find points whose surrounding window gave large corner responses ($\theta >$ threshold)
3. Take the points of local maxima, i.e. perform non-maximum suppression

Example

Original image

Example

Compute corner response at every pixel



Example

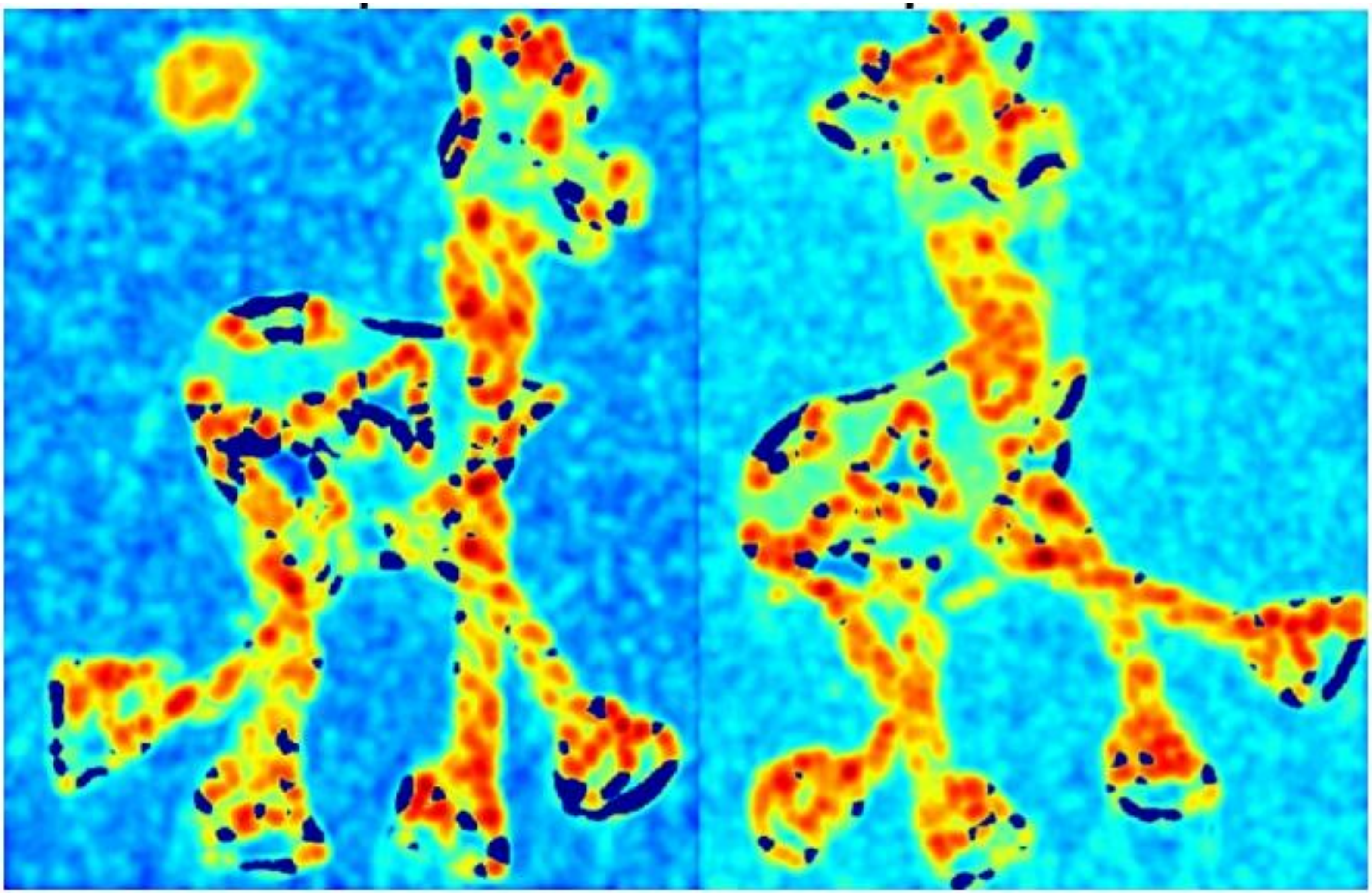
Detected corners marked in blue

Another Example

Original images

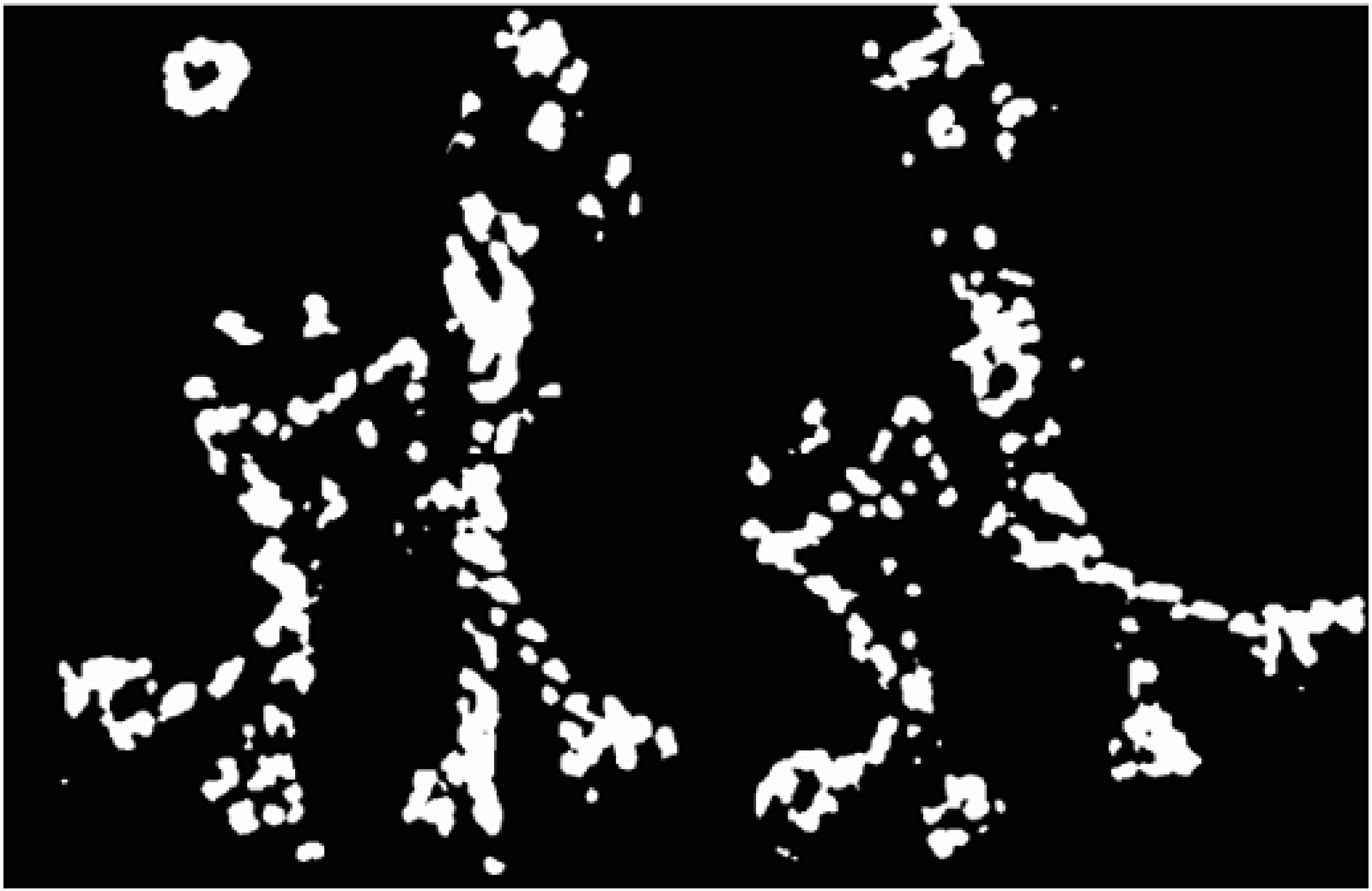
Another Example

Compute corner responses θ



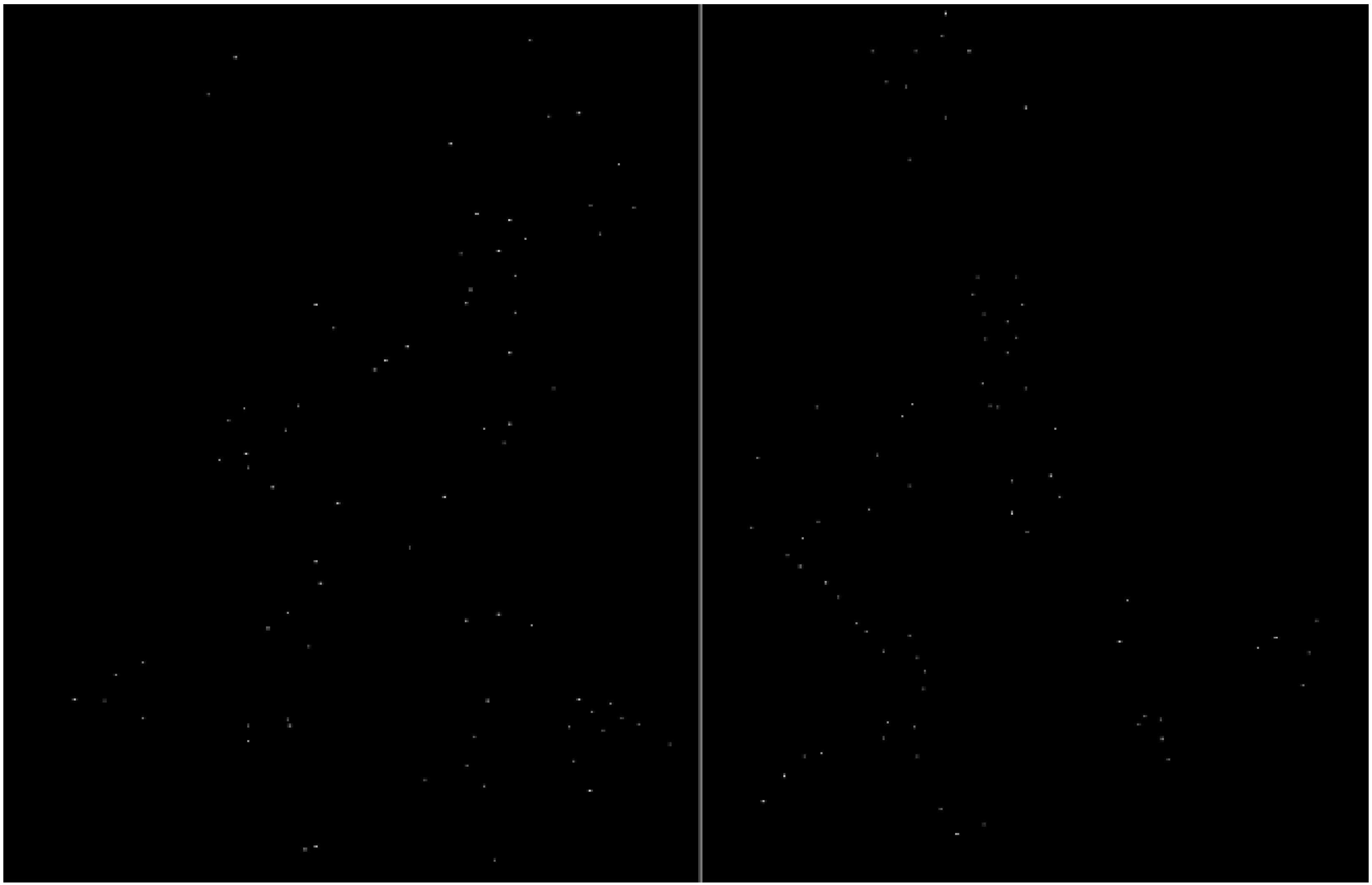
Another Example

Find points with large corner responses: θ > threshold



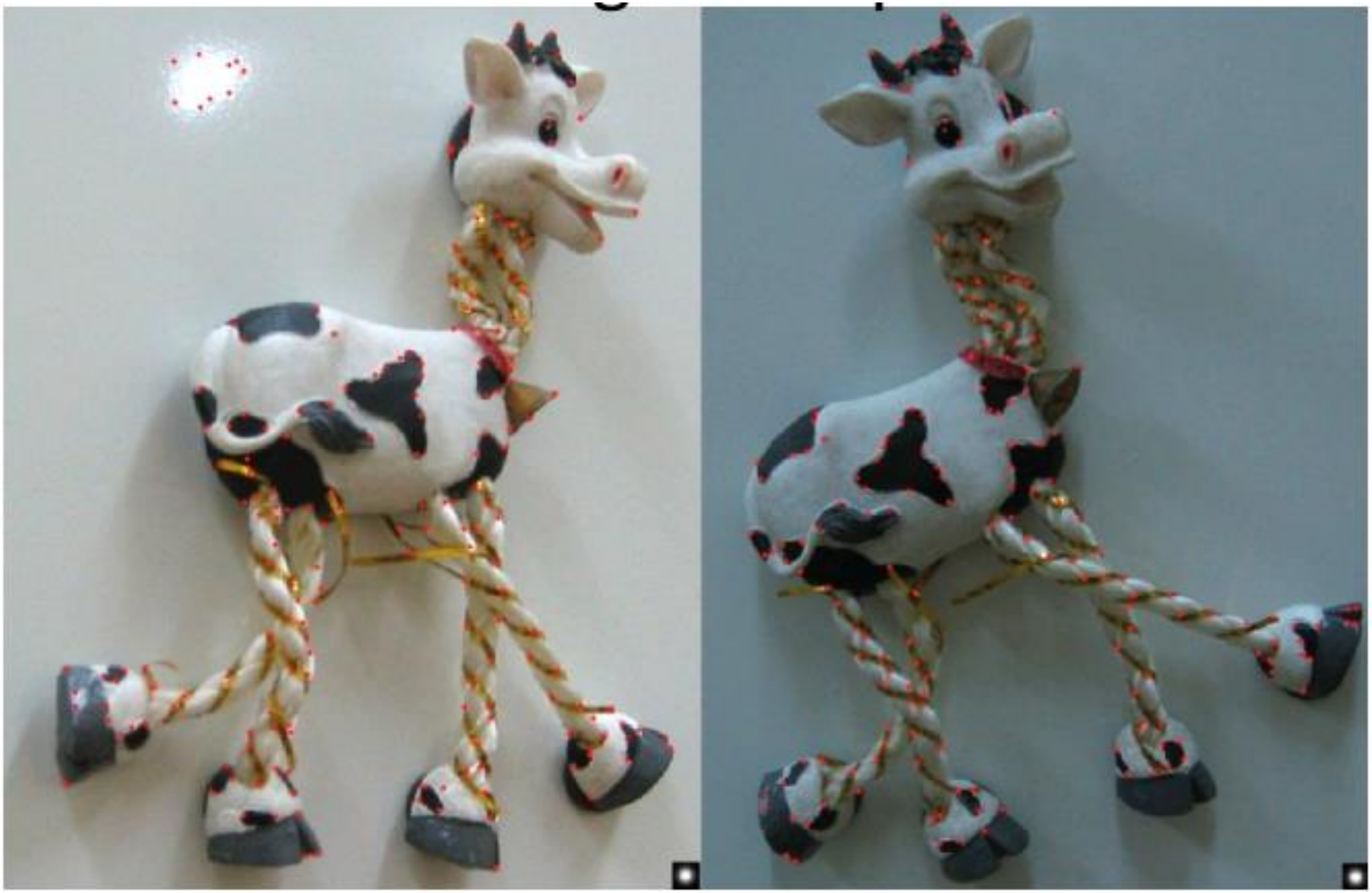
Another Example

Take only the points of local maxima of θ

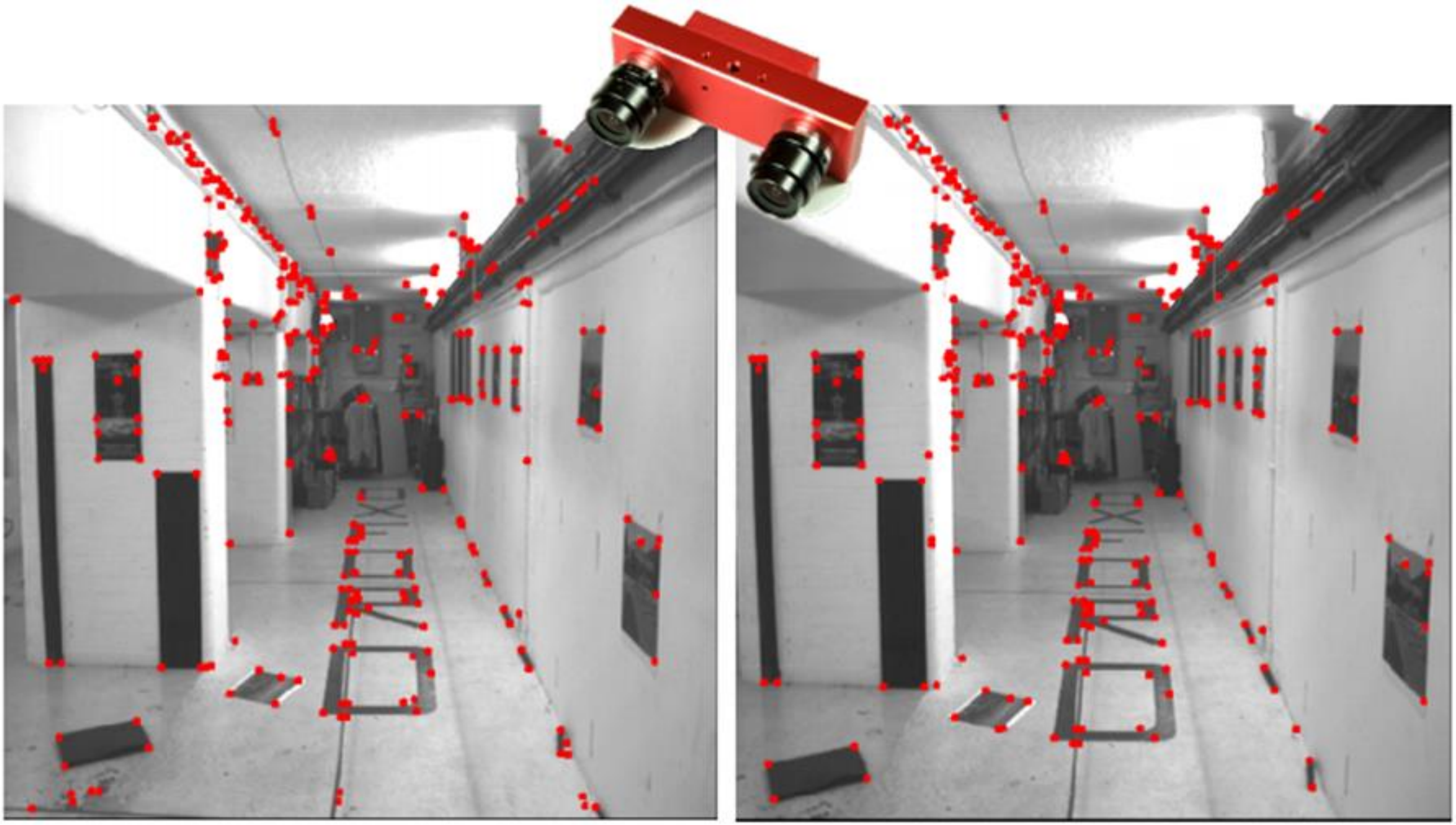


Another Example

Detected corners marked in red



Another Example



These corners are well suited for finding stereo correspondences

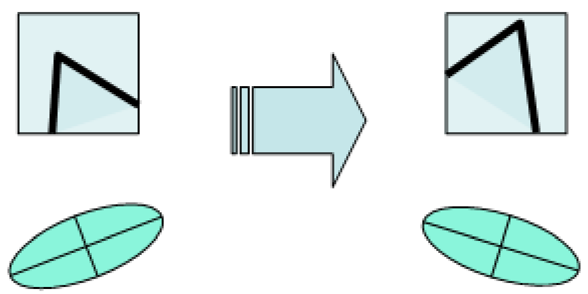
Properties of Harris corner detector

- Translation invariance? **YES**

  If image shifted, still can locate corners

- Rotation invariant? **YES**

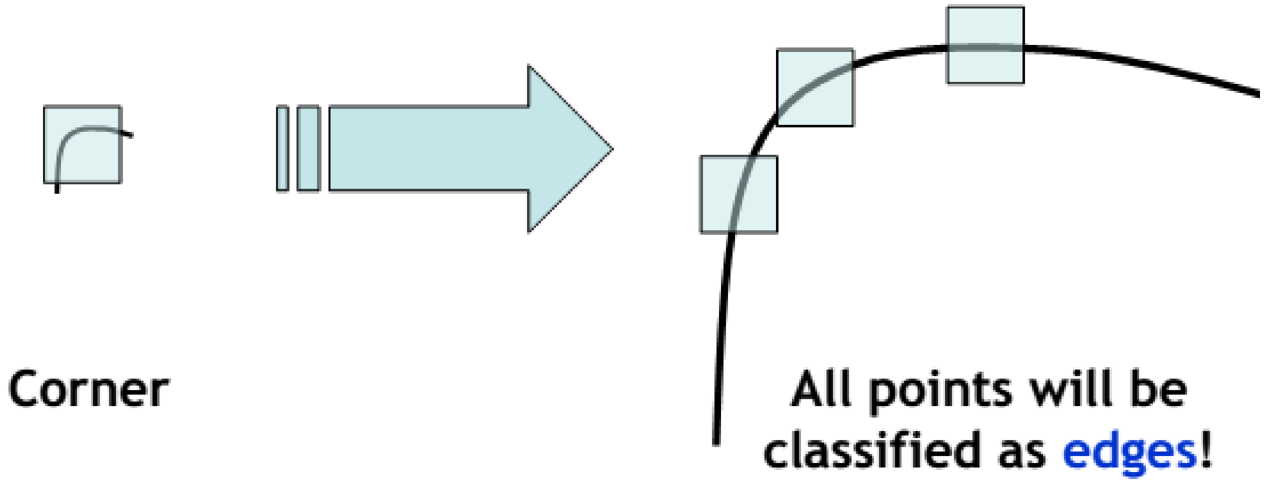
$$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Properties of Harris corner detector

- Translation invariance? **YES**
- Rotation invariant? **YES**
- Scale invariant? **NO**

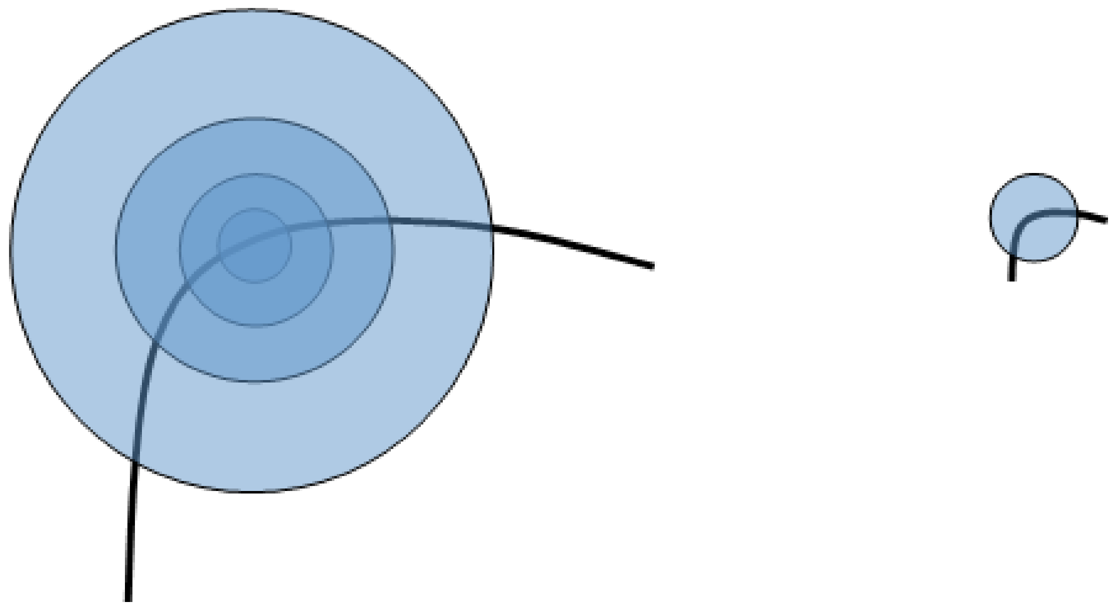


Scale invariant interest points

- How can we independently select interest points in each image, such that the detections are repeatable across different scales?

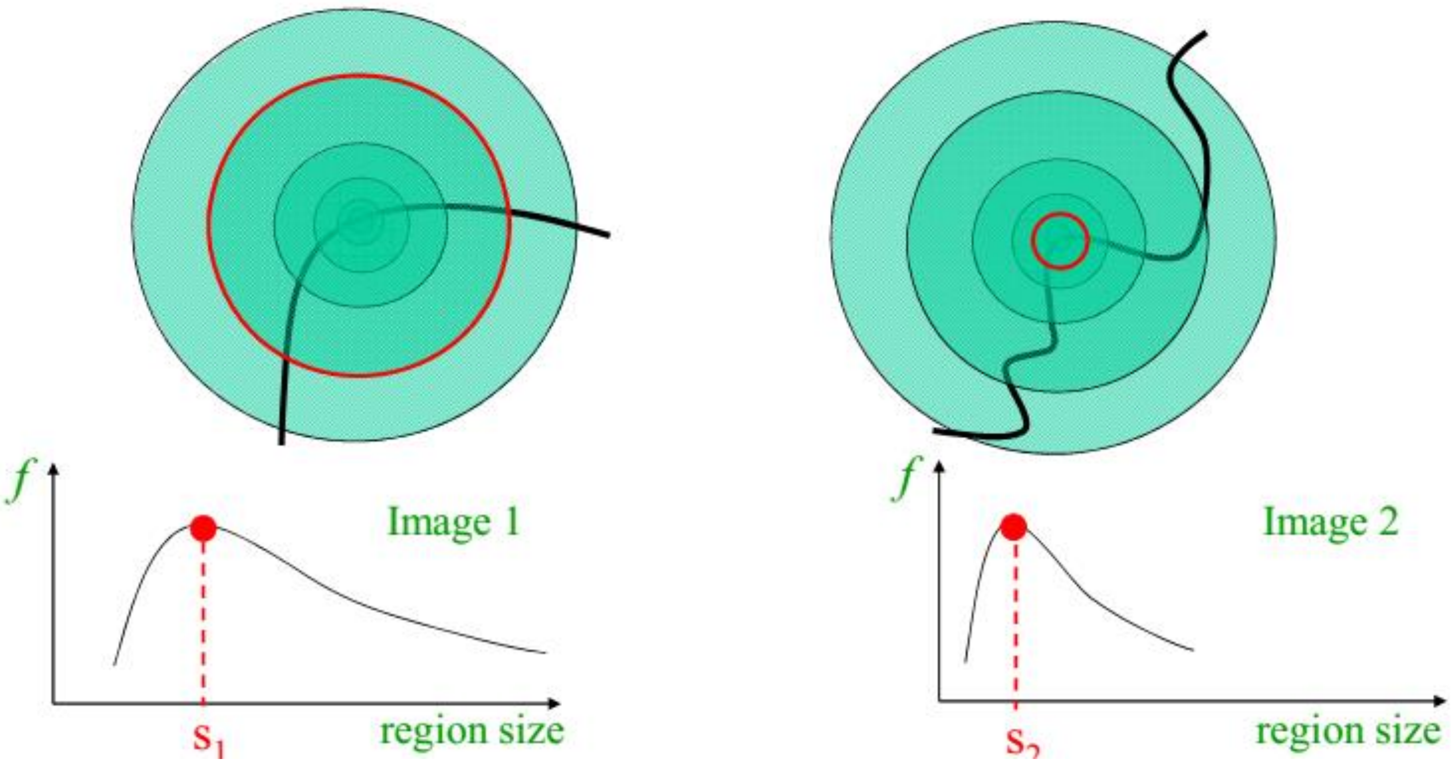
Scale Invariant Detection

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



Automatic Scale Selection

- **Intuition**
  - Find scale that gives local maxima of some function *f* in both position and scale



  - What can be the “signature” function to do this?

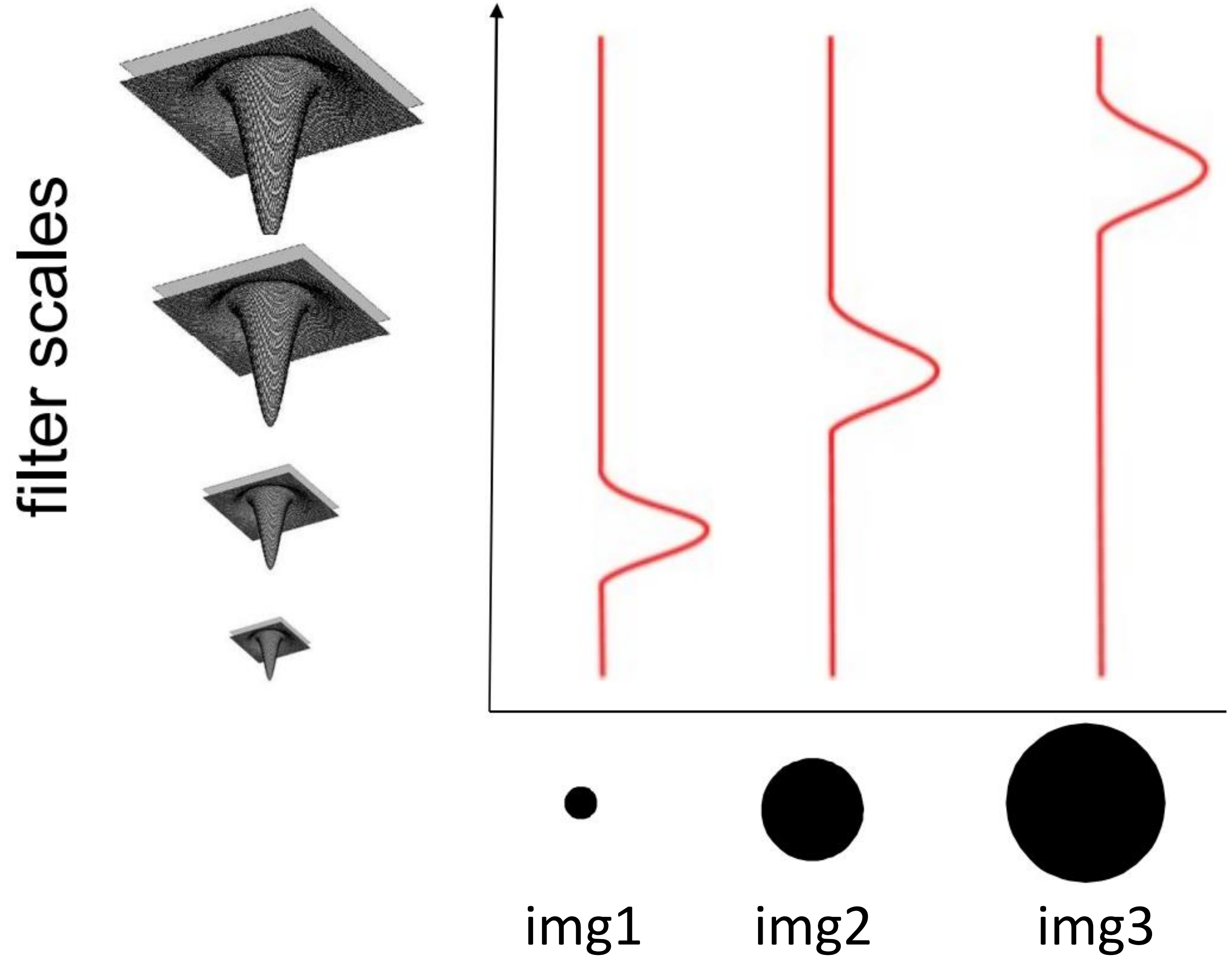
What is a good function?

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

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

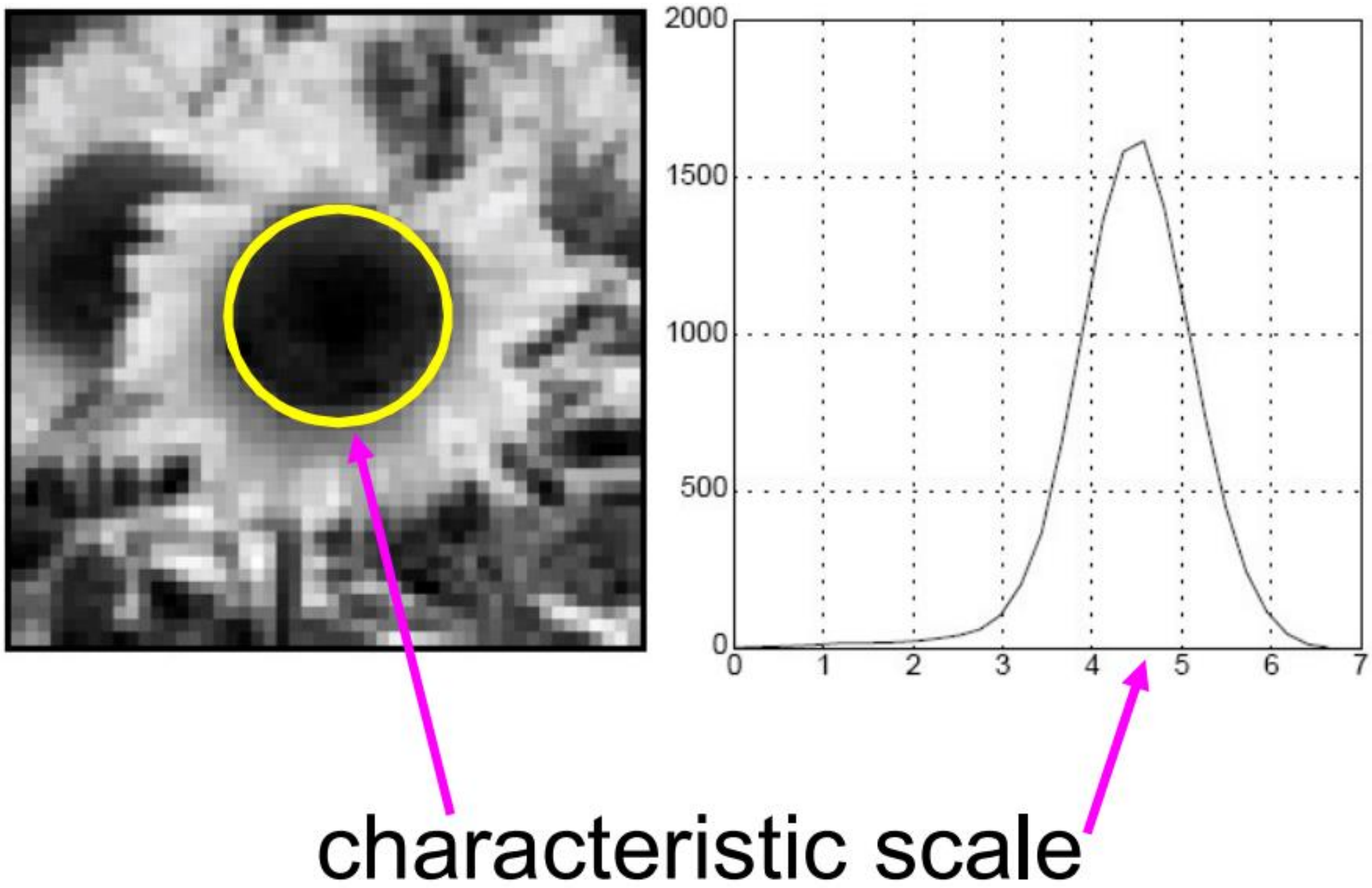
Blob detection in 2D

- **Laplacian-of-Gaussian** = "blob" detector



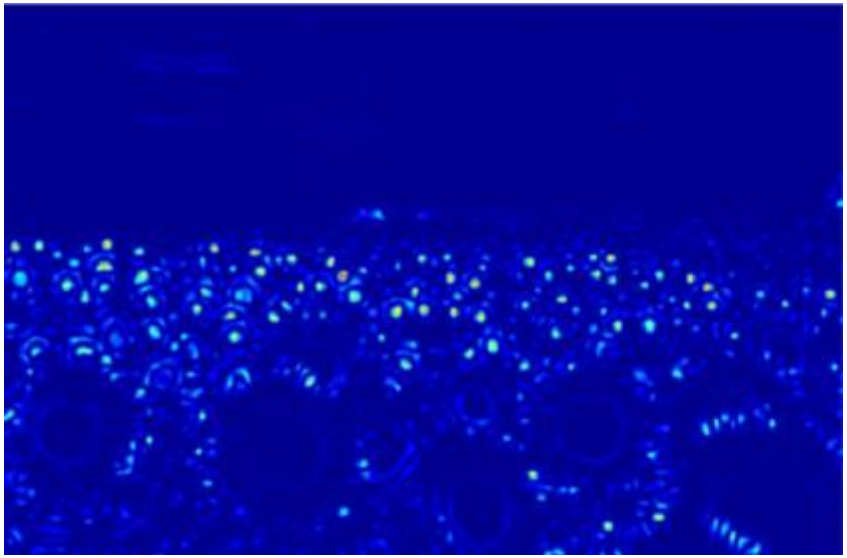
Blob detection in 2D

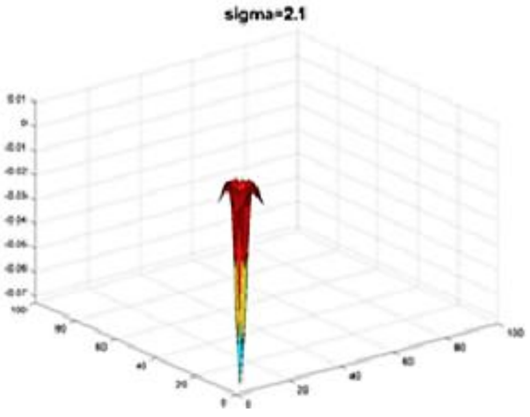
- We define the *characteristic scale* as the scale that produces the peak of Laplacian response



Example

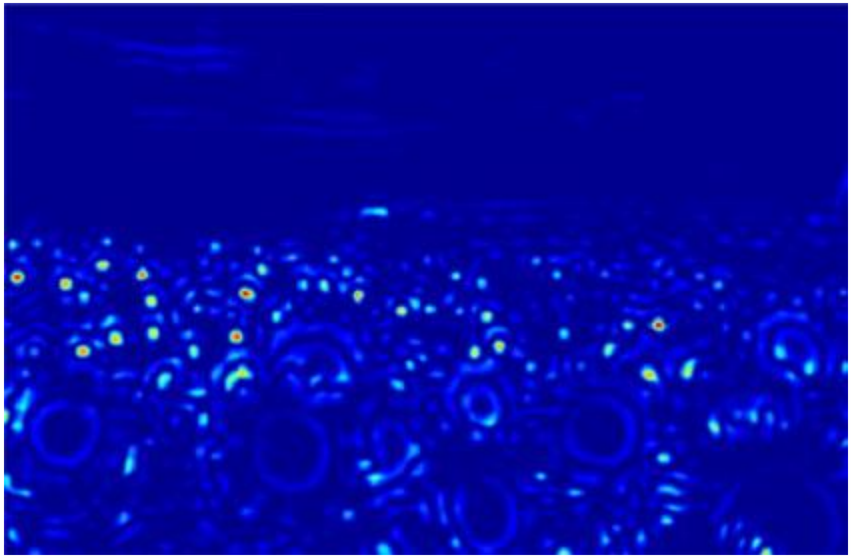
- Image and Laplacian response at scale σ=2.1

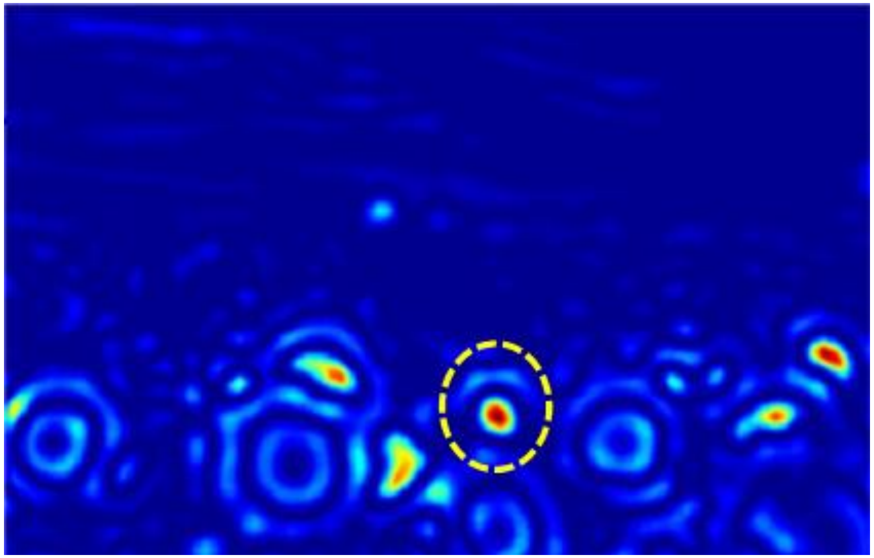


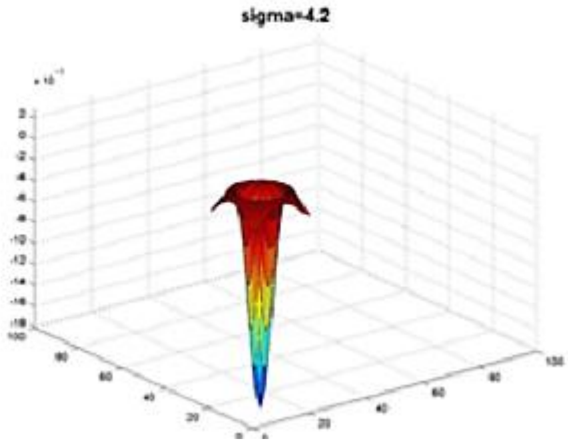


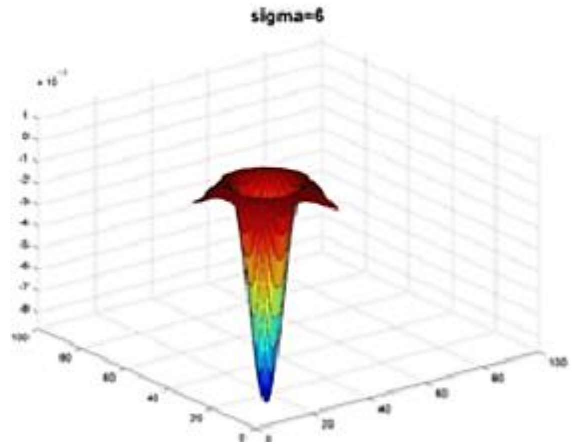
Example

- Laplacian response at scales σ=4.2 and σ=6



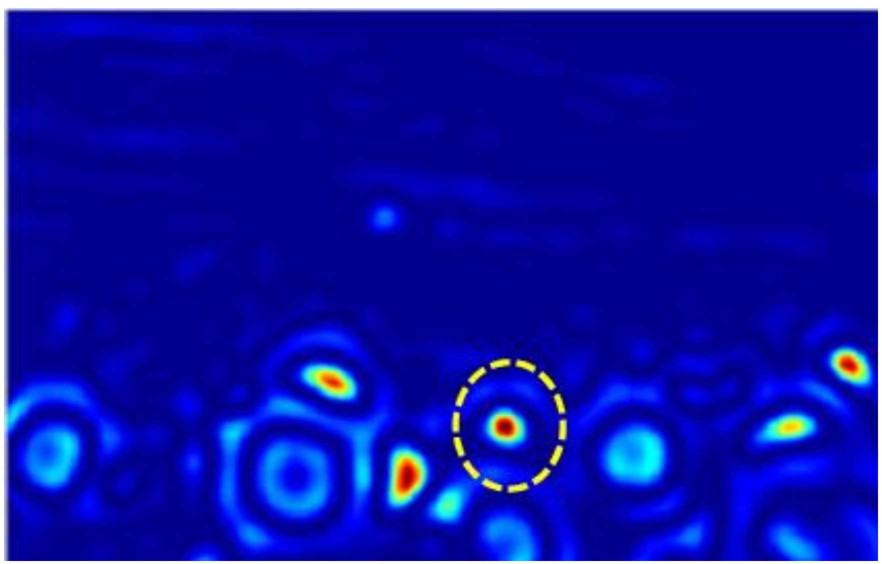


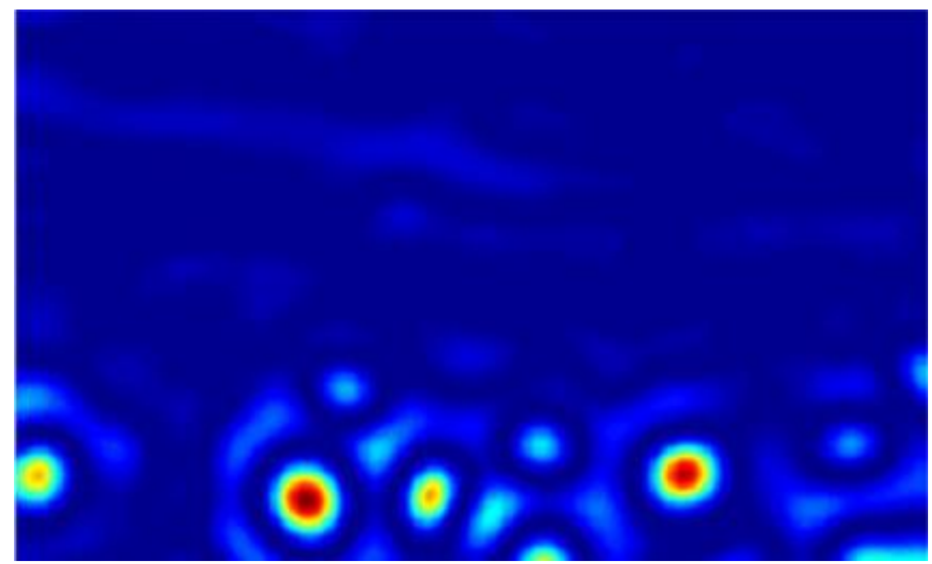


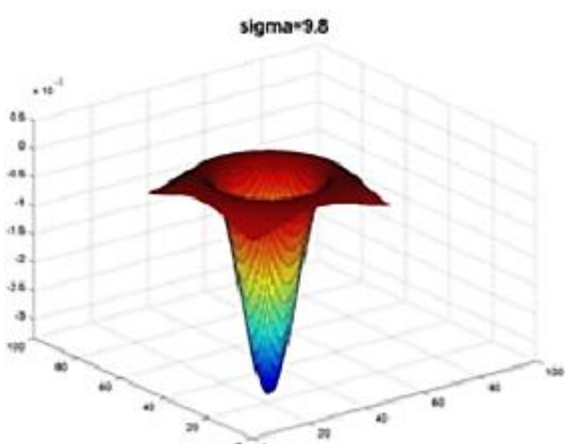


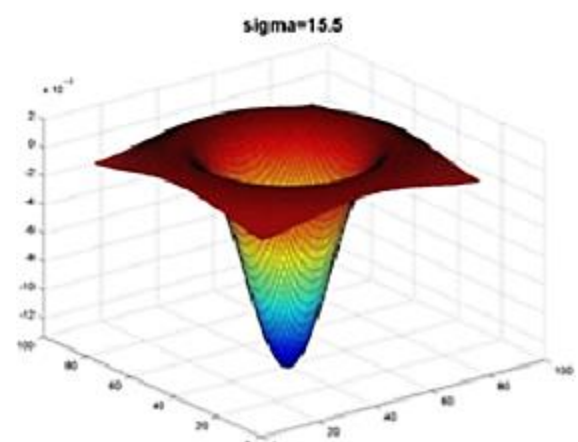
Example

- Laplacian response at scales σ=9.8 and σ=15.5



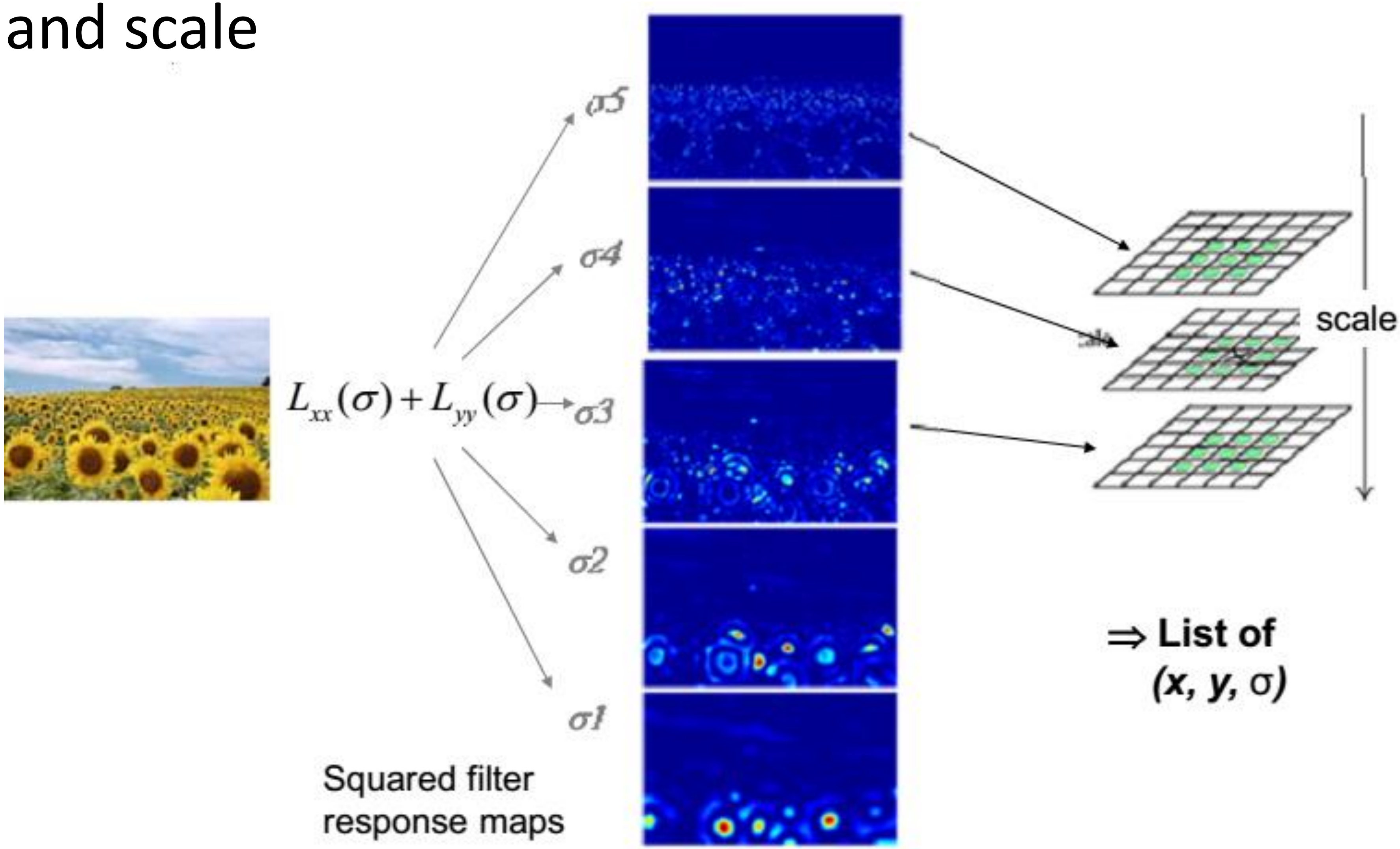






Scale invariant interest points

- Interest points are local maximas in **both** position and scale



Scale-space blob detector: Example

DoG – More efficient

- We can approximate the Laplacian with a **Difference of Gaussians (DoG)** → More efficient to implement

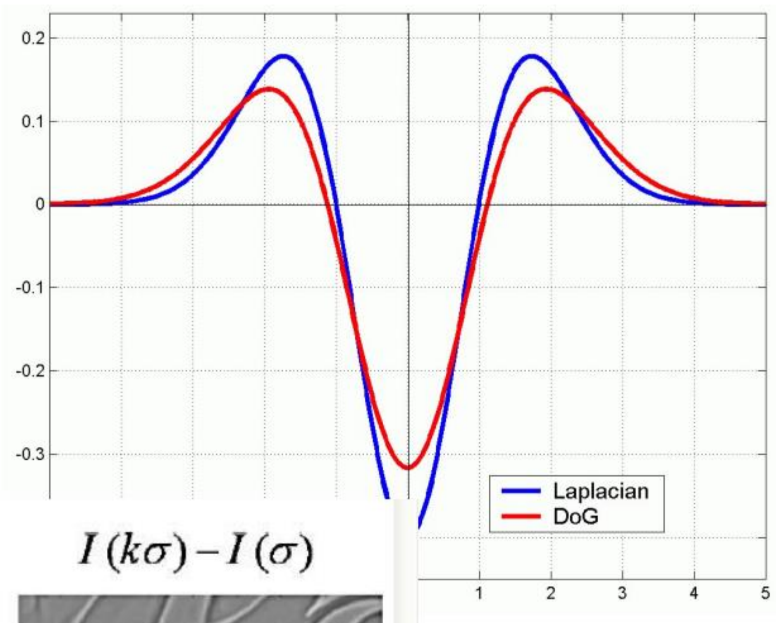
$$f = \text{Kernel} * \text{Image}$$

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

(Laplacian)

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

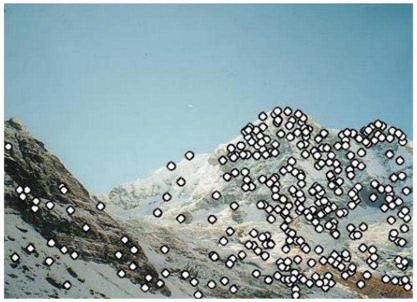
(Difference of Gaussians)



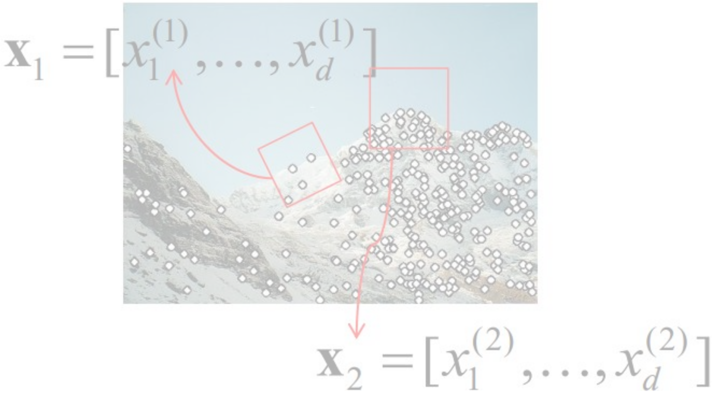
$I(k\sigma)$ - $I(\sigma)$ = $I(k\sigma) - I(\sigma)$

Local Features: Main Components

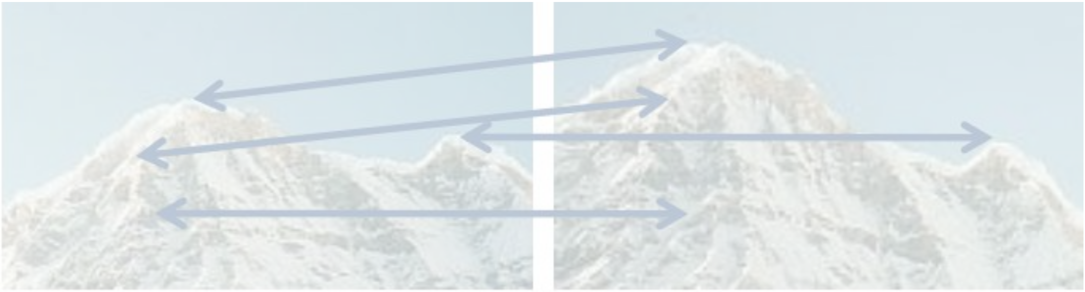
1. **Detection**: Identify the interest points



2. **Description**: Extract vector feature descriptor surrounding each interest point



3. **Matching**: Determine correspondence between descriptors in two views



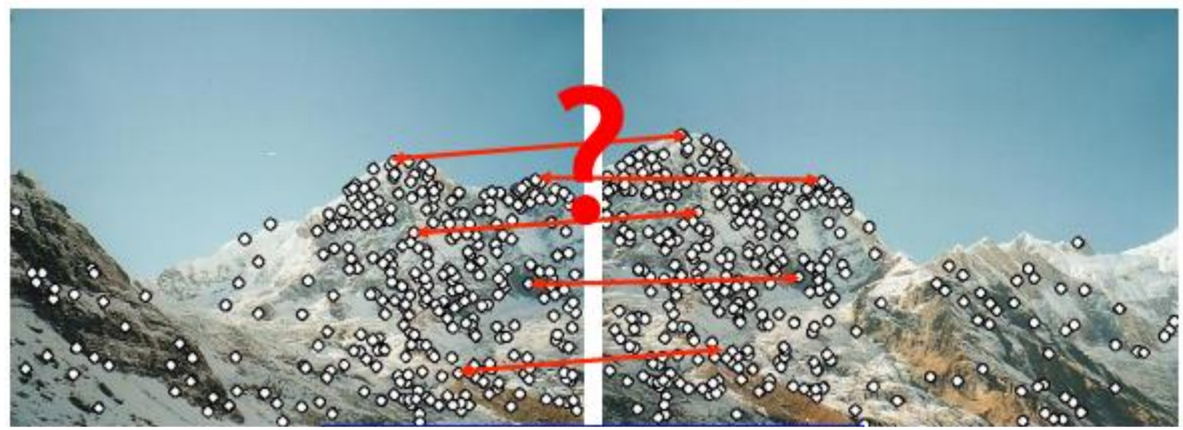
Interest Point Description

Geometric Transformations

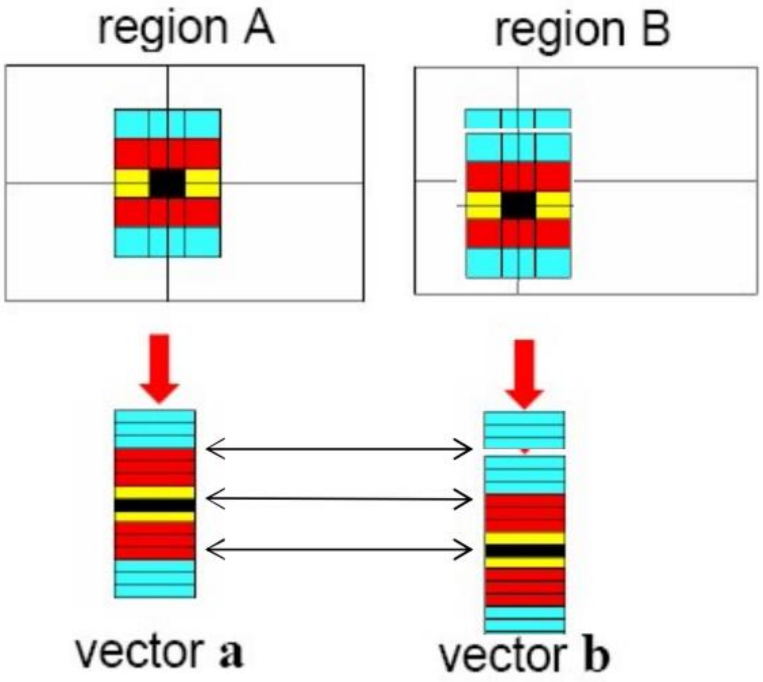
e.g. scale,
translation,
rotation

Descriptors: "Describing" points

- We now know how to detect interest points
- **NEXT**: How to describe them for matching / recognition / etc. ?



Raw patches as descriptors



- Simplest way: Describe the neighbourhood around an interest point by getting the list of intensities to form a feature vector
- But, this is very sensitive to even small shifts, rotations

Raw patches as descriptors

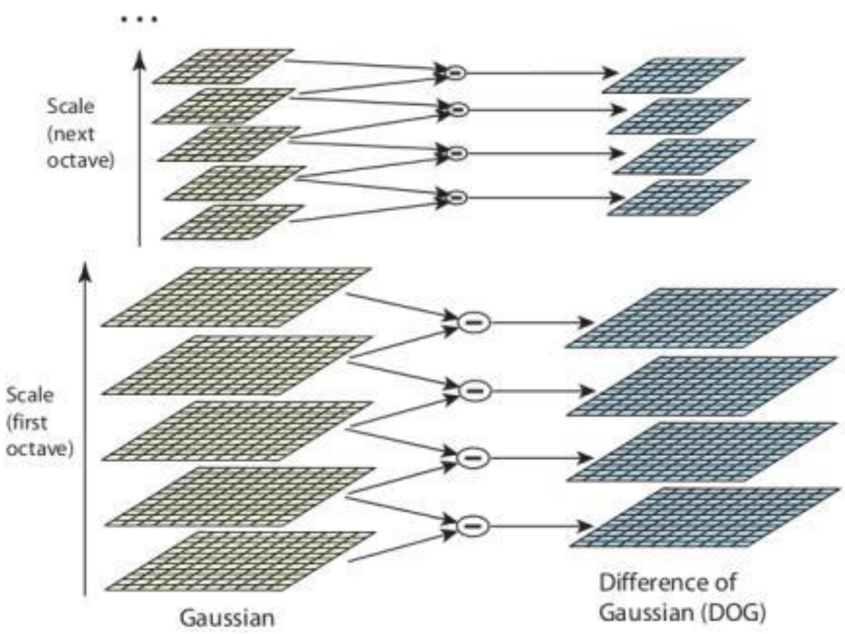
- “Patches” with similar content should have similar descriptors
- Notice how these patches are invariant towards rotation and scale
  - Roughly the same object orientation and size!

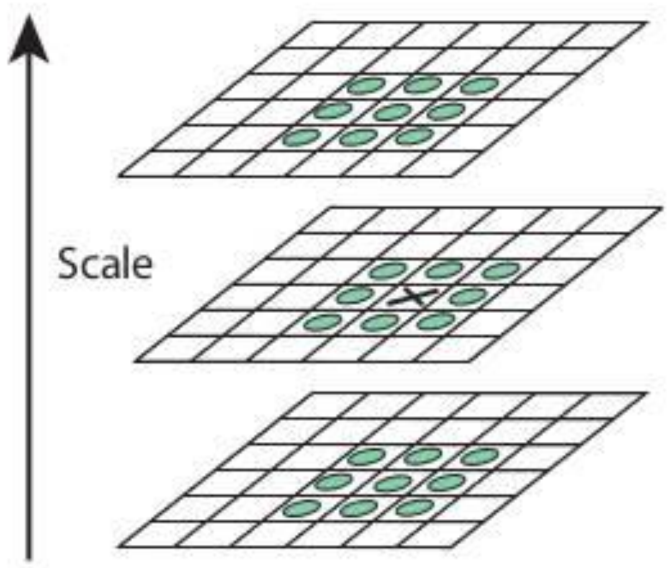
Scale Invariant Feature Transform [Lowe 2004]

1. Detect an interesting patch with an interest operator. Patches are translation invariant.
2. Determine its dominant orientation.
3. Rotate the patch so that the dominant orientation points upward. This makes the patches rotation invariant.
4. Do this at multiple scales, converting them all to one scale through sampling.
5. Convert to illumination “invariant” form

SIFT detector [Lowe 2004]

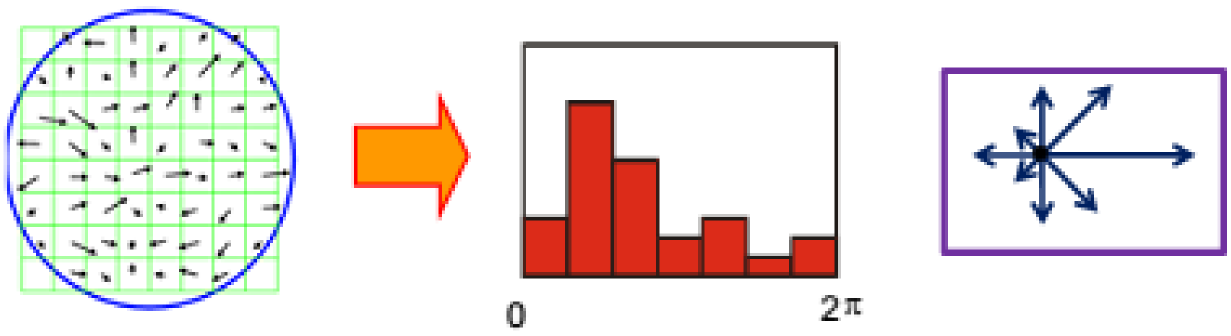
- Scale-space extrema (or “blob”) detection
  - Get DoGs on different octaves of an image in Gaussian pyramid
  - Search for local extrema over scale and space

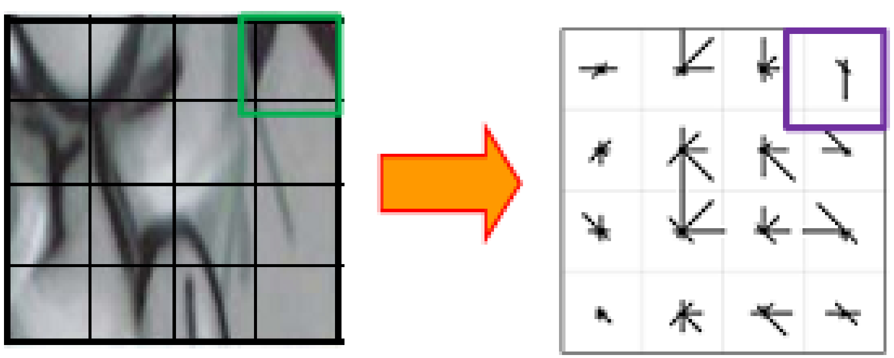




SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation





4x4 grid

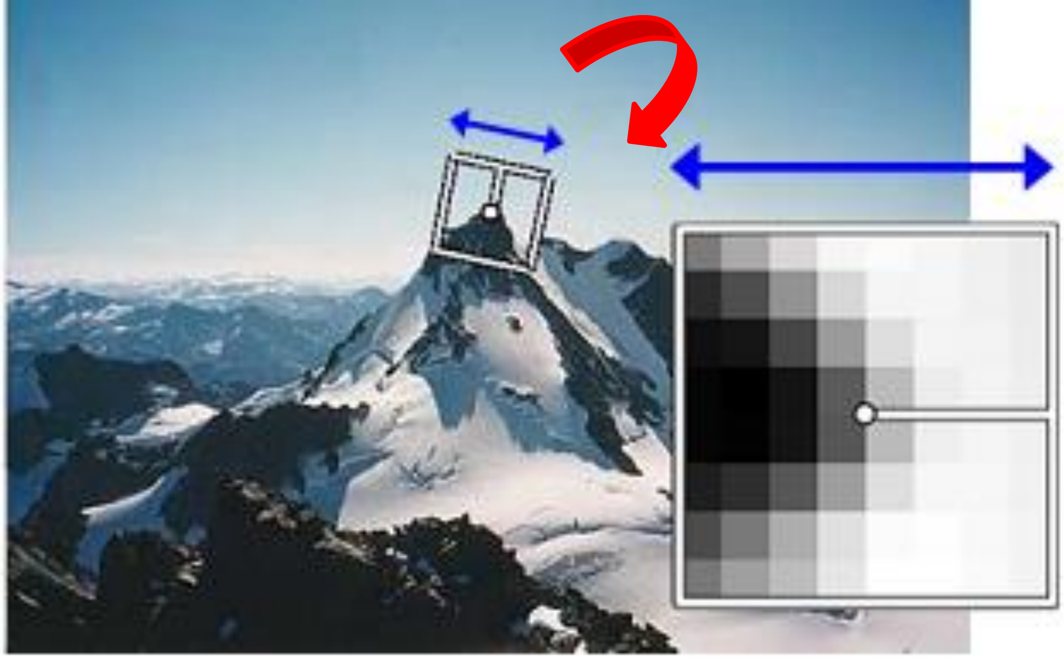
8 orientations per block

Why subpatches?

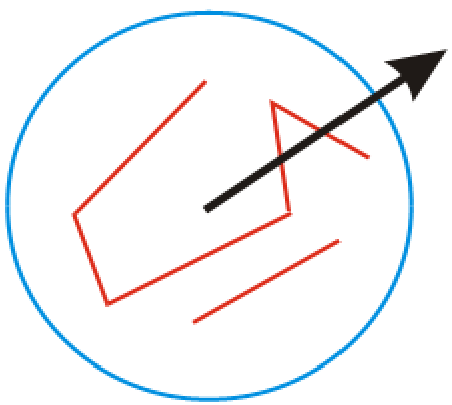
Why does SIFT have some illumination invariance?

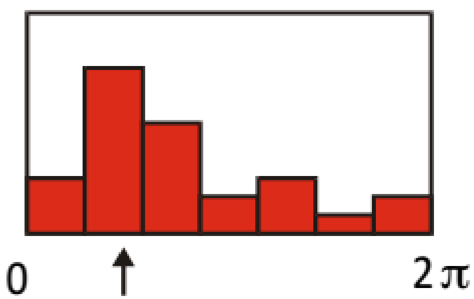
Making descriptor rotation invariant

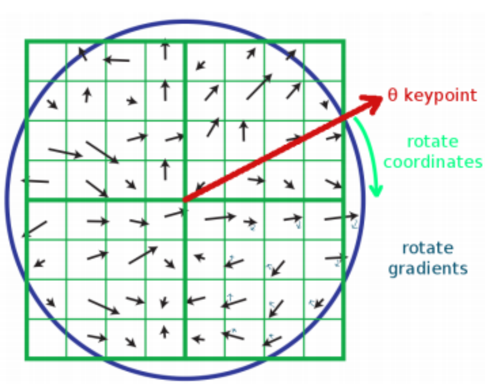
- Rotate patch according to its dominant gradient orientation
  - How to find the dominant direction?



SIFT descriptor formation



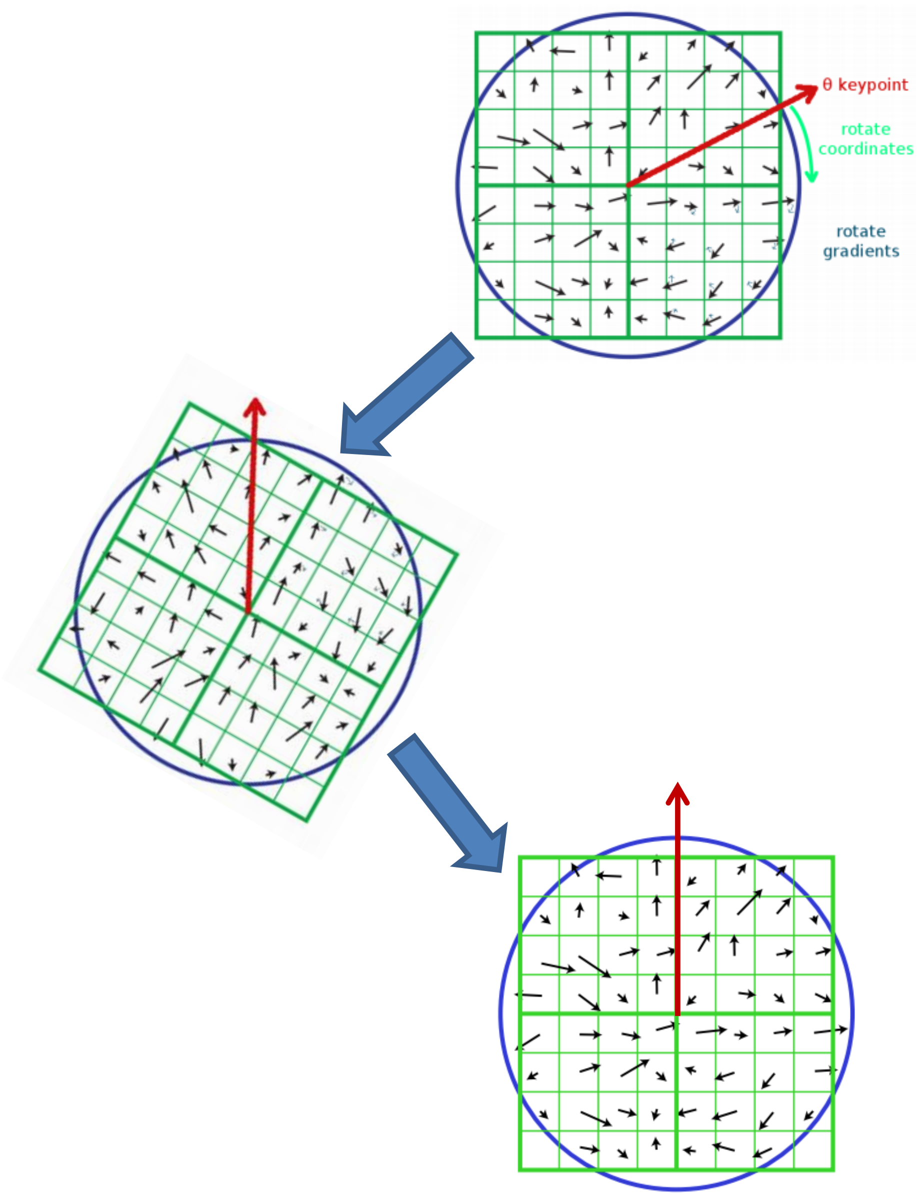




- How to find dominant orientation θ ?
  - Consider a 16x16 window
  - Create an orientation histogram to add up the magnitudes of gradients at each quantized orientation
  - The original SIFT uses 36 angular bins, so the bin width is 360/36 = 10 degrees each

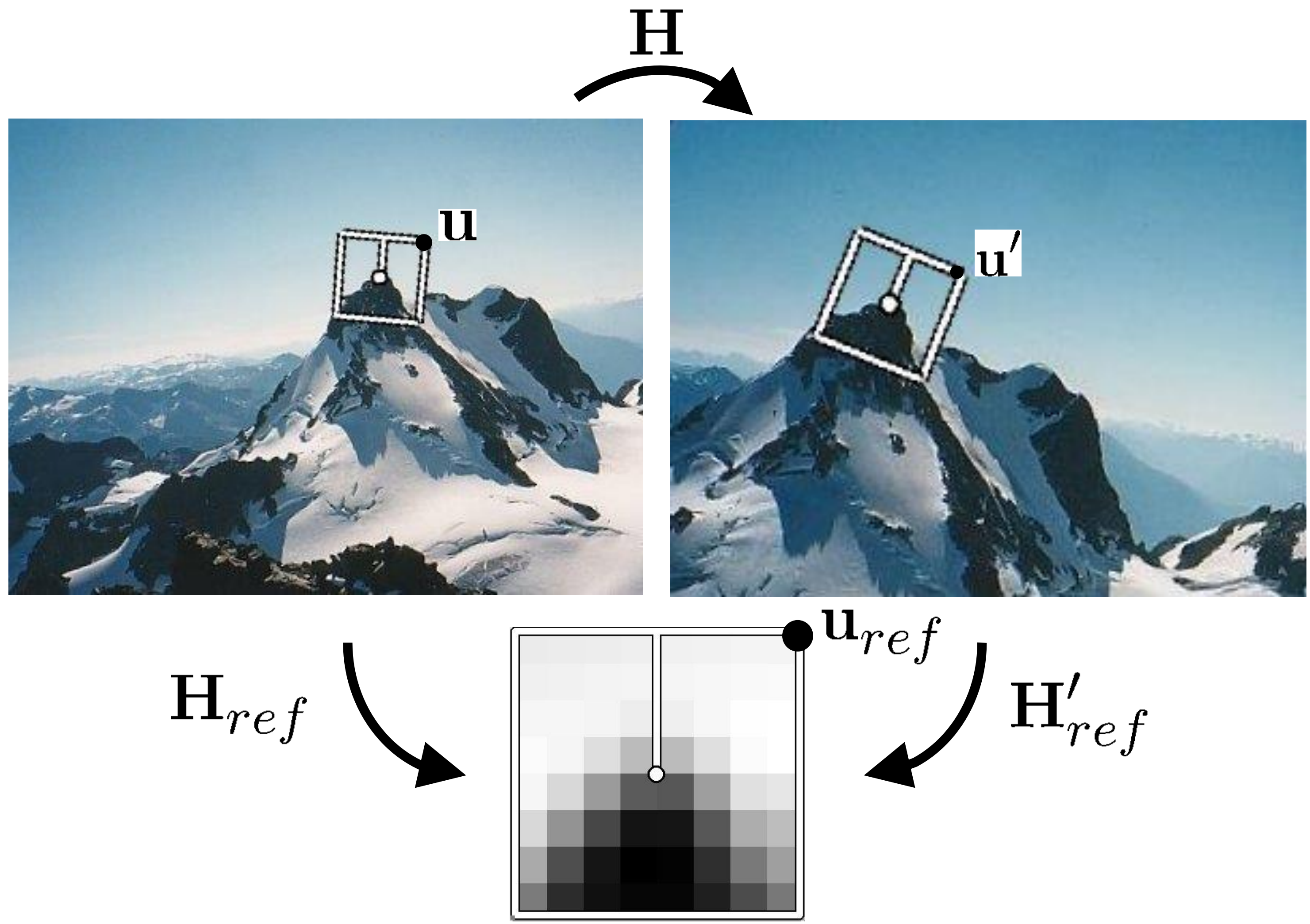
- **Dominant orientation** = bin with the maximum orientation count

SIFT descriptor formation



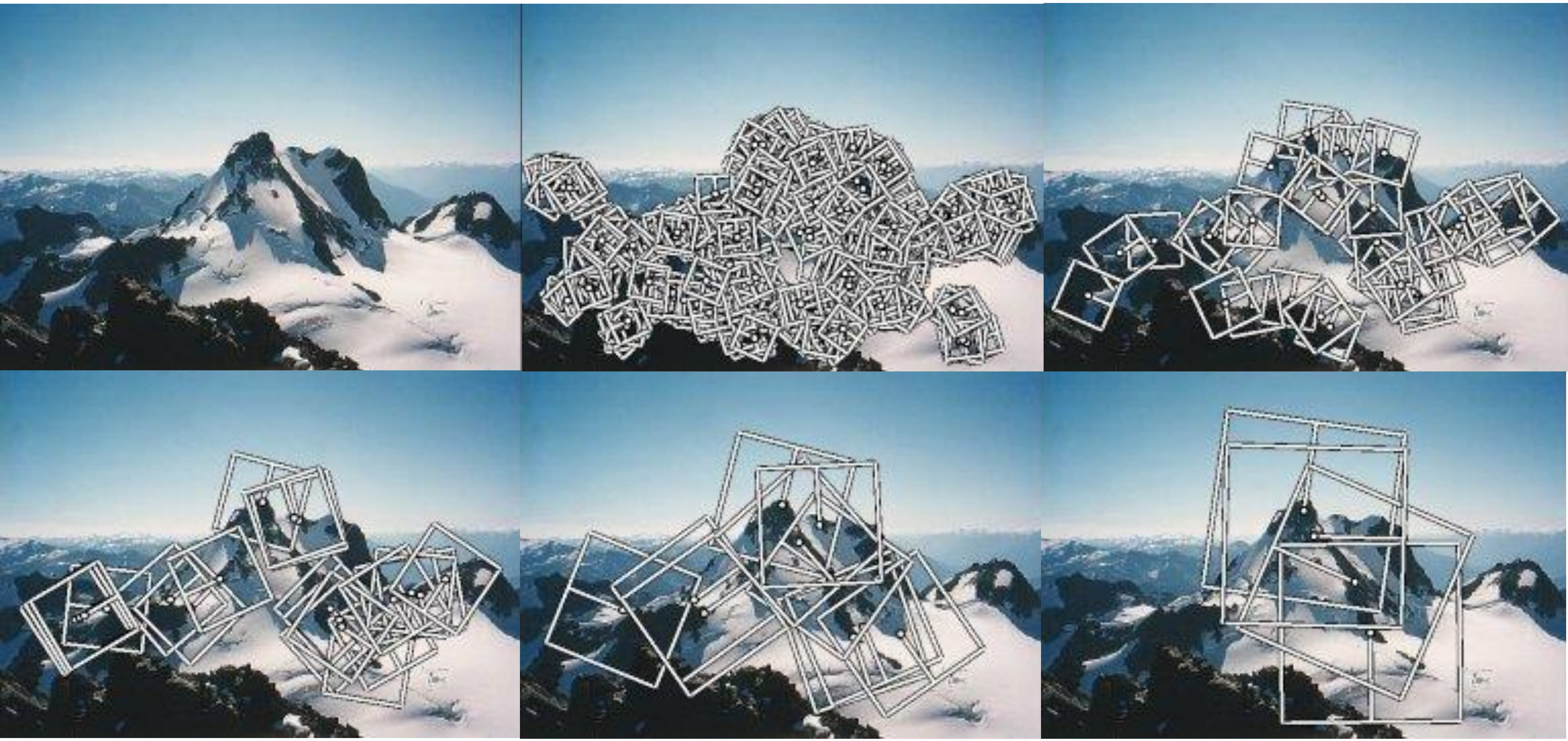
- A 16x16 neighbourhood around a keypoint is taken
- Find the image gradients on the 16x16 array of locations
- To be rotation invariant, rotate the gradient directions AND locations by -θ to align the locations parallel to the dominant direction
- Patches are now in a canonical (or "standard") orientation

Review: Matt Brown's Canonical Frames



[Brown, Szeliski, Winder CVPR 2005]

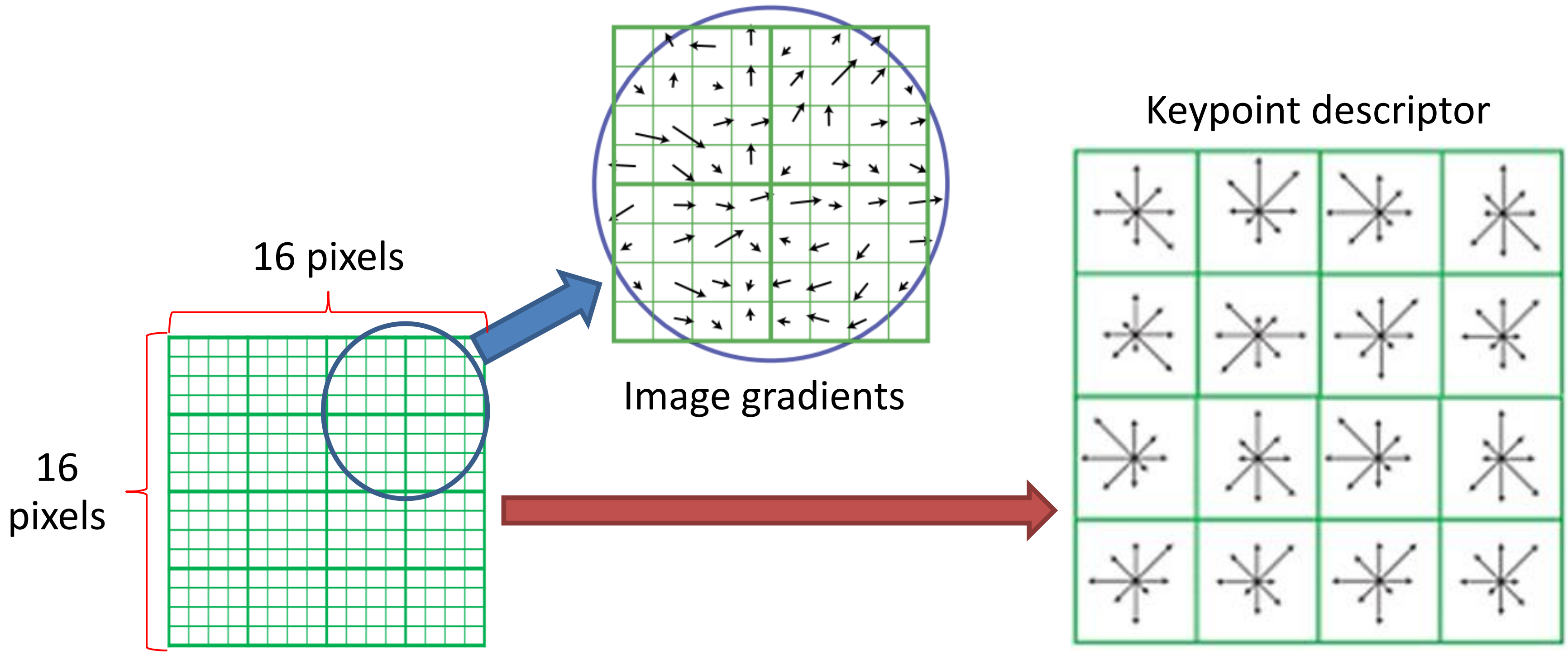
Multi-Scale Oriented Patches



- Extract oriented patches at multiple scales

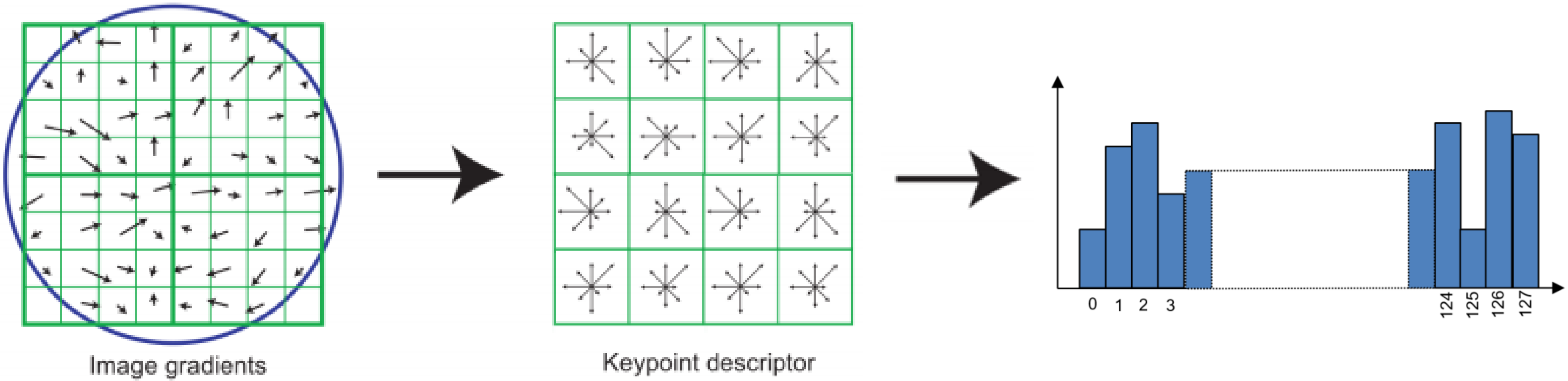
[Brown, Szeliski, Winder CVPR 2005]

SIFT descriptor formation



- Using precise gradients are fragile and inefficient, use a simpler representation
  - Create array of orientation histograms (4x4 array)
  - Put the rotated gradients into their local orientation histograms
  - The SIFT authors found that the best results were with 8 orientation bins per histogram, and a 4x4 histogram array

SIFT descriptor formation

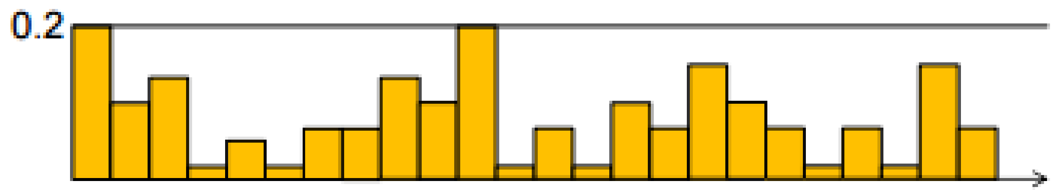


- **8 orientation bins per histogram**, a 4x4 histogram array, yields 8x4x4 = 128 numbers
- SIFT descriptor is a **vector of length 128**, which is invariant to **rotation** (because **descriptor is rotated**) and **scale** (**interest points derived from DoG**)

SIFT descriptor formation

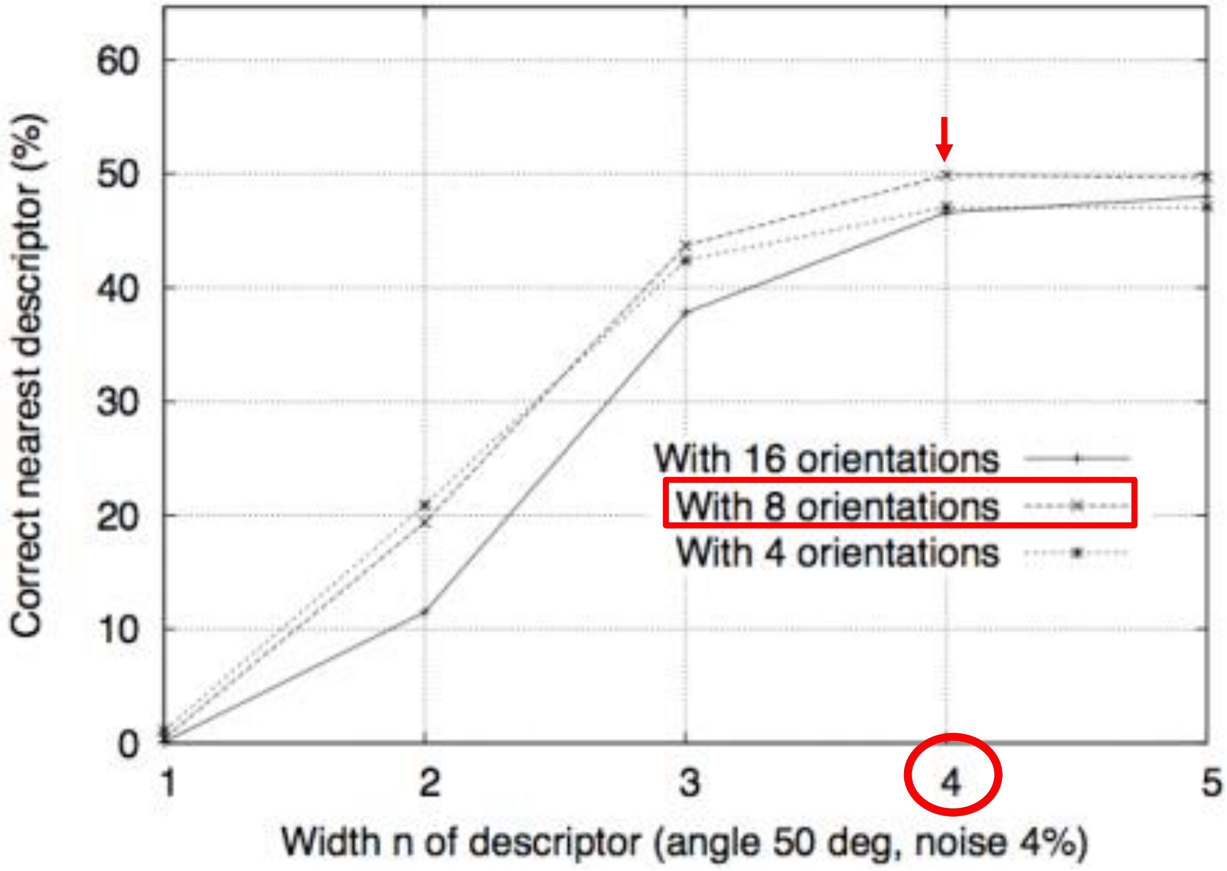
- Finally, the descriptor values are normalized:

$$\sum_i d_i^2 = 1 \quad \text{such that:} \quad d_i < 0.2$$



  – **Reason**: Very large image gradients are usually from unreliable illumination effects (glare, etc.), so this will help clamp the values down → we want some slight invariance towards illumination too!

SIFT descriptor formation



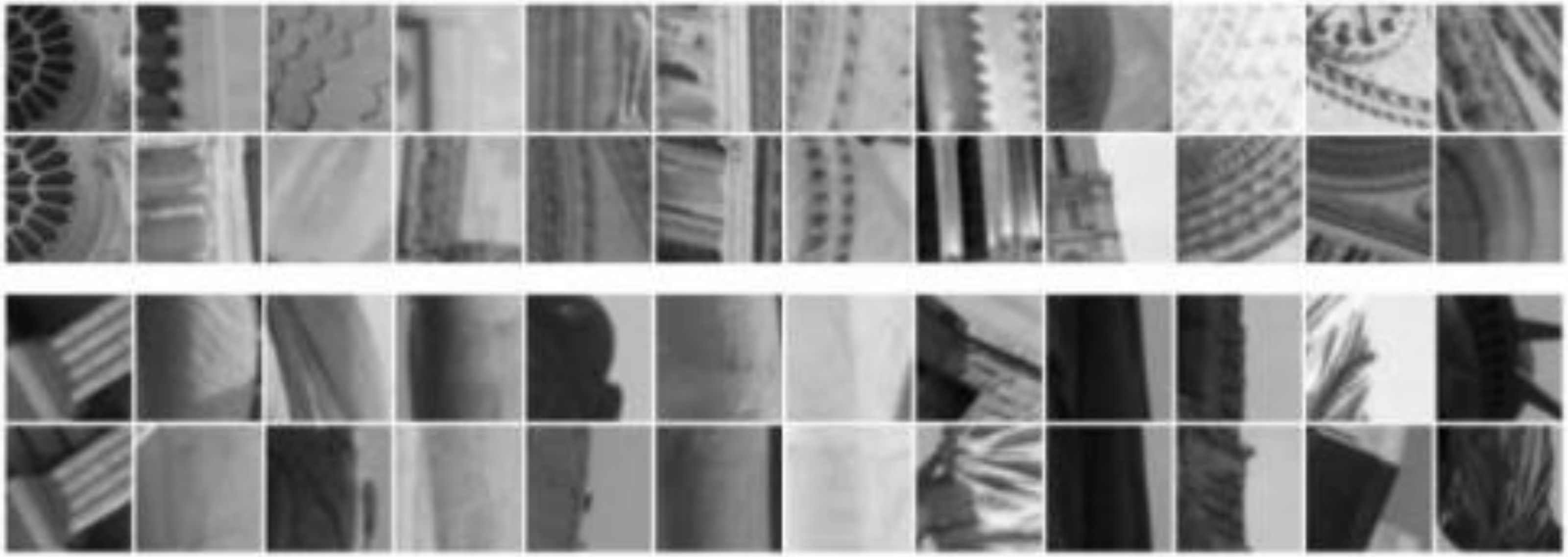
- How did Lowe came up with the use of a **4x4** keypoint descriptor and **8** orientations? By some experiments…

Properties of SIFT

- An extraordinary descriptor that is still popular today
  - Can handle **changes in viewpoint**
    - Up to about 30 degrees out of plane rotation
  - Can handle **significant changes in illumination**
    - Sometimes even day vs. night (below)
  - Fast and efficient – **can run in real-time**
  - Lots of code available (an alternative is the SURF)

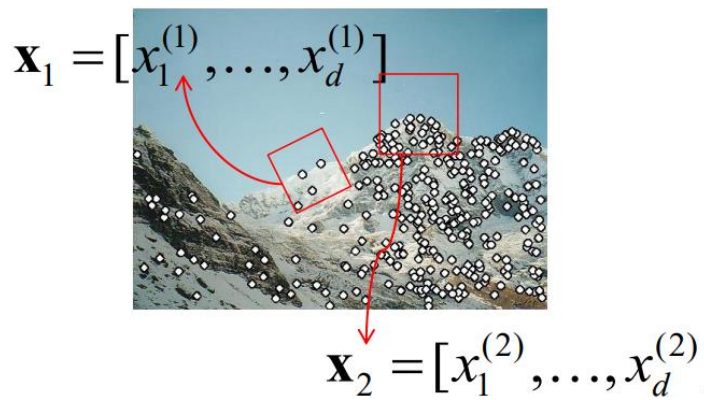
When does SIFT fail?

- Some SIFT patches that are “thought” to be the same, but are not!

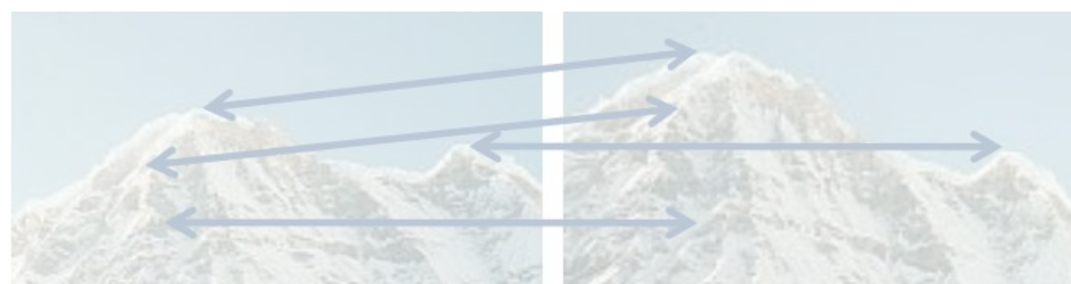


Local Features: Main Components

1. Detection: Identify the interest points



2. Description: Extract vector feature descriptor surrounding each interest point



3. Matching: Determine correspondence between descriptors in two views

Applications of Local Features

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition

Application: Detection by matching

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Application: Automatic Mosaicing

http://matthewalunbrown.com/autostitch/autostitch.html

Application: Wide baseline stereo

Application: Recognition

Schmid and Mohr 1997

Sivic and Zisserman, 2003

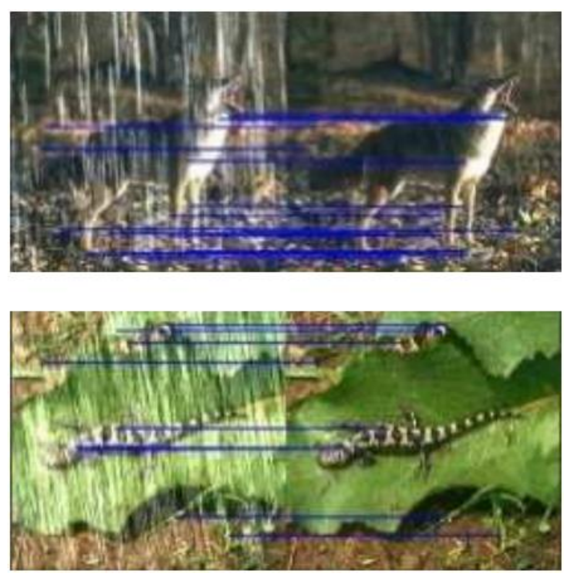
Rothganger et al. 2003

Lowe 2002

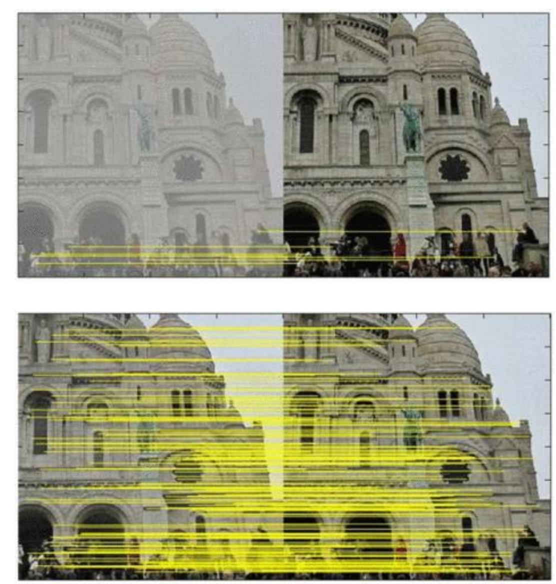
Application: Support High-Level Vision Models

- Keypoint retrieval to support:
  - Image enhancement
  - Object re-identification
  - Object matching

SIFT recovery from dark
(Loh et al., 2019)



SIFT recovery from rain
(Wang et al., 2021)



SIFT recovery from haze
(Huang et al., 2022)

Summary

- **Local invariant features**
  - Why local not global?
  - Why invariant?
- **Detection:** Corners as good distinctive features
  - Harris corner detector
  - Scale-space extrema (blob) detector
- **Description:** Describing features in local "patches"
  - SIFT