Lecture 5
Image Features

Christopher Brunner

AMME4710 – Computer Vision and Image Processing

Semester 2, 2013

http://www.acfr.usyd.edu.au/courses/amme4710/

By the end of this lecture you will be able to:

- Recognize the importance of extracting features from images in computer vision.
- Identify the main kinds of image features and their properties.
- Understand the intuition behind the main approaches to detecting features

Resources

- Readings
  - Szeliski, Computer Vision, Ch. 4

  http://szeliski.org/Book/
- Additional material (optional)
  - Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” 2004
  - Mikolajczyk, “A performance evaluation of local descriptors,” 2007
- Slides credits
  - A. Torralba, S. Seitz, T. Darrell, K. Grauman, R. Fergus

1.4 Fundamental Steps in Digital Image Processing

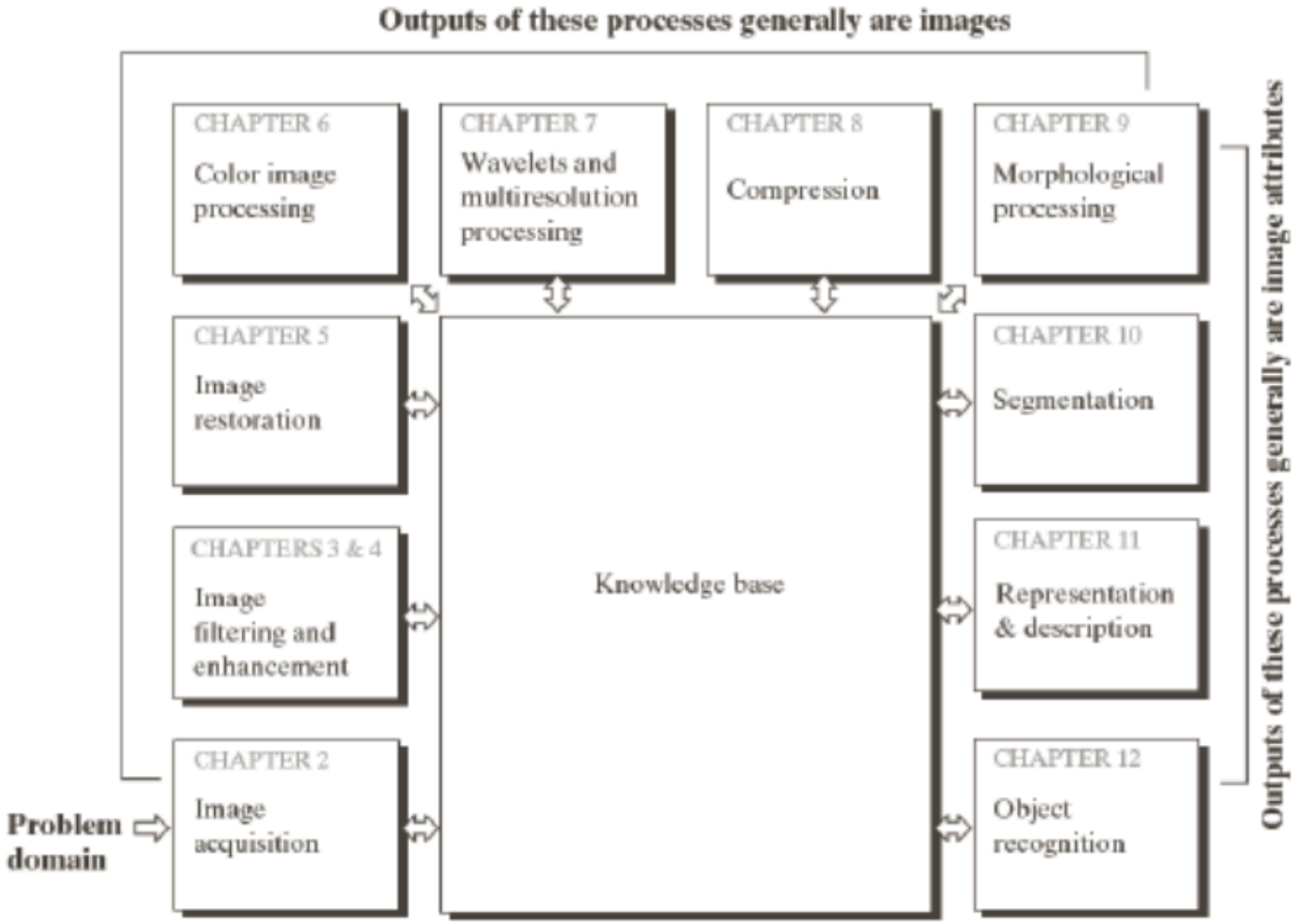
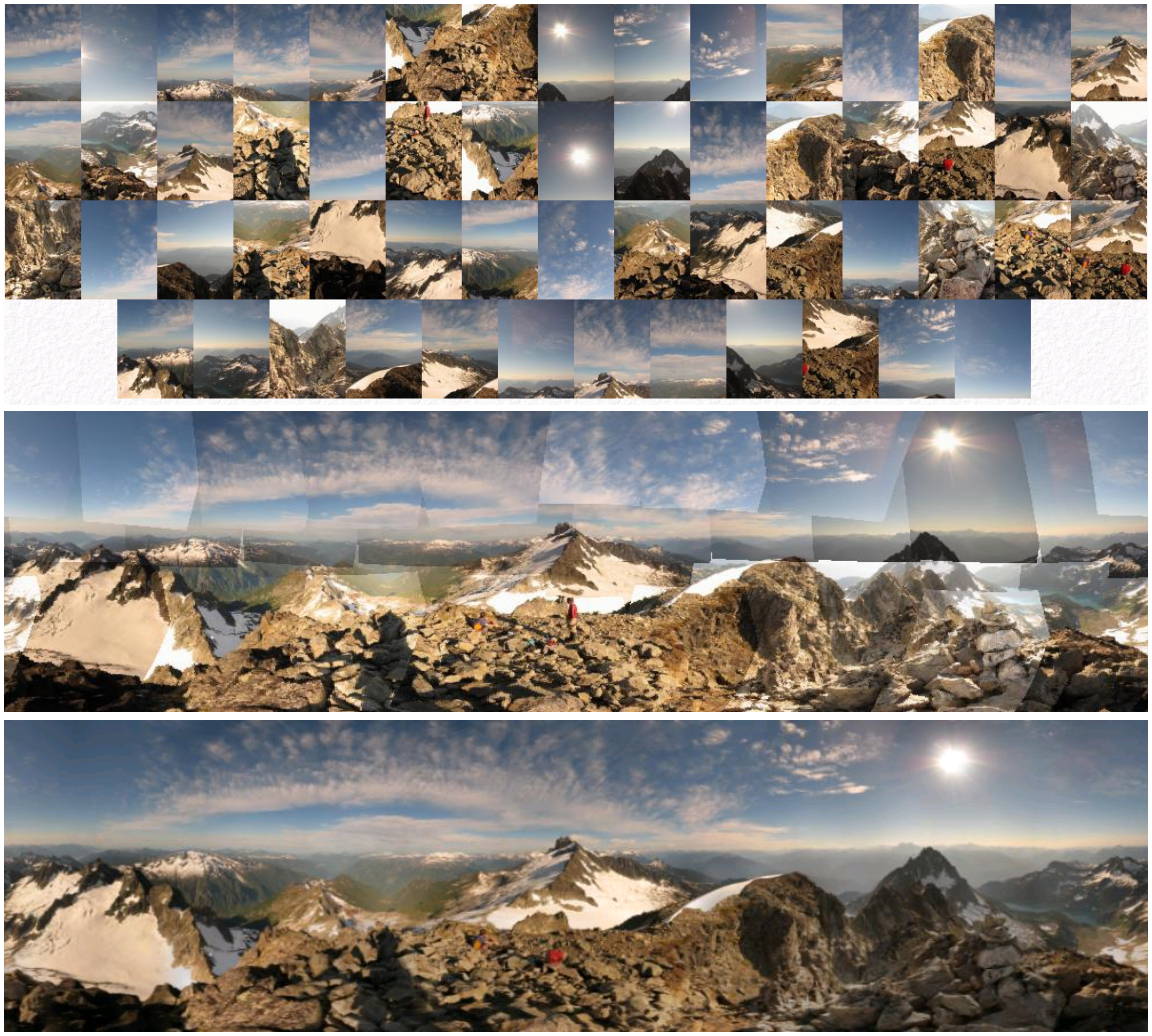


FIGURE 1.23 Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed. *(in the book)*

Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

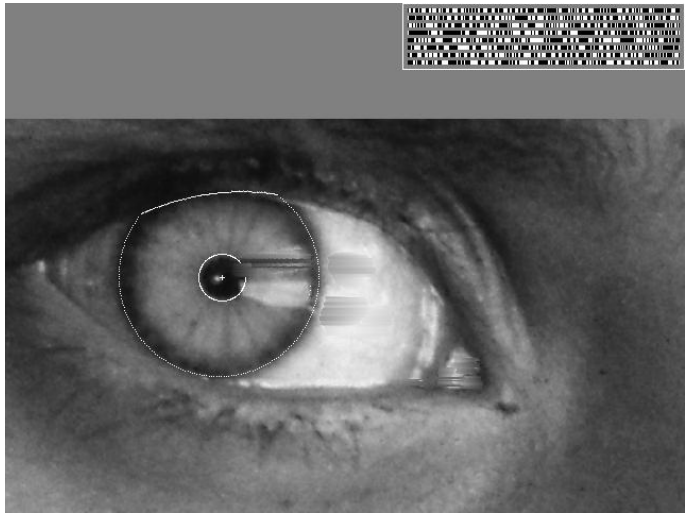
Lowe 2002

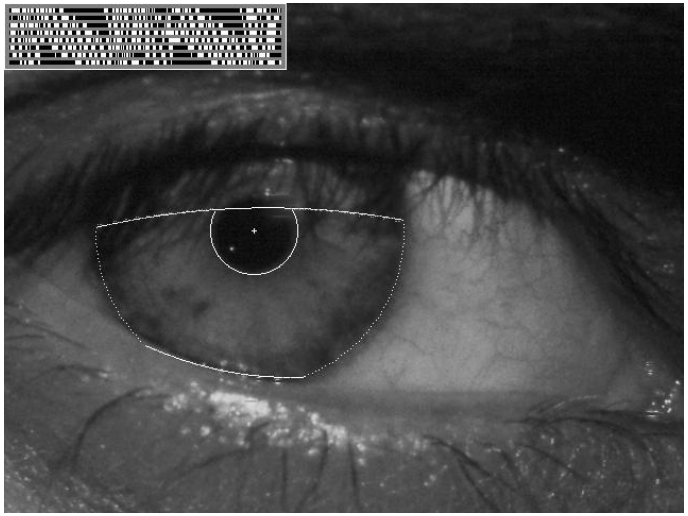
Very hard case

1985, age 12

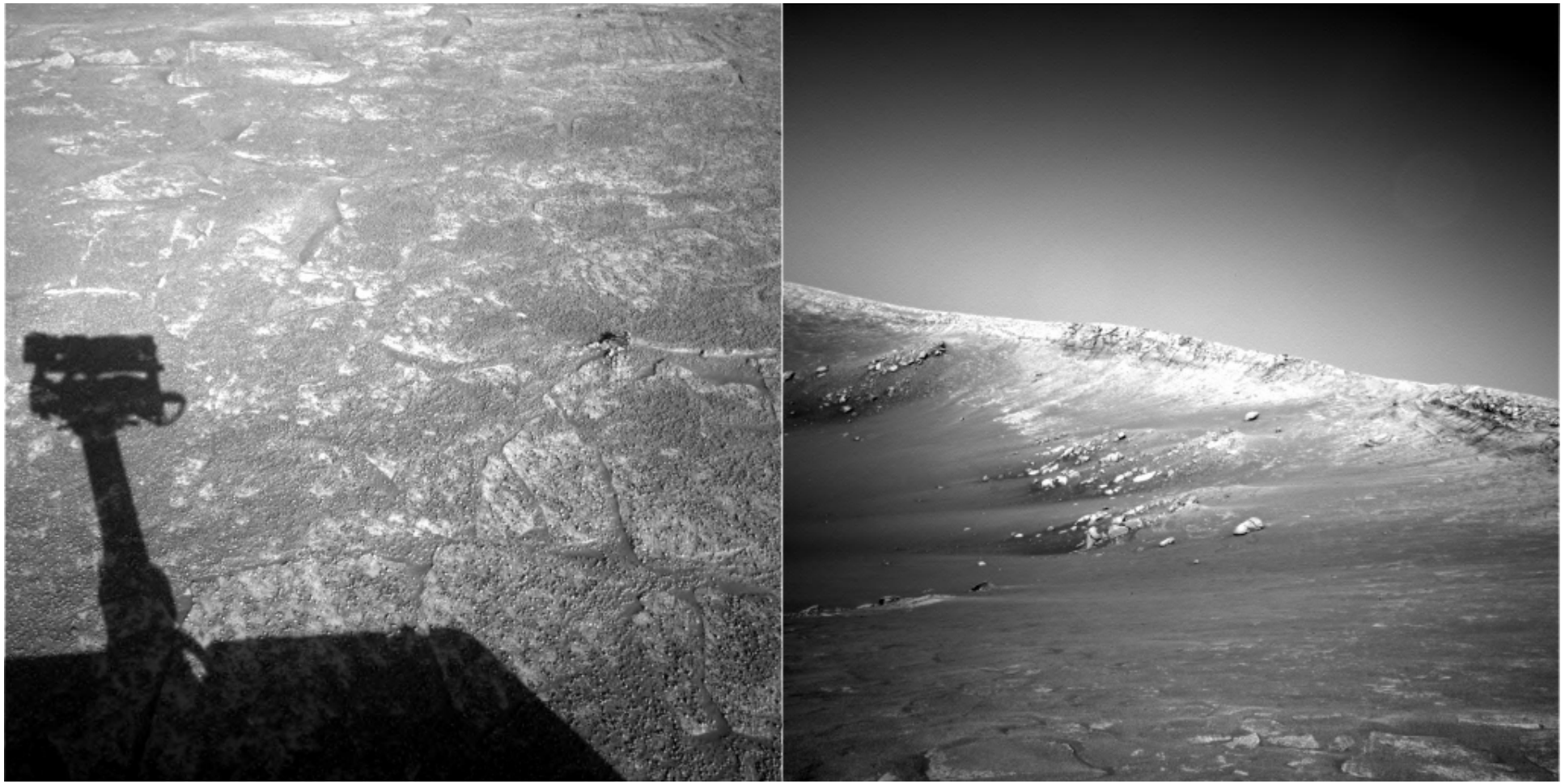
2002, age 30

"*How the Afghan Girl was Identified by Her Iris Patterns*" (J. Daugman, 2002)



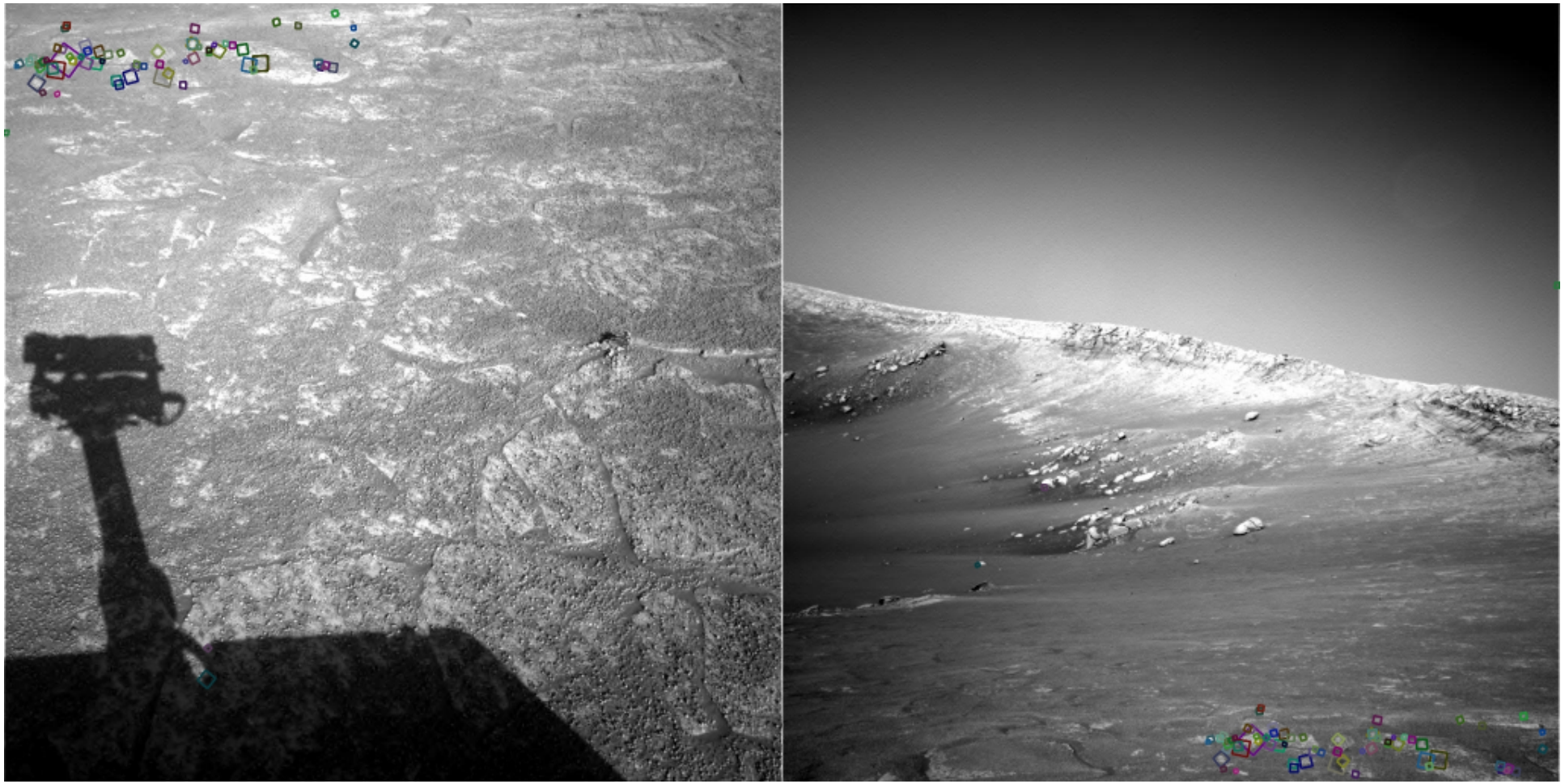


Harder still?



NASA Mars Rover images

Answer below (look for tiny colored squares…)



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

More motivation…

- Feature points are used also for:
  - Image alignment (e.g., mosaics)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - Other …

Outline

- Introduction
- Corners (Harris detector)
- Edges (Canny edge detector)
- Lines (RANSAC, Hough transform)
- Invariant Descriptors (SIFT)
- Matching (ICP)

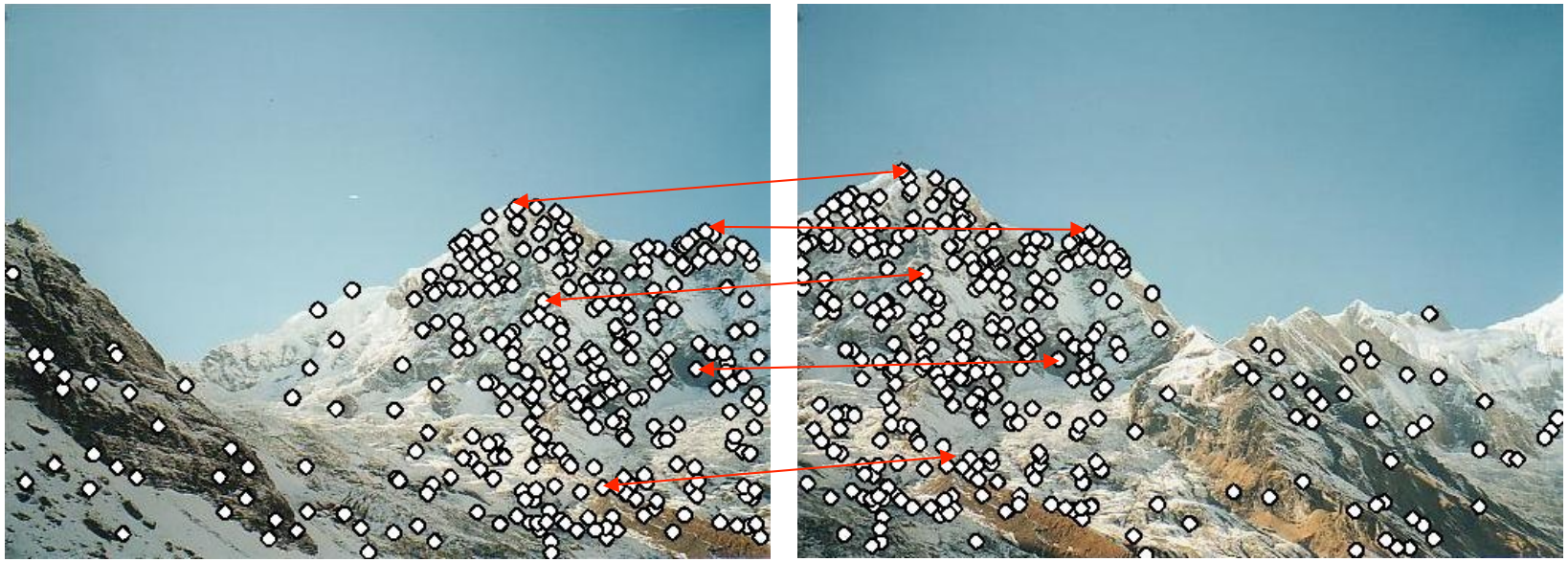
INTRODUCTION

How do we build a panorama?

- We need to match (align) images
- Global methods sensitive to occlusion, lighting, parallax effects. So look for local features that match well.
- How would you do it by eye?

Matching with Features

- Detect feature points in both images
- Find corresponding pairs

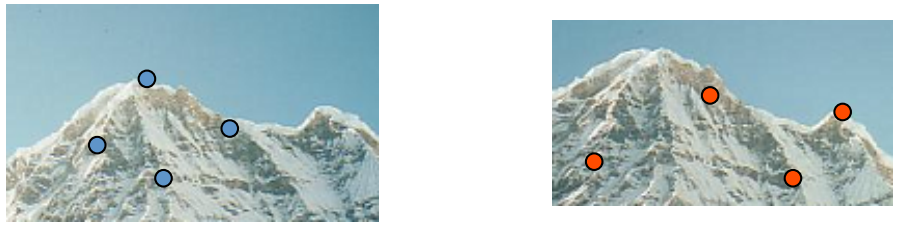


Matching with Features

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

Matching with Features

- Problem 1:
  - Detect the *same* point *independently* in both images

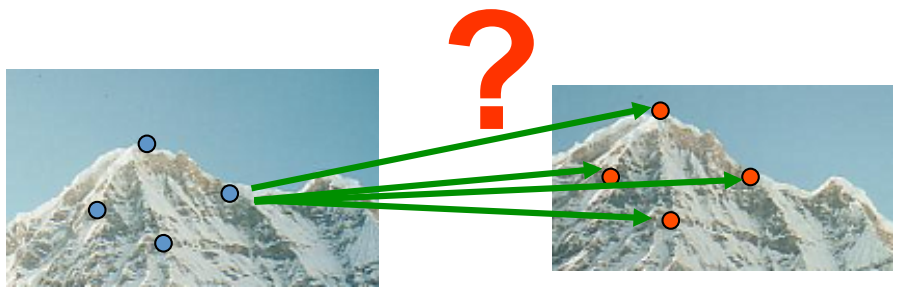


no chance to match!

We need a repeatable detector

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

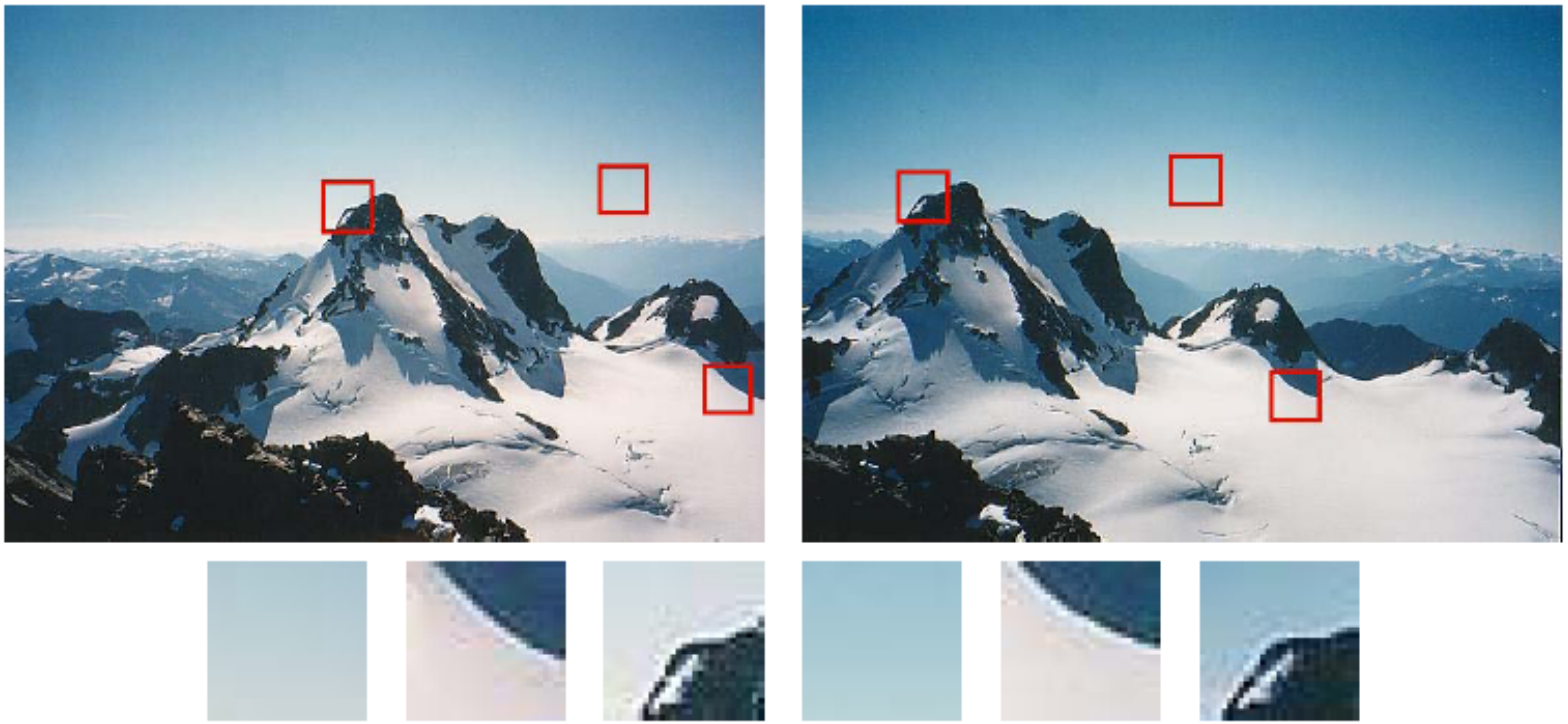
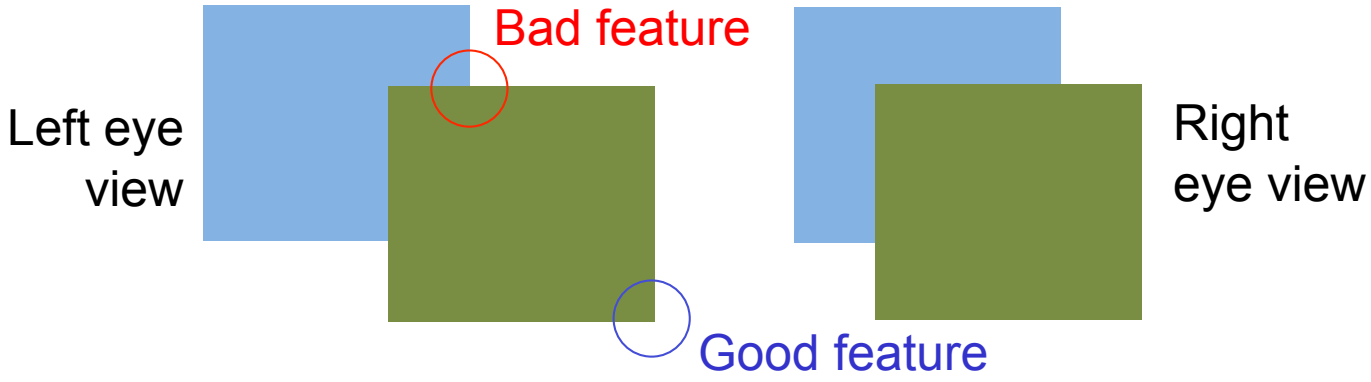


Figure 4.3: *Image pairs with extracted patches below. Notice how some patches can be localized or matched with higher accuracy than others.*

Selecting Good Features

- What's a "good feature"?
  - Satisfies brightness constancy—looks the same in both images
  - Has sufficient texture variation
  - Does not have too much texture variation
  - Corresponds to a "real" surface patch—see below:



  - Does not deform too much over time

Advantages of local features

Locality

- features are local, so robust to occlusion and clutter

Distinctiveness:

- can differentiate a large database of objects

Quantity

- hundreds or thousands in a single image

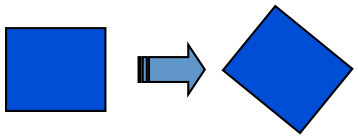
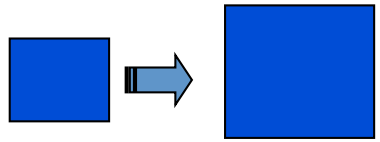
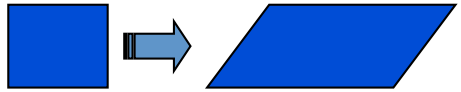
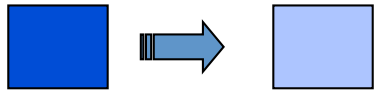
Efficiency

- real-time performance achievable

Generality

- exploit different types of features in different situations

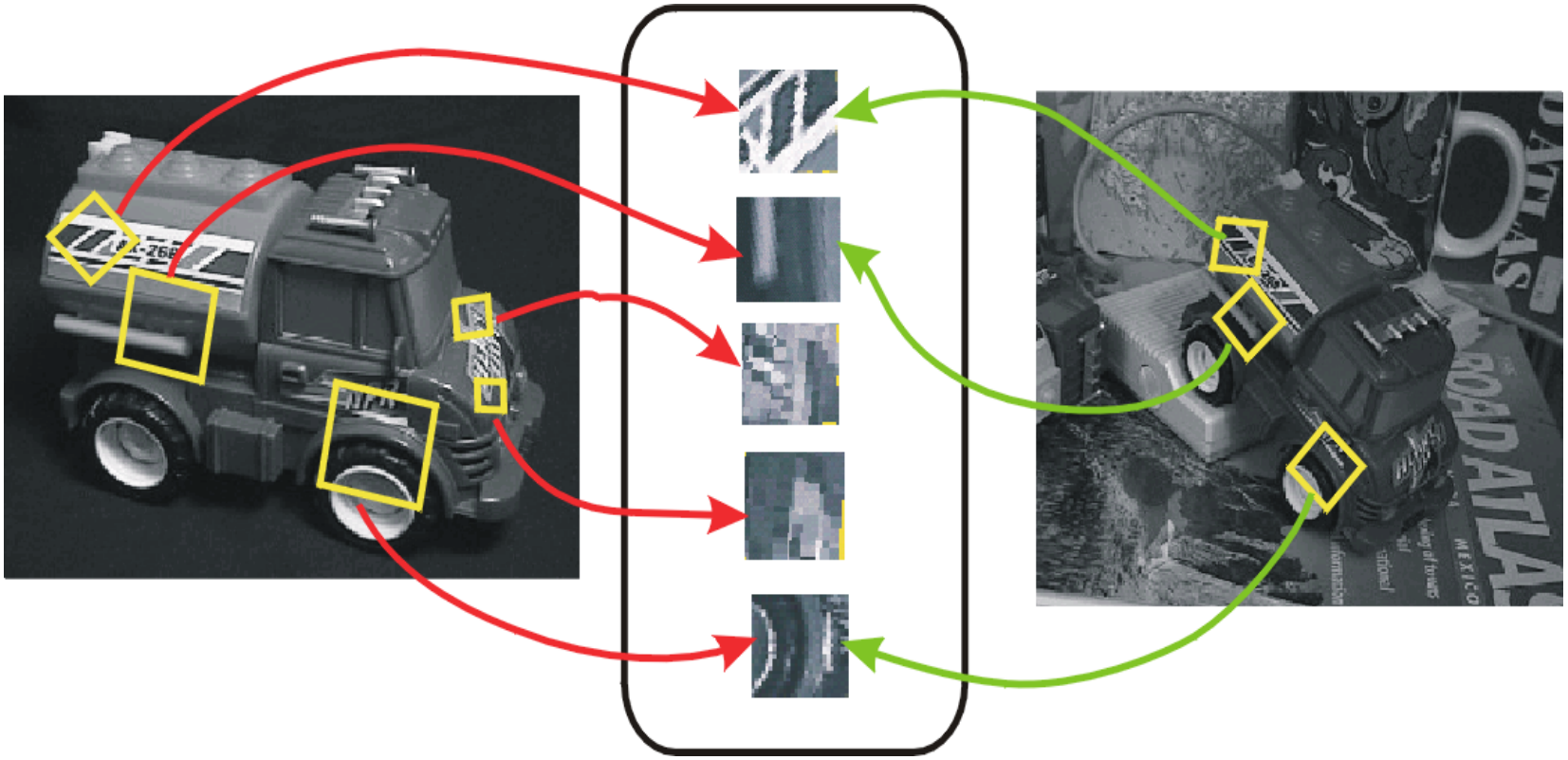
Models of Image Change

- Geometry
  - Rotation 
  - Similarity (rotation + uniform scale) 
  - Affine (scale dependent on direction) 
    valid for: orthographic camera, locally planar object
- Photometry
  - Affine intensity change ($I \rightarrow a\,I + b$) 

Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …

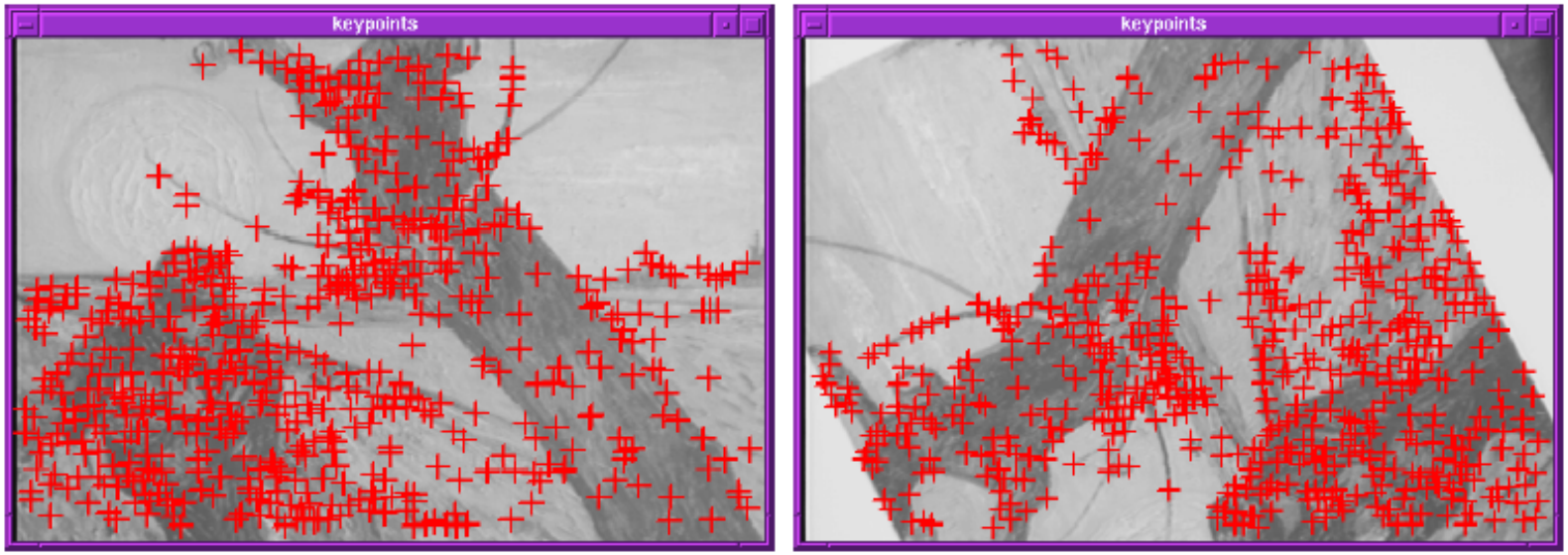


Feature Descriptors

CORNERS

HARRIS DETECTOR

Finding Corners



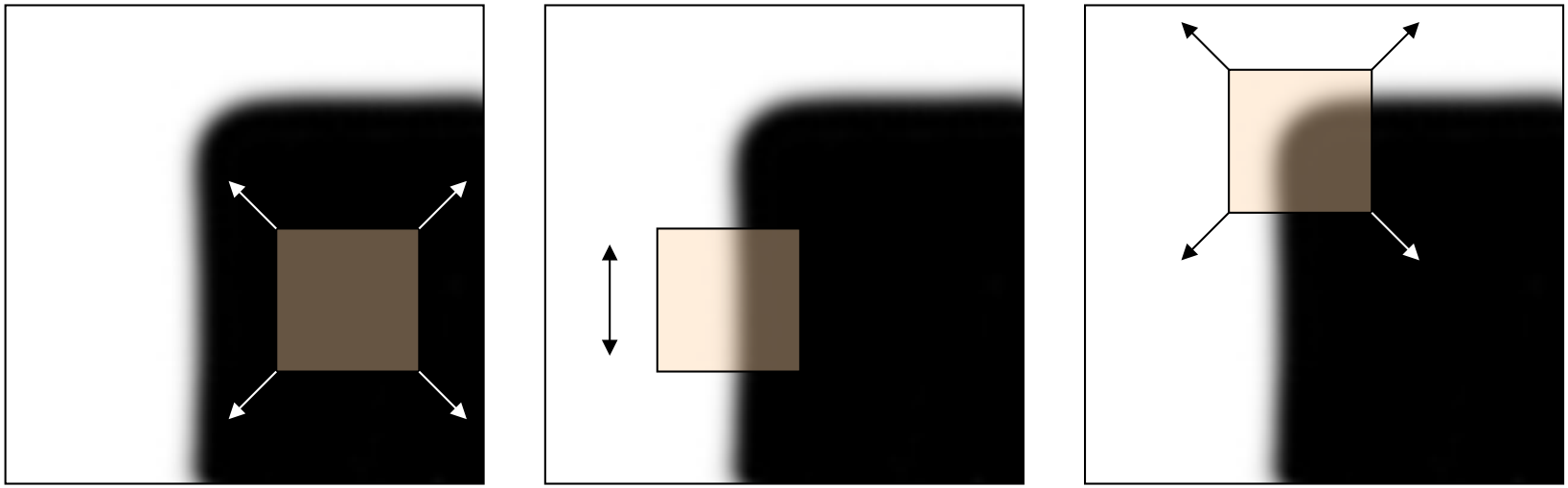
- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and **distinctive**

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Source: Lana Lazebnik

Corners as distinctive interest points

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Source: A. Efros

Harris Detector formulation

Change of intensity for the shift [*u,v*]:

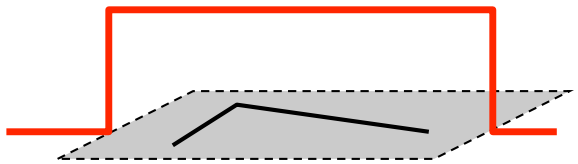
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

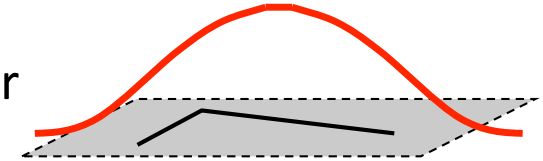
Intensity

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

Source: R. Szeliski

Harris Detector formulation

Expanding I(x,y) in a Taylor series expansion, we have, for small shifts [u,v], a bilinear approximation:

$$E(u,v) \cong \begin{bmatrix} u,v \end{bmatrix} \; M \; \begin{bmatrix} u \\ v \end{bmatrix}$$

where *M* is a 2×2 matrix computed from image derivatives:

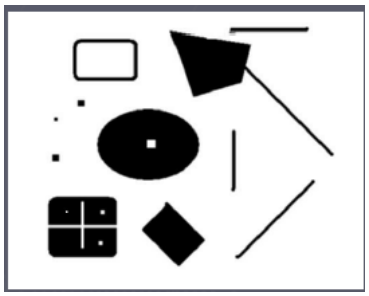
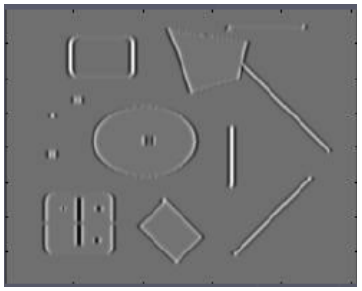
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

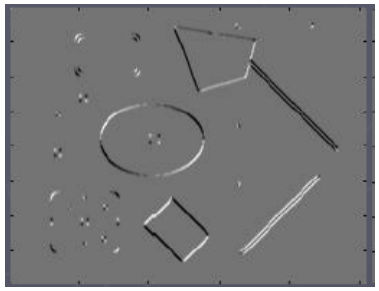
Gradient with respect to x, times gradient with respect to y

Sum over image region – area we are checking for corner

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

Harris Detector formulation



where *M* is a 2×2 matrix computed from image derivatives:

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

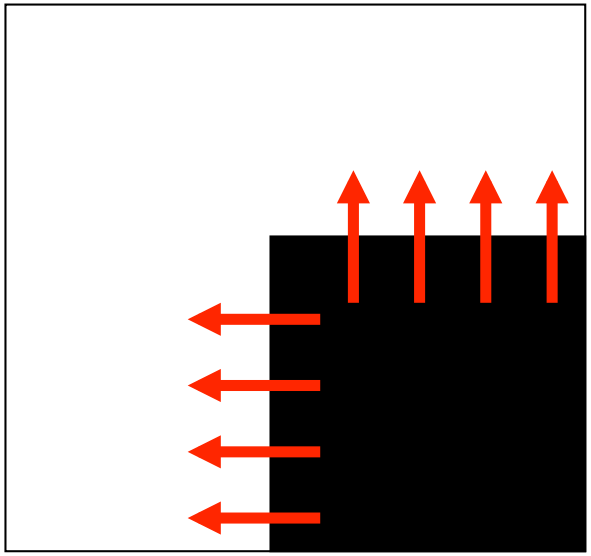
Gradient with respect to x, times gradient with respect to y

Sum over image region – area we are checking for corner

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

What does this matrix reveal?

First, consider an axis-aligned corner:



What does this matrix reveal?

First, consider an axis-aligned corner:

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

This means dominant gradient directions align with x or y axis

If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

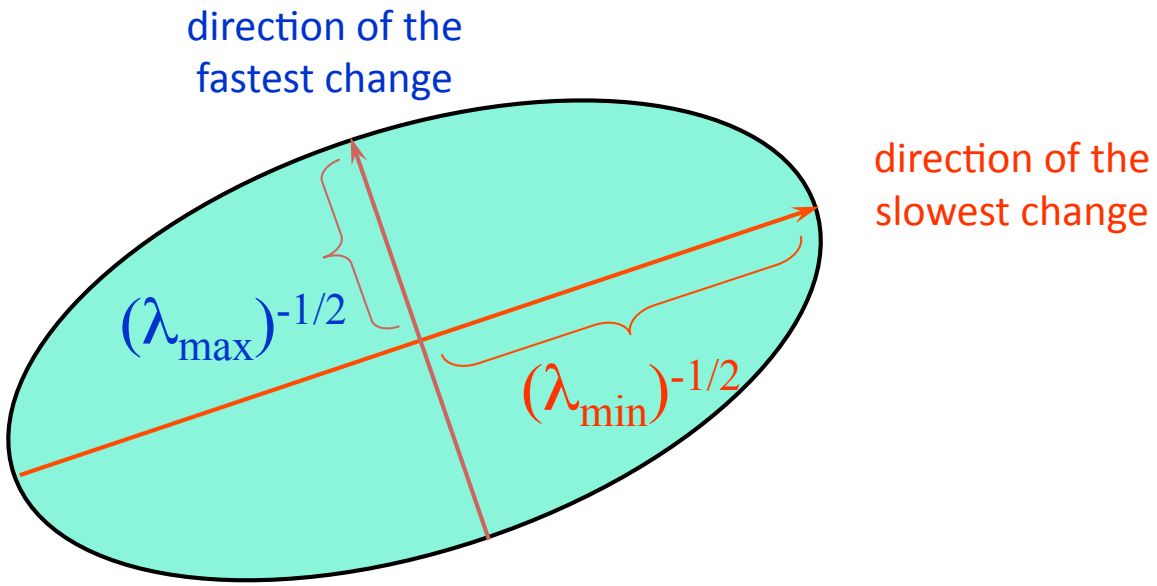
What if we have a corner that is not aligned with the image axes?

Slide credit: David Jacobs

General Case

Since M is symmetric, we have $M = R^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}R$

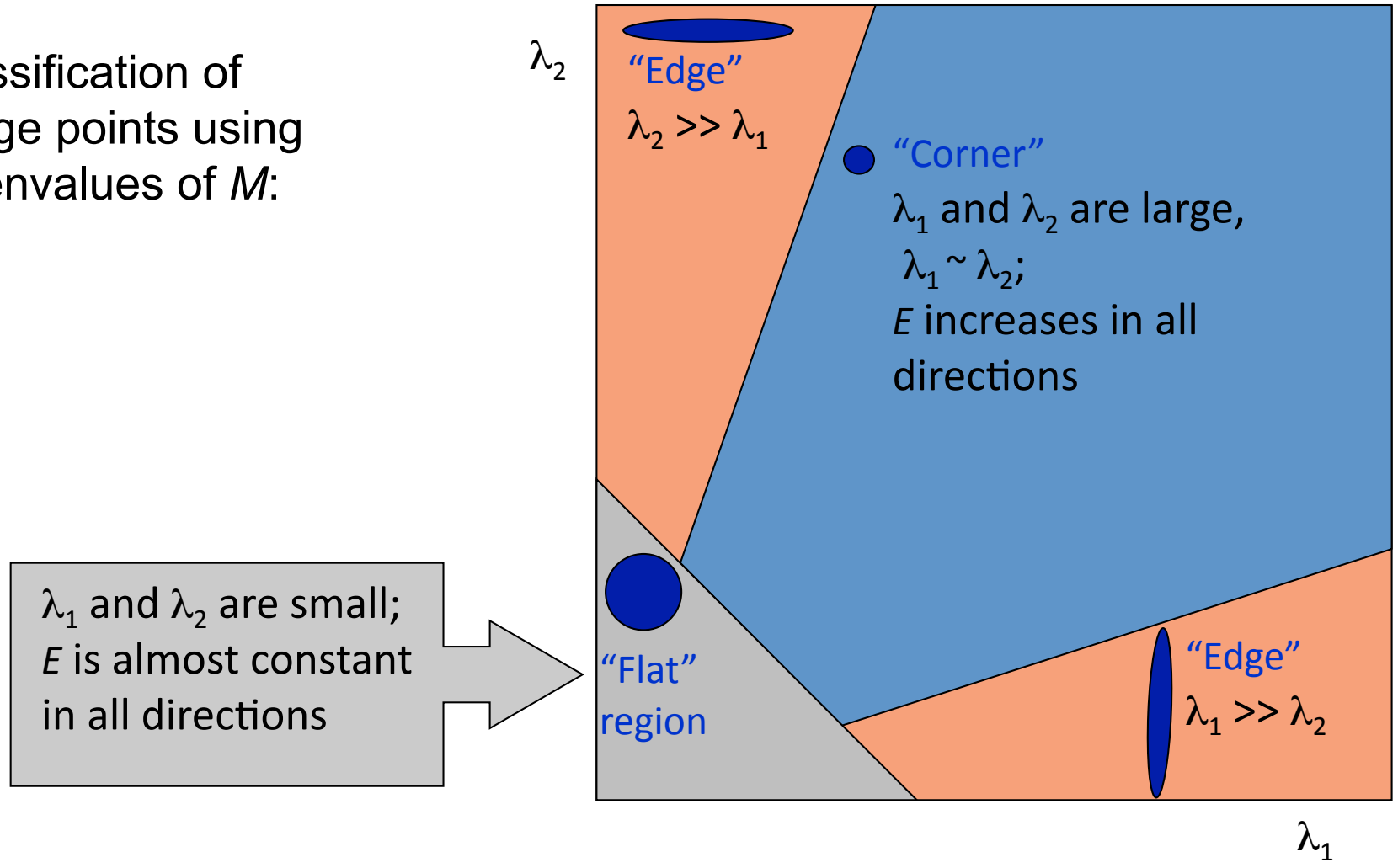
We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*



Slide adapted form Darya Frolova, Denis Simakov.

Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:



Harris Detector: Mathematics

Measure of corner response:

$$R = \det M - k\left(\text{trace } M\right)^2$$

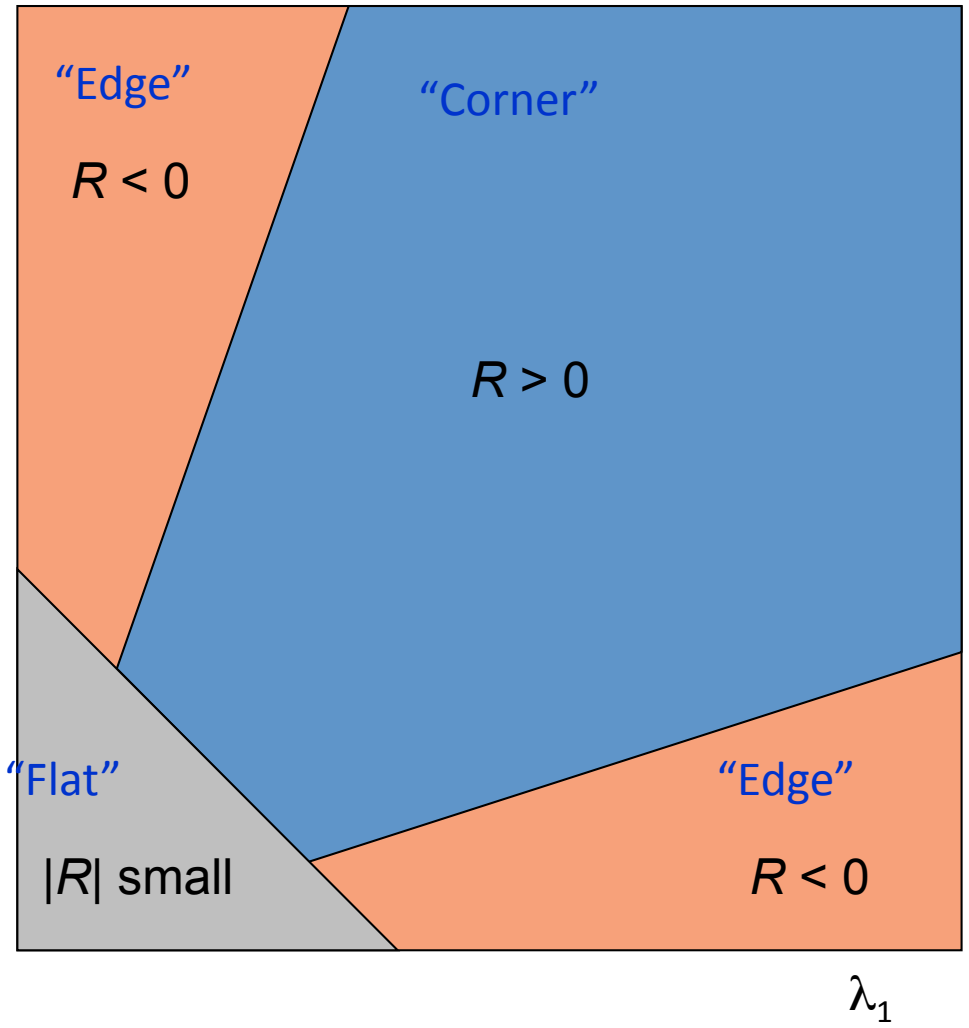
This expression does not requires computing the eigenvalues.

$$\det M = \lambda_1 \lambda_2$$
$$\text{trace } M = \lambda_1 + \lambda_2$$

(k – empirical constant, k = 0.04-0.06)

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Harris corner detector

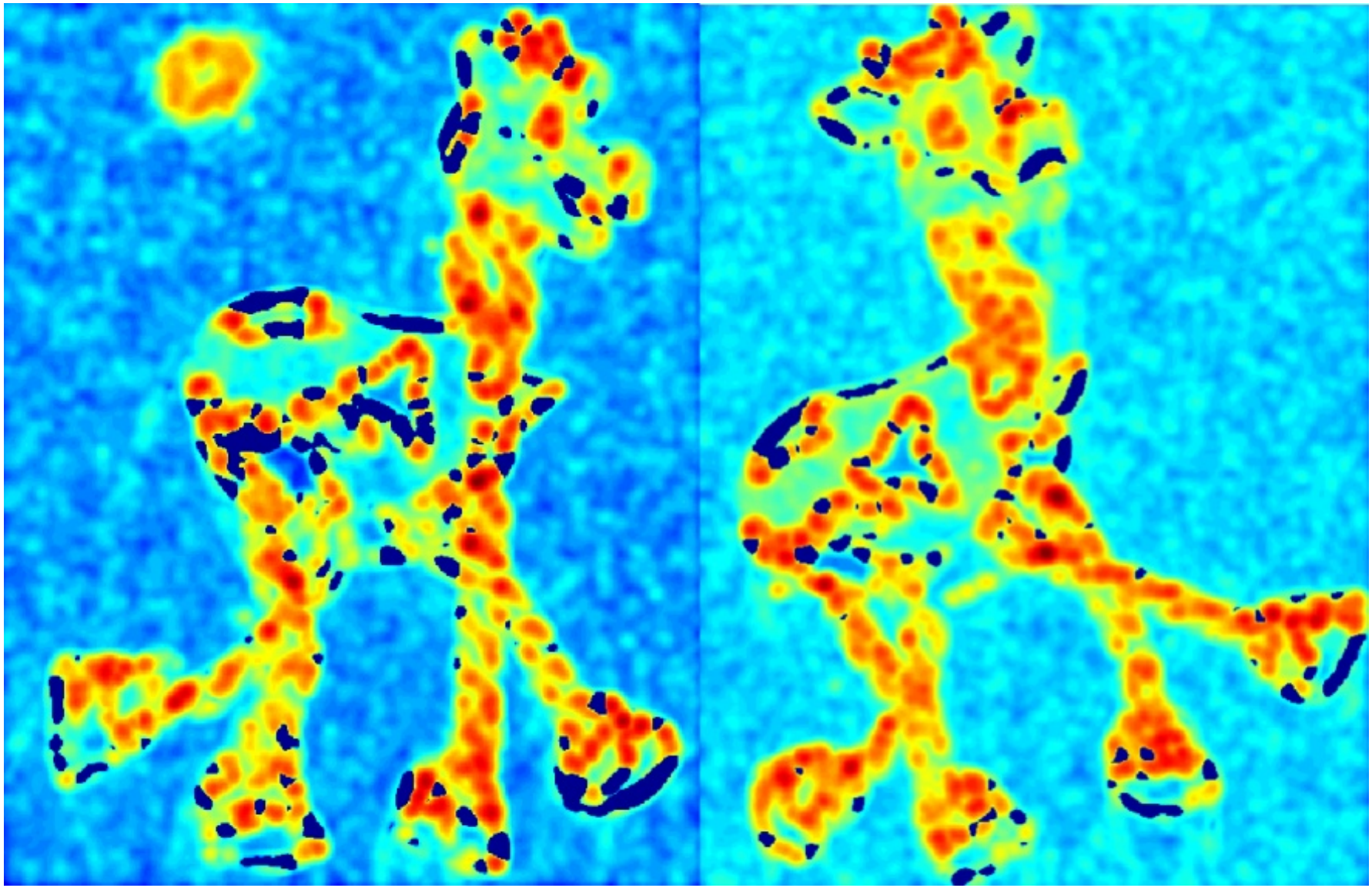
- Algorithm steps
  1. Compute Gaussian derivatives at each pixel
  2. Compute second moment matrix *M* in a Gaussian window around each pixel
  3. Compute corner response function *R*
  4. Find points with large corner response (R > threshold)
  5. Take the points of local maxima of R

Harris Detector: Workflow

Slide adapted form Darya Frolova, Denis Simakov, Weizmann Institute.

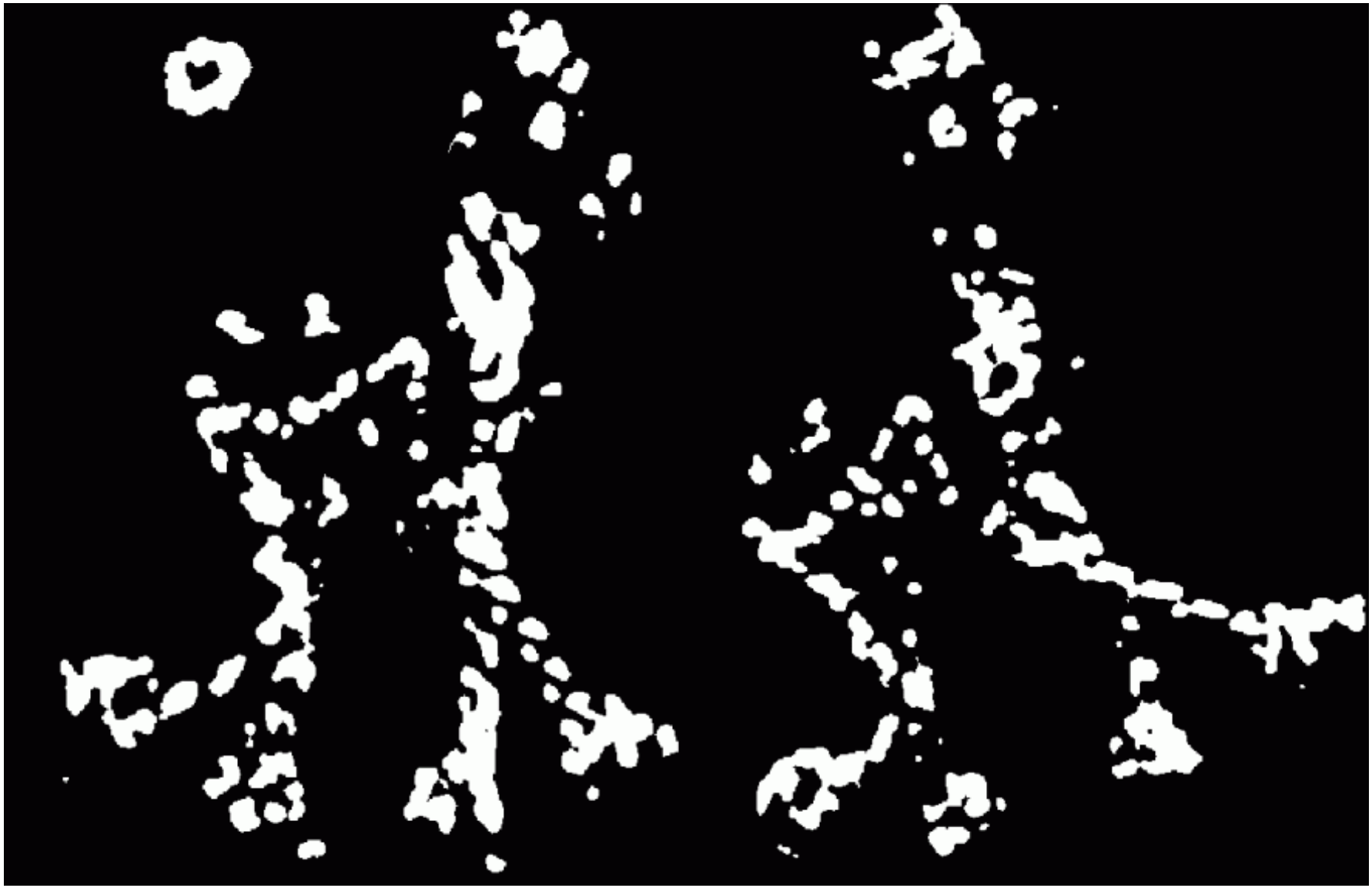
Harris Detector: Workflow

Compute corner response R



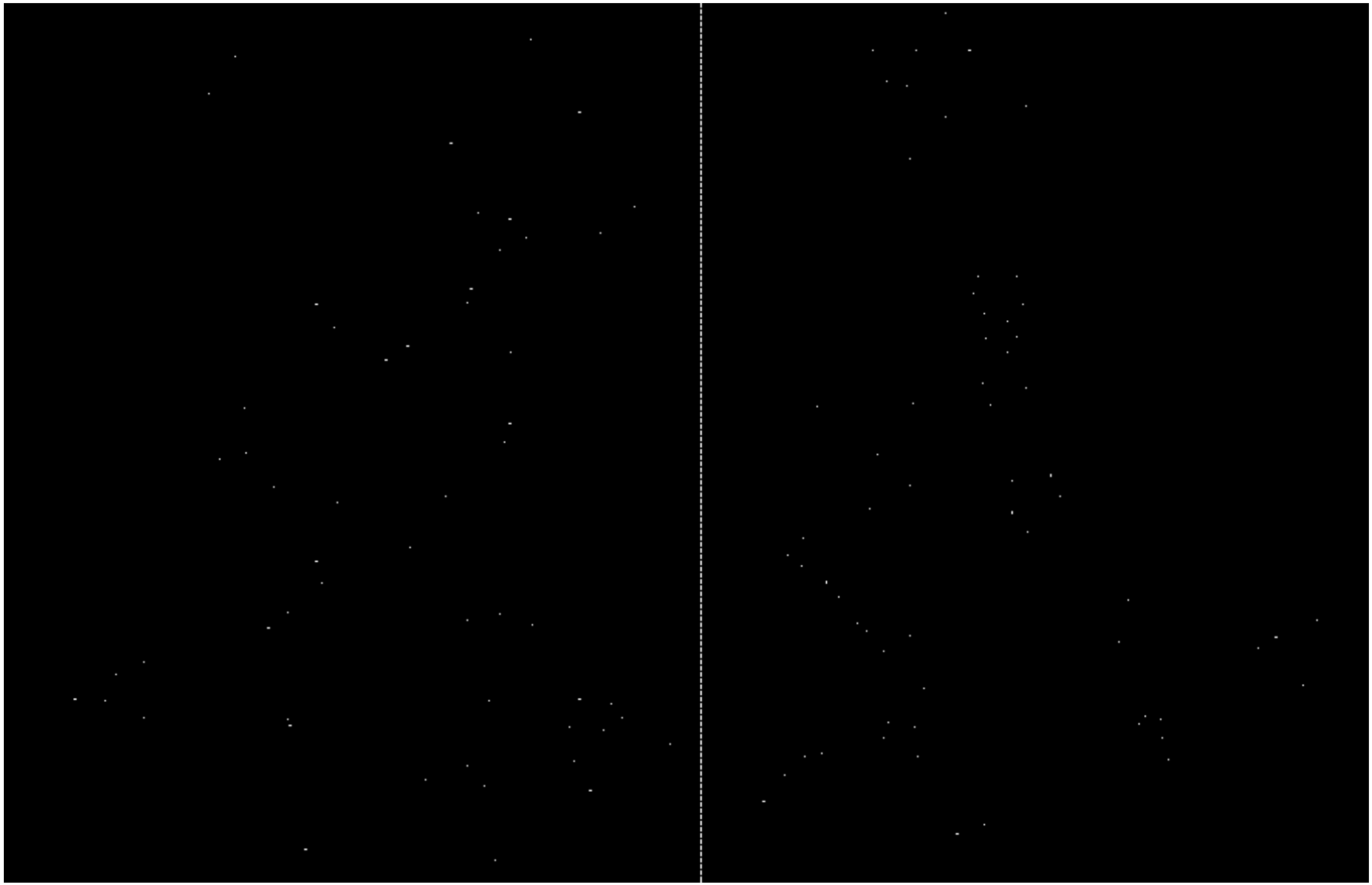
Harris Detector: Workflow

Find points with large corner response: R>threshold



Harris Detector: Workflow

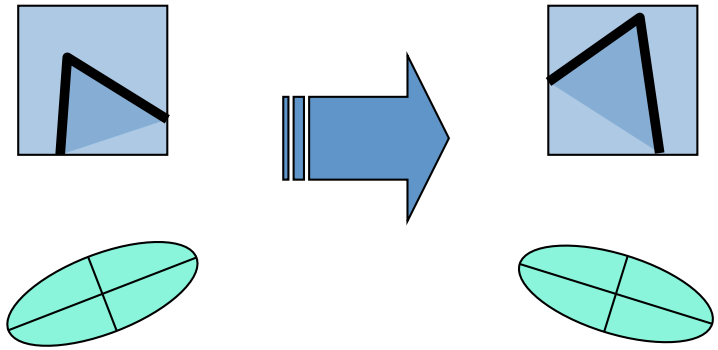
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Properties

- Rotation invariance

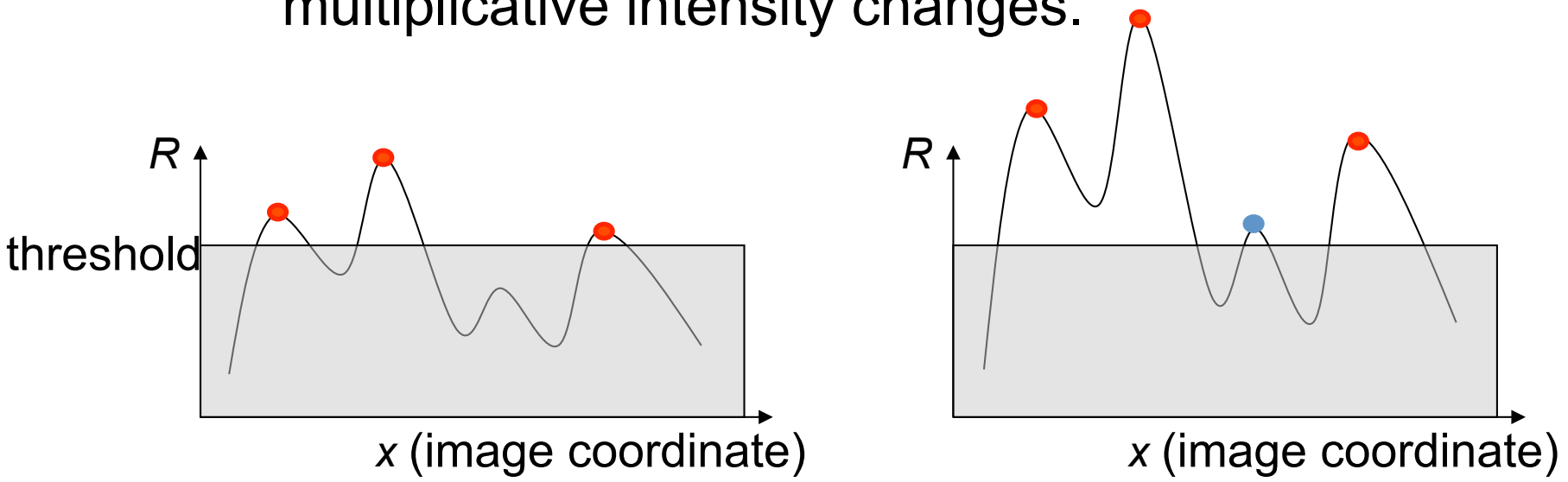


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

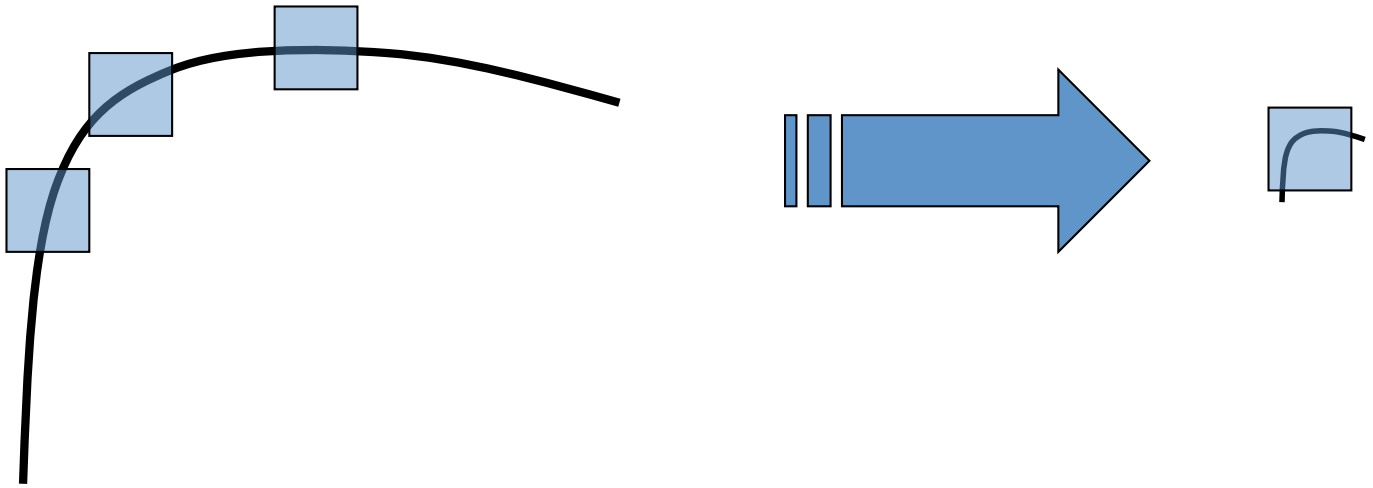
Harris Detector: Properties

- Partial invariance to additive and multiplicative intensity changes
  - Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
  - Intensity scale: $I \rightarrow a\, I$ Because of fixed intensity threshold on local maxima, only partial invariance to multiplicative intensity changes.



Harris Detector: Properties

- Not invariant to image scale



All points will be classified as edges

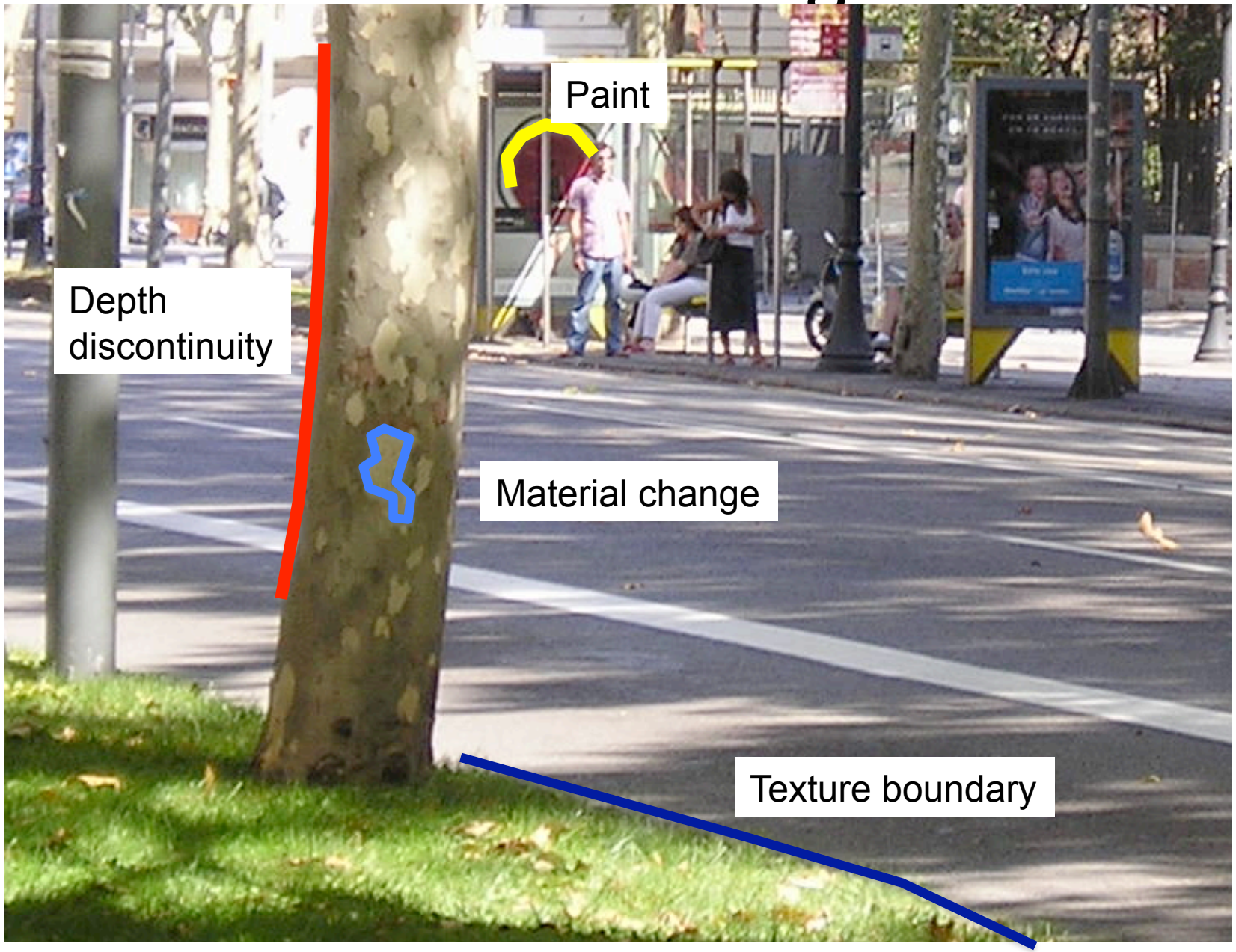
Corner !

EDGES

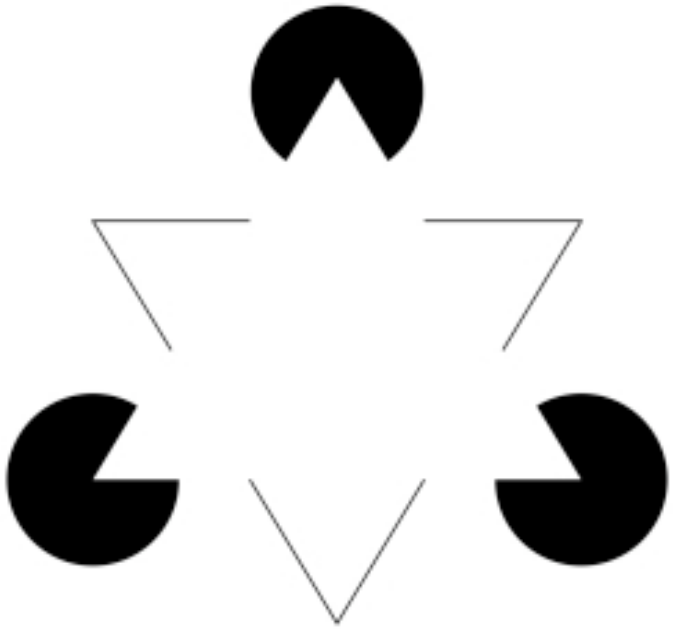
CANNY EDGE DETECTOR

What is an edge?

What is an edge?



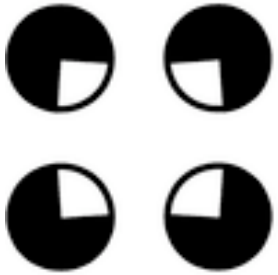
What is an edge?



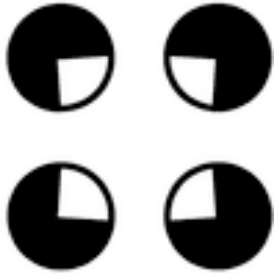
Kanizsa triangle

(a)

(b)



(c)



(d)

Edge detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels – can compress a lot of visual information by using just the edges.
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

Source: D. Lowe

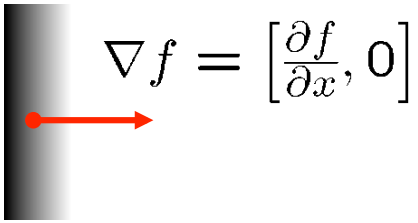
Gradients and edges (Forsyth, ch. 8)

- Points of sharp change in an image are interesting:
  - change in reflectance
  - change in object
  - change in illumination
  - noise
- Sometimes called **edge points**

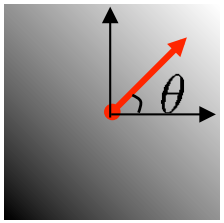
- General strategy
  - determine image gradient
  - now mark points where gradient magnitude is particularly large wrt neighbours (ideally, curves of such points).

Image gradient

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

-  $\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient points in the direction of most rapid increase in intensity

- How does this direction relate to the direction of the edge?

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

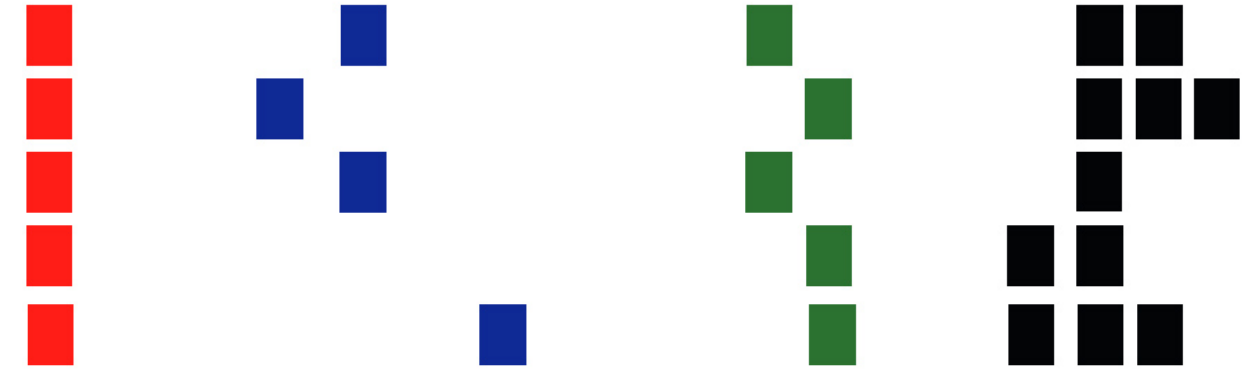
The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Designing an edge detector

- Criteria for an "optimal" edge detector:
  - **Good detection:** the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges)
  - **Good localization:** the edges detected must be as close as possible to the true edges
  - **Single response:** the detector must return one point only for each true edge point; that is, minimize the number of local maxima around the true edge



Source: L. Fei-Fei

Canny edge detector

- This is probably the most widely used edge detector in computer vision.
- Theoretical model: step-edges corrupted by additive Gaussian noise.
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization.
- MATLAB: edge(image, 'canny')

J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

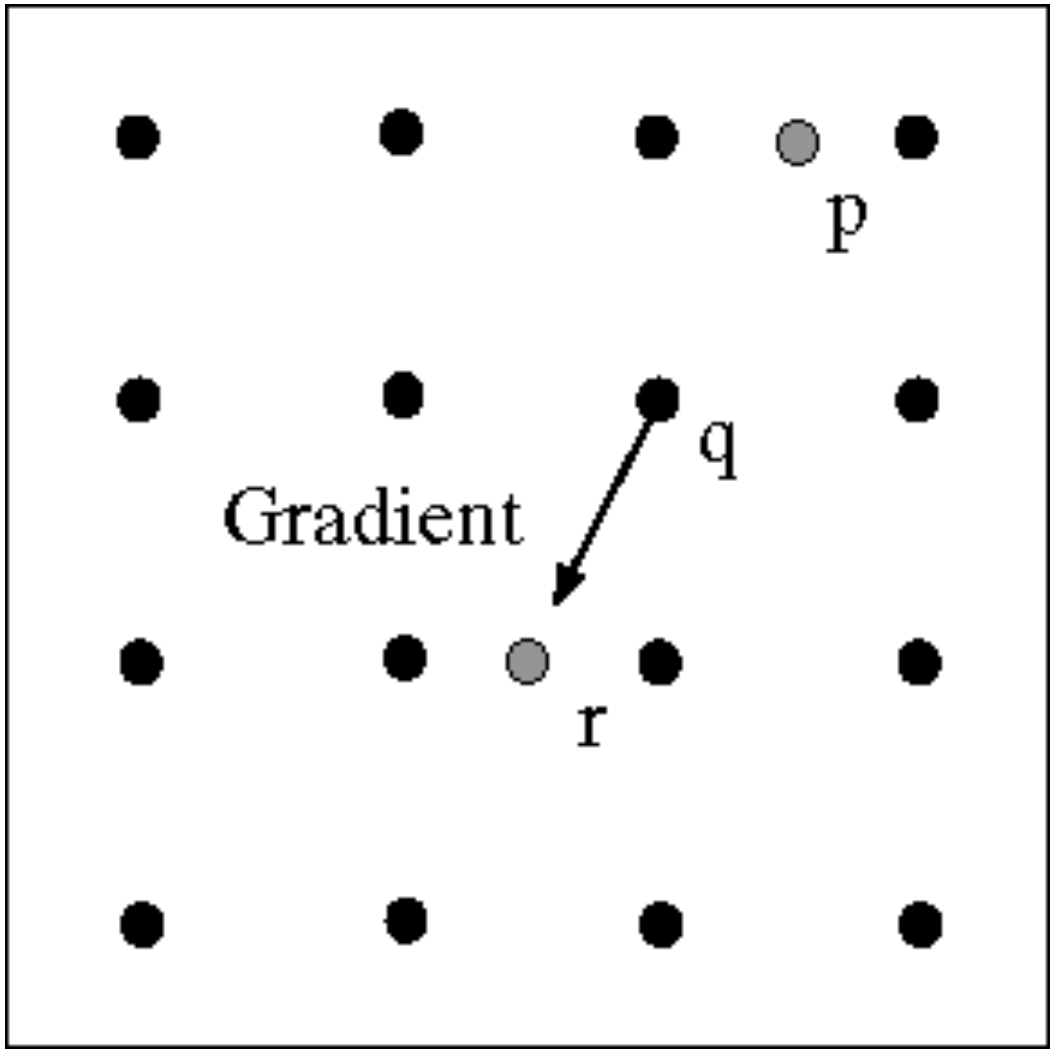
Source: L. Fei-Fei

Canny edge detector

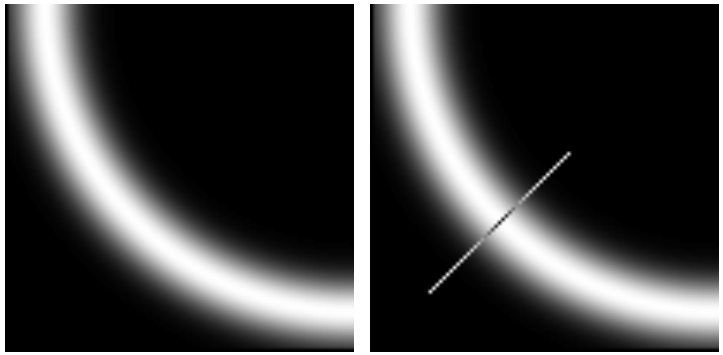
1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” down to single pixel width

Source: D. Lowe, L. Fei-Fei

Non-maximum suppression



At q, we have a maximum if the value is larger than those at both p and at r.
(Interpolate to get these values.)



Source: D. Forsyth

Example

- original image (Lena, 1972)

Example

Norm (magnitude) of the gradient

Example

thresholding

Example

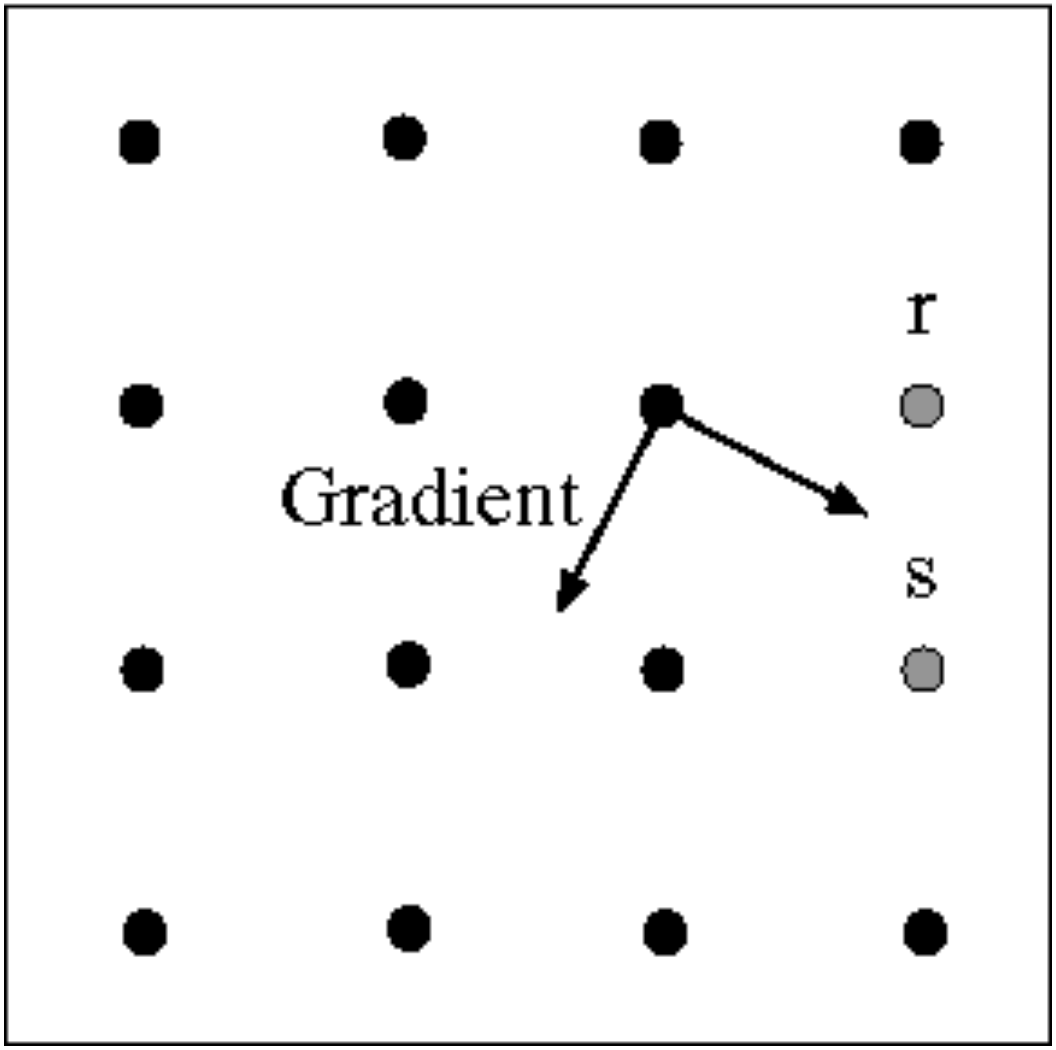
Non-maximum suppression

Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points

Source: D. Lowe, L. Fei-Fei

Edge linking



Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).

Source: D. Forsyth

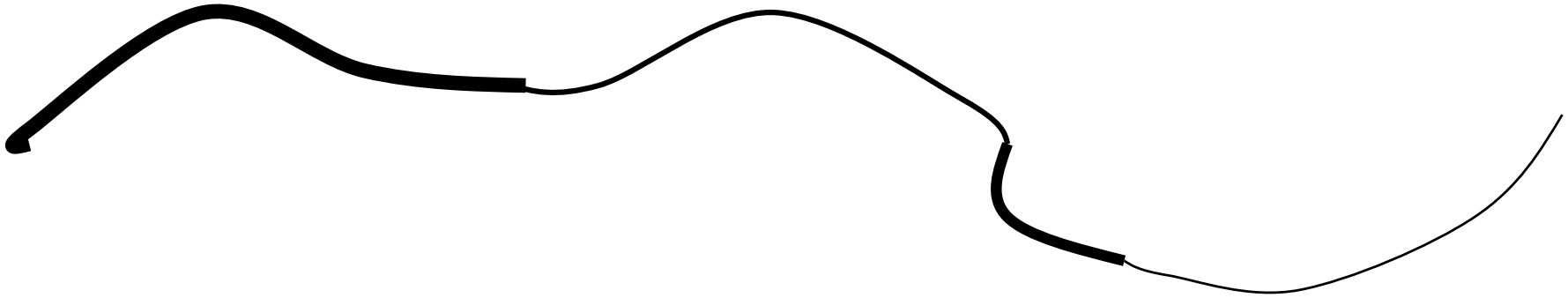
Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points
   - Hysteresis thresholding: use a higher threshold to start edge curves and a lower threshold to continue them

Source: D. Lowe, L. Fei-Fei

Hysteresis thresholding

- Use a high threshold to start edge curves and a low threshold to continue them
  - Reduces *drop-outs*



Source: S. Seitz

Hysteresis thresholding

original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Source: L. Fei-Fei

Effect of σ (Gaussian kernel spread/size)

original Canny with $\sigma = 1$ Canny with $\sigma = 2$

The choice of σ depends on desired behavior

- large σ detects large scale edges
- small σ detects fine features

Source: S. Seitz

Auto-correlation surfaces