EEE-6512: Image Processing and Computer Vision

November 5, 2018
Lecture #11: Features/Feature Extraction II
Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

Outline

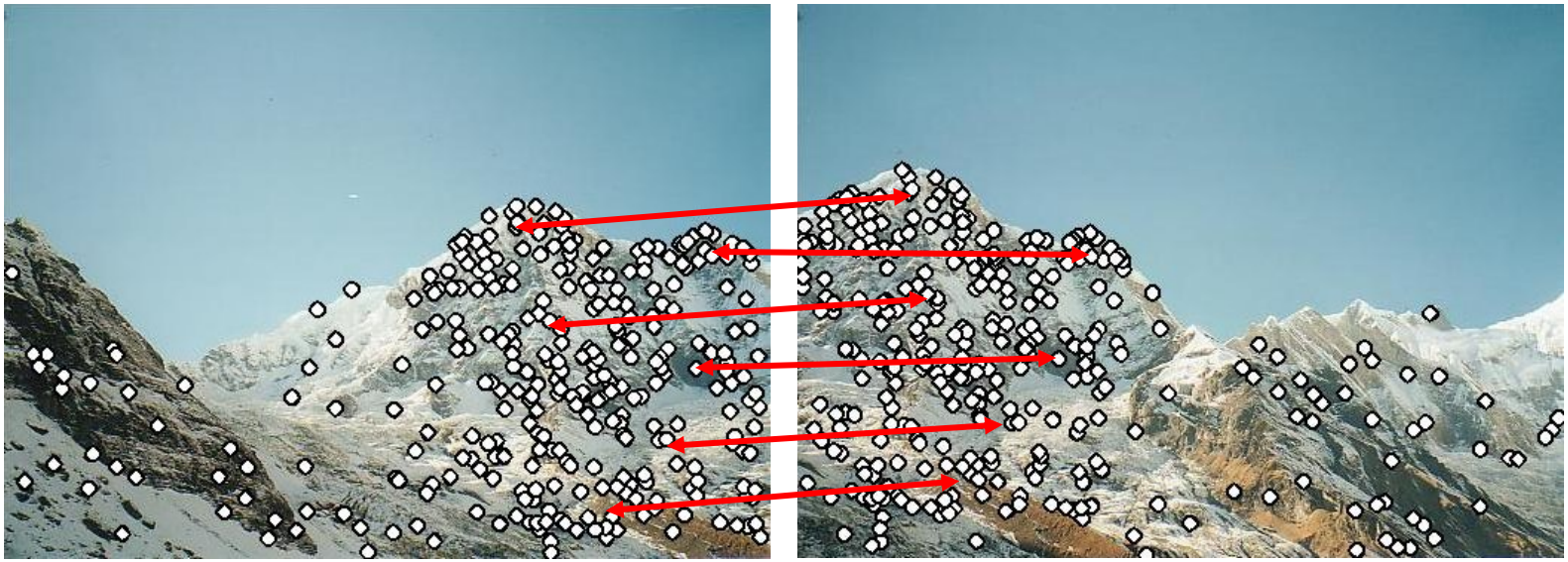
- Local Features
  - Motivation
  - Applications
  - Advantages
- Corners as Keypoints
  - Moravec Corner Detector
  - Harris Corner Detector
- Scale Invariant Feature Transform (SIFT) Descriptor

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?



Step 1: extract features
Step 2: match features

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
Step 3: align images

Image matching

by Diva Sian

by swashford

Local Feature Advantages

Locality

Features are local, so robust to occlusion and clutter

Distinctiveness

Can differentiate a large database of objects

Quantity

Hundreds or thousands in an image

Efficiency

Real-time performance achievable

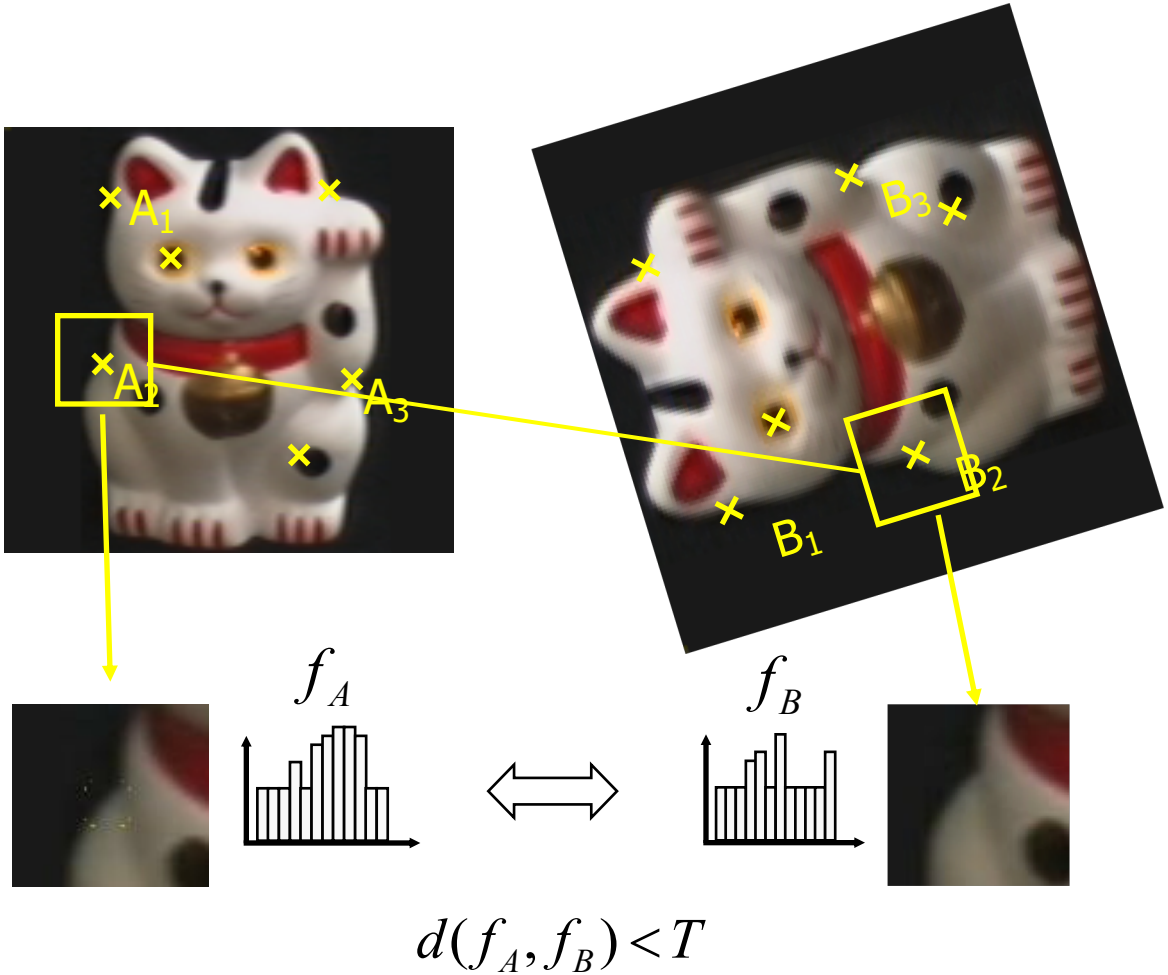
Generality

Exploit different types of features in different situations

Applications

- Keypoints are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition

Overview of Keypoint 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 a local descriptor from the normalized region
5. Match local descriptors

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Key trade-offs

Detection

More Repeatable

Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

Minimize wrong matches

More Flexible

Robust to expected variations
Maximize correct matches

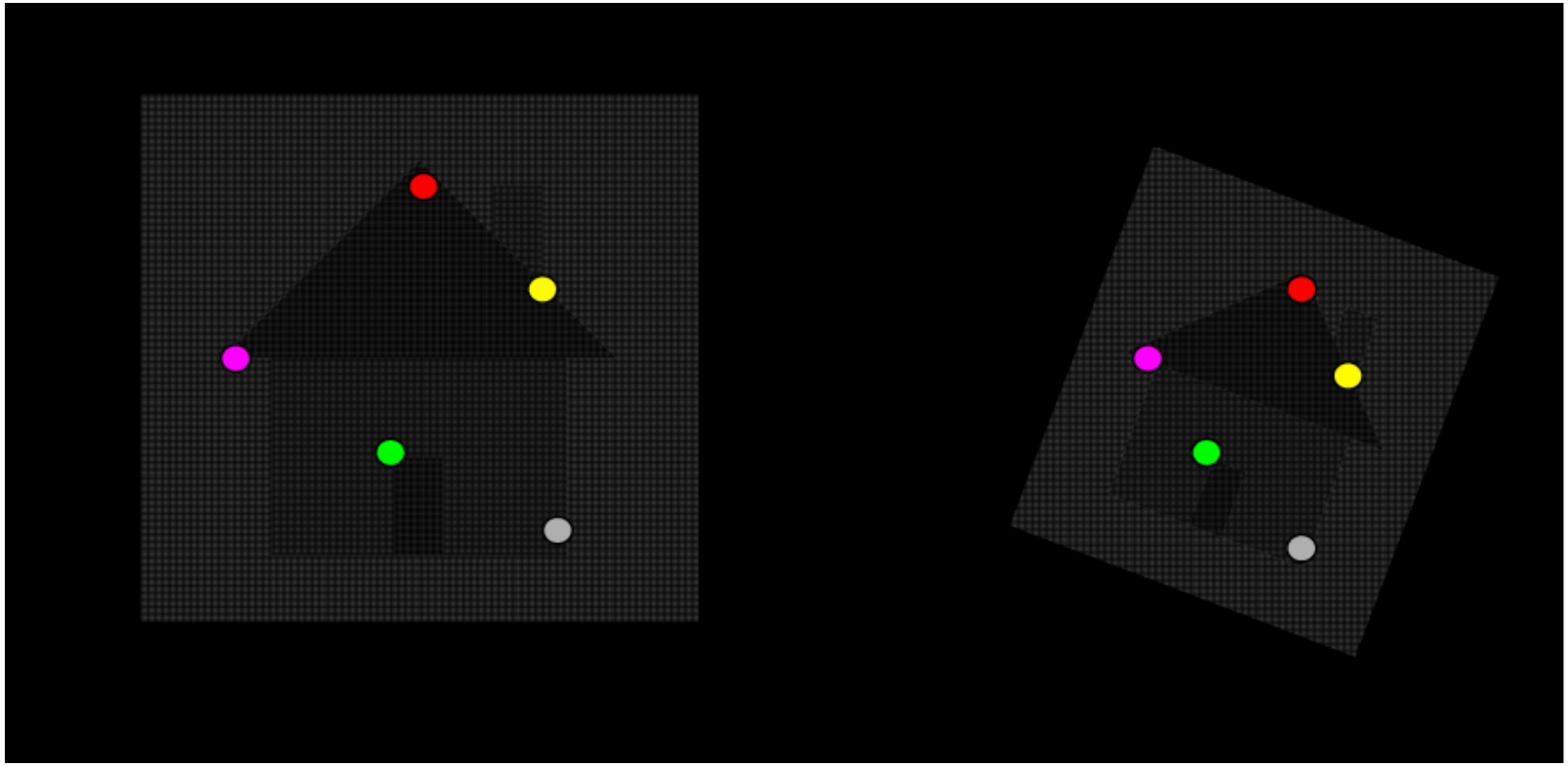
Corners as Keypoint Features

Interest Points

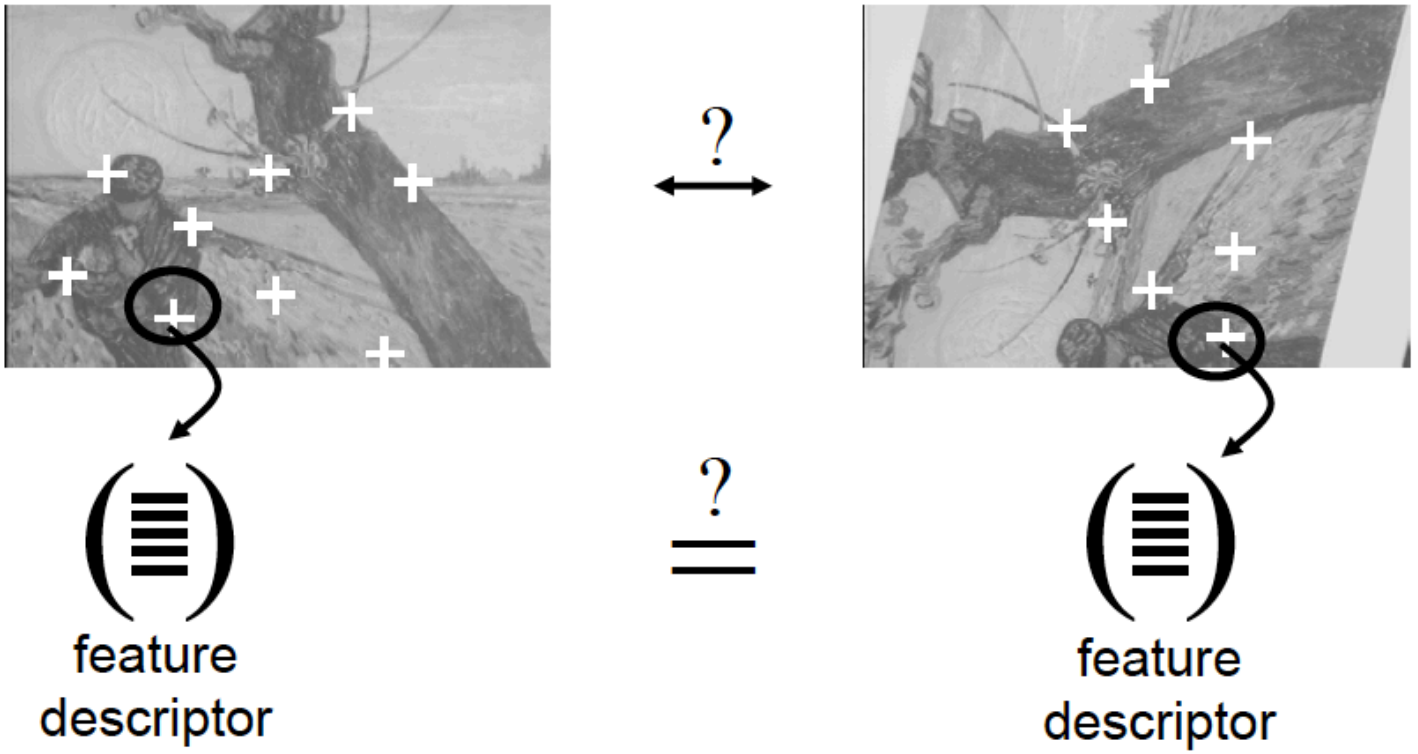
- Interest points are local features associated with the significant change of an image property or several properties simultaneously (intensity, color, texture)

Why extract interest points?

- Corresponding points or features between images enable the estimation of parameters describing geometric transformation between the images.



What if we don't know the correspondences?

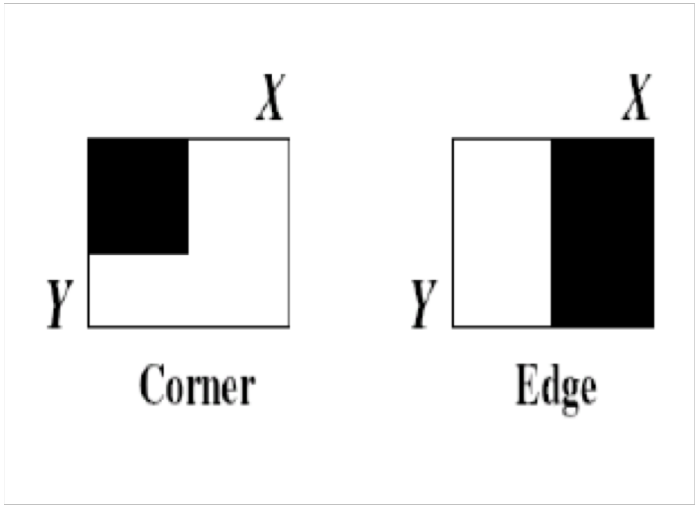


Need to compare feature descriptors of local patches around interest points

Corners vs. Edges

- **Corners** are locations where variations of intensity *f(x,y)* in both X and Y are high
  - Both partial derivatives f_x and f_y are large

- **Edges** are locations where variation of *f(x,y)* in certain direction is high, while variation in the orthogonal direction is low.
  - When edge is oriented along Y, f_x is large and f_y small

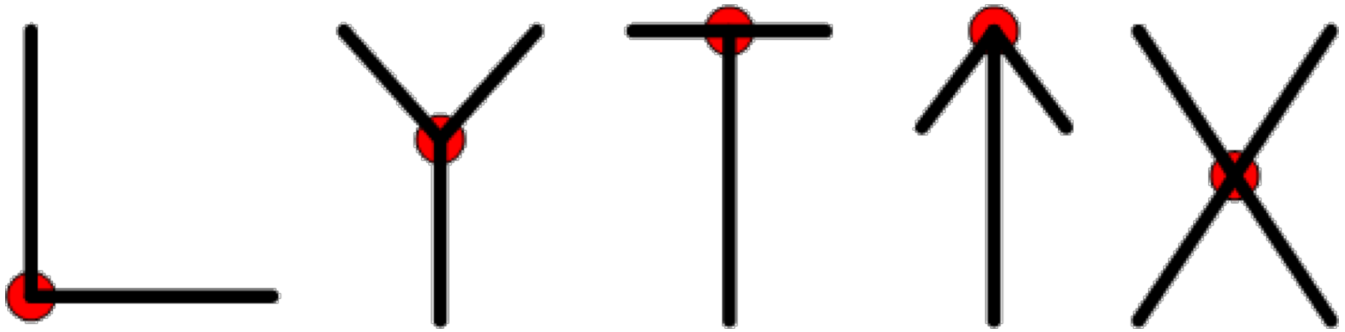


Corners in Images

- Corner is sharp turn of contour
- Corners are used in shape analysis and motion analysis
- They are dominant in human perception of 2D shapes
- Two types of corner detection:
  - In digital curves (assumes extracted contours)
  - In digital images (does not assume extracted contours)
- We only consider corner detection in grayscale images

Corner

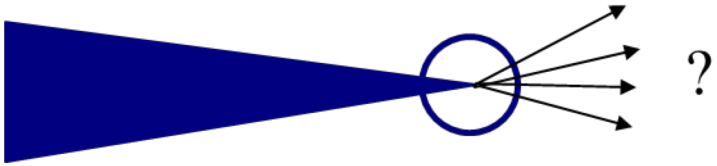
- A corner can be defined as the intersection of two or more edges.



- Examples of corner types include L-junctions, Y-junctions, T-junction, Arrow-junctions, and X-junctions.

Corners as Keypoint Features

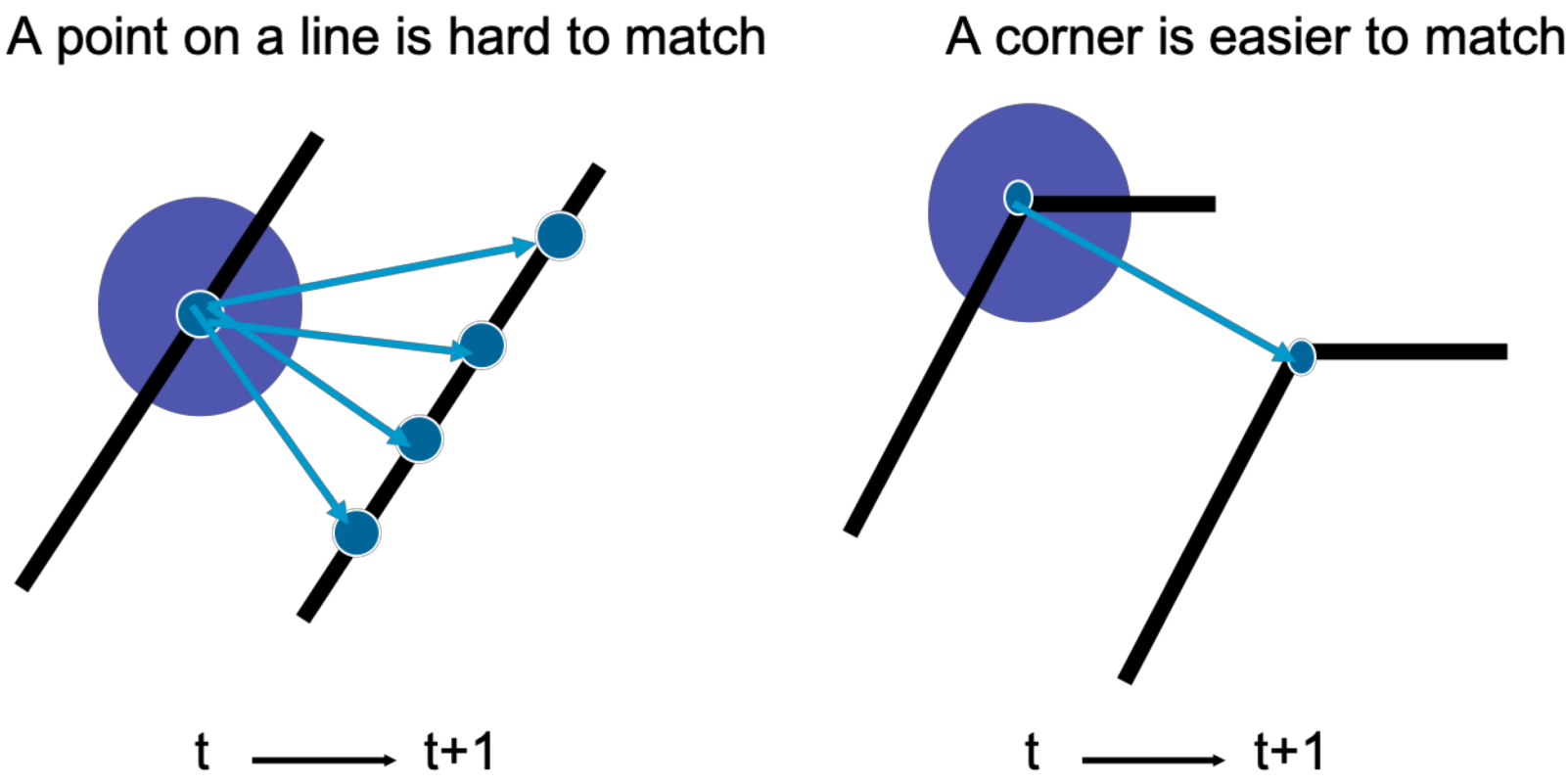
- Challenges
  - Gradient computation is less reliable near a corner due to ambiguity of edge orientation



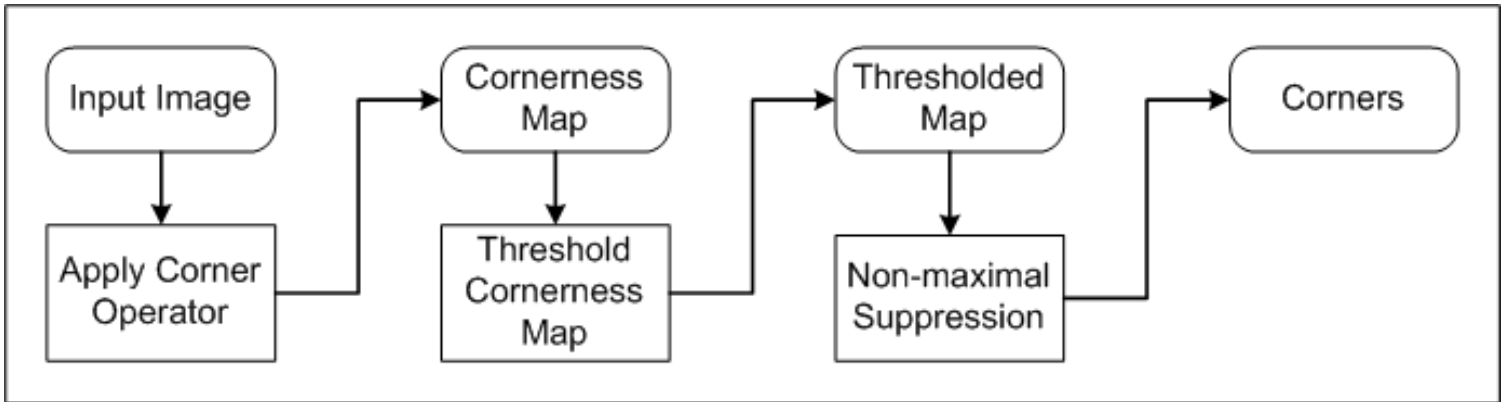
  - Corner detector are usually not very robust. This deficiency is overcome either by manual intervention or large redundancies.

Aperture Problem

- In a motion sensor has a finite receptive field, it perceives the through something resembling an aperture, making the the motion of a homogeneous contour ambiguous.



Steps in Corner Detection

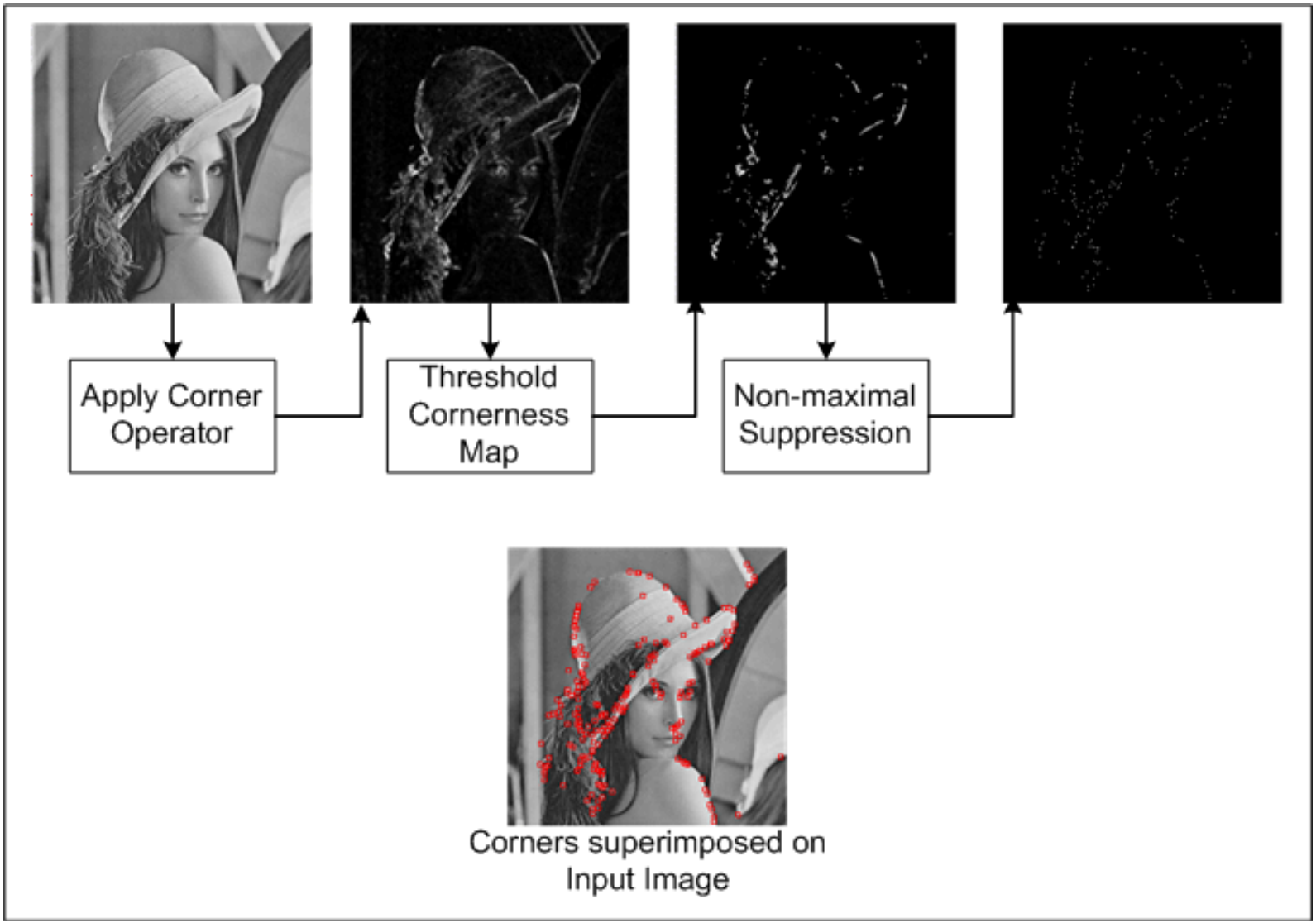


1. For each pixel, the corner operator is applied to obtain a **cornerness** measure for the pixel.

2. Threshold **cornerness** map to eliminate weak corners.

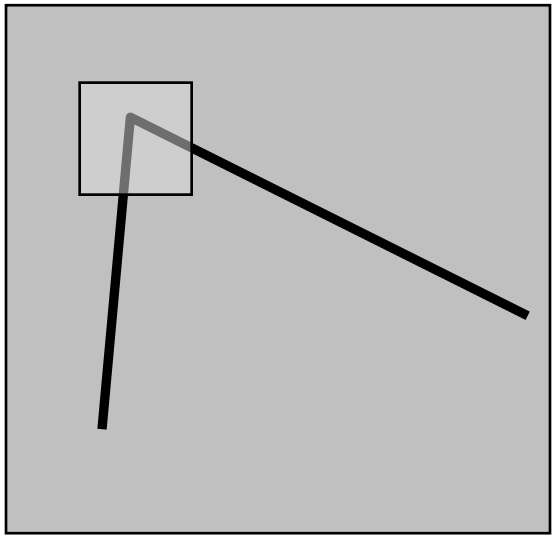
3. Apply non-maximal suppression to eliminate points whose **cornerness** measure is not larger than the **cornerness** measure values of all points within a certain distance.

Steps in Corner Detection (cont.)

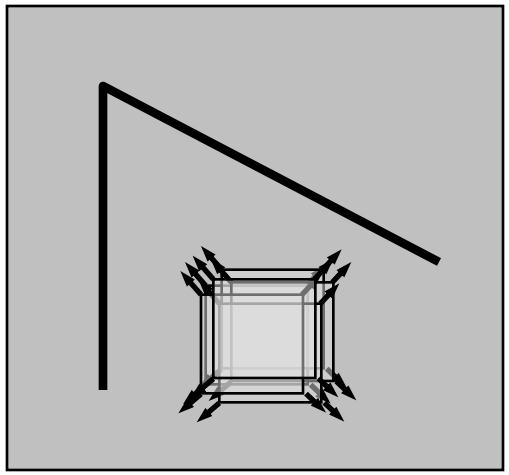


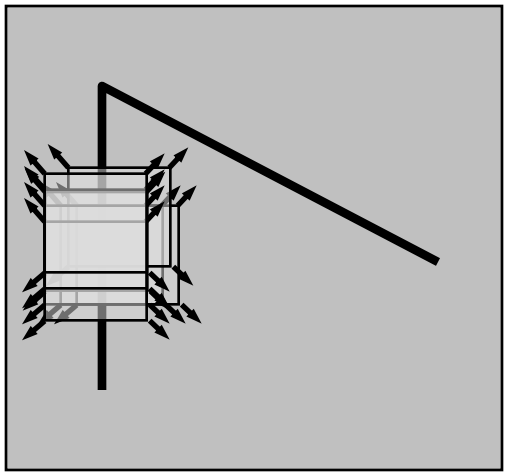
The Basic Idea

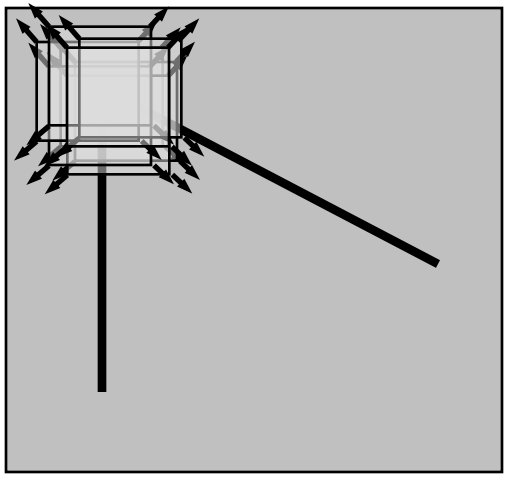
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Basic Idea







"flat" region: no change in all directions

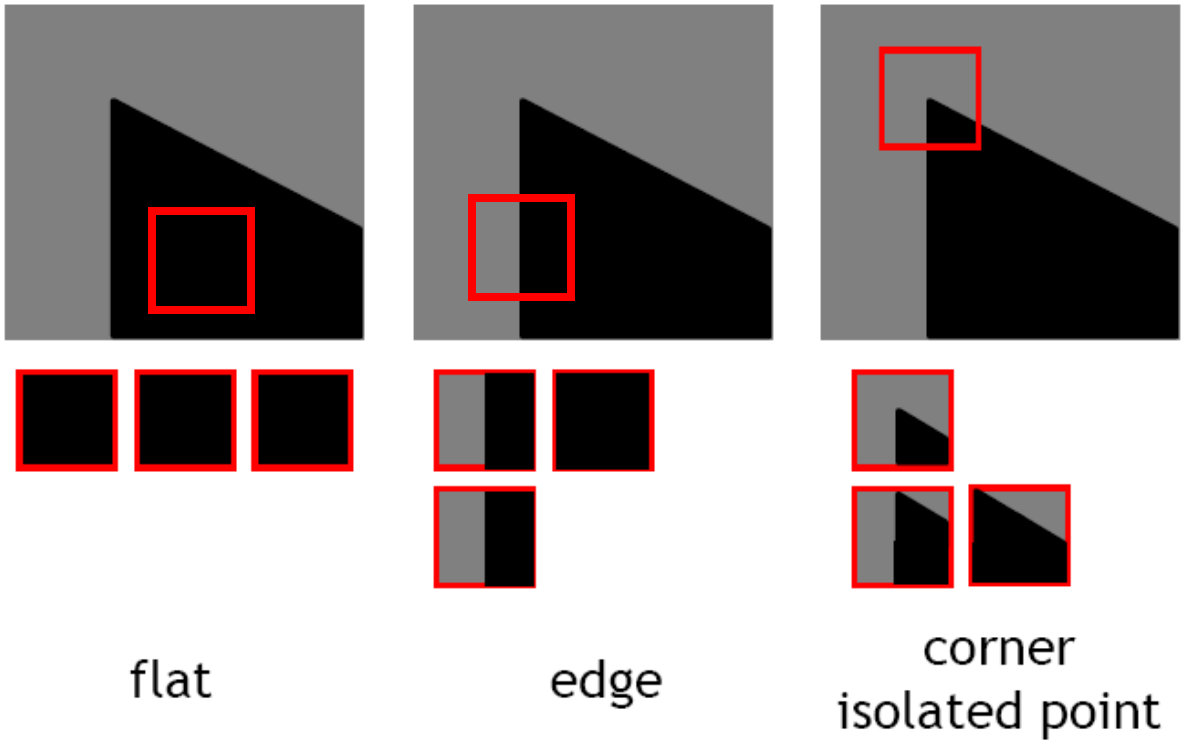
"edge": no change along the edge direction

"corner": significant change in all directions

Moravec Corner Detector

Moravec Corner Detector

- Shift in any direction would result in a significant change at a corner.



Algorithm:
•Shift in horizontal, vertical, and diagonal directions by one pixel.
•Calculate the absolute value of the MSE for each shift.
•Take the minimum as the cornerness response.

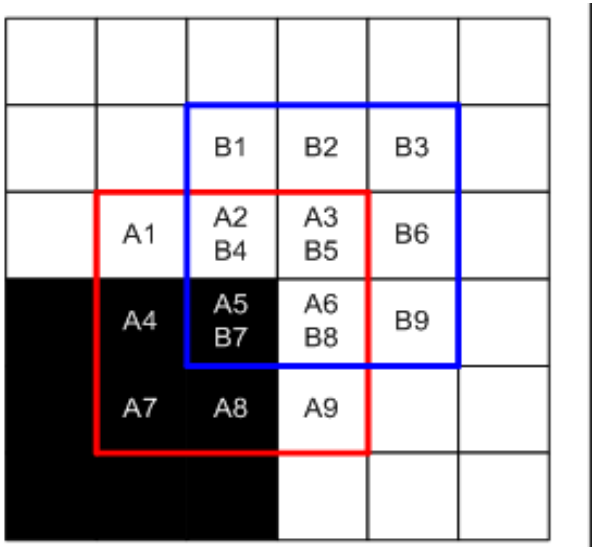
Moravec Detector

- **Idea:** measure intensity variation at (x,y) by shifting a small window (3x3 or 5x5) by one pixel in each of the **eight** principal directions (horizontally, vertically, and four diagonals).

Moravec Detector

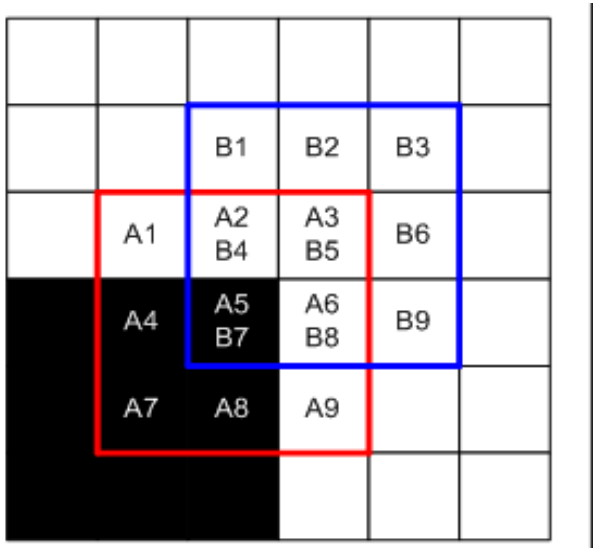
- The intensity variation S_w in a given direction (Δx, Δy) can be calculated as follows:

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y) \right)^2 .$$



Moravec Detector

- Moravec's detector calculates S_w in all 8 directions:



$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y) \right)^2 .$$

$\Delta x, \Delta y$ in {-1,0,1}

$S_W(-1,-1), S_W(-1,0), \ldots, S_W(1,1)$

Moravec Detector

- The "cornerness" of a pixel is the minimum intensity variation found over the eight shift directions:

  Cornerness(x,y) = min{$S_W(-1,-1)$, $S_W(-1,0)$, ...$S_W(1,1)$}

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Cornerness Map (normalized)

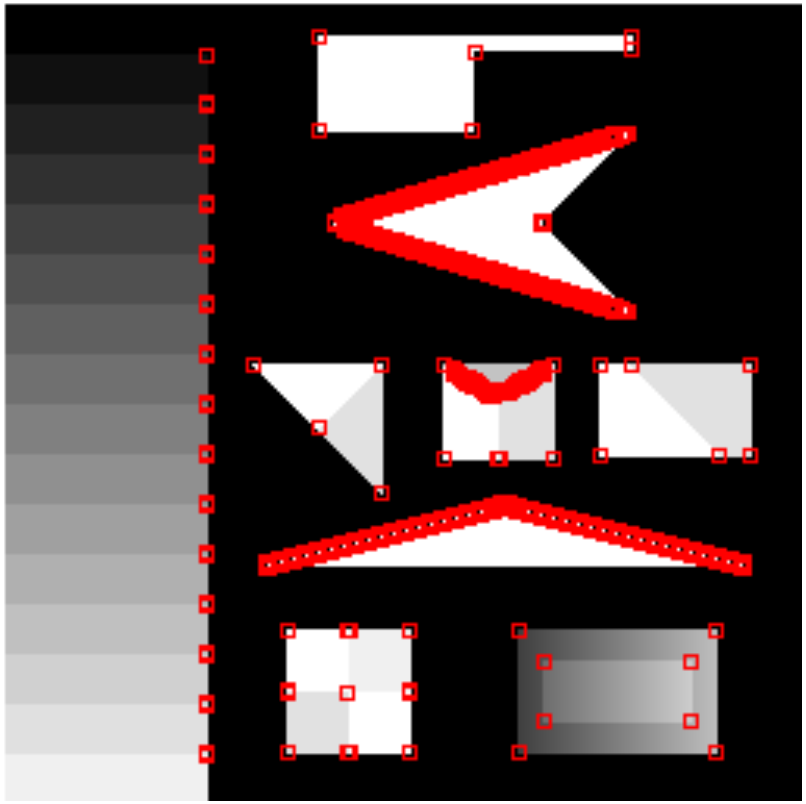
Note response to isolated points!

Moravec Detector

- Non-maximal suppression will yield final corners

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

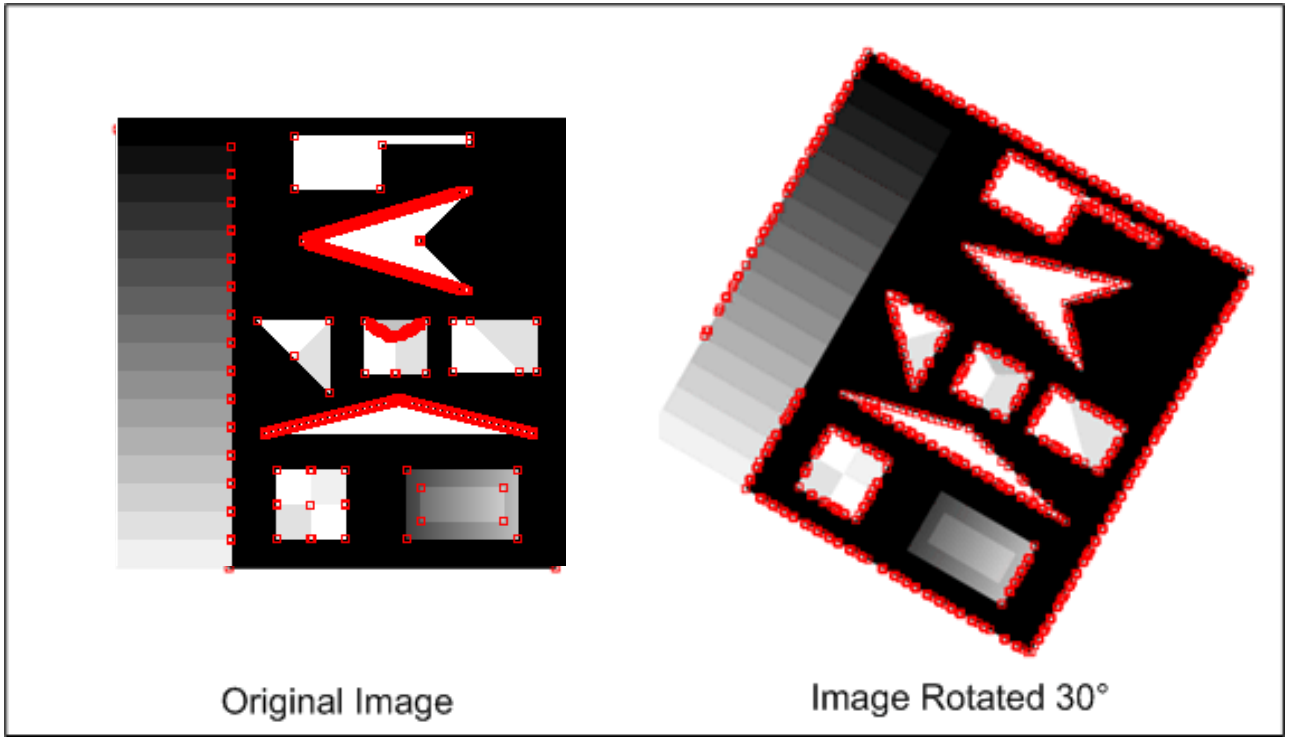
Moravec Detector



- Does a reasonable job in finding the majority of true corners.
- Edge points not in one of the eight principal directions will be assigned a relatively large cornerness value.

Moravec Detector

The response is anisotropic as the intensity variation is only calculated at a discrete set of directions and shifts (i.e., not rotationally invariant)



Harris Corner Detector

Harris Corner Detector

- Originally developed as features for motion tracking
- Improves the Moravec operator by avoiding the use of discrete directions and shifts.
- Uses a Gaussian window instead of a square window.
- Unlike Moravec detector, Harris detector is rotationally invariant.

Harris Detector: Mathematics

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

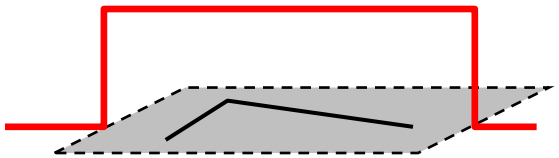
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

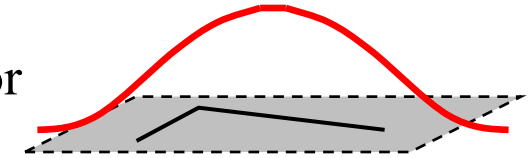
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector: Mathematics

For small shifts $[u,v]$ we have a *bilinear* approximation:

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

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

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

Describes the gradient distribution (i.e. local structure) in window.

Intuitive Way to Understand Harris Detector

1. Treat gradient vectors as a set of points with the center of mass defined as being at (0,0).

2. Fit an ellipse to that set of points via scatter matrix

3. Analysis ellipse parameters for varying cases

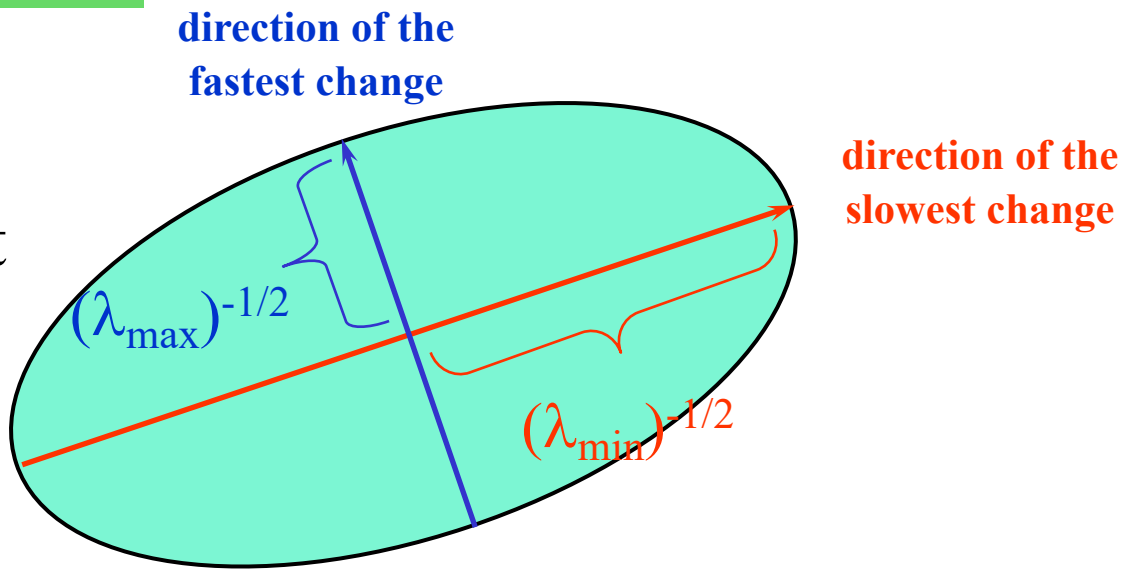
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

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

λ_1, λ_2 – eigenvalues of M

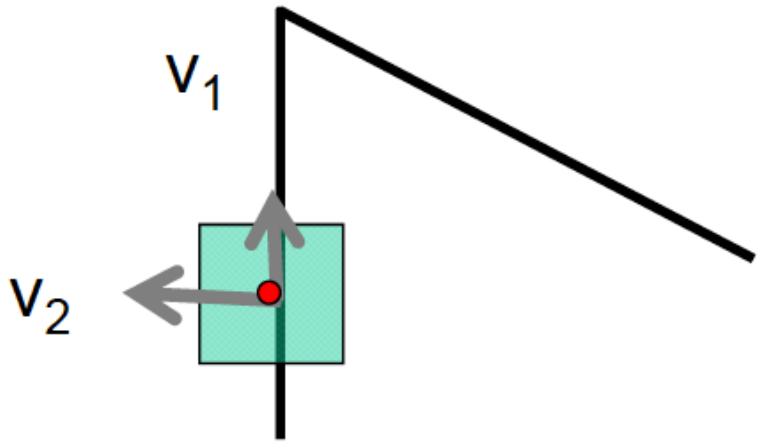
Ellipse $E(u,v) = \text{const}$



Harris Detector: Mathematics

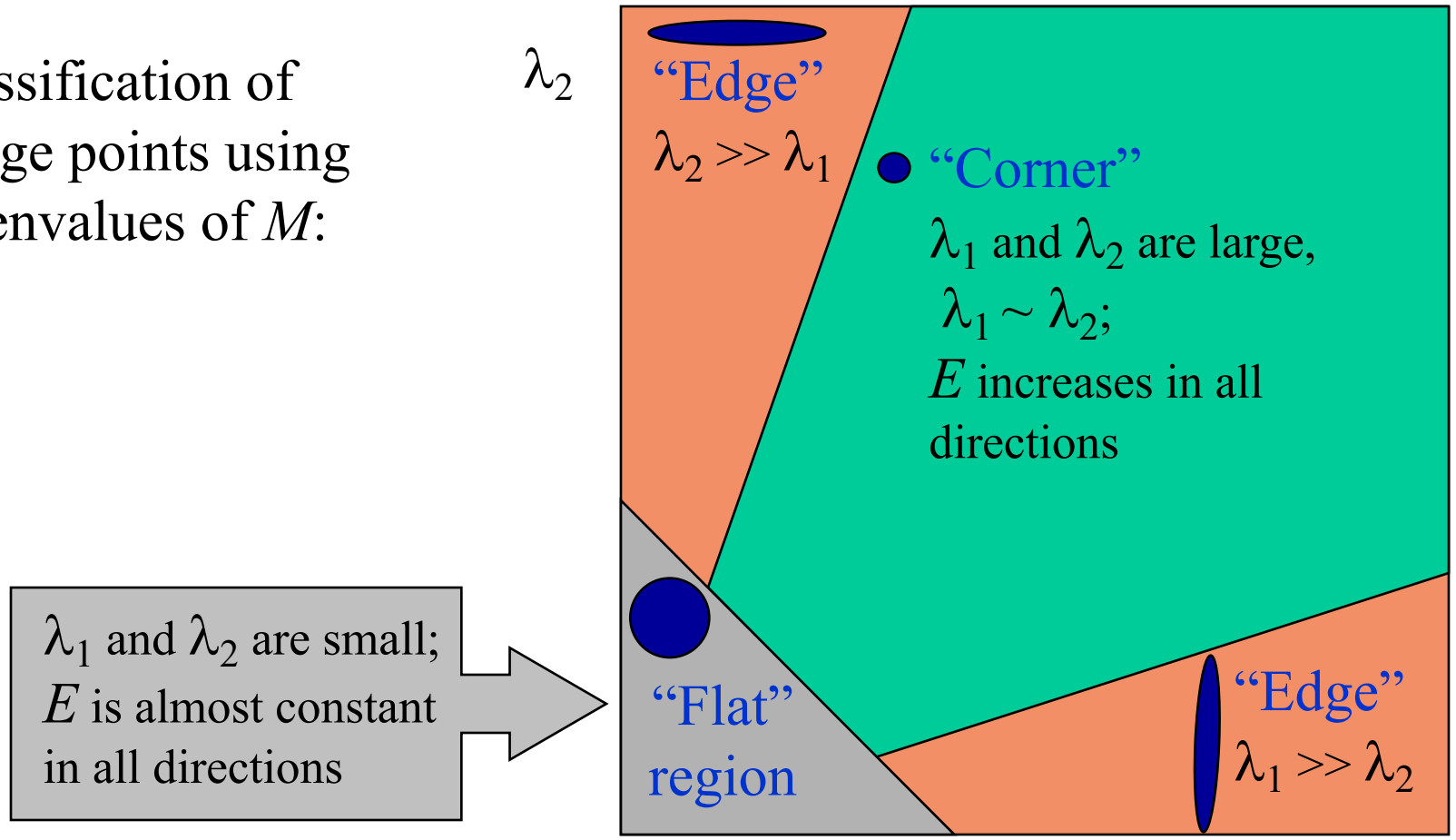
The eigenvectors of M **encode** the direction of intensity change

The eigenvalues of M encode the **strength** of intensity change.



Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector: Mathematics

Measure of corner response:

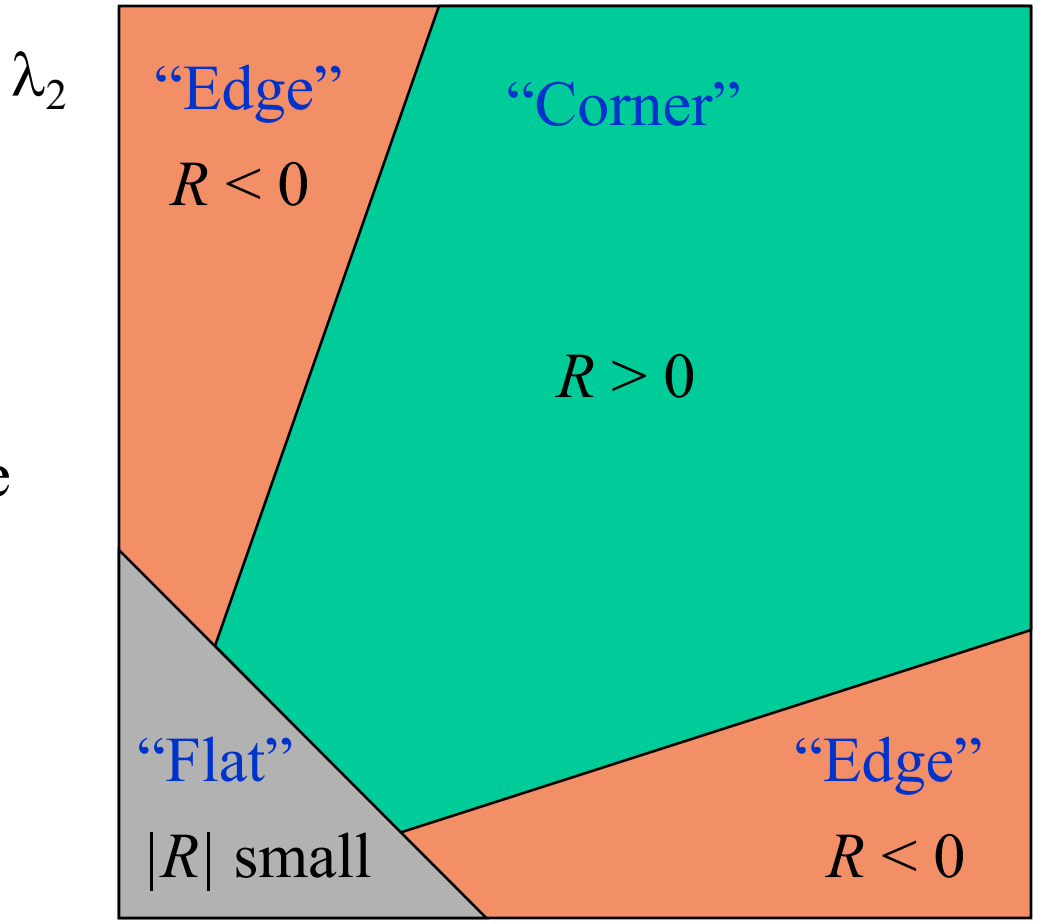
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

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

(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 Detector

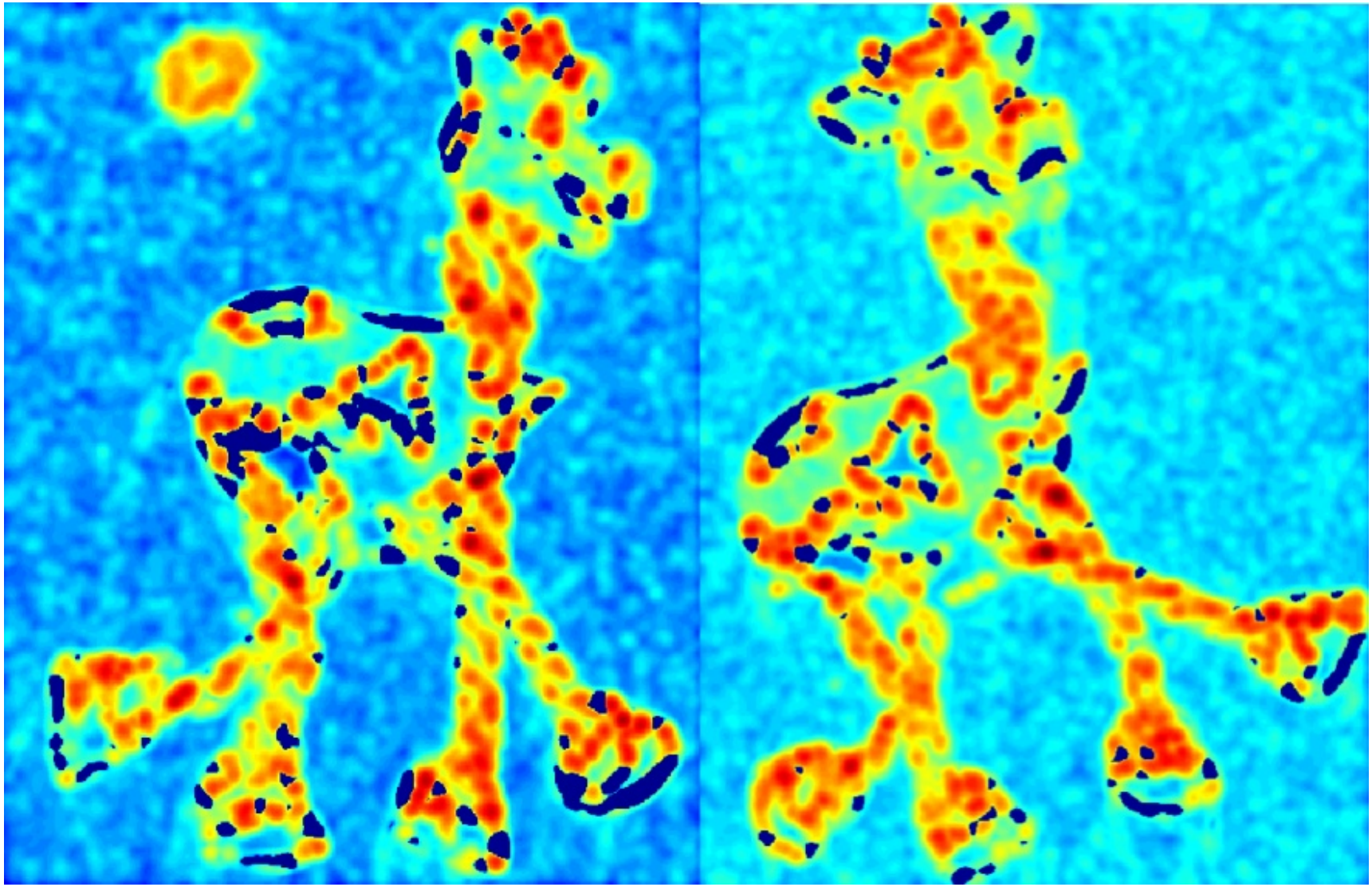
The Algorithm:

- Find points with large corner response function R (R > threshold)
- Take the points of local maxima of R

Harris Detector: Workflow

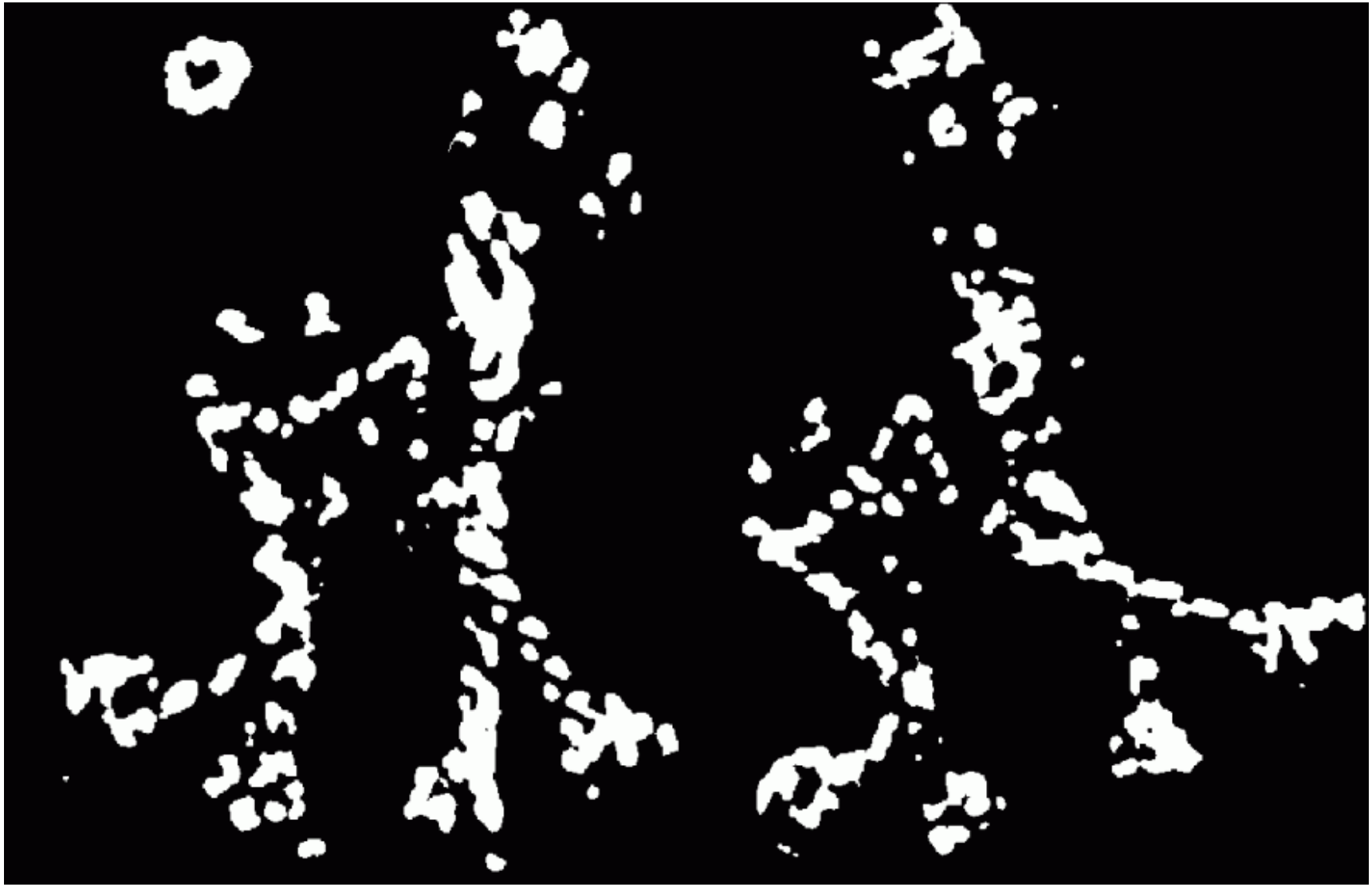
Harris Detector: Workflow

Compute corner response R



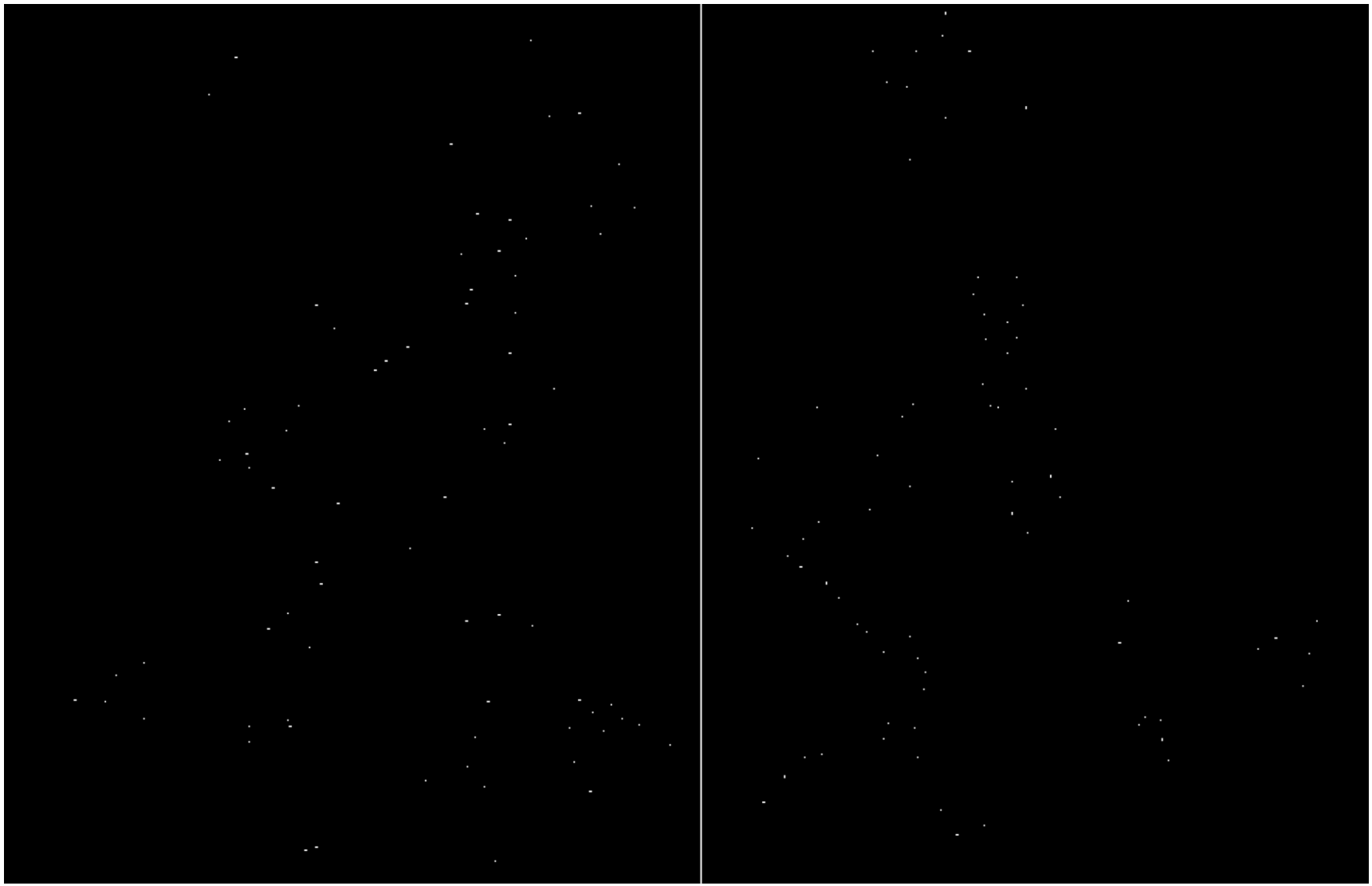
Harris Detector: Workflow

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: Summary

Average intensity change in direction [*u,v*] can be expressed as a bilinear form:

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

Describe a point in terms of eigenvalues of *M*:

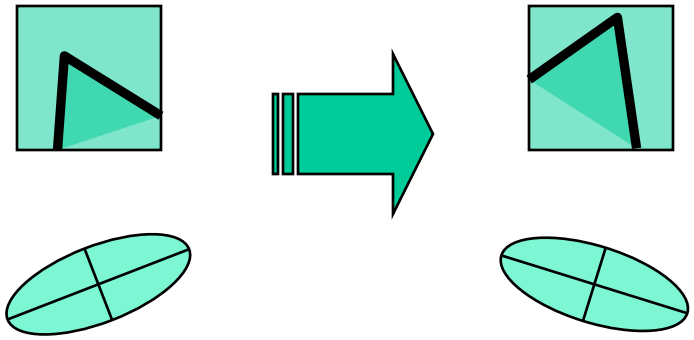
measure of corner response

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

A good (corner) point should have a *large intensity change* in *all directions*, i.e. *R* should be large positive

Harris Detector: Some Properties

Rotation invariance

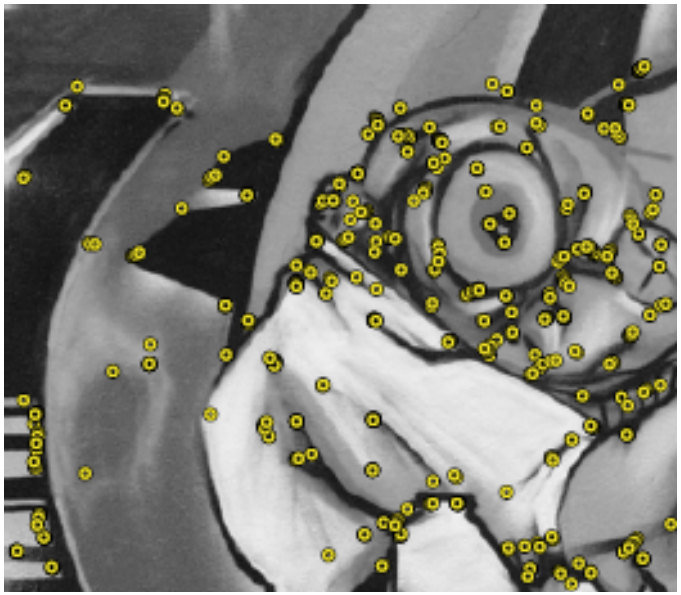


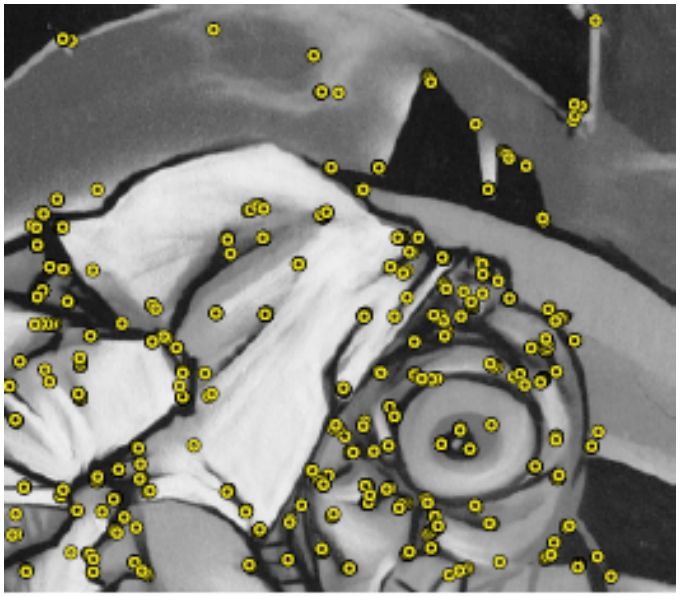
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris Detector: Some Properties

- 2D shift invariant.
- Invariant to rotation



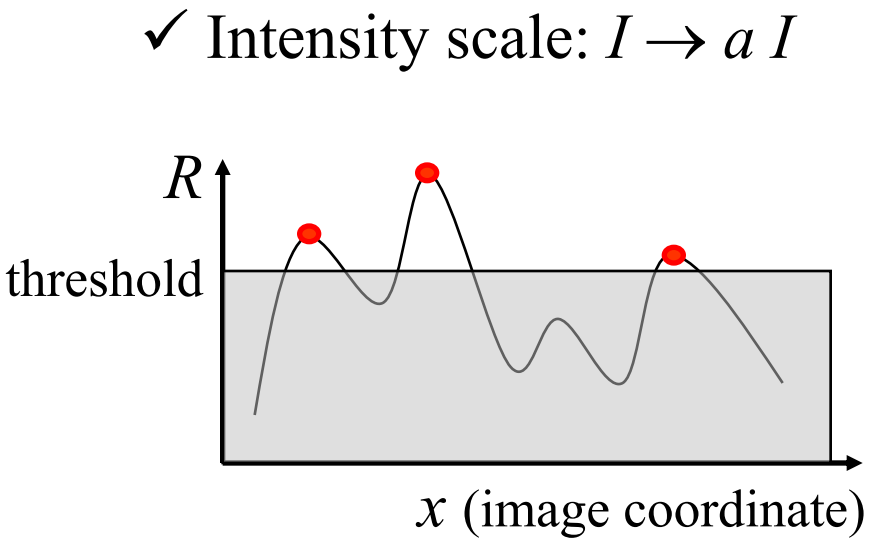


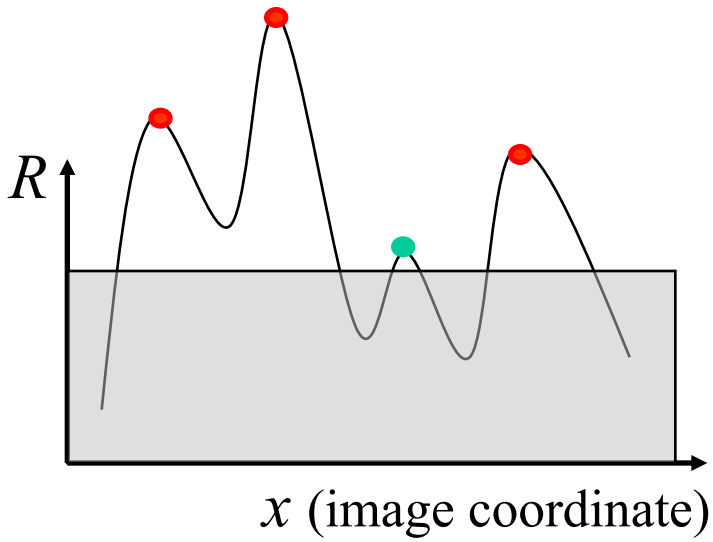
Harris Detector: Some Properties

Partial invariance to *affine intensity* change

✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

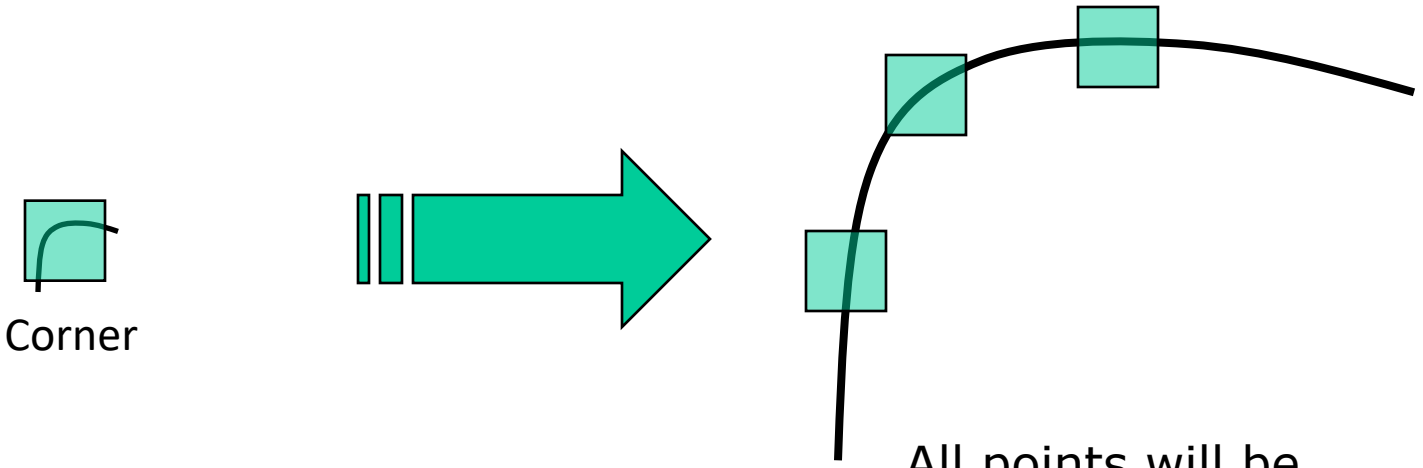
✓ Intensity scale: $I \rightarrow a\, I$





Harris Detector: Invariance Properties

Scaling

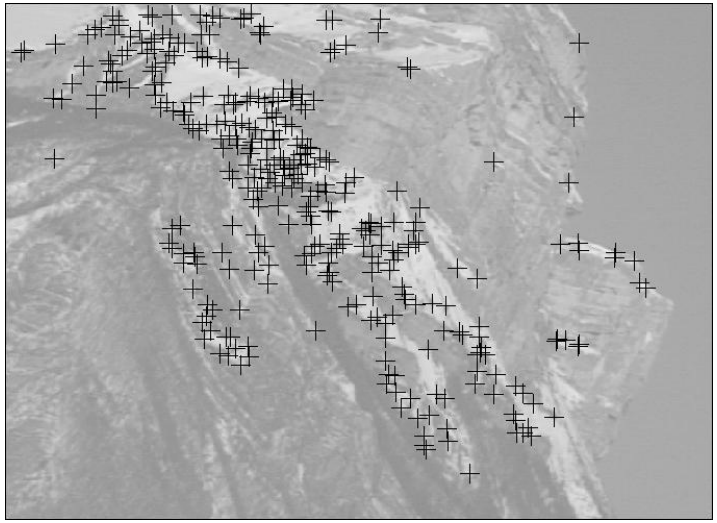


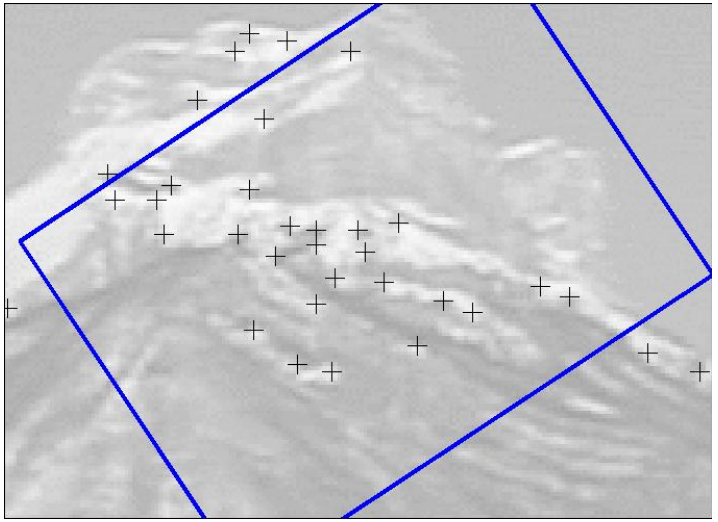
Not invariant to scaling

Harris Detector: Disadvantages

Sensitive to:

- Scale changes
- Significant view point changes (orientation/position of camera)
- Significant contrast changes





Summary of the Harris detector

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x.I_x \quad I_{y2} = I_y.I_y \quad I_{xy} = I_x.I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma\prime} * I_{x2} \quad S_{y2} = G_{\sigma\prime} * I_{y2} \quad S_{xy} = G_{\sigma\prime} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

6. Threshold on value of R. Compute nonmax suppression.

SIFT Feature Descriptor

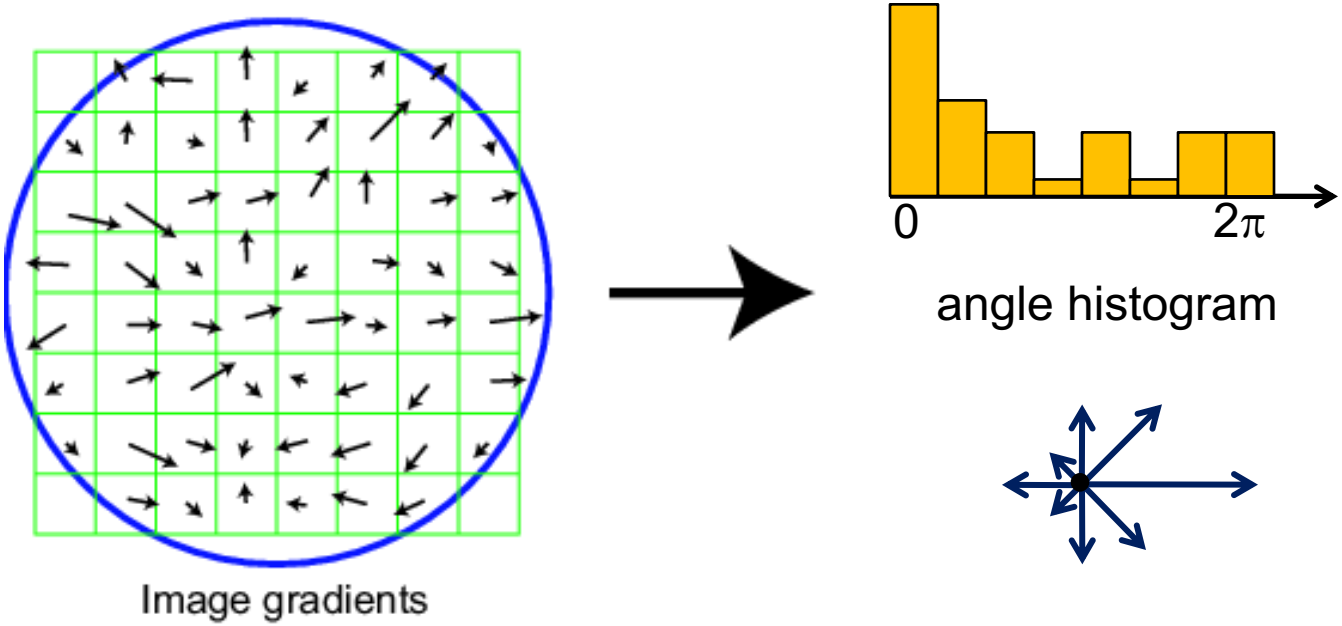
Local Descriptors

- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient
- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations



SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

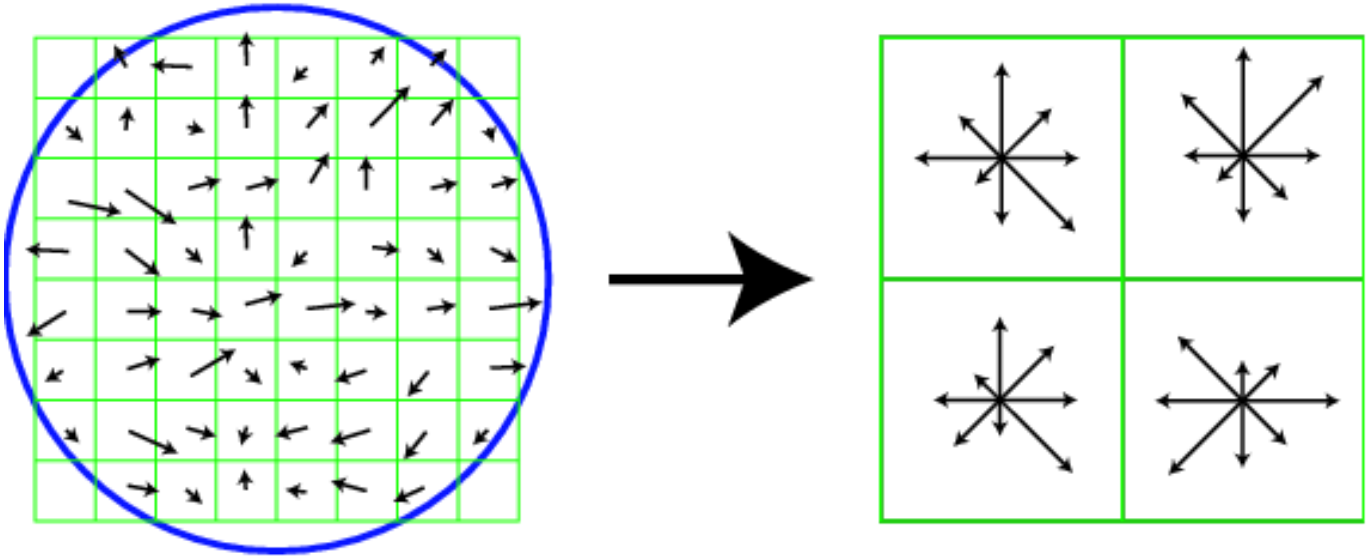
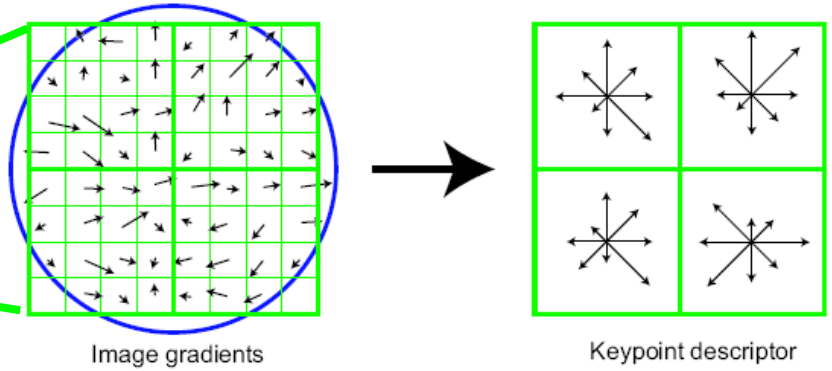


Image gradients

Keypoint descriptor

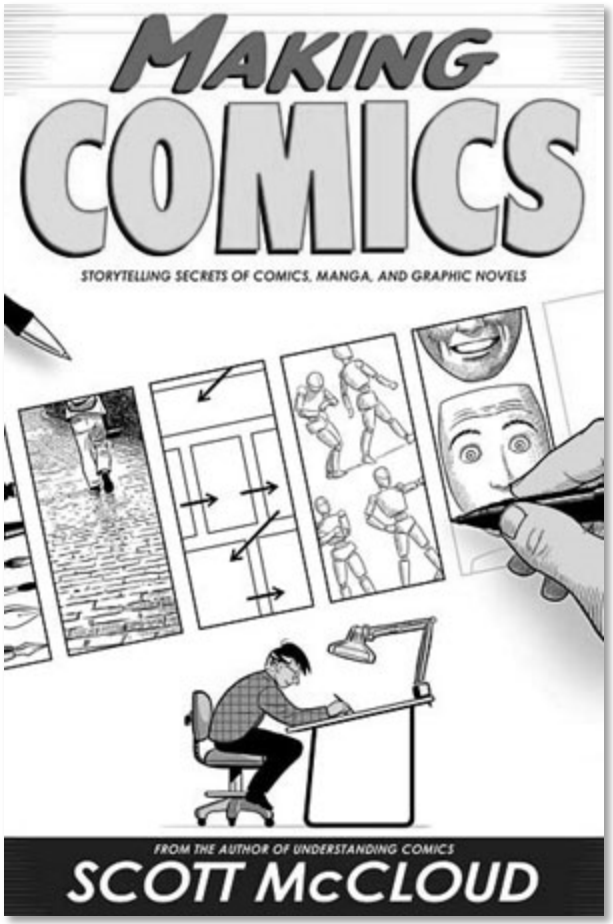
Local Descriptors: SIFT Descriptor



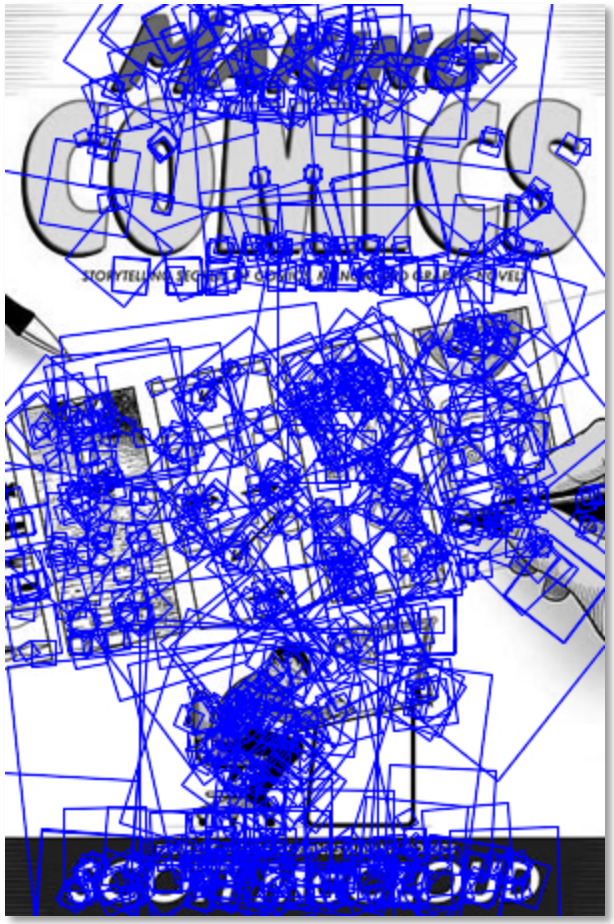
Histogram of oriented gradients

- Captures important texture information
- Robust to small translations / affine deformations

SIFT Example



sift



868 SIFT features

Feature matching

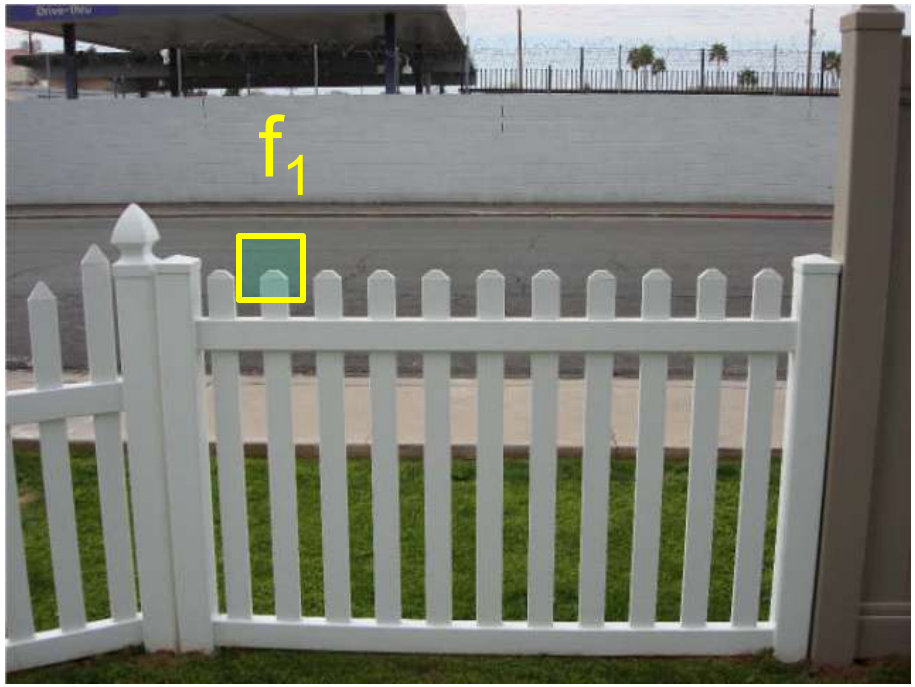
Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

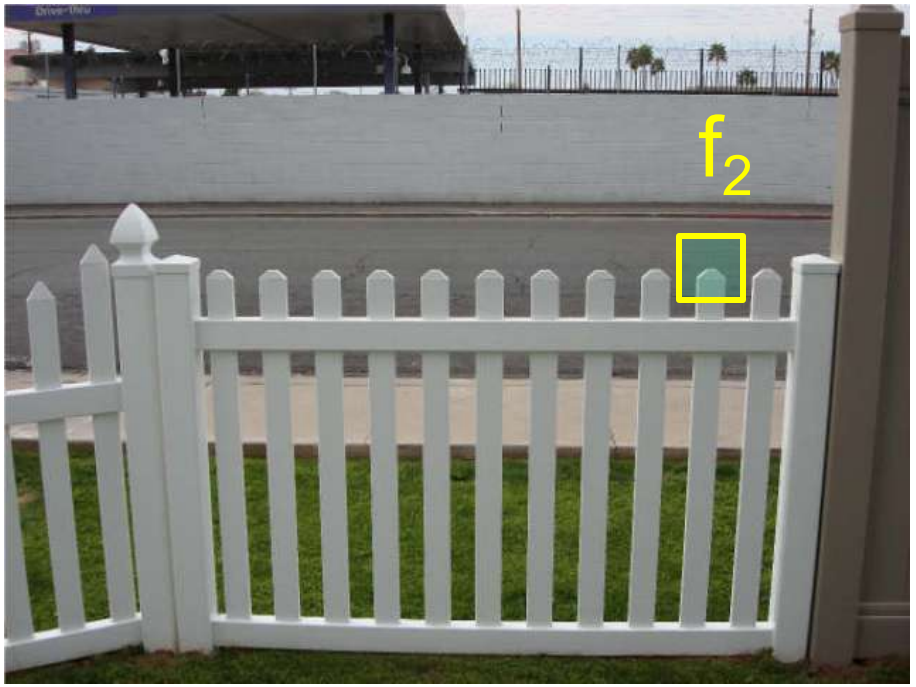
Feature distance

How to define the difference between two features f_1, f_2?

- Simple approach: L_2 distance, $||f_1 - f_2||$
- can give good scores to ambiguous (incorrect) matches



I_1

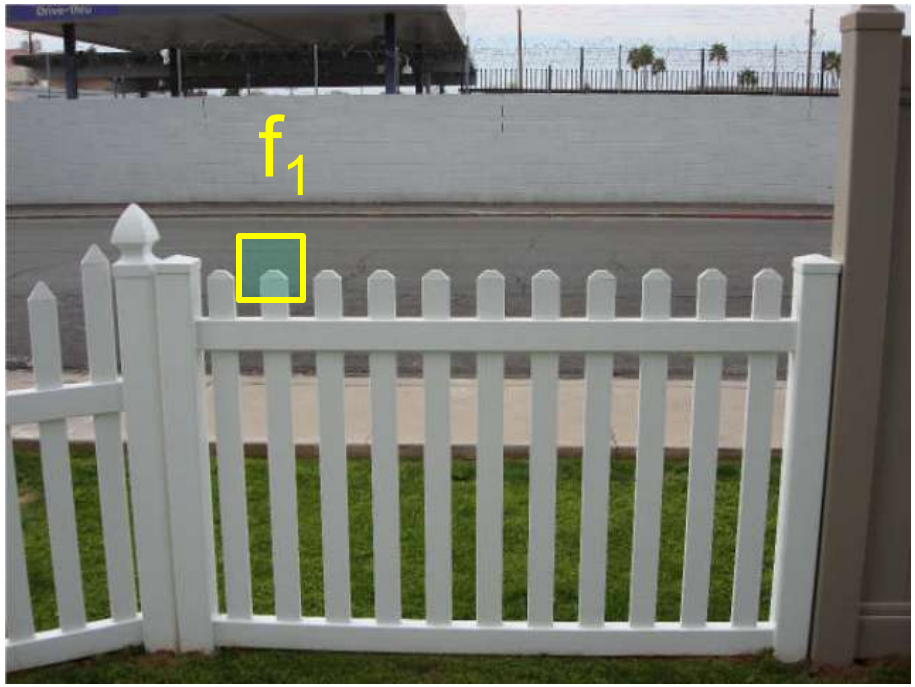


I_2

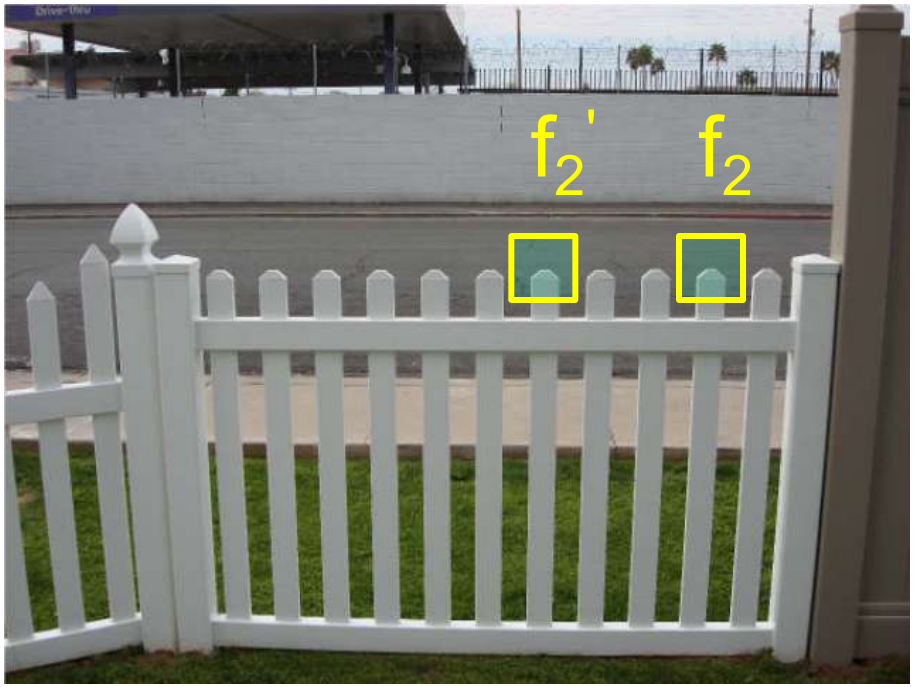
Feature distance

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $||f_1 - f_2|| / ||f_1 - f_2'||$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2nd best SSD match to f_1 in I_2
  - gives large values for ambiguous matches

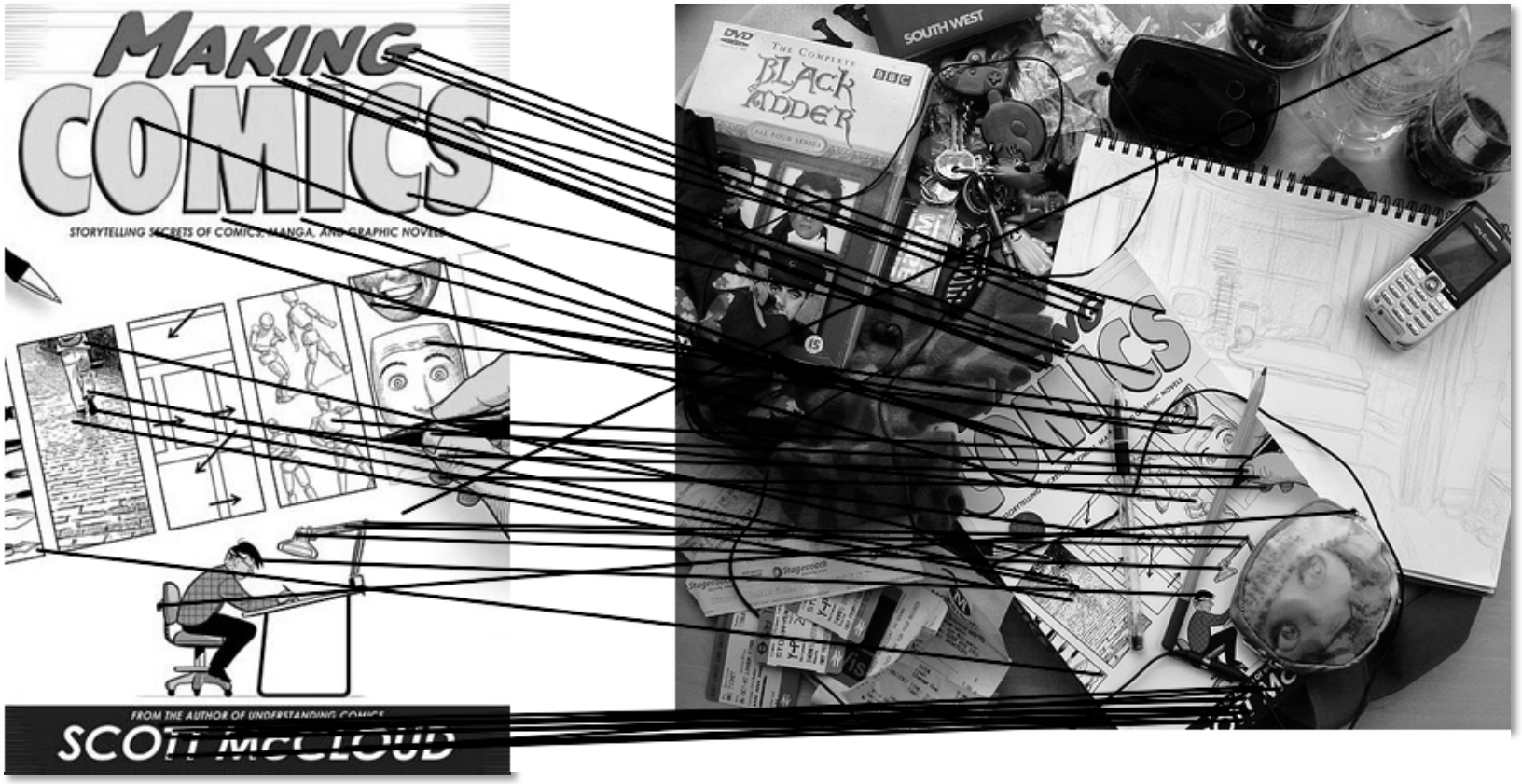


I_1



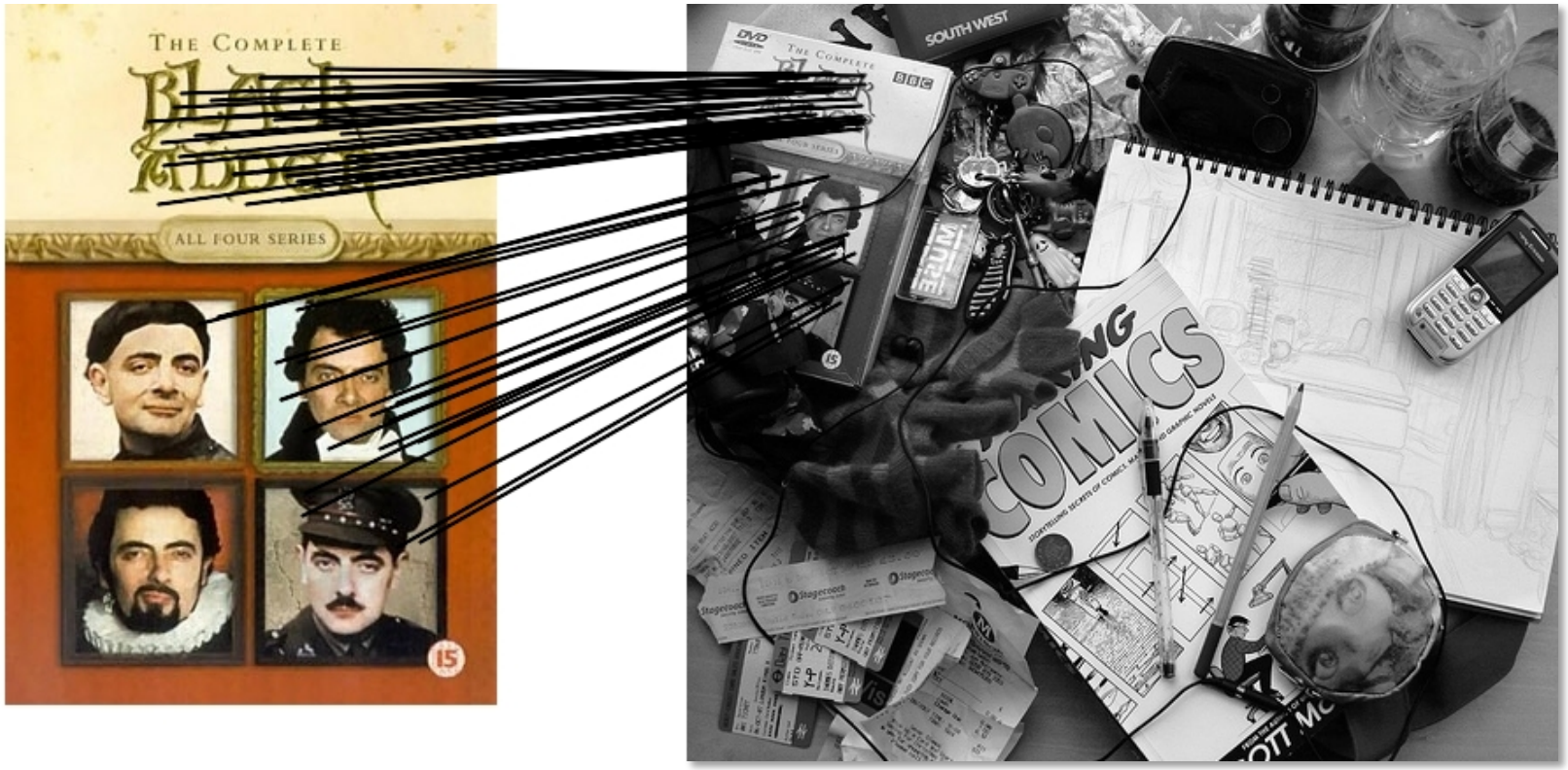
I_2

Feature matching example



51 matches

Feature matching example



58 matches

Summary / Things to Remember

- For keypoint detection the features should be repeatable and distinctive
- Precise localization in x-y for Harris Corner Detector
- Harris not invariant to scale changes
- For feature descriptors should be robust, distinctive, compact, and efficient.
- For object recognition or stitching, SIFT descriptor is good choice.

Questions?

Next Time: Model Fitting

Slide Credits

Images taken from Digital Image Processing by Gonzalez and Woods Text.

Material taken from Jen-Chang Liu lecture slides

Material taken from Saad Bedros lecture slides

Material taken from Jia-Bin Huang lecture slides

Material taken from David Lowe lecture slides