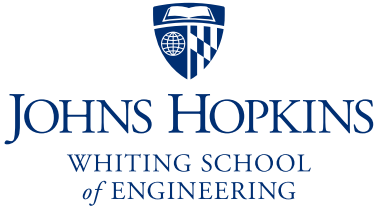
Johns Hopkins Engineering

Computer Vision

Corner Detection and Matching



Corner Detection and Matching

- Topics:
  - Motivating Feature Detection
  - Harris Corner Detector Theory
  - Matching With Patches As Features

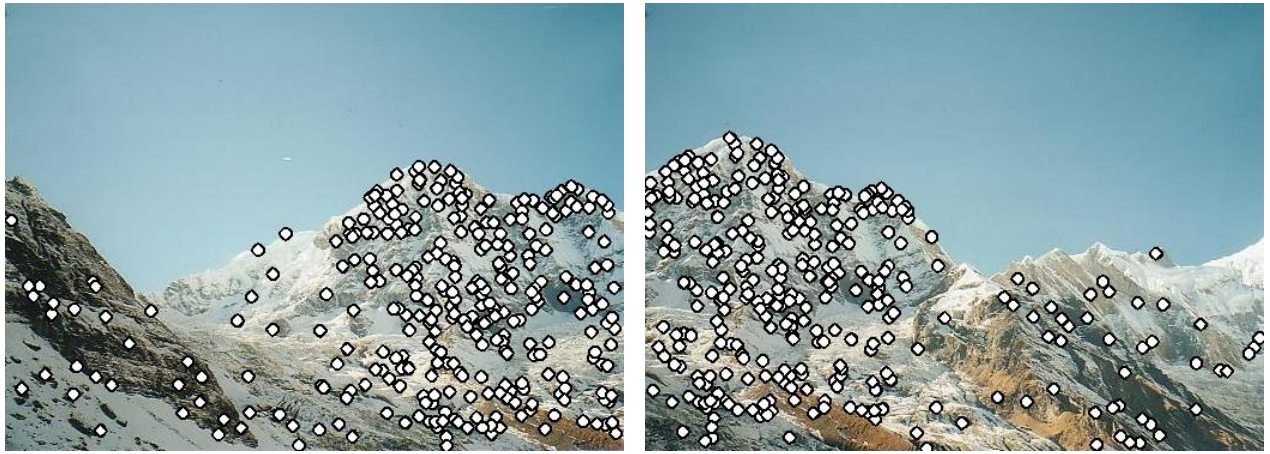
Motivation: Build a Panorama

How do we build a panorama?

- We need to match (align) images

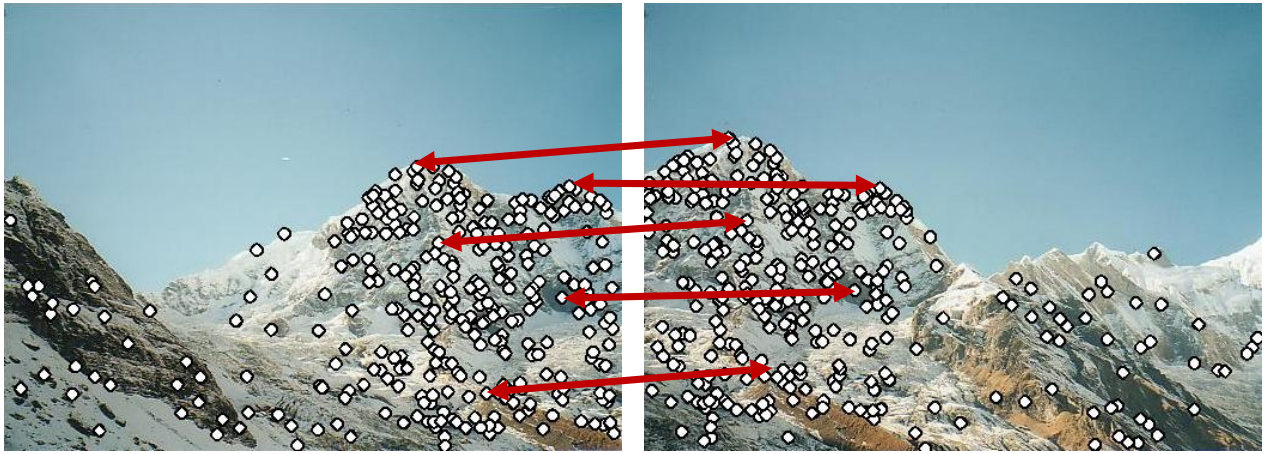
Matching with Features (1)

- Detect feature points in both images



Matching with Features (2)

- Detect feature points in both images
- Find corresponding pairs

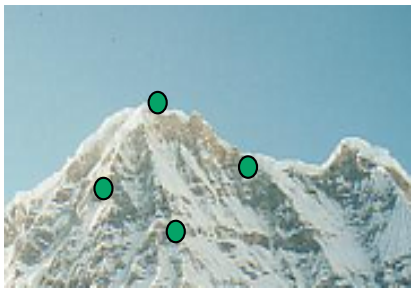


Matching with Features (3)

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

Matching with Features (4)

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

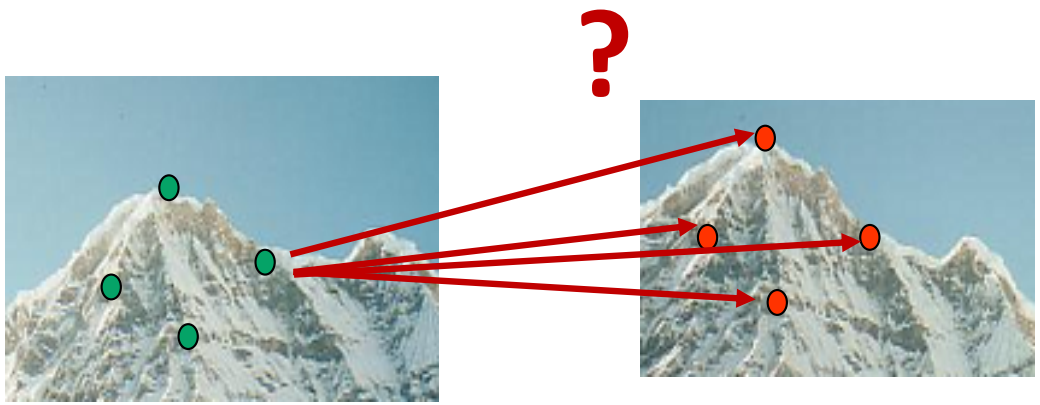


no chance to match!

We need a repeatable detector

Matching with Features (5)

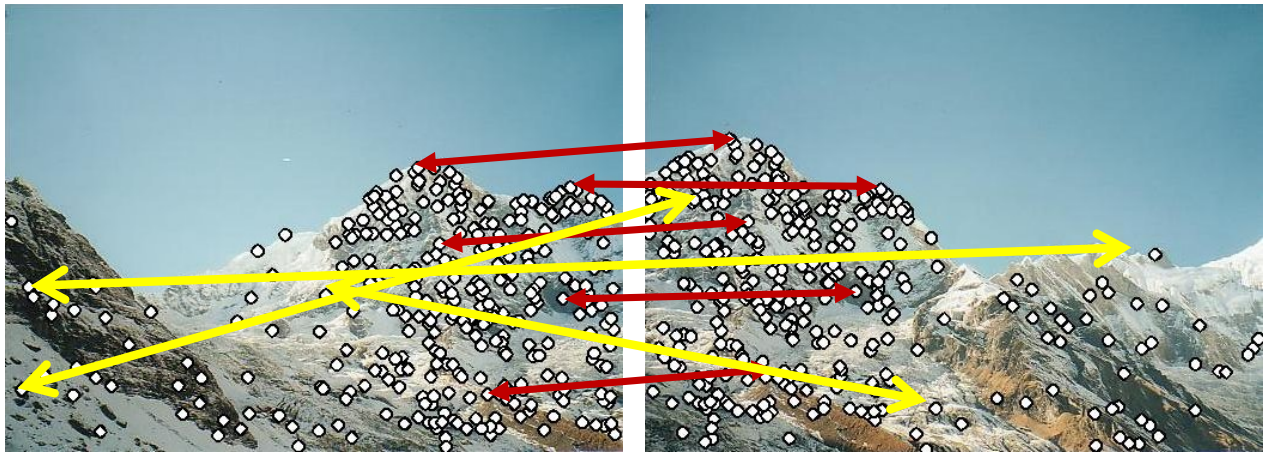
- Problem 2:
  - For each point correctly recognize the corresponding one



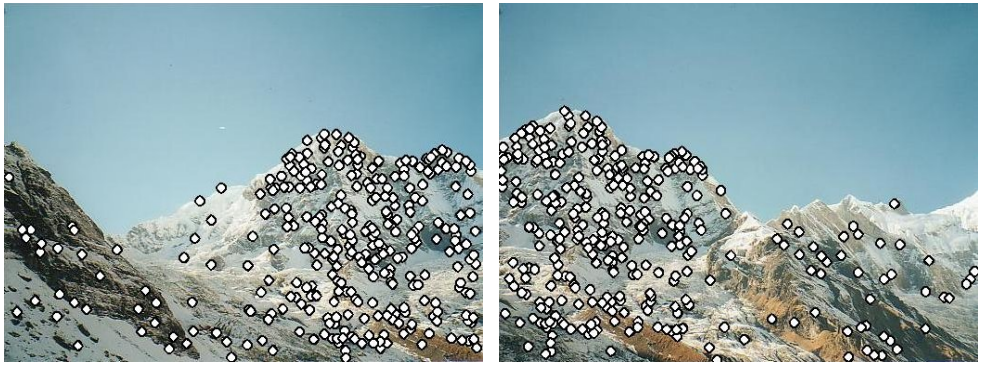
We need a reliable and distinctive descriptor

Matching with Features

- Problem 3:
  - Need to estimate transformation between images, despite erroneous correspondences.



Characteristics of Good Features

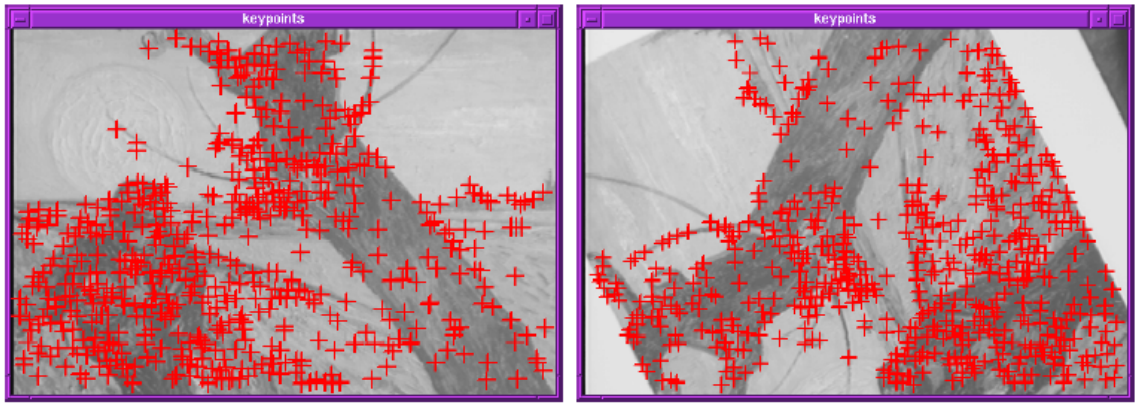


- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Applications

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

Corners

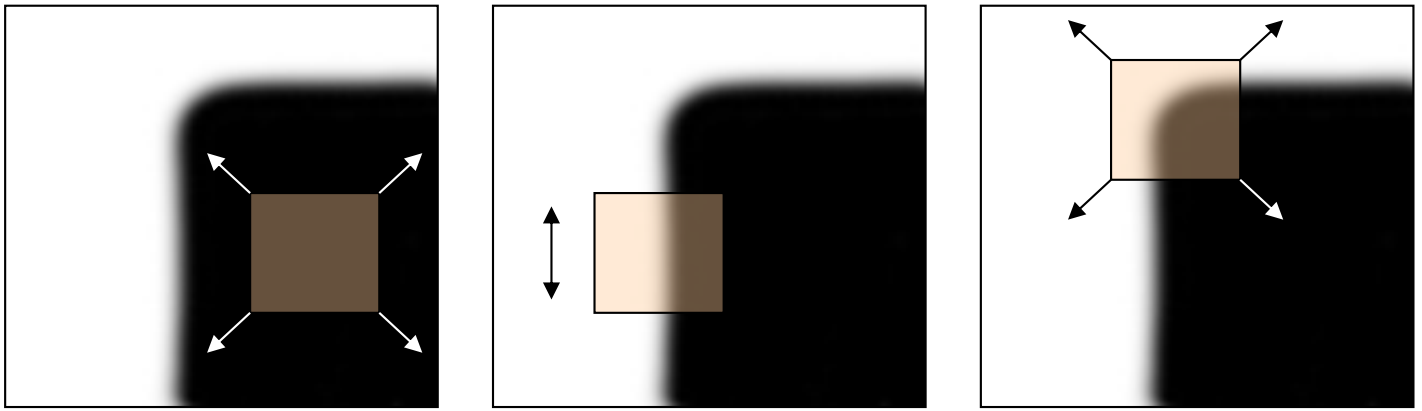


- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

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

Corner Detection: 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



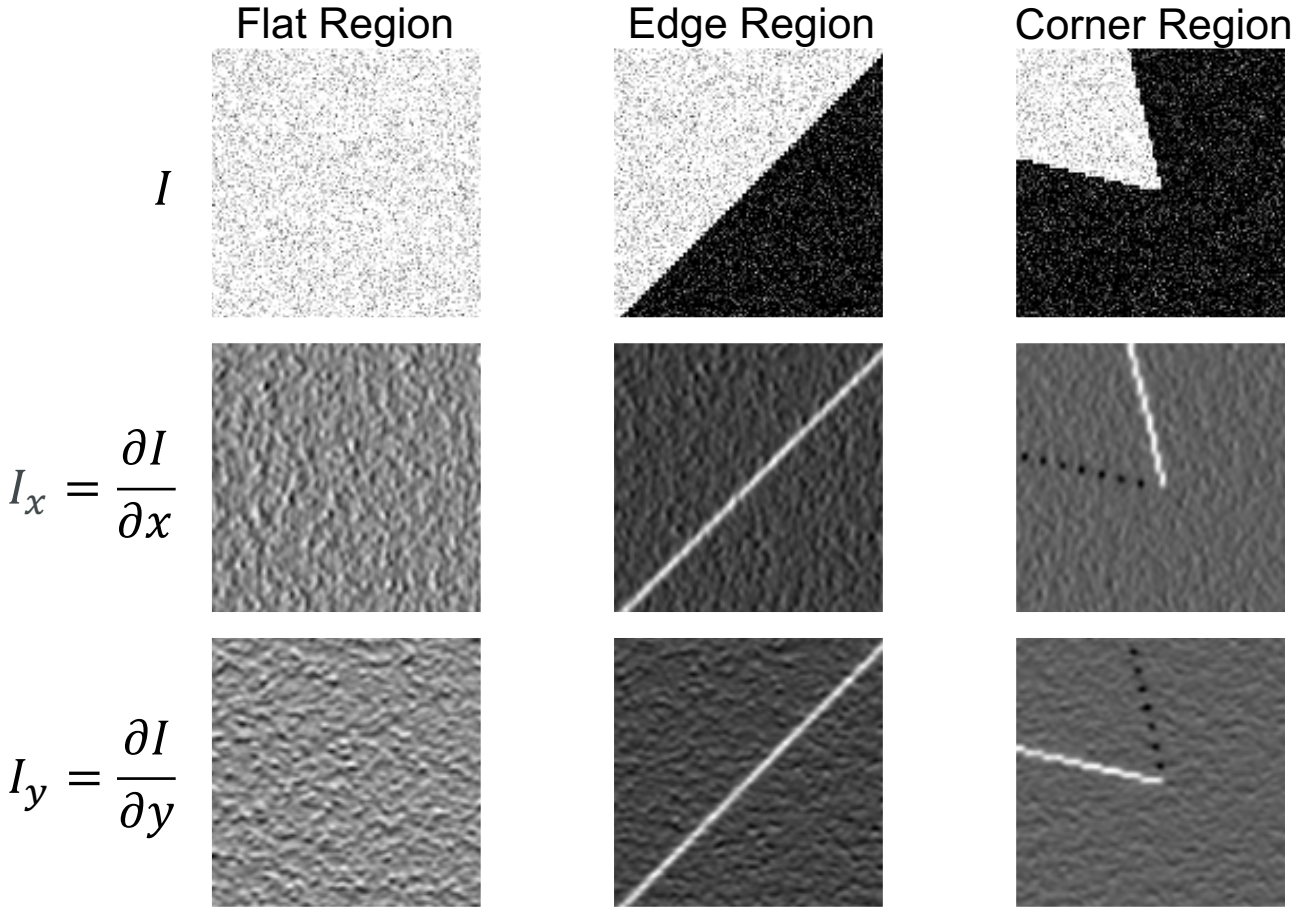
"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

"corner":
significant change
in all directions

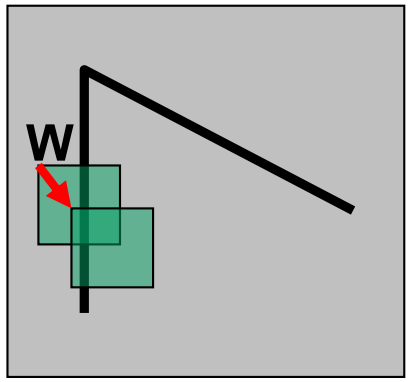
Source: A. Efros

Image Gradients



Feature Detection: The Math

- Consider shifting the window **W** by (u,v)
  - How do the pixels in **W** change?
  - Compare each pixel before and after by summing up the squared differences (SSD)
  - This defines an SSD "error" of *E(u,v)*:



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

Source: S. Seitz

Small Motion Assumption

- Taylor Series expansion of I:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \quad \text{…higher order terms…}$$

- If the motion (u,v) is small, then first order approx. is good

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

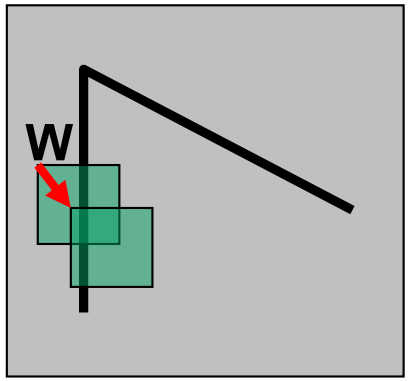
shorthand: $I_x = \frac{\partial I}{\partial x}$

- Plugging this into the formula on the previous slide…

Source: S. Seitz

Feature Detection: The Math (2)

- Consider shifting the window **W** by (u,v)
  - How do the pixels in **W** change?
  - Compare each pixel before and after by summing up the squared differences (SSD)
  - This defines an SSD "error" of *E(u,v)*:



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

$$\approx \sum_{(x,y)\in W} \left[I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y) \right]^2$$

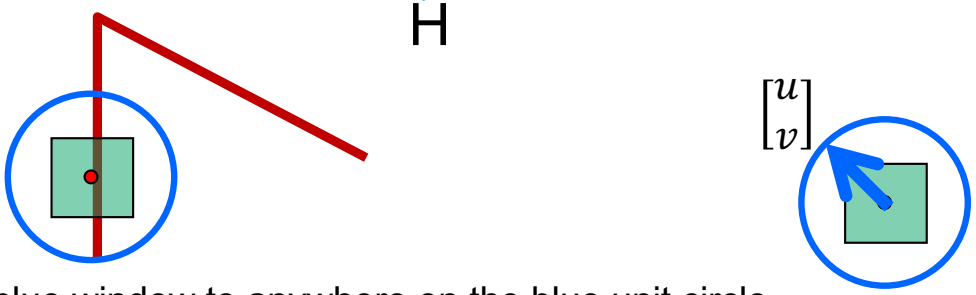
$$\approx \sum_{(x,y)\in W} \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$

Source: S. Seitz

Feature Detection: The Math (3)

- This can be rewritten:

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} \sum_{(x,y)\in W} \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{\text{H}} \begin{bmatrix} u \\ v \end{bmatrix}$$



For the example above

- You can move the center of the blue window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of ***H***

Source: S. Seitz

Eigenvalue/Eigenvector Review

The **eigenvectors** of a matrix **A** are the vectors **x** that satisfy: $Ax = \lambda x$

The scalar λ is the **eigenvalue** corresponding to **x**

- The eigenvalues are found by solving: $\det(A - \lambda I) = 0$
- In our case, ***A*** = ***H*** is a 2x2 matrix, so we have: $det\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$
- The solution: $\lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$

Once you know λ, you find **x** by solving $\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 0$

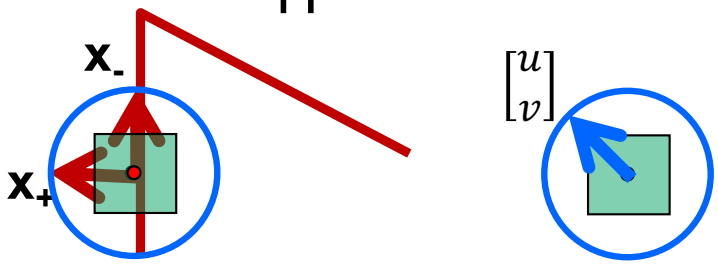
Source: S. Seitz

Feature Detection: The Math (4)

- This can be rewritten:

$$E(u,v) = [u \quad v] \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

H



Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of **largest** increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of **smallest** increase in E.
- λ_- = amount of increase in direction x_-

$$Hx_+ = \lambda_+ x_+$$
$$Hx_- = \lambda_- x_-$$

Source: S. Seitz

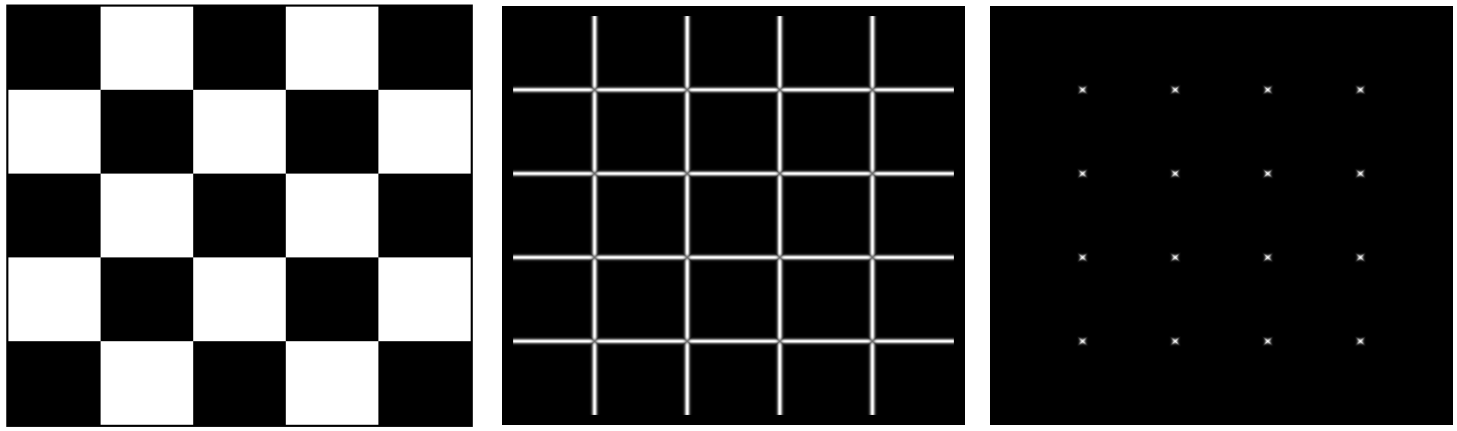
Feature Detection: The Math (5)

- How are λ_+, $\mathbf{x}_+$, λ_-, and $\mathbf{x}_-$ relevant for feature detection?
  - What's our feature scoring function?

Source: S. Seitz

Feature Detection: The Math (6)

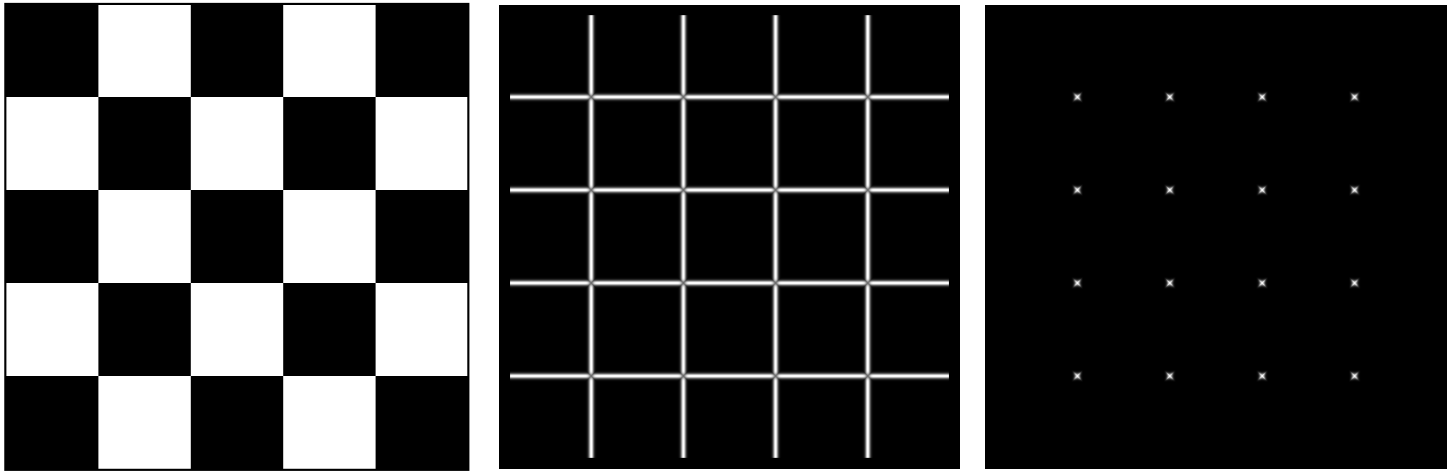
- How are λ_+, $\mathbf{x}_+$, λ_-, and $\mathbf{x}_-$ relevant for feature detection?
  - What's our feature scoring function?
- Want *E(u,v)* to be ***large*** for small shifts in ***all*** directions
  - The *minimum* of *E(u,v)* should be large, over all unit vectors [u v]
  - This minimum is given by the smaller eigenvalue (λ_-) of ***H***



I λ_+ λ_-

Feature Detection: The Math (7)

- Here's what you do
  - Compute the gradient at each point in the image
  - Create the ***H*** matrix from the entries in the gradient
  - Compute the eigenvalues.
  - Find points with large response (λ_- > threshold)
  - Choose those points where λ_- is a local maximum as features



I λ_+ λ_-

Summary

- **Corners**: Are one way to think about stable feature locations in images
- Essential concepts in this lecture:
  - What we mean by corners as a distribution of gradients
  - Formulating the corner detection problem
  - Deriving a measure for corners

Johns Hopkins Engineering

Computer Vision

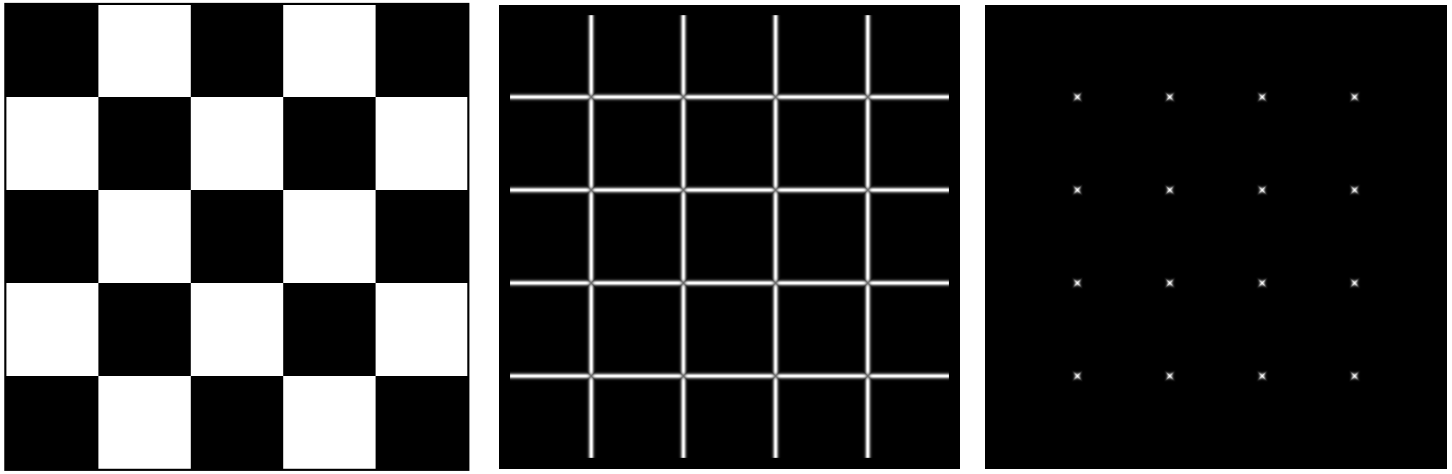
Corner Detection and Matching

Corner Detection and Matching

- Topics:
  - Harris Corner Detector Properties
  - Harris Corner Detector Examples

Feature Detection: The Math (7)

- Here's what you do
  - Compute the gradient at each point in the image
  - Create the ***H*** matrix from the entries in the gradient
  - Compute the eigenvalues.
  - Find points with large response (λ_- > threshold)
  - Choose those points where λ_- is a local maximum as features



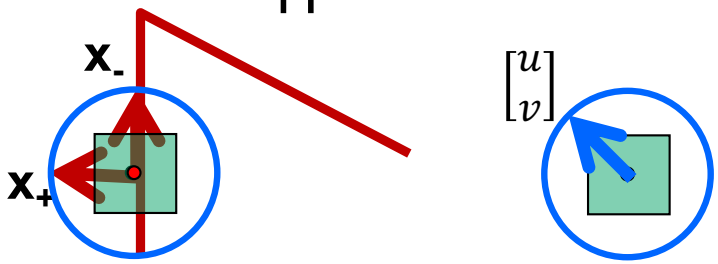
I λ_+ λ_-

Feature Detection: The Math (4)

- This can be rewritten:

$$E(u,v) = [u \quad v] \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

H



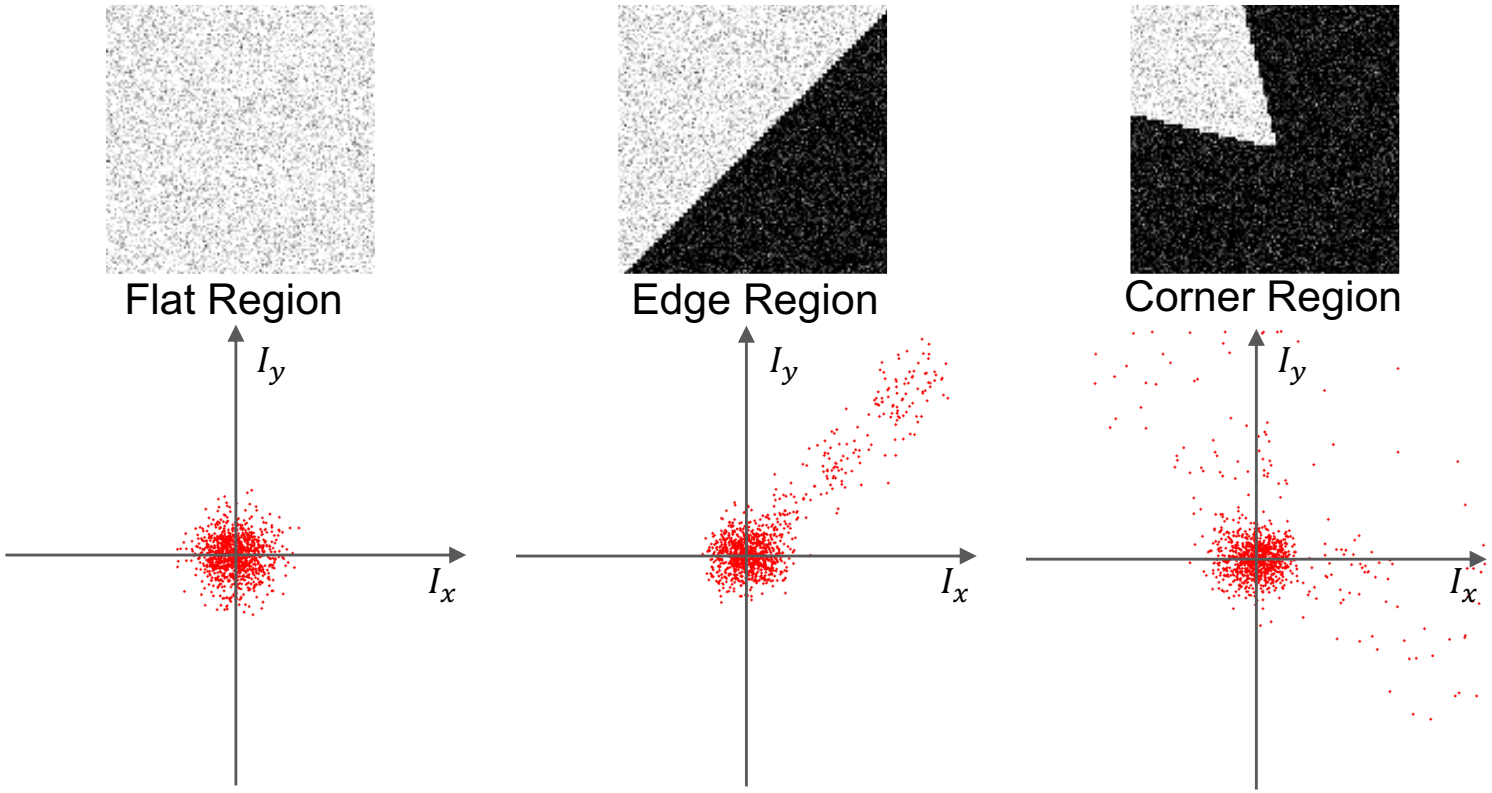
Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of **largest** increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of **smallest** increase in E.
- λ_- = amount of increase in direction x_-

$$Hx_+ = \lambda_+ x_+$$
$$Hx_- = \lambda_- x_-$$

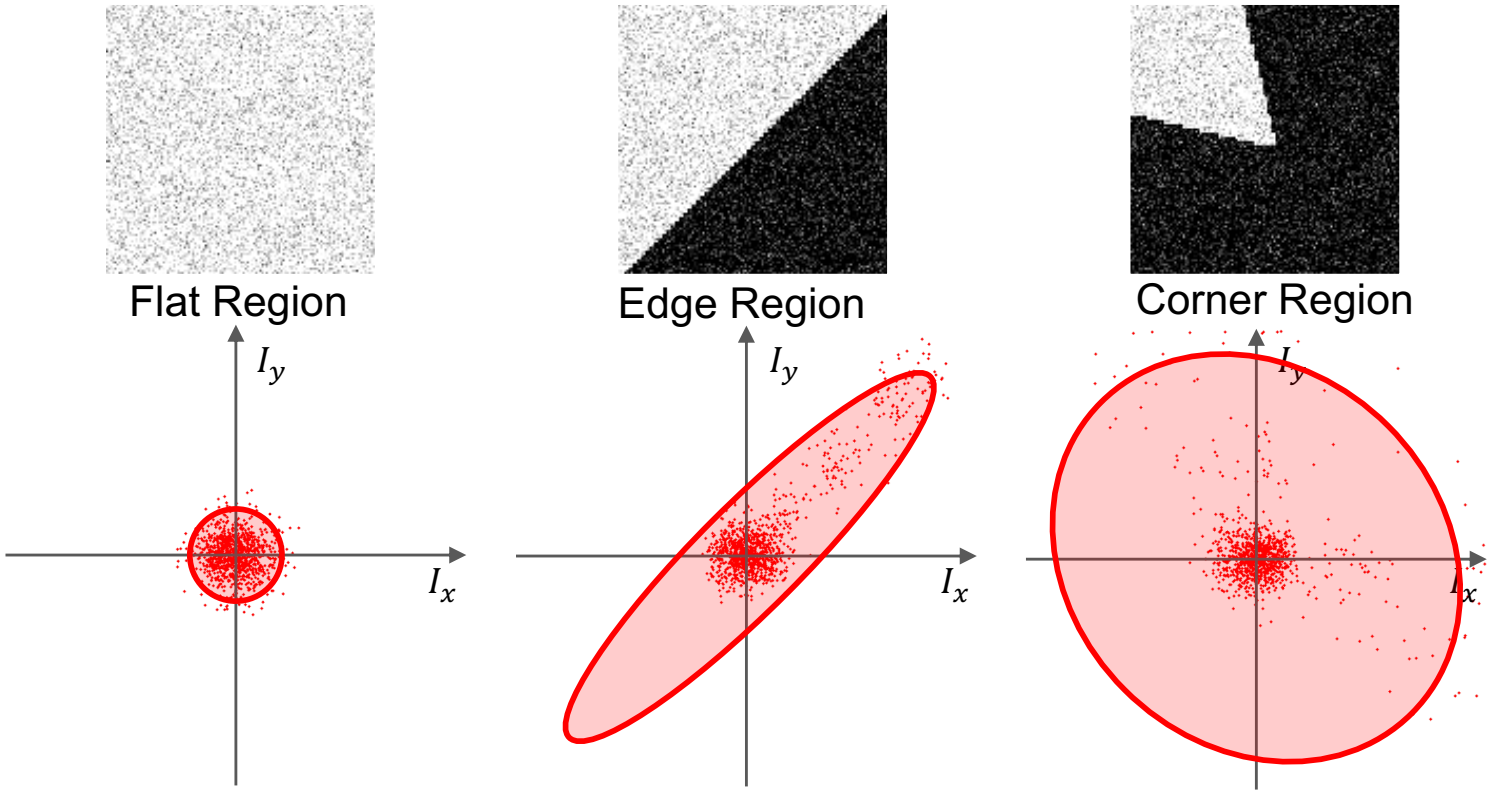
Source: S. Seitz

Distribution of Image Gradients



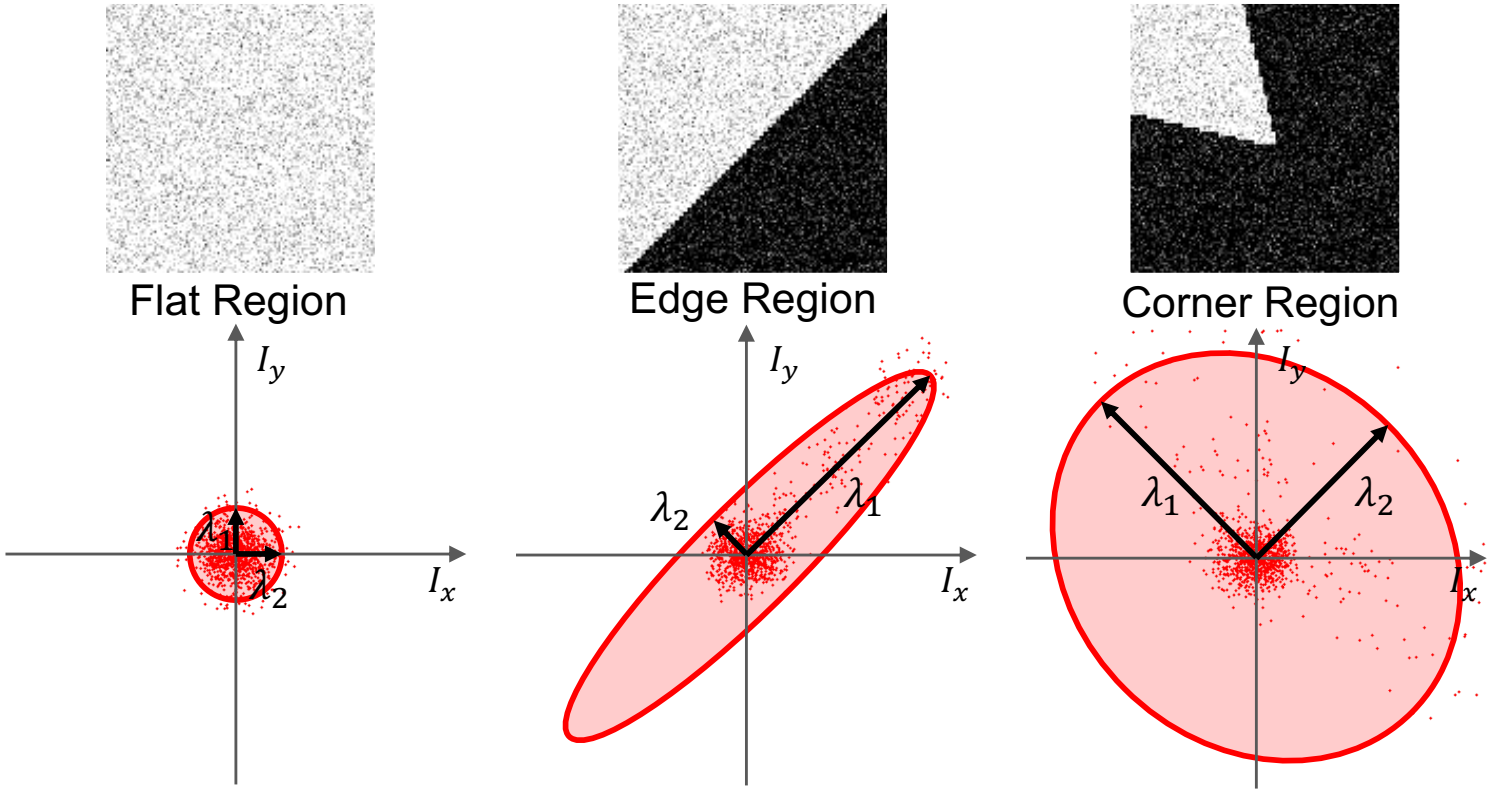
Distribution of I_x and I_y is different for all three regions.

Fitting Elliptical Disk to Distribution



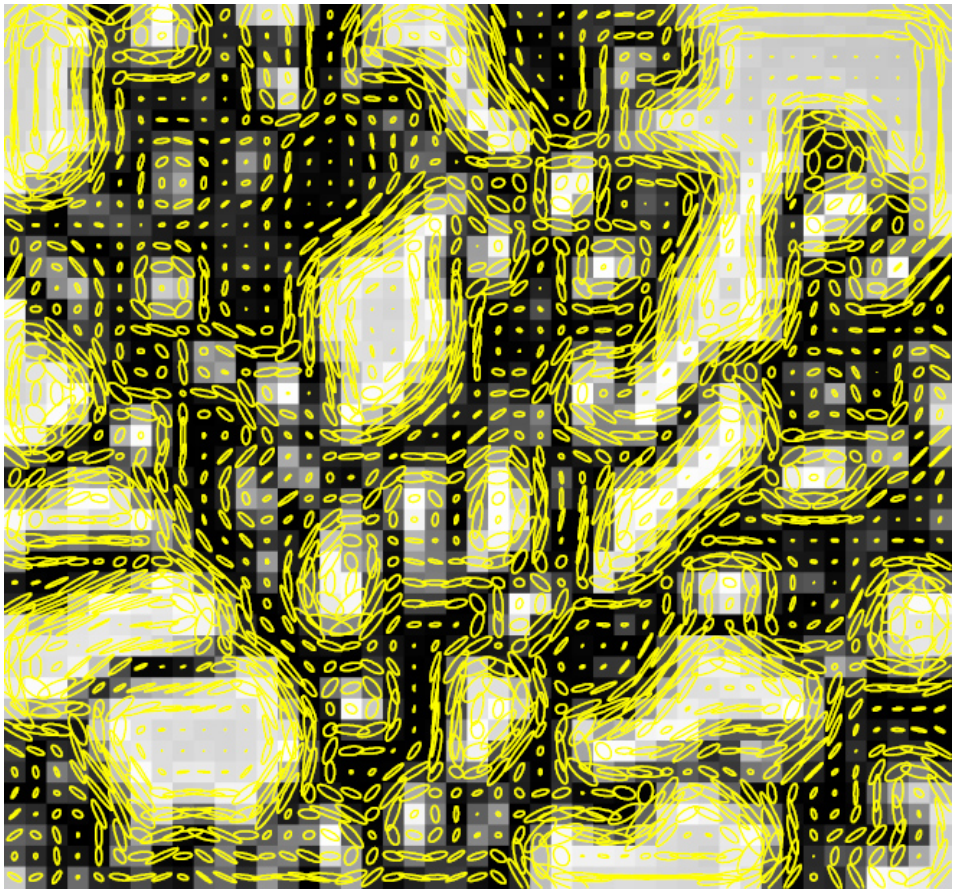
Distribution of I_x and I_y is different for all three regions.

Fitting Elliptical Disk to Distribution (cont.)



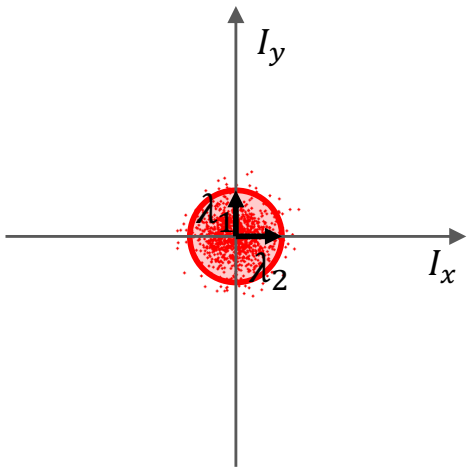
Visualization of Second Moments

Visualization of Second Moments



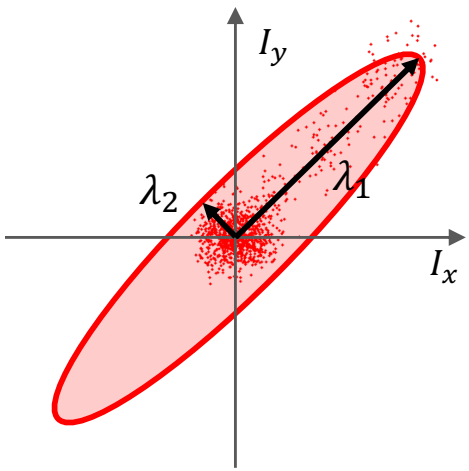
Interpretation of λ_1 and λ_2

Flat Region



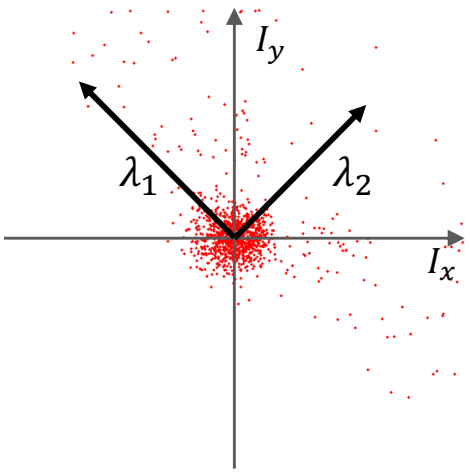
$\lambda_1 \sim \lambda_2$
Both are Small

Edge Region



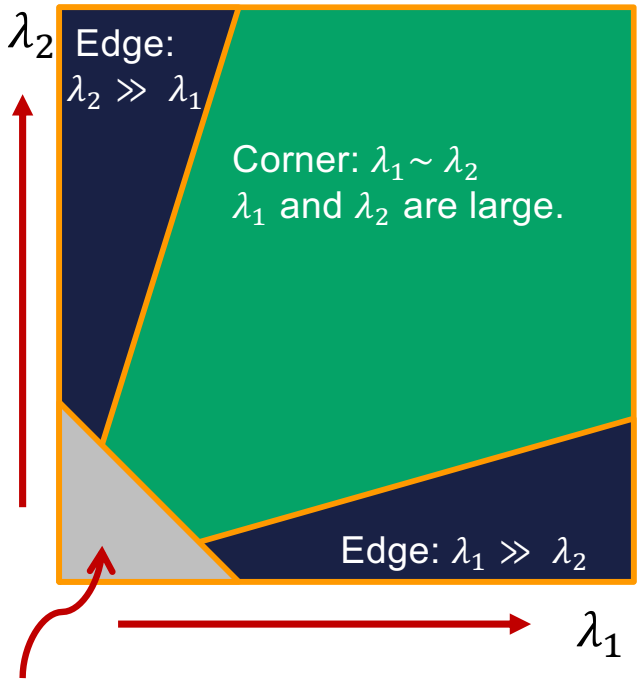
$\lambda_1 \gg \lambda_2$
λ_1 is Large
λ_2 is Small

Corner Region

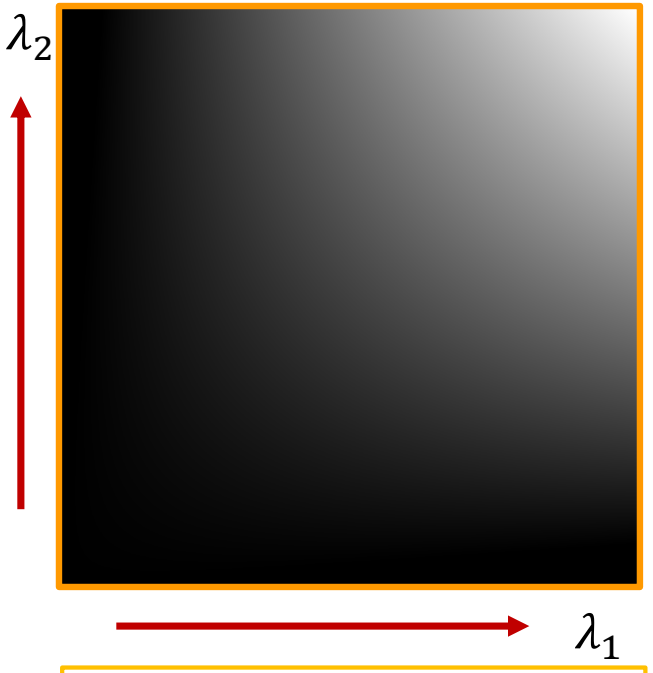


$\lambda_1 \sim \lambda_2$
Both are Large

Harris Corner Response Function



Flat: $\lambda_1 \sim \lambda_2$
λ_1 and λ_2 are small.

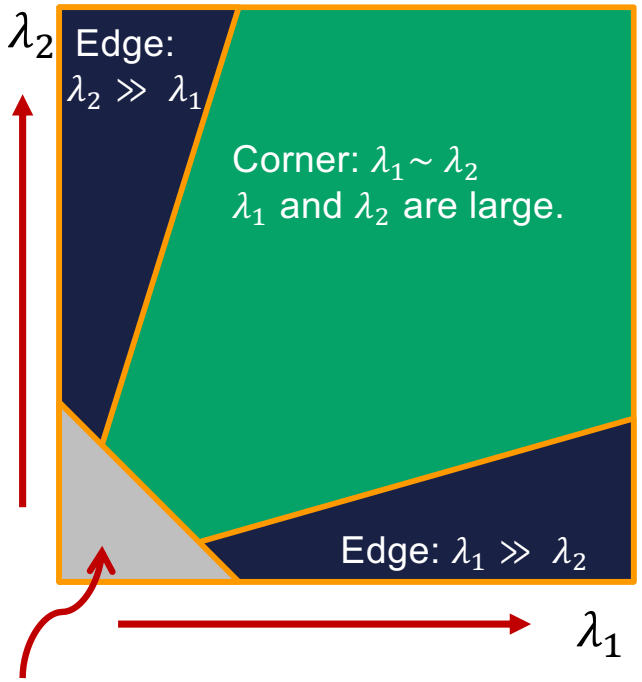


$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

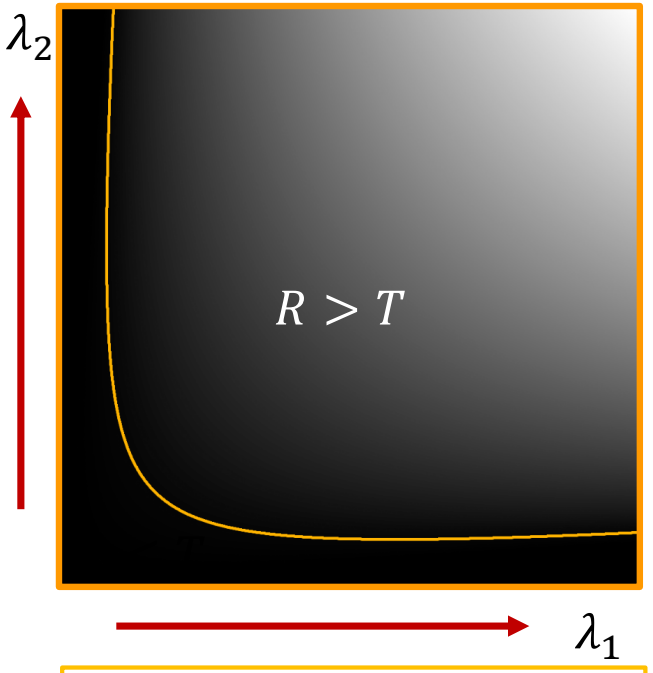
where: $0.04 \leq k \leq 0.06$
(Designed Empirically)

[Harris1988]

Harris Corner Response Function (cont.)



Flat: $\lambda_1 \sim \lambda_2$
λ_1 and λ_2 are small.



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

where: $0.04 \leq k \leq 0.06$
(Designed Empirically)

[Harris1988]

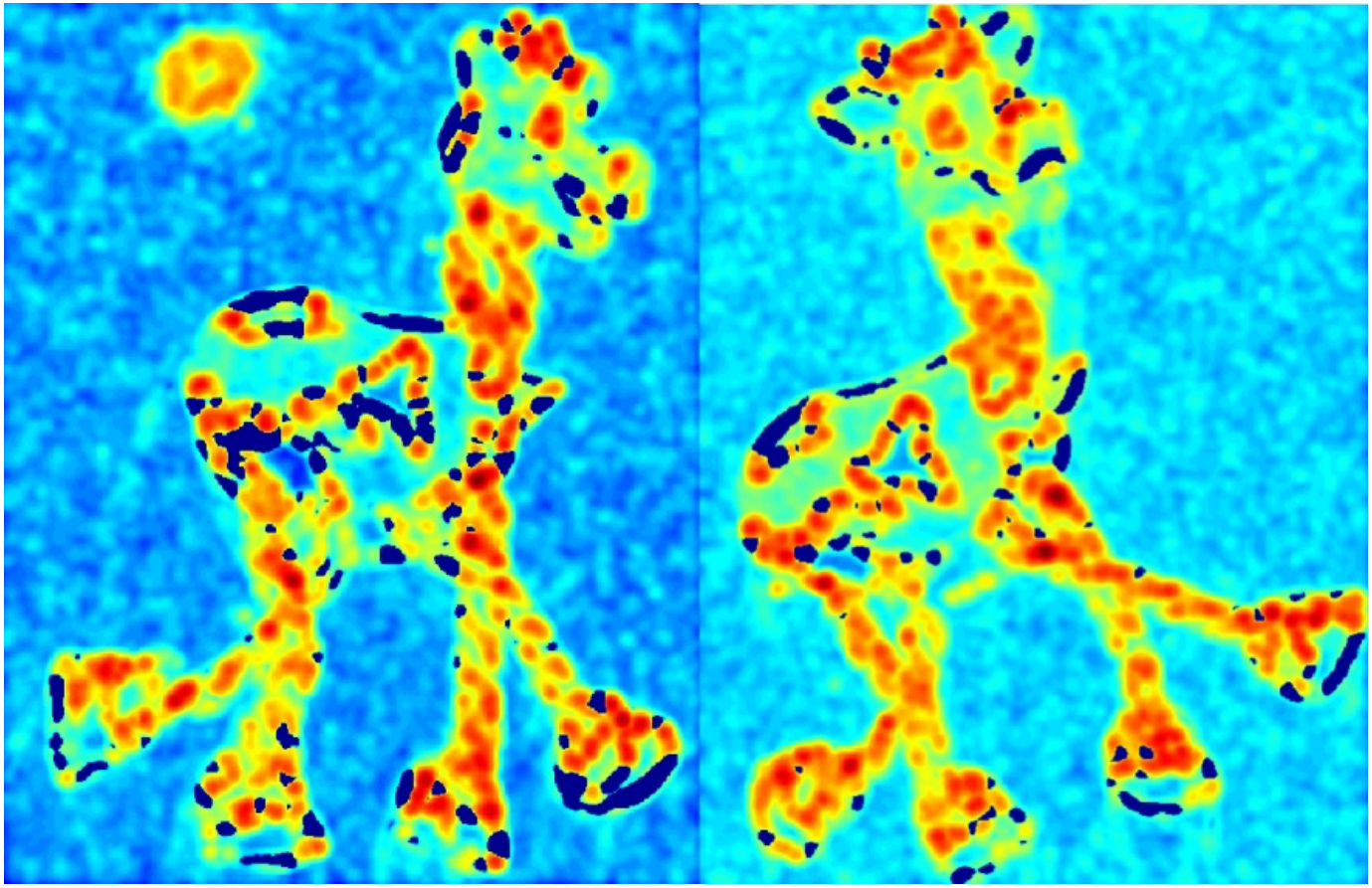
Harris Detector: Steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix *H* in a Gaussian window around each pixel
3. Compute corner response function *R*
4. Threshold *R*
5. Find local maxima of response function (nonmaximum suppression)

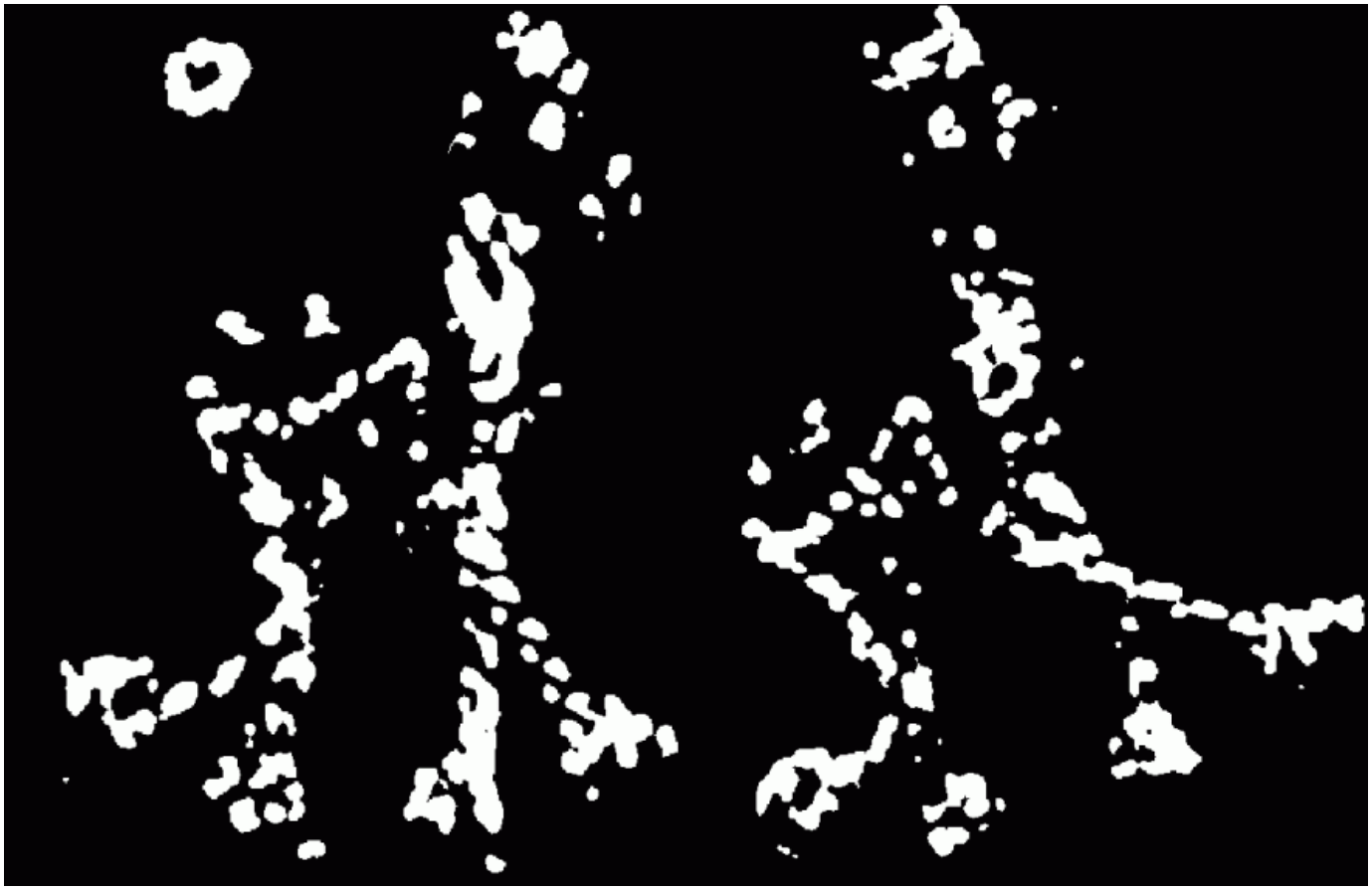
C.Harris and M.Stephens. "A Combined Corner and Edge Detector." Proceedings of the 4th Alvey Vision Conference: pages 147—151, 1988.

Harris Detector: Steps (cont.)

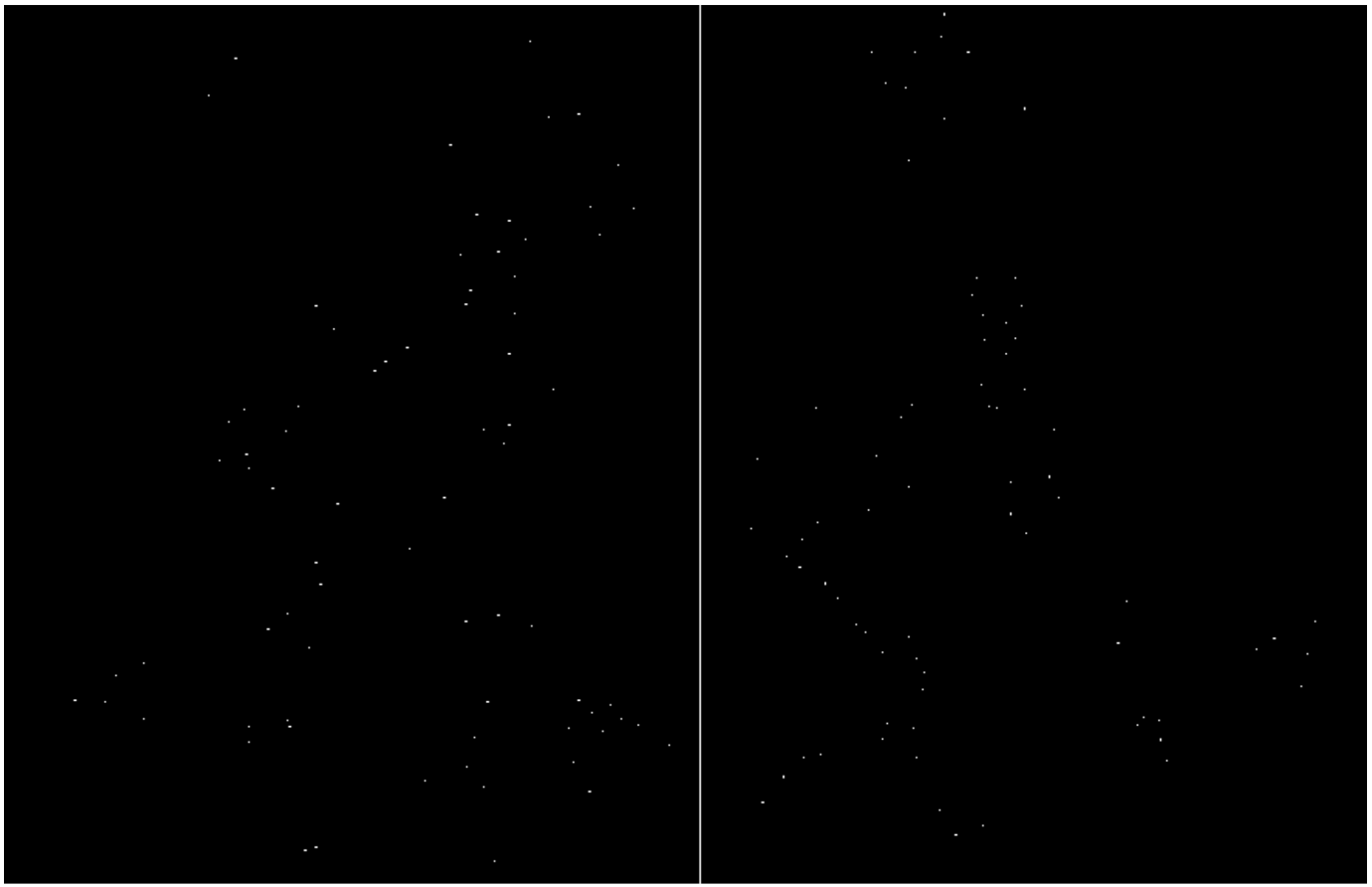
Harris: Corner Response



Harris: Threshold Response



Harris: Local Maxima

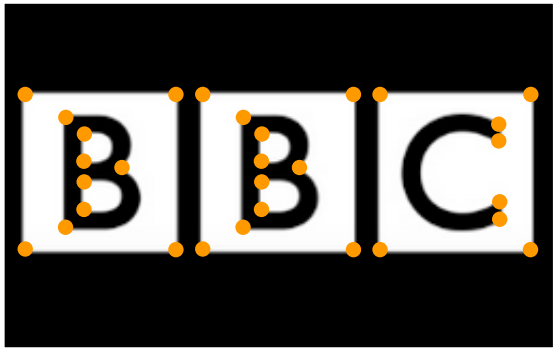


Harris Corner Locations

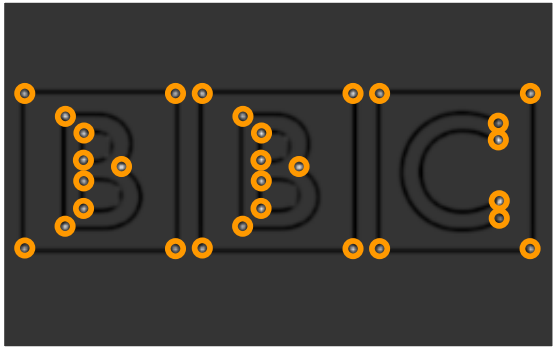
Harris Corner Detection Example

Image

Corner Response R

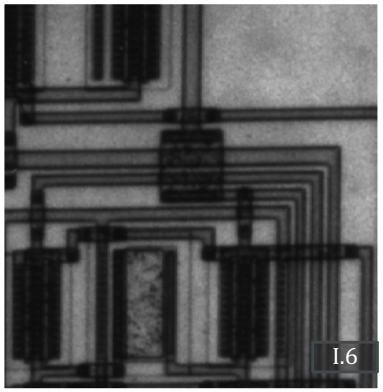


Detected Corners



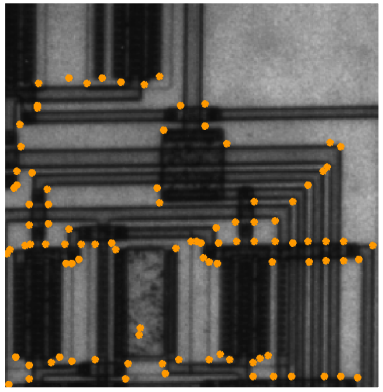
Corner Response $R > T$

Harris Corner Detection Example (cont.)

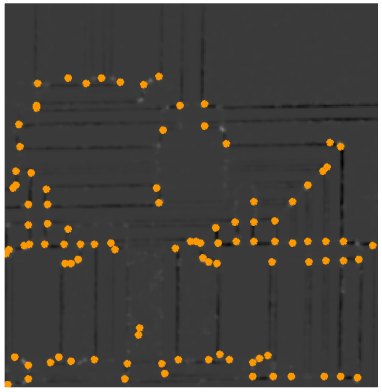


Image

Corner Response R



Detected Corners



Corner Response $R > T$

Invariance and Covariance

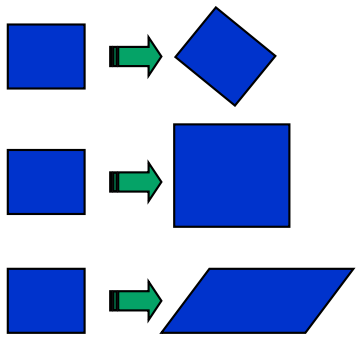
- We want features to be *invariant* to photometric transformations and *covariant* to geometric transformations
  - **Invariance:** image is transformed, and features do not change
  - **Covariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations

Transformations

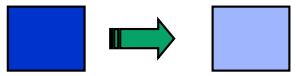
- Geometric
  - **Rotation**

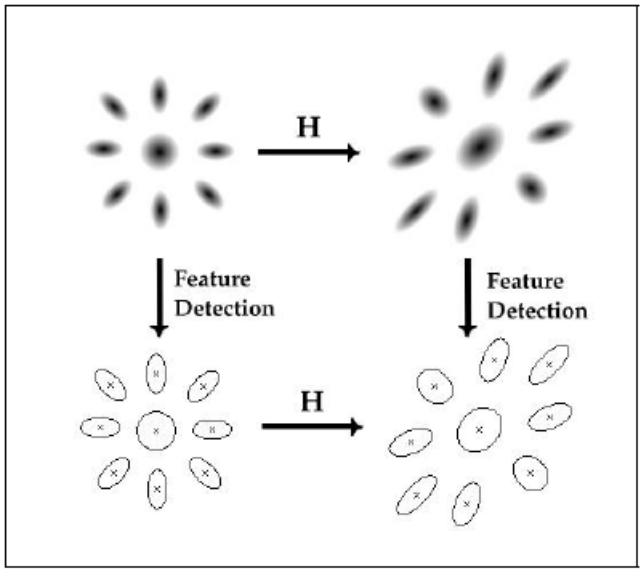
    Scale

  - **Affine**
    valid for:
    orthographic camera,
    locally planar object



- Photometric
  - **Affine intensity change** ($I \rightarrow a\,I + b$)

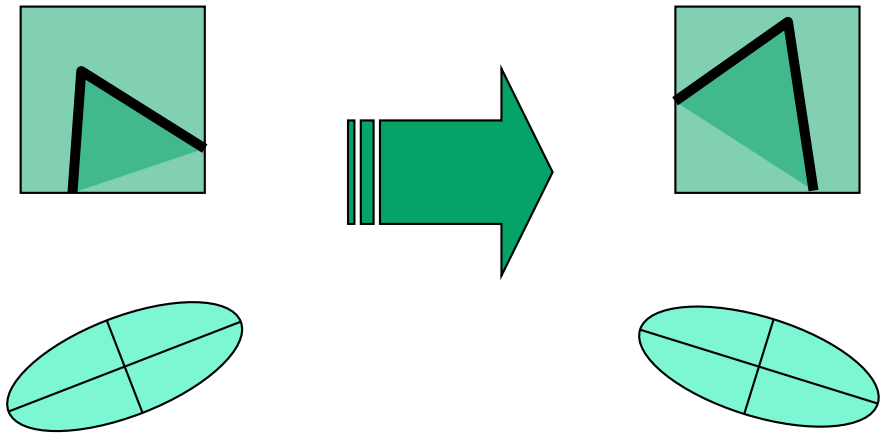




T. Kadir, A. Zisserman, and M. Brady, An Affine Invariant Salient Region Detector, ECCV 2004

Slide credit: S. Lazebnik

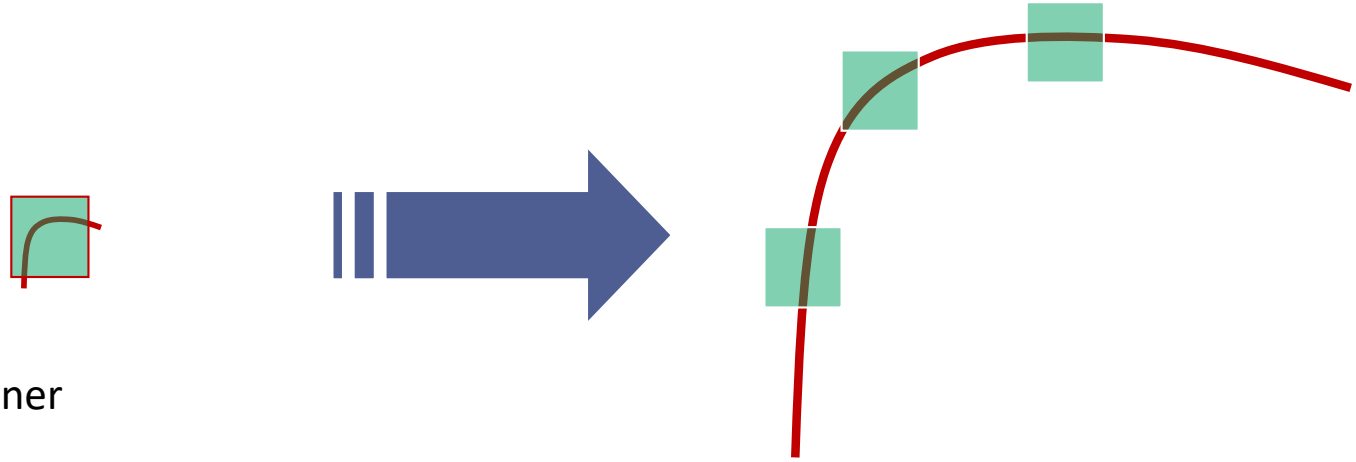
Image Rotation



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

Corner response R is invariant w.r.t. rotation and *corner location* is covariant

Scaling



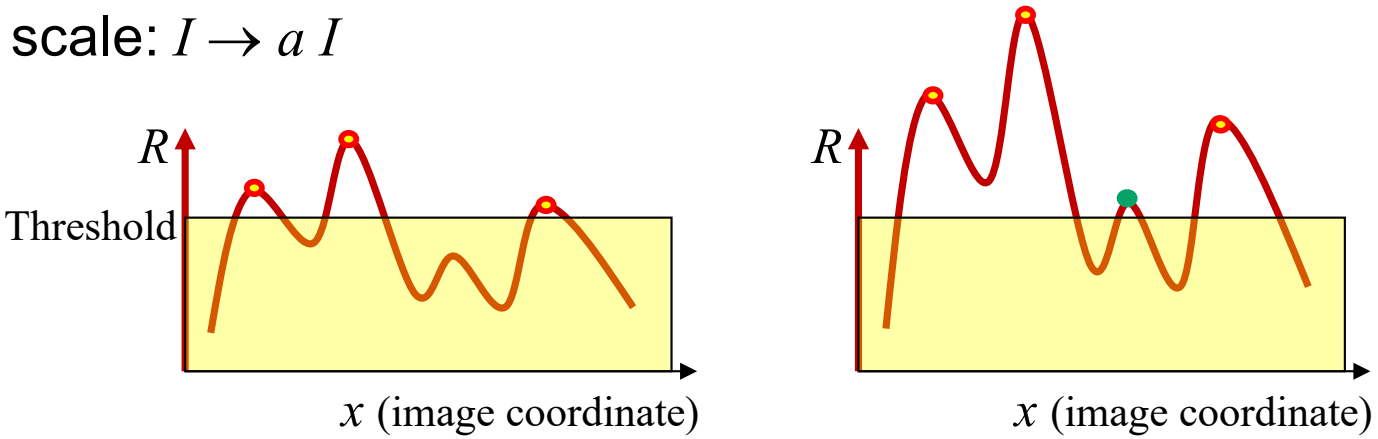
Corner

All points will be classified as edges

Not invariant to scaling

Affine Intensity Change

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scale: $I \rightarrow a\, I$



Partially invariant to affine intensity change

Summary

- **Corners**: Are one way to think about stable feature locations in images
- Essential concepts in this lecture:
  - How corners are expressed in real images
  - Picking corner points from the cornerness map
  - The idea of invariance and covariance

Johns Hopkins Engineering

Computer Vision

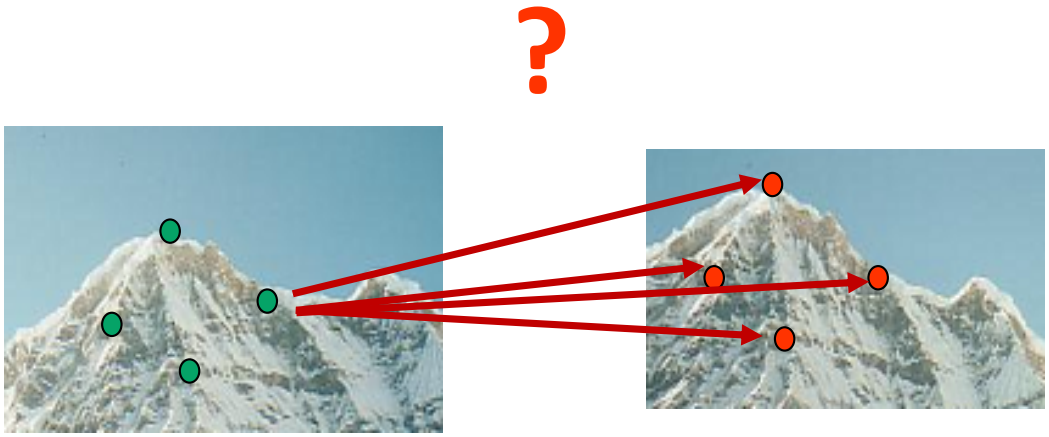
Corner Detection and Matching

Corner Detection and Matching

- Topics:
  - Matching With Patches As Features

Matching With Features

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

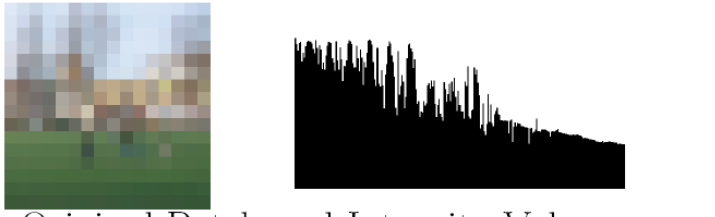


We need a reliable and distinctive descriptor

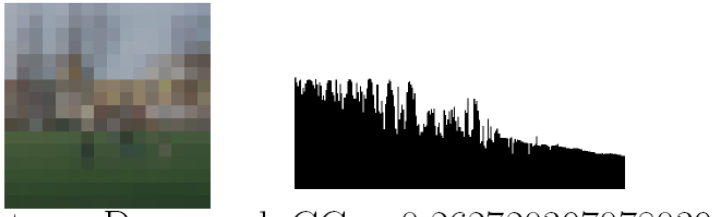
Cross-Correlation

- $CC(P_1, P_2) = \frac{1}{N}\sum_{i}^{N} P_1[i]P_2[i]$
- Output value depends on contrast and brightness of pixels
- Not invariant to changes in a,b

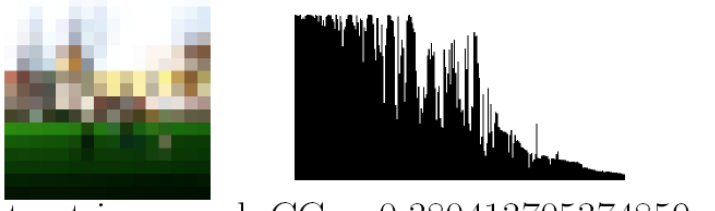
Affine photometric transformation:
$I \rightarrow a\,I + b$



Original Patch and Intensity Values



Brightness Decreased, CC = 0.262720397078039



Contrast increased, CC = 0.380413705374859

Normalized Cross-Correlation

- Make each patch zero mean:

$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$

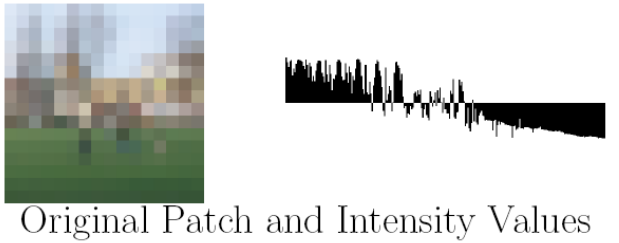
$$Z(x,y) = I(x,y) - \mu$$

- Then make unit variance:

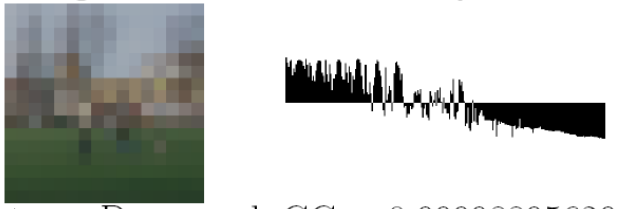
$$\sigma^2 = \frac{1}{N}\sum_{x,y} Z(x,y)^2$$

$$ZN(x,y) = \frac{Z(x,y)}{\sigma}$$

Affine photometric transformation:
$I \rightarrow a\,I + b$



Original Patch and Intensity Values

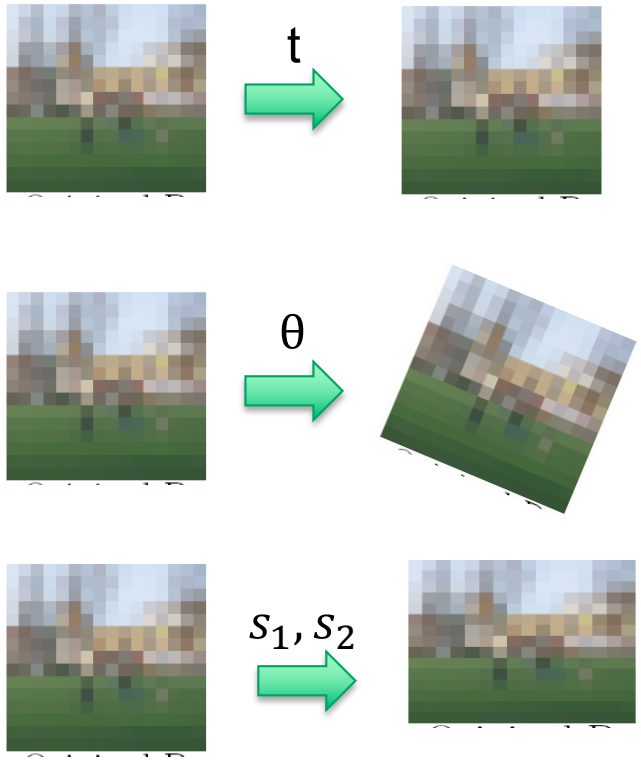


Brightness Decreased, CC = 0.99998895629

Contrast increased, CC = 0.969868160814

Normalized Cross-Correlation: Geometric Change

- Translation: Covariant
- Rotation
  - Not invariant
  - Can use eigenvectors to rotate patches
- Scale/Aspect ratio
  - Not invariant
  - Could be made invariant if there were a way to measure these changes in the detector



Feature Matching

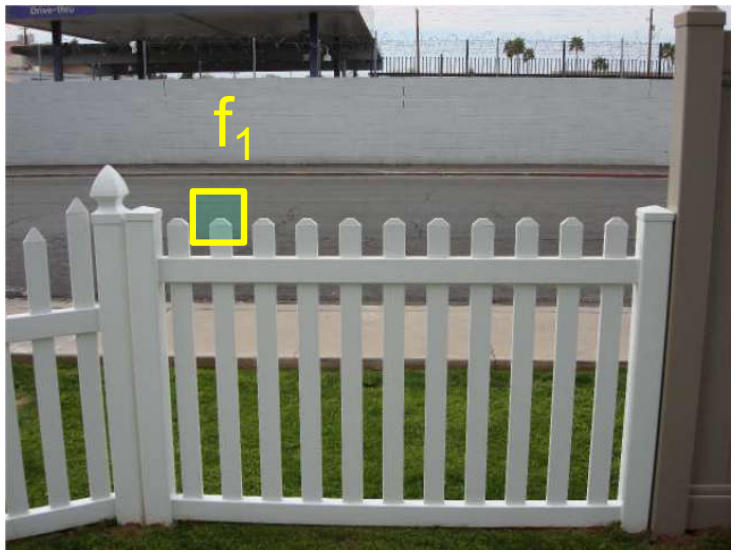
Given a feature in I_1, how to find the best match in I_2?

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

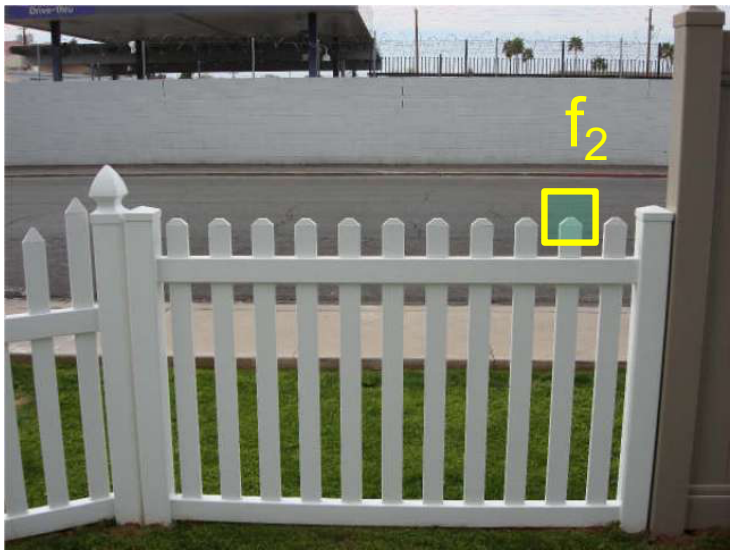
Feature Distance

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

- Simple approach is CC(f_1, f_2)
  - Cross-Correlation between entries of the two descriptors
  - Can give good scores to very ambiguous (bad) matches



I_1



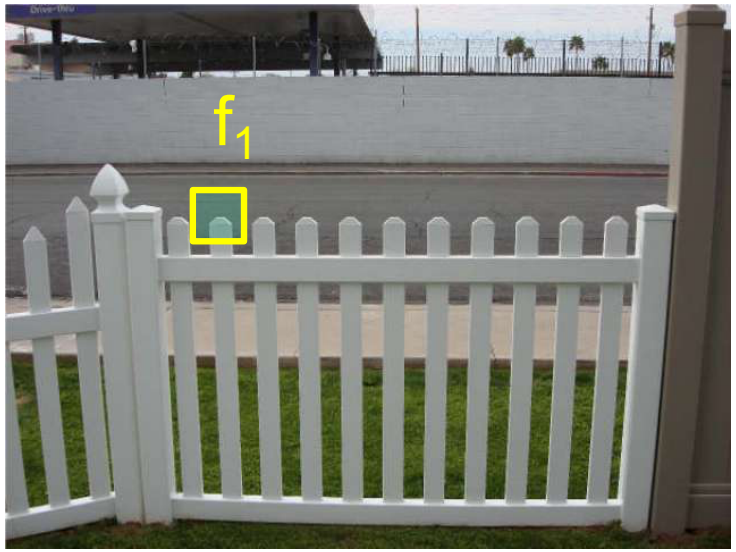
I_2

Slide: S. Seitz

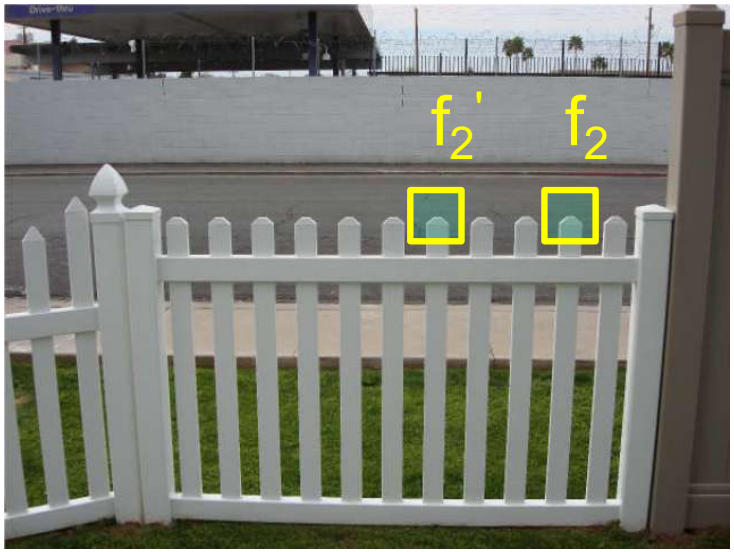
Feature Distance

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

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



I_1

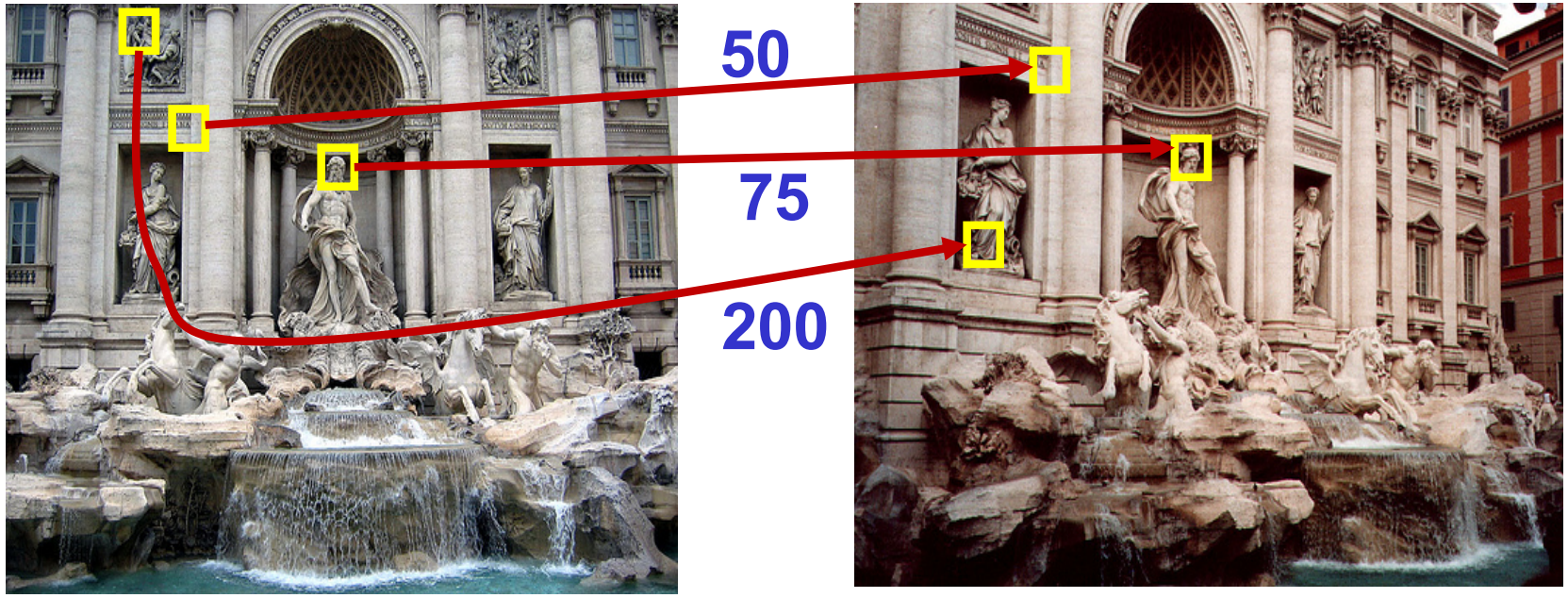


I_2

Slide: S. Seitz

Evaluating The Results

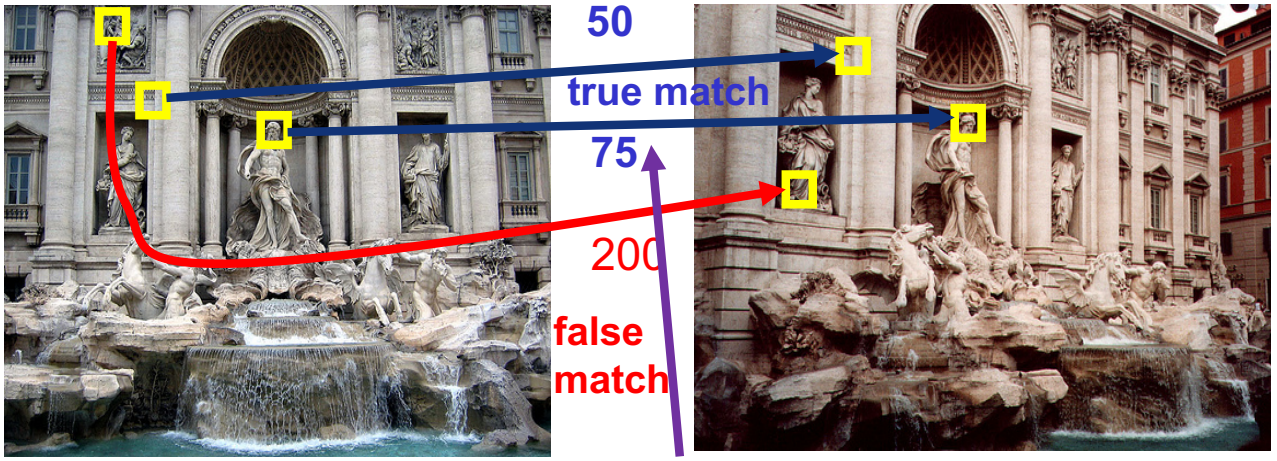
How can we measure the performance of a feature matcher?



Feature Distance

Slide: S. Seitz

True/False Positives



The distance threshold affects performance

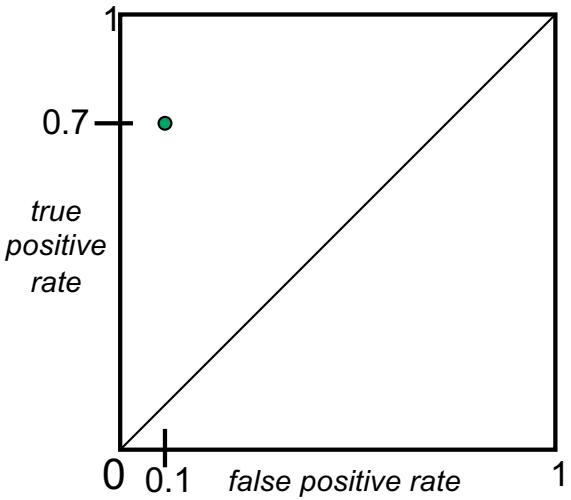
- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?

Slide: S. Seitz

Evaluating The Results

How can we measure the performance of a feature matcher?

$$\frac{\text{\# true positives}}{\text{\# matching features (positives)}}$$



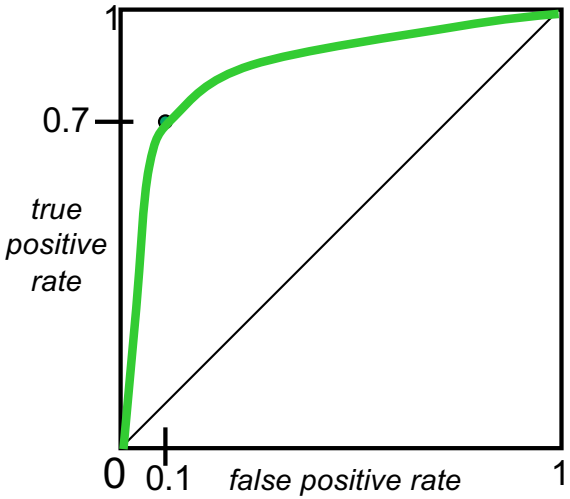
$$\frac{\text{\# false positives}}{\text{\# unmatched features (negatives)}}$$

Slide: S. Seitz

Evaluating The Results

How can we measure the performance of a feature matcher?

ROC Curve
("Receiver Operator Characteristic")



$$\frac{\text{\# true positives}}{\text{\# matching features (positives)}}$$

$$\frac{\text{\# false positives}}{\text{\# unmatched features (negatives)}}$$

- ROC Curves
  - Generated by counting # current/incorrect matches, for different thresholds
  - Want to maximize area under the curve (AUC)
  - Useful for comparing different feature matching methods

Slide: S. Seitz

Summary

- **Corners**: Are one way to think about stable feature locations in images
- Essential concepts in this lecture:
  - Using correlation as matching – pros and cons
  - Tabulating matching results

References

- Textbooks:
  - Robot Vision (Chapter 8) Horn, B. K. P., MIT Press
  - Computer Vision: Algorithms and Applications (Chapter 4.1) Szelinski, 2011 (available online)
  - Computer Vision: A Modern Approach (Chapter 8) Forsyth, D and Ponce, J., Prentice Hall
  - Digital Image Processing (Chapter 3) González, R and Woods, R., Prentice Hall
- Papers:
  - [Canny1986] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
  - [Harris1988] Harris, C. and Stephens, M., A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference. pp. 147–151.
  - [Marr1980] Marr, D. and Hildreth, E., Theory of Edge Detection," Proc. R. Soc. London,B 207, 187-217, 1980.

Image Credits

- I.1 Adapted from Fig 3.1, Nalwa, V., A Guided Tour of Computer Vision.
- I.2 Adapted from Fig 3.3, Nalwa, V., A Guided Tour of Computer Vision.
- I.3 Matlab Demo Image
- I.4 http://en.wikipedia.org/wiki/File:Caf%C3%A9_wall.svg
- I.5 http://www.michaelbach.de/ot/geom_KitaokaBulge/index.html
- I.6 Matlab Demo Image