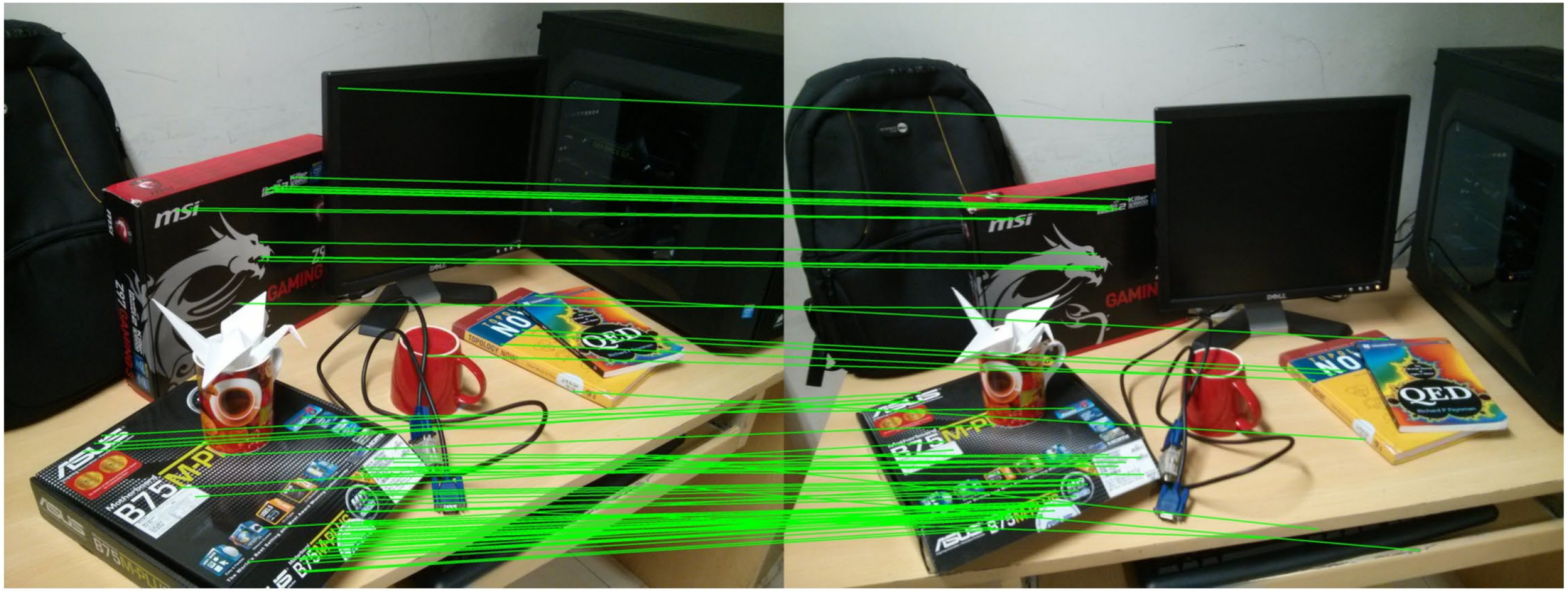
Keypoint Features: Image Convolution and Harris Corner Detector

CS 6384 Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Feature Detection and Matching

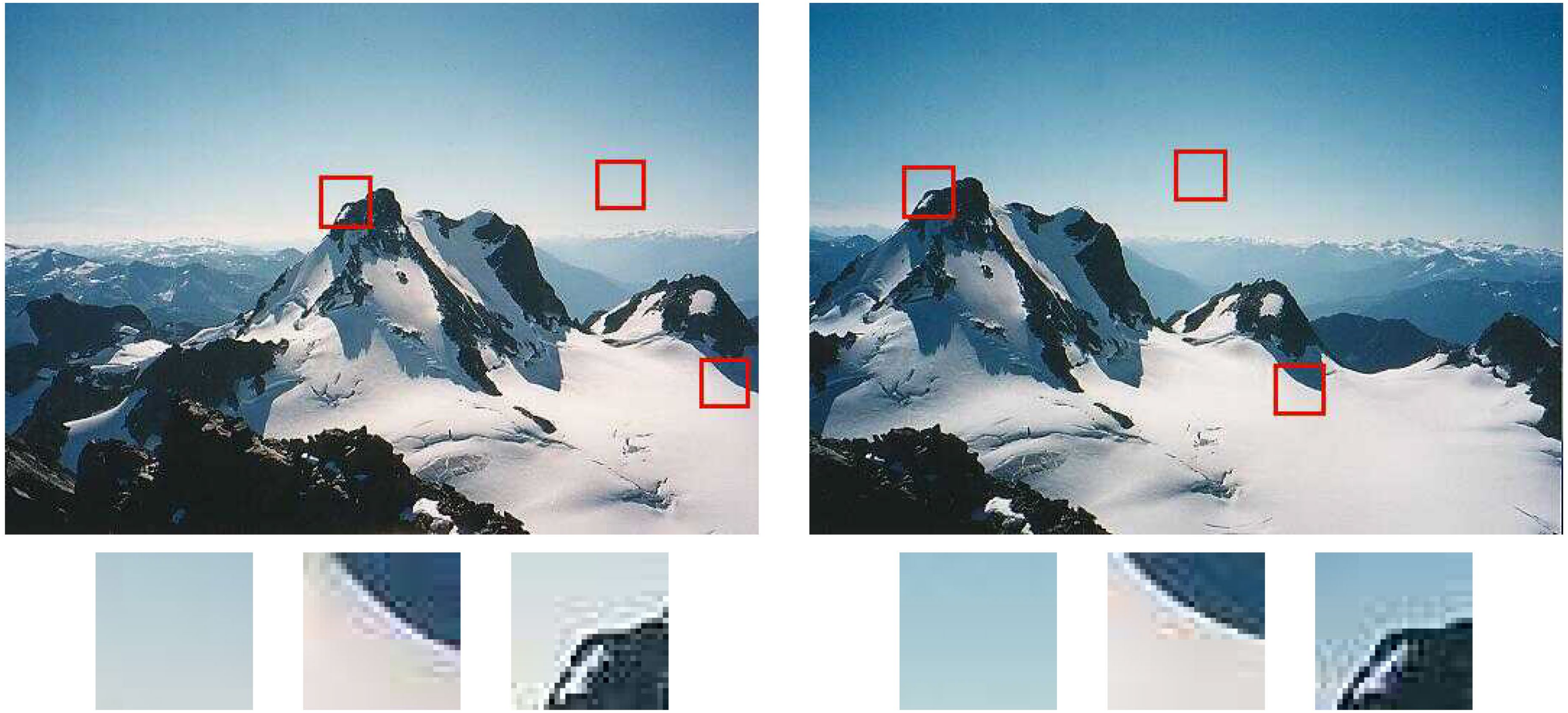


Geometry-aware Feature Matching for Structure from Motion Applications. Shah et al, WACV'15

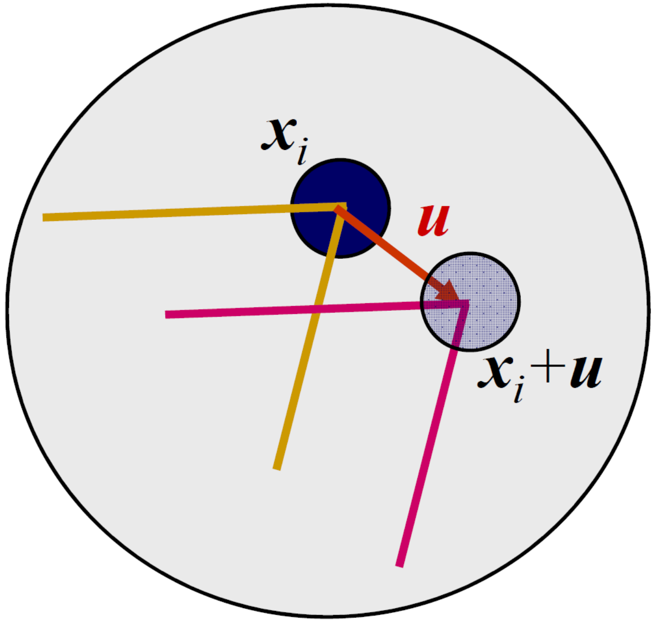
Applications: stereo matching, image stitching, 3D reconstruction, camera pose estimation, object recognition

Feature Detectors

- How to find image locations that can be reliably matched with images?

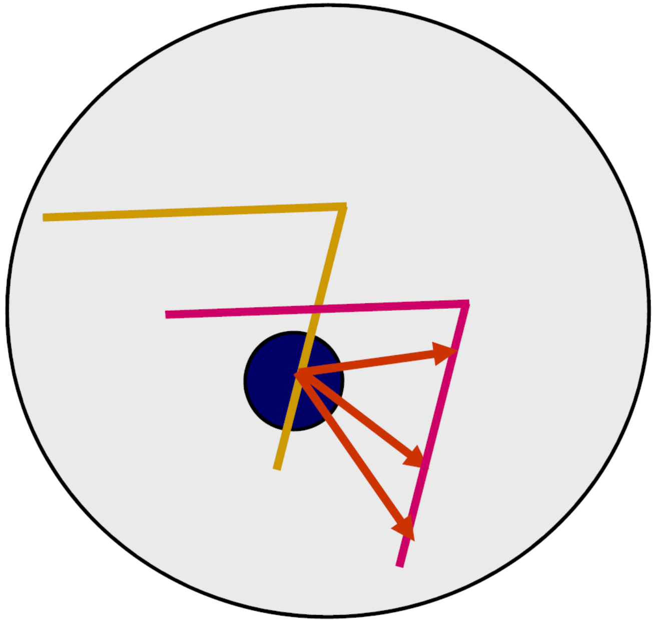


Feature Detectors



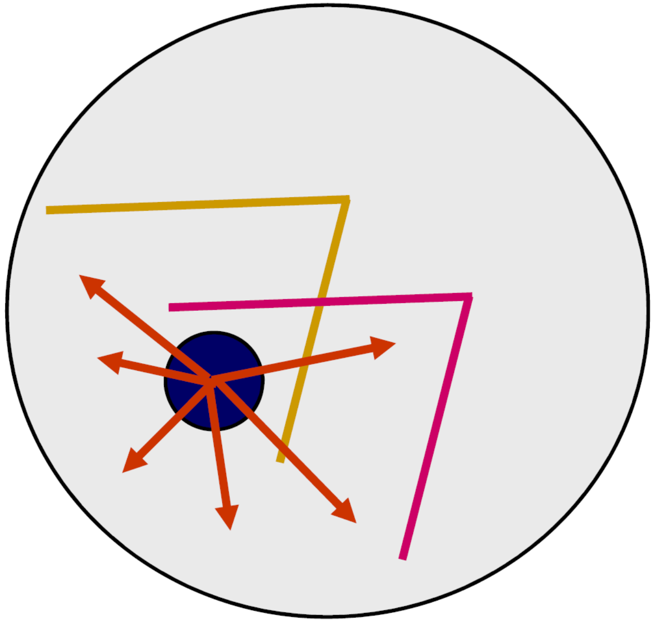
(a)

Corner



(b)

Edge



(c)

Textureless region

Image Data

width

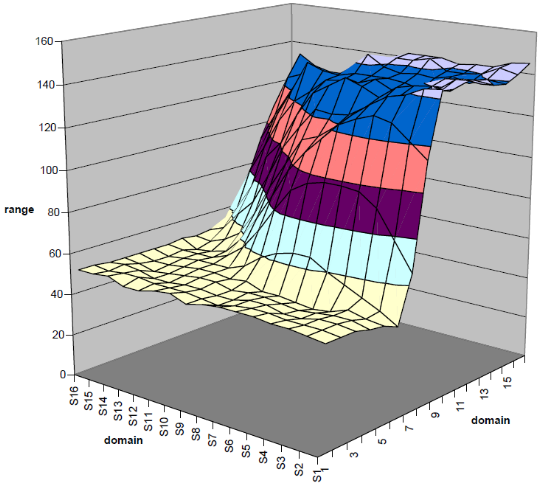
height

$H \times W \times 3$

$H \times W$

Grayscale
[0, 255]

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120



Function $I(\mathbf{x})$ $f(\mathbf{x})$

$I(x, y)$ $f(x, y)$

RGB color space
[0, 255]

```
0.2989 * R + 0.5870 * G + 0.1140 * B
```

Linear Filtering

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

$f(x,y)$

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$h(x,y)$

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

$g(x,y)$

k \ l	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

Correlation $$g(i,j) = \sum_{k,l} f(i+k, j+l) h(k,l)$$

Kernel

$$g = f \otimes h$$

Filtering vs. Convolution

k \ l	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

- Filtering $$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l)$$

What is the difference?

- Convolution $$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l)$$

$$g = f * h$$

Filter flipped vertically and horizontally

$h(x,y)$

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

Properties of Convolution

Commutative

$$a \star b = b \star a$$

Associative

$$(((a \star b_1) \star b_2) \star b_3) = a \star (b_1 \star b_2 \star b_3)$$

Distributes over addition

$$a \star (b + c) = (a \star b) + (a \star c)$$

Scalars factor out

$$\lambda a \star b = a \star \lambda b = \lambda (a \star b)$$

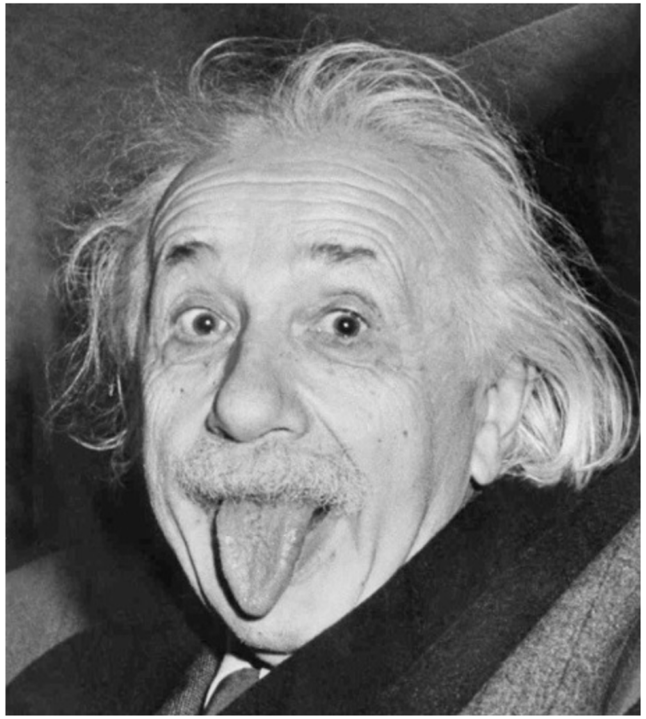
Derivative Theorem of Convolution

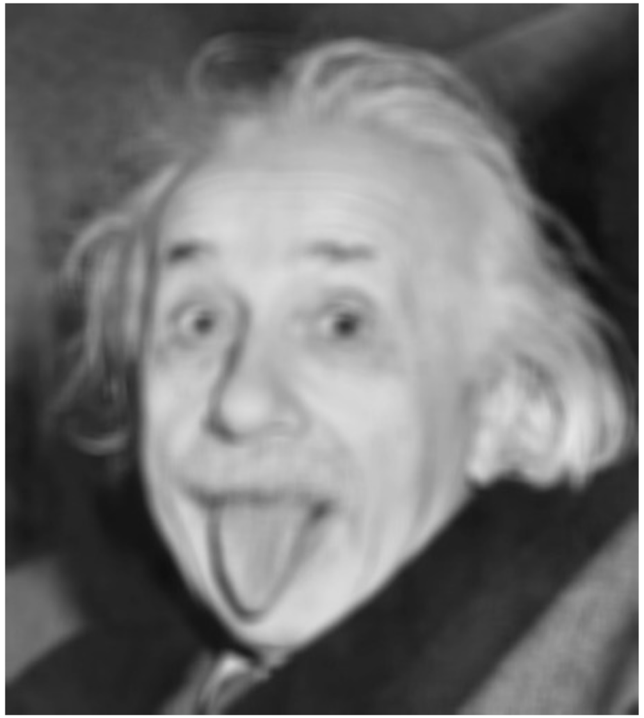
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x} h) \star f$$

Box Filter

- Replace a pixel with a local average (smoothing)

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

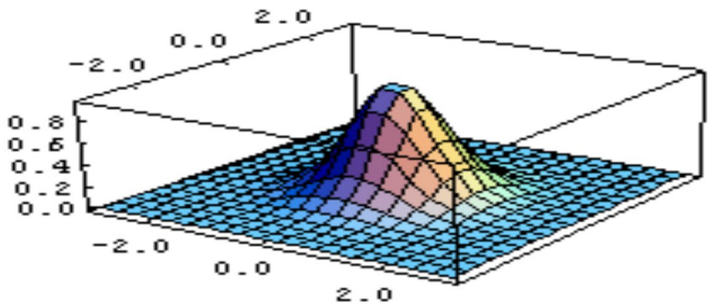




Gaussian Filter

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

Unit: pixels



$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

Standard deviation σ

- Pixels at a distance of more than 3σ are small
- Typical filter dimension $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$
- Large σ , large filter size

Separable Filtering

- A 2D convolution can be performed by a 1D horizontal convolution followed a 1D vertical convolution

$$\mathbf{K} = \mathbf{v}\mathbf{h}^T$$

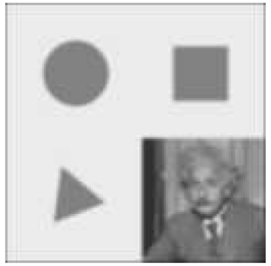
$$n \times n \qquad n \times 1 \qquad 1 \times n$$

Outer product

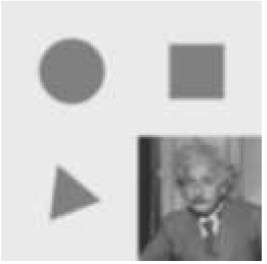
Separable Filtering

$$\frac{1}{K^2}\begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline 1 & 1 & \cdots & 1 \\ \hline \vdots & \vdots & 1 & \vdots \\ \hline 1 & 1 & \cdots & 1 \\ \hline \end{array} \qquad \frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \qquad \frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

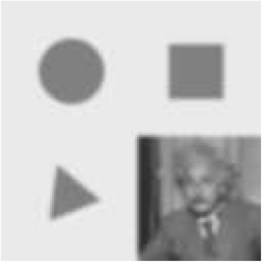
$$\frac{1}{K}\begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline \end{array} \qquad \frac{1}{4}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline \end{array} \qquad \frac{1}{16}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$



(a) box, $K = 5$

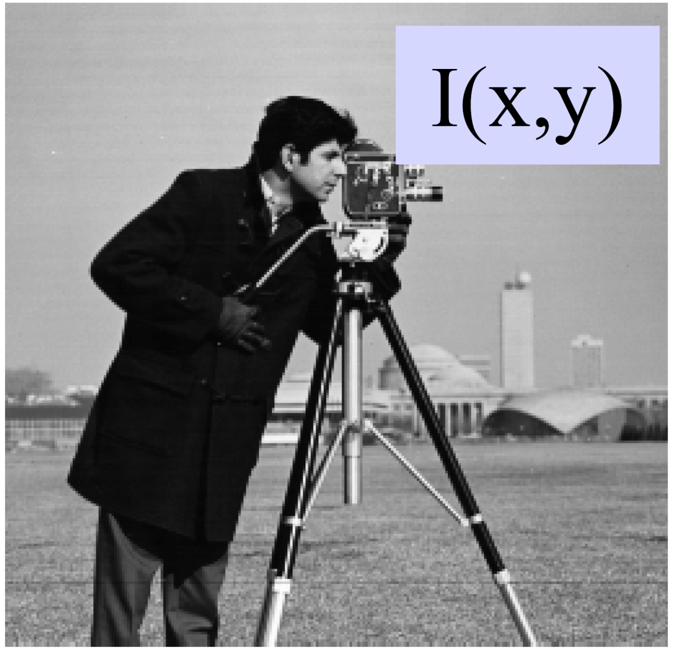


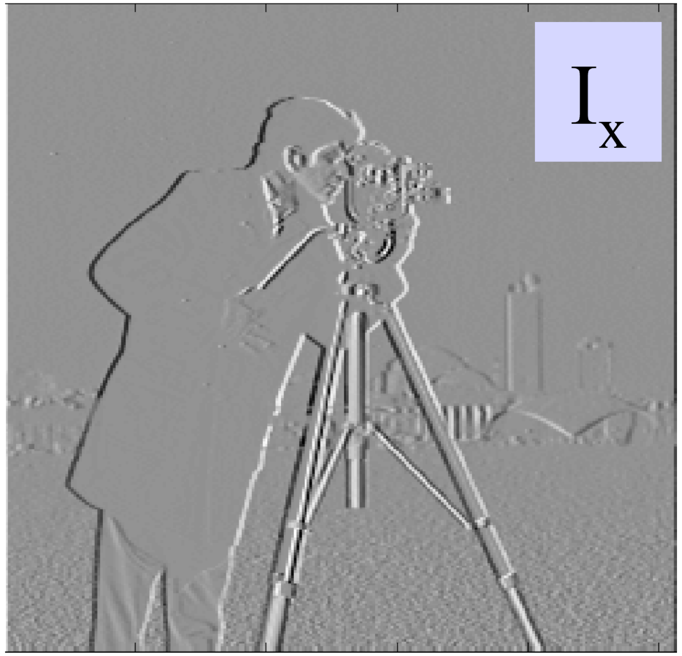
(b) bilinear



(c) “Gaussian”

Image Gradient





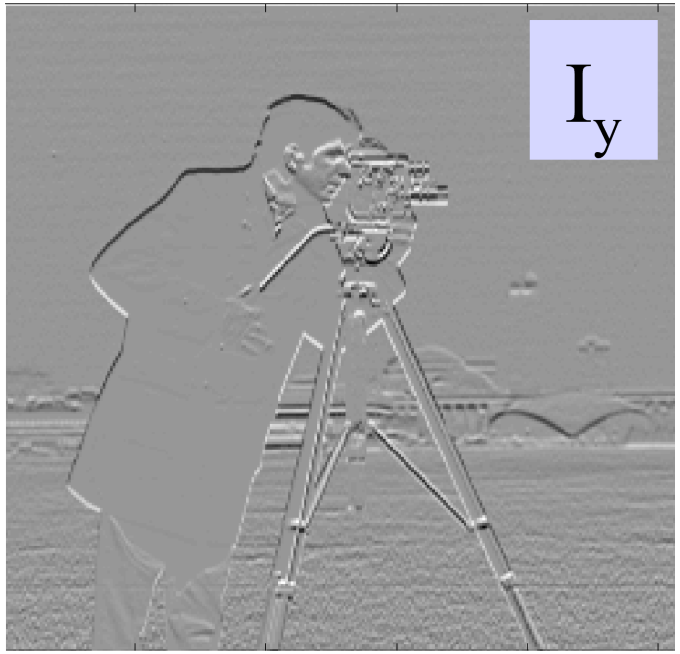


Image Gradient

- Derivative of a function

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Central difference is more accurate

$$f'(x) = \lim_{h \to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$$

- Image gradient with central difference
  - Applying a filter <-> a convolutional kernel

1	0	-1

X derivative

1
0
-1

Y derivative

Image Gradient

- Sobel Filter

1	0	-1
2	0	-2
1	0	-1

Sobel

=

1
2
1

weighted average and scaling

1	0	-1

x-derivative

$$\boldsymbol{S}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \boldsymbol{S}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

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

Convolution $\frac{\partial \boldsymbol{f}}{\partial x} = \boldsymbol{S}_x \star \boldsymbol{f}$ $\qquad \frac{\partial \boldsymbol{f}}{\partial y} = \boldsymbol{S}_y \star \boldsymbol{f}$

Common Derivative Filters

Sobel

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Scharr

3	0	-3
10	0	-10
3	0	-3

3	10	3
0	0	0
-3	-10	-3

Prewitt

1	0	-1
1	0	-1
1	0	-1

1	1	1
0	0	0
-1	-1	-1

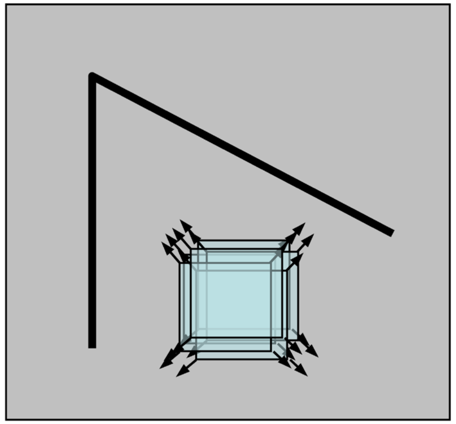
Roberts

0	1
-1	0

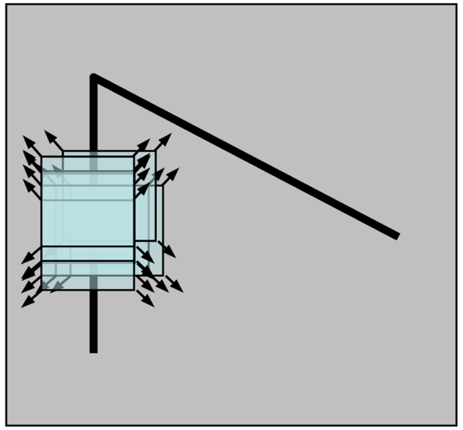
1	0
0	-1

Harris Corner Detector

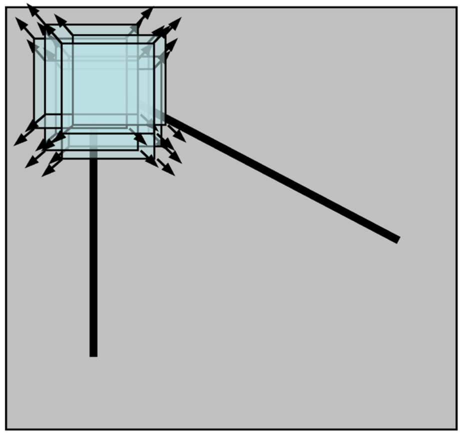
- Corners are regions with large variation in intensity in all directions



"flat" region:
no change in
all directions

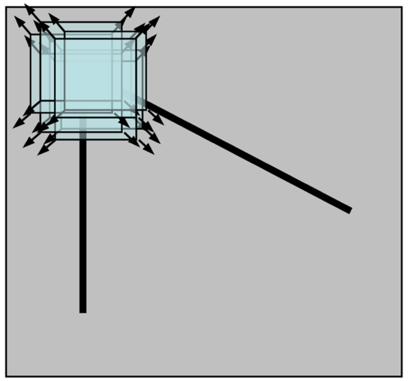


"edge":
no change
along the edge
direction



"corner":
significant
change in all
directions

Harris Corner Detector



Grayscale image $I(x, y)$

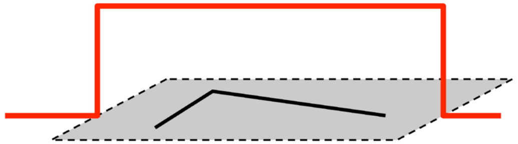
Image patch inside the window

$$f(\Delta x, \Delta y) = \sum_{x_k, y_k} w(x_k, y_k)\left(I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y)\right)^2$$

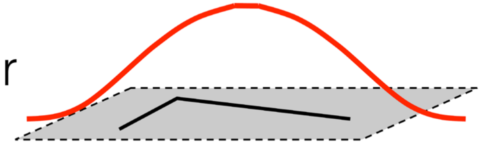
sum of squared differences (SSD)

Shift (offset)

Window function



or



1 in window, 0 outside

Gaussian

Idea: if $f(\Delta x, \Delta y)$ is large for all $(\Delta x, \Delta y)$, the patch has a corner

Harris Corner Detector

- Taylor series

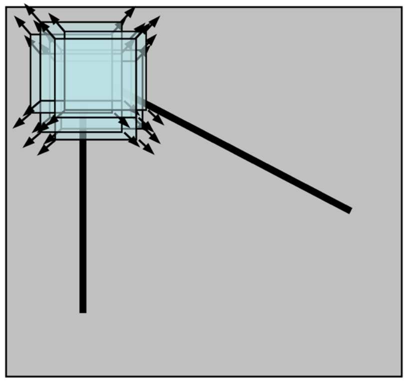
One dimension about x_0

$$f(x_0 + \Delta x) = f(x_0) + \Delta x \, f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) + \ldots$$

Two dimension about (x, y)

$$f(x + \Delta x, y + \Delta y) = f(x, y) + \left[f_x(x, y)\,\Delta x + f_y(x, y)\,\Delta y\right] + \frac{1}{2!}\left[(\Delta x)^2 f_{xx}(x, y) + 2\,\Delta x\,\Delta y\, f_{xy}(x, y) + (\Delta y)^2 f_{yy}(x, y)\right] + \frac{1}{3!}\left[(\Delta x)^3 f_{xxx}(x, y) + 3\,(\Delta x)^2 \Delta y\, f_{xxy}(x, y) + 3\,\Delta x\,(\Delta y)^2 f_{xyy}(x, y) + (\Delta y)^3 f_{yyy}(x, y)\right] + \ldots$$

Harris Corner Detector



Sum of squared differences $f(\Delta x, \Delta y) = \sum_{x_k, y_k} w(x_k, y_k)(I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$

First order approximation $I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$

X derivative Y derivative

$$f(\Delta x, \Delta y) \approx \sum_{x,y} w(x, y)(I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2$$

$$f(\Delta x, \Delta y) \approx (\Delta x \quad \Delta y) M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \qquad M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{x,y} w(x, y) I_x^2 & \sum_{x,y} w(x, y) I_x I_y \\ \sum_{x,y} w(x, y) I_x I_y & \sum_{x,y} w(x, y) I_y^2 \end{bmatrix}$$

Idea: if $f(\Delta x, \Delta y)$ is large for all $(\Delta x, \Delta y)$, the patch has a corner

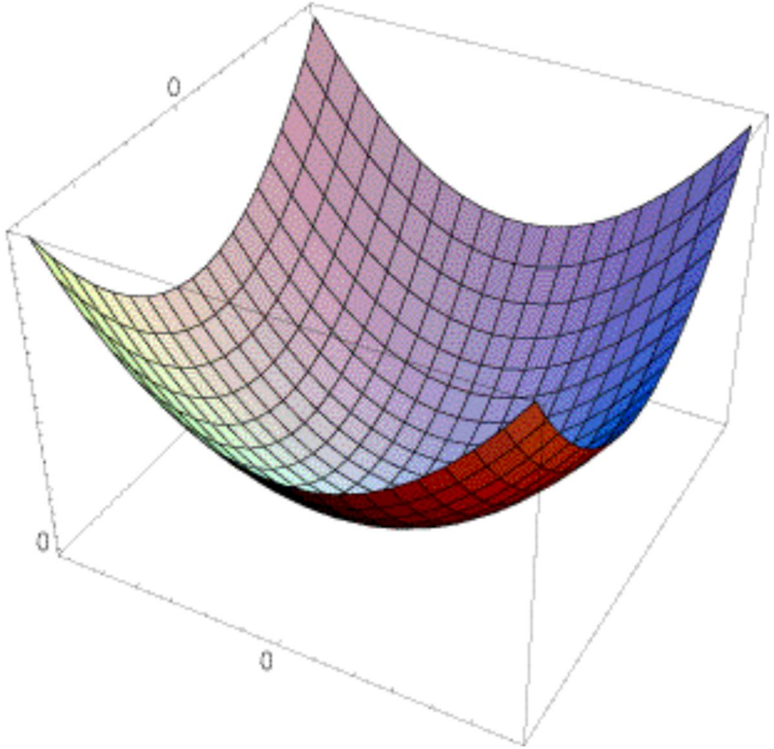
Harris Corner Detector

- A quadratic function

$$f(\Delta x, \Delta y) \approx \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

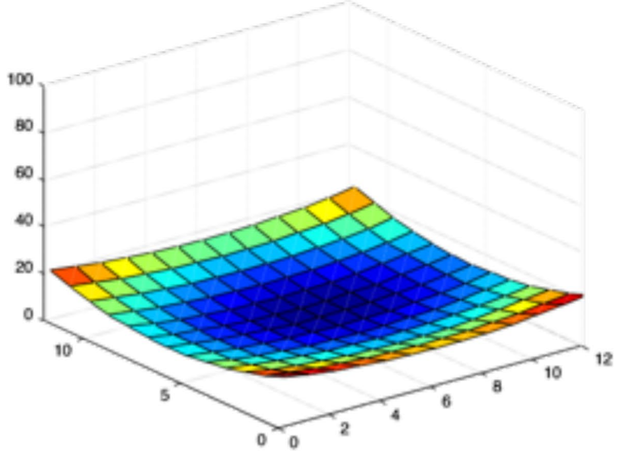
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{x,y} w(x,y) I_x^2 & \sum_{x,y} w(x,y) I_x I_y \\ \sum_{x,y} w(x,y) I_x I_y & \sum_{x,y} w(x,y) I_y^2 \end{bmatrix}$$

Gradient covariance matrix

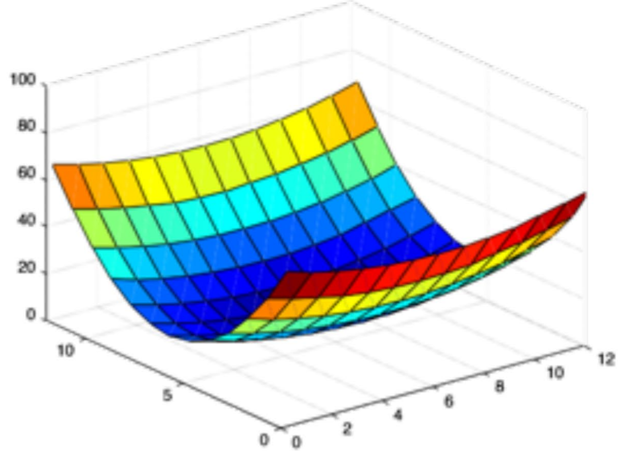


Harris Corner Detector

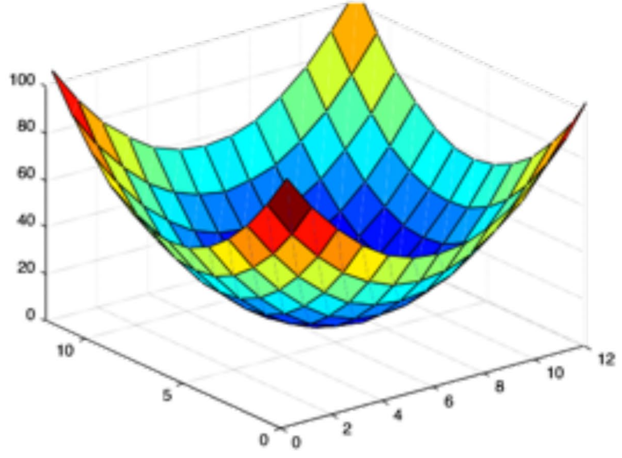
- A quadratic function $$f(\Delta x, \Delta y) \approx (\Delta x \quad \Delta y) M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$



Flat



Edge



Corner

Idea: if $f(\Delta x, \Delta y)$ is large for all $(\Delta x, \Delta y)$, the patch has a corner

Harris Corner Detector

- Compute the eigenvalues and eigenvectors of M

eigenvalue

$$M\boldsymbol{e} = \lambda \boldsymbol{e}$$

eigenvector

Eigenvalues: find the roots of $\det(M - \lambda I) = 0$

Eigenvectors: for each eigenvalue, solve $(M - \lambda I)\boldsymbol{e} = 0$

Harris Corner Detector

- Real symmetric matrices
  - All eigenvalues of a real symmetric matrix are real
  - Eigenvectors corresponding to distinct eigenvalues are orthogonal

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

- Since M is symmetric, we have

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

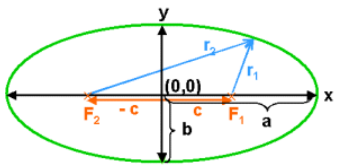
Harris Corner Detector

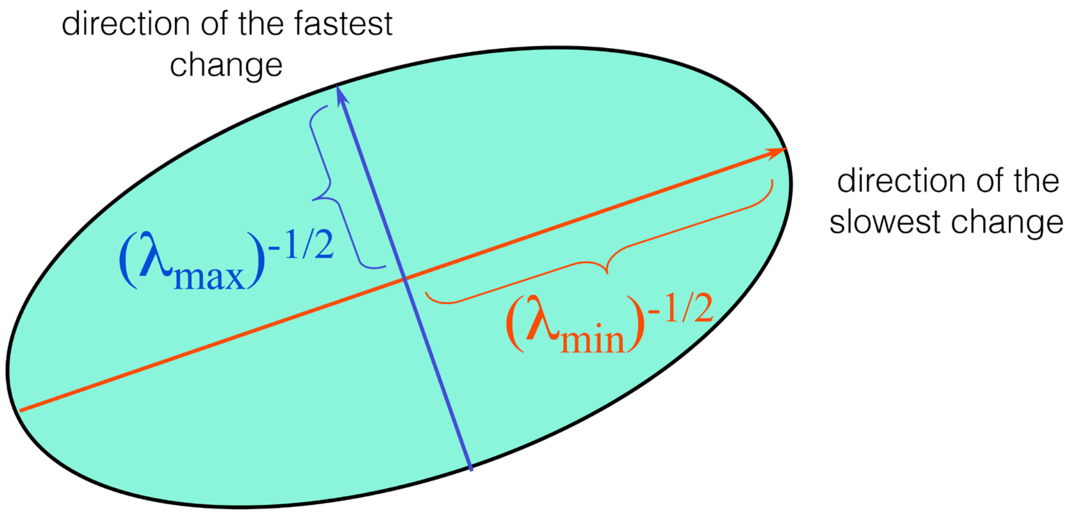
- Since M is symmetric, we have $M = R^{-1}\begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}R$
- We can visualize M as ellipse with axis lengths determined by the eigenvalues and orientation determined by R

Ellipse equation:

$$[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$





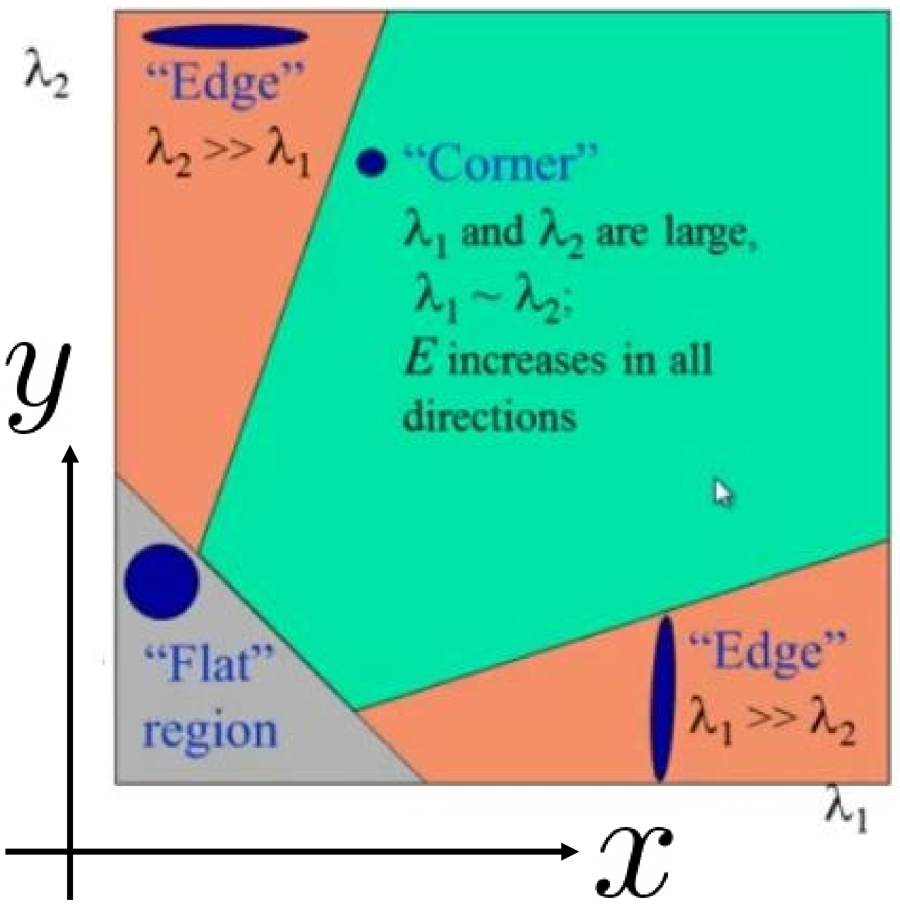
Harris Corner Detector

- Interpreting Eigenvalues

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

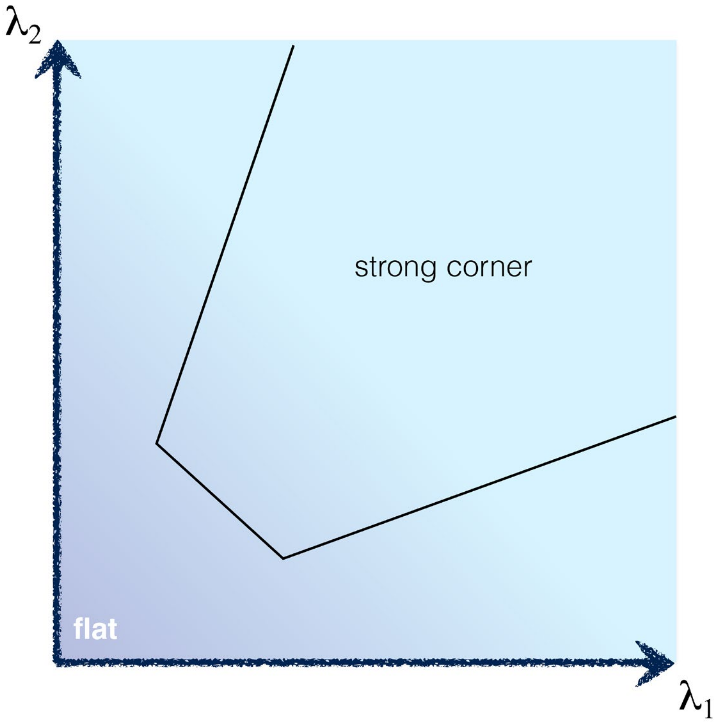
$$f(\Delta x, \Delta y) \approx (\Delta x \quad \Delta y) M \begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

λ_1 X direction gradient λ_2 Y direction gradient



Harris Corner Detector

- Define a score to detect corners



Option 1 Kanade & Tomasi (1994)

$$R = \min(\lambda_1, \lambda_2)$$

Option 2 Harris & Stephens (1988)

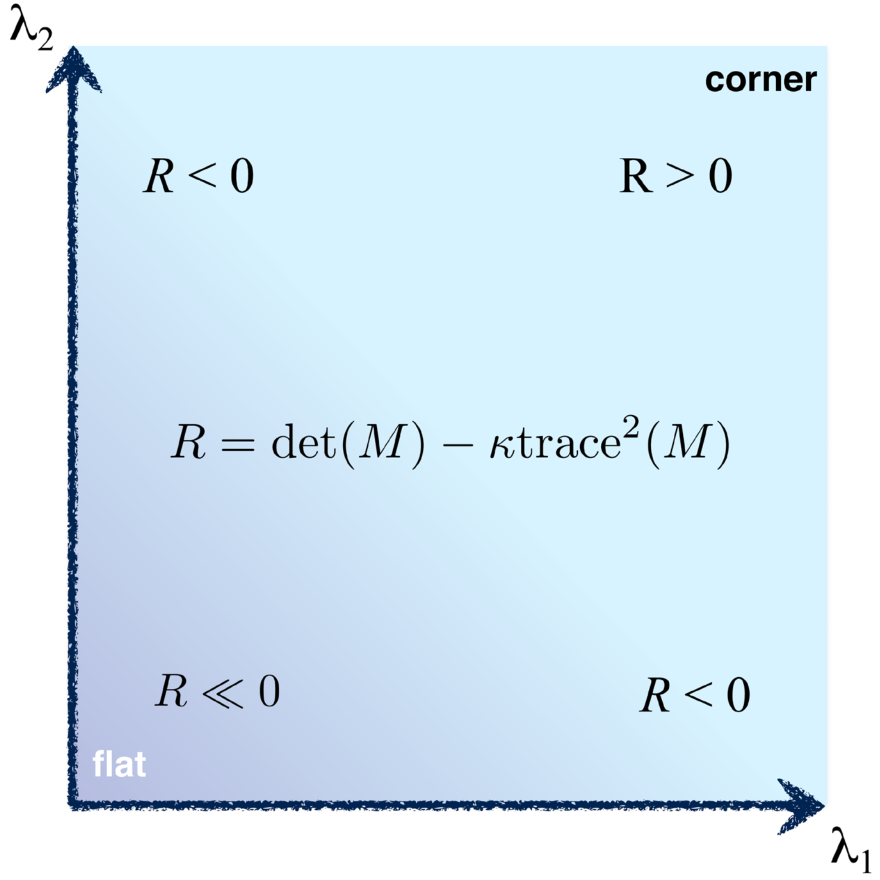
$$R = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$$

Can compute this more efficiently…

Harris Corner Detector

- Define a score to detect corners

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



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

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$

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

$$\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$$
$$\text{tr}(\mathbf{P}^{-1}\mathbf{AP}) = \text{tr}(\mathbf{APP}^{-1}) = \text{tr}(\mathbf{A})$$

Harris Corner Detector

1. Compute x and y derivatives of image

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

Sobel filter

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x \qquad I_{y^2} = I_y \cdot I_y \qquad I_{xy} = I_x \cdot I_y$$

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

Gaussian window

$$S_{x^2} = G_{\sigma'} * I_{x^2} \qquad S_{y^2} = G_{\sigma'} * I_{y^2} \qquad S_{xy} = G_{\sigma'} * I_{xy}$$

Harris Corner Detector

3. Determine the matrix at every pixel

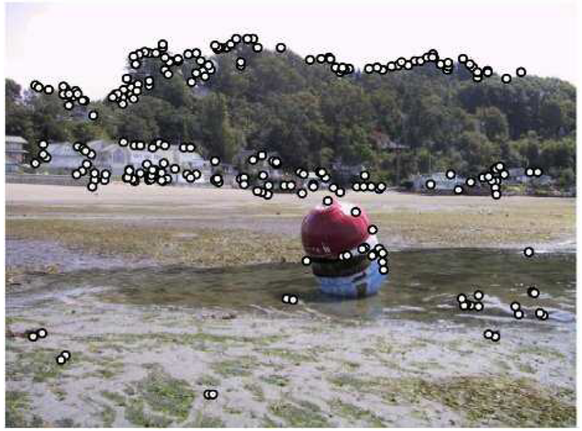
$$M(x,y) = \begin{bmatrix} S_{x^2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y^2}(x,y) \end{bmatrix}$$

4. Compute the response of the detector at each pixel

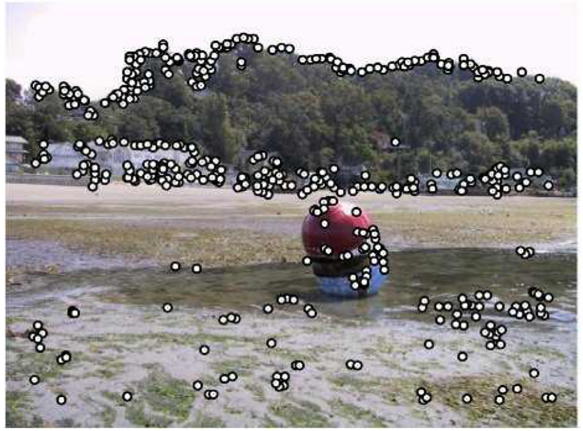
$$R = \det M - k(\text{trace} M)^2$$

5. Threshold on R and perform non-maximum suppression

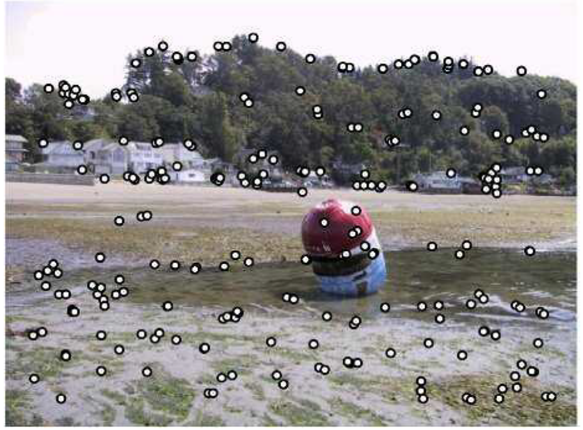
Non-Maximum Suppression (NMS)



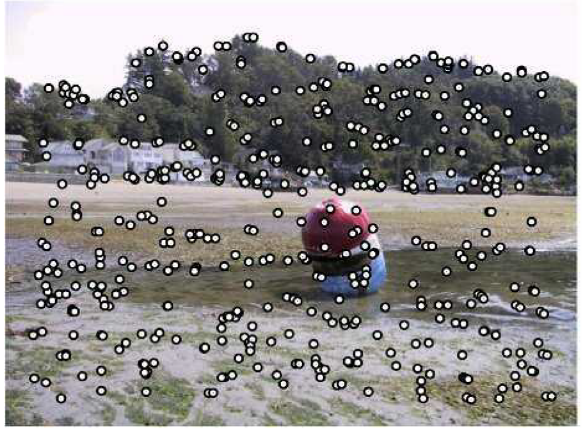
(a) Strongest 250



(b) Strongest 500

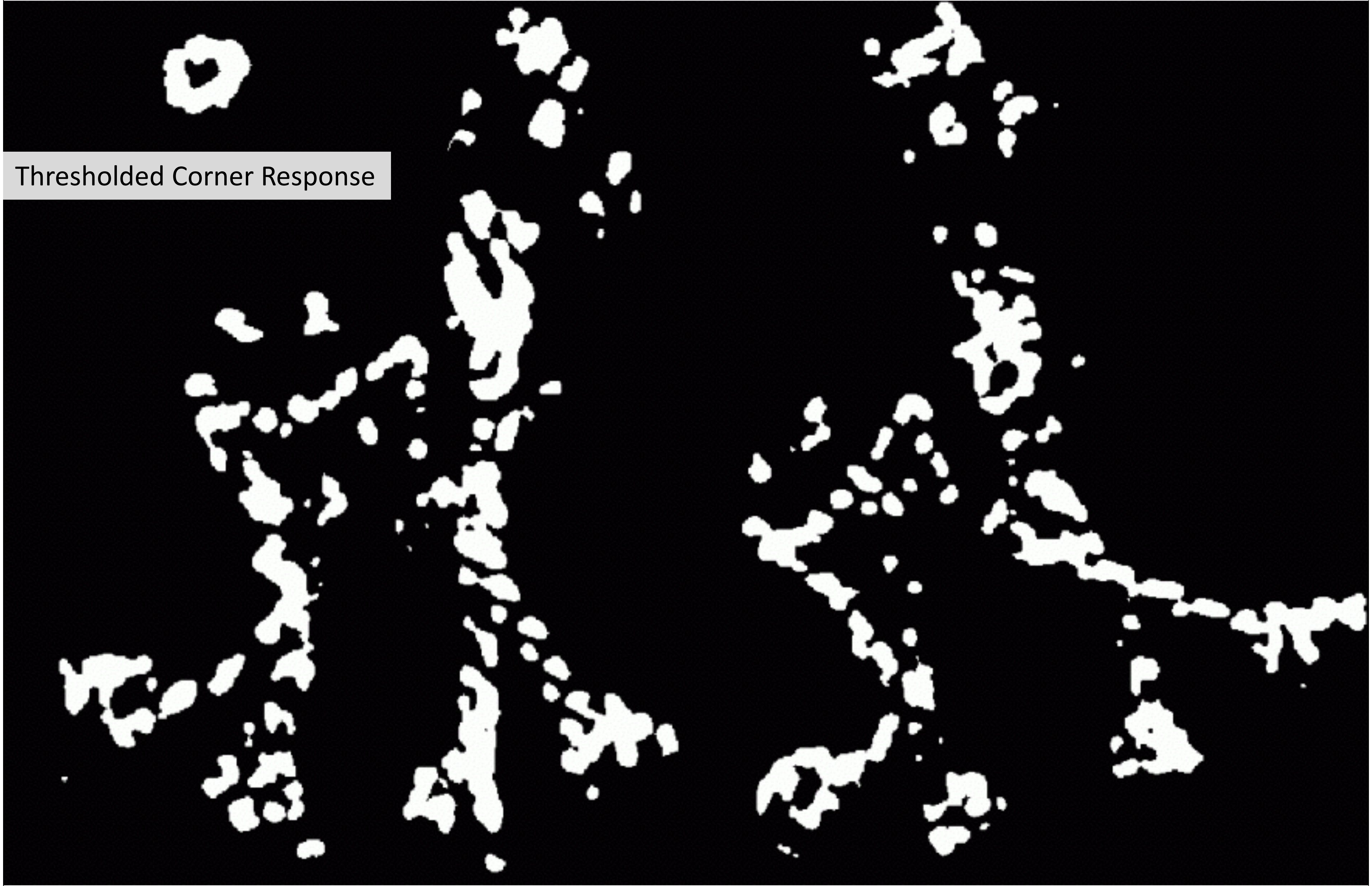


(c) ANMS 250, $r = 24$



(d) ANMS 500, $r = 16$

Adaptive non-maximal suppression
Suppression radius r



Thresholded Corner Response


NMS

Further Reading

- Section 3.2, 7.1, Computer Vision, Richard Szeliski
- A COMBINED CORNER AND EDGE DETECTOR. Chris Harris & Mike Stephens. http://www.bmva.org/bmvc/1988/avc-88-023.pdf