Lecture 4

Derivatives and edges

Administrative

A1 is out

- It is graded
- Due **April 18**

Administrative

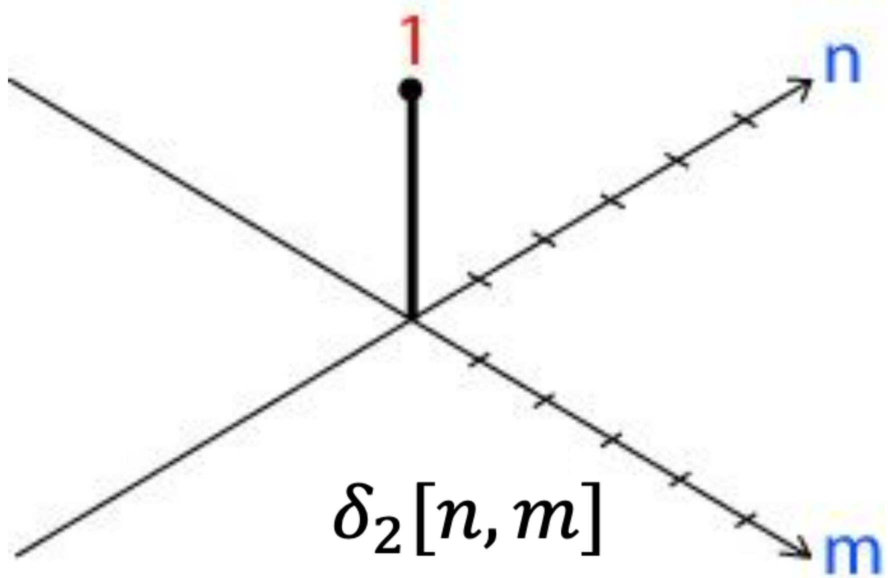
Recitation

- Friday 1:30-2:20pm @ BAG 154

This week:
We will go over Python & Numpy basics
- will have polls
- prep for final exam

So far: 2D impulse function

- A special function
- 1 at the origin [0,0].
- 0 everywhere else



So far: We get the impulse response when we pass an impulse function through a LSI system

- The moving average filter equation again: $g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$

$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

Pass in an impulse function — Record its response

$$h[n,m] = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & \underline{1/9} & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

So far: write down f as a sum of impulses

Let's say our input f is a 3x3 image:

f[0,0]	f[0,1]	f[1,1]
f[1,0]	f[1,1]	f[1,2]
f[2,0]	f[2,1]	f[2,2]

=

f[0,0]	0	0
0	0	0
0	0	0

+

0	f[0,1]	0
0	0	0
0	0	0

+ … +

0	0	0
0	0	0
0	0	f[2,2]

= f[0,0] ×

1	0	0
0	0	0
0	0	0

+ f[0,1] ×

0	1	0
0	0	0
0	0	0

+ … + f[2,2] ×

0	0	0
0	0	0
0	0	1

$$= f[0,0] \cdot \delta_2[n,m] + f[0,1] \cdot \delta_2[n,m-1] + \ldots + f[2,2] \cdot \delta_2[n-2,m-2]$$

So far: write down f as a sum of impulses

- Superposition:

$$S\{\alpha f_1[n,m] + \beta f_2[n,m]\} = \alpha S\{f_1[n,m]\} + \beta S\{f_2[n,m]\}$$

$$\mathcal{S}[\sum_i \alpha_i f_i[n,m]] = \sum_i \alpha_i \mathcal{S}[f_i[n,m]]$$

- We can now use superposition to see what the output g is:

$$f[n,m] \xrightarrow{S} g[n,m]$$

$$f[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \delta_2[n-k, m-l]$$

$$\xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot S\{\delta_2[n-k, m-l]\}$$

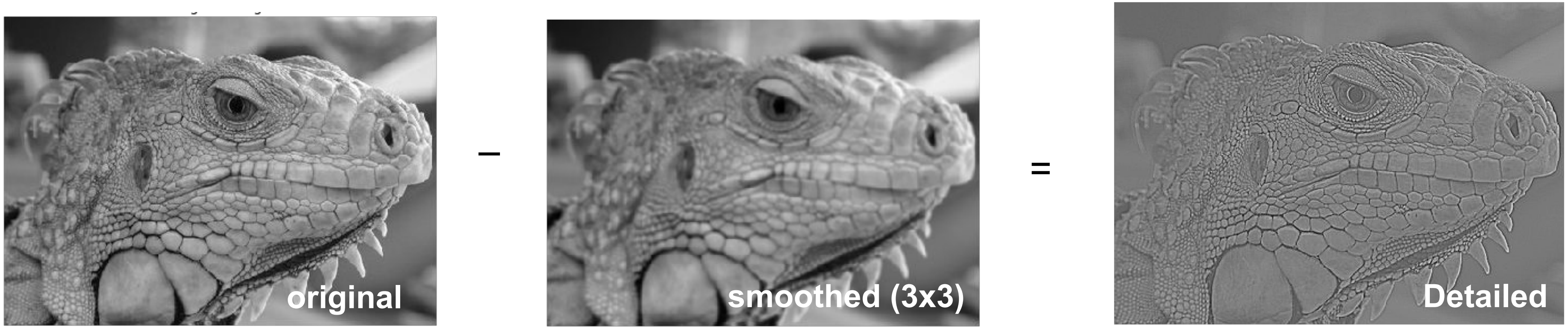
So far: We derived convolutions

- An LSI system is completely specified by its impulse response.
    - For any input f, we can compute the output g in terms of the impulse response h.

Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

So far: We created a sharpening system by combining filters



Let's add it back to get a sharpening system:

(Cross) correlation – symbol: $**$

Cross correlation of two 2D signals f[n,m] and h[n,m]

$$f[n,m] ** h[n,m] = \sum_{k} \sum_{l} f[k,l] h[n+k, m+l]$$

Equivalent to a convolution without the flip

Today’s agenda

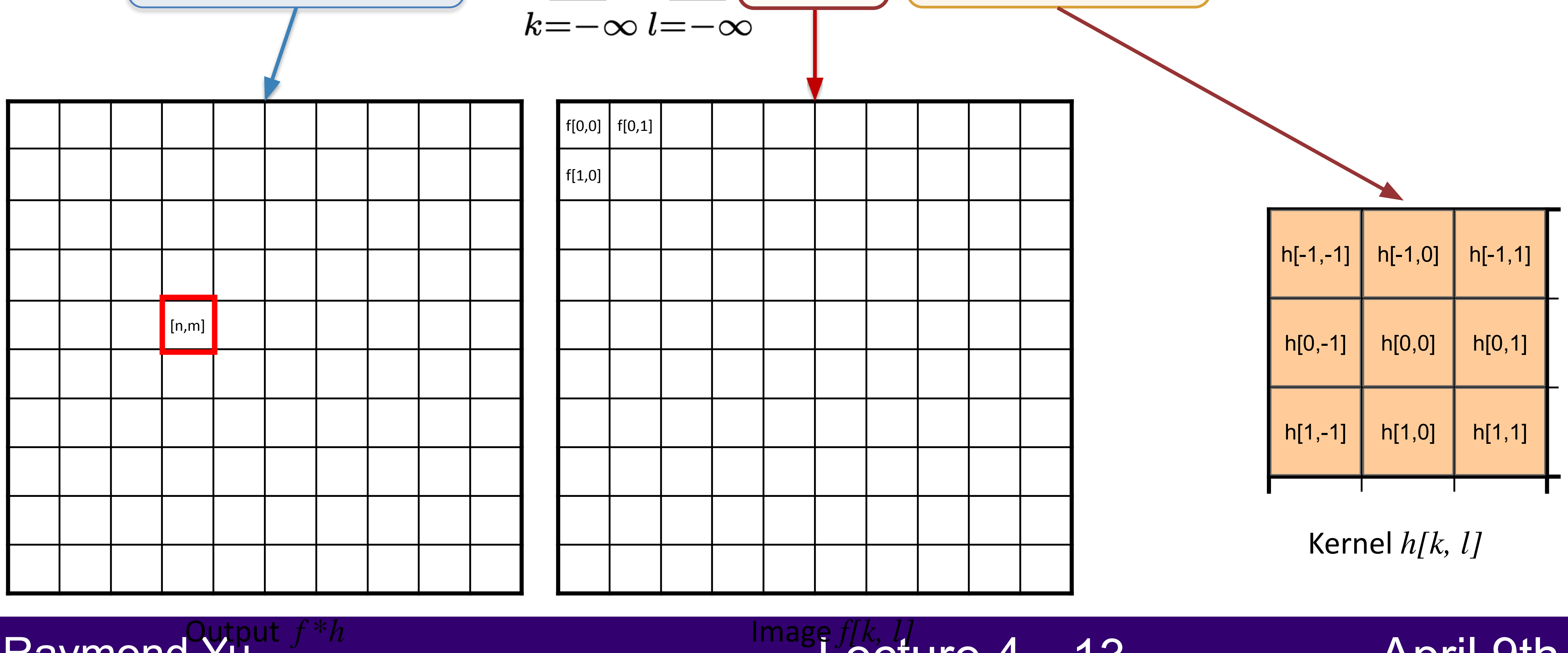
- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

Today's agenda

- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

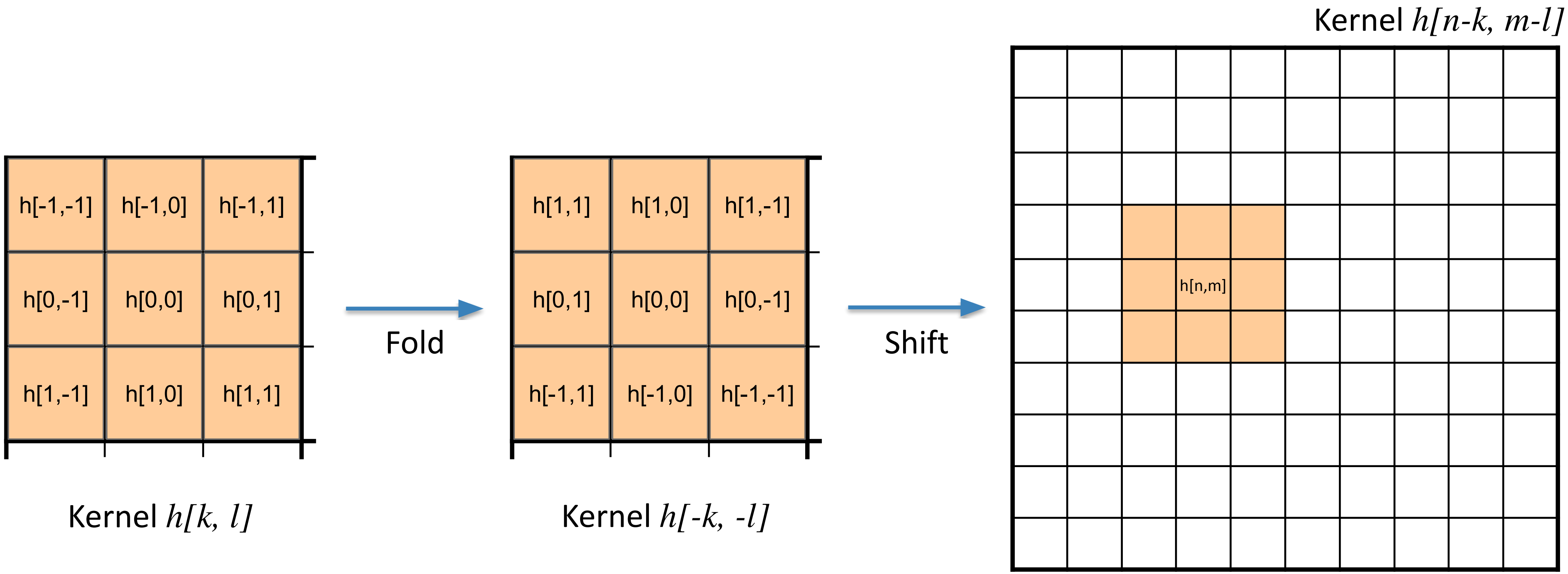
2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



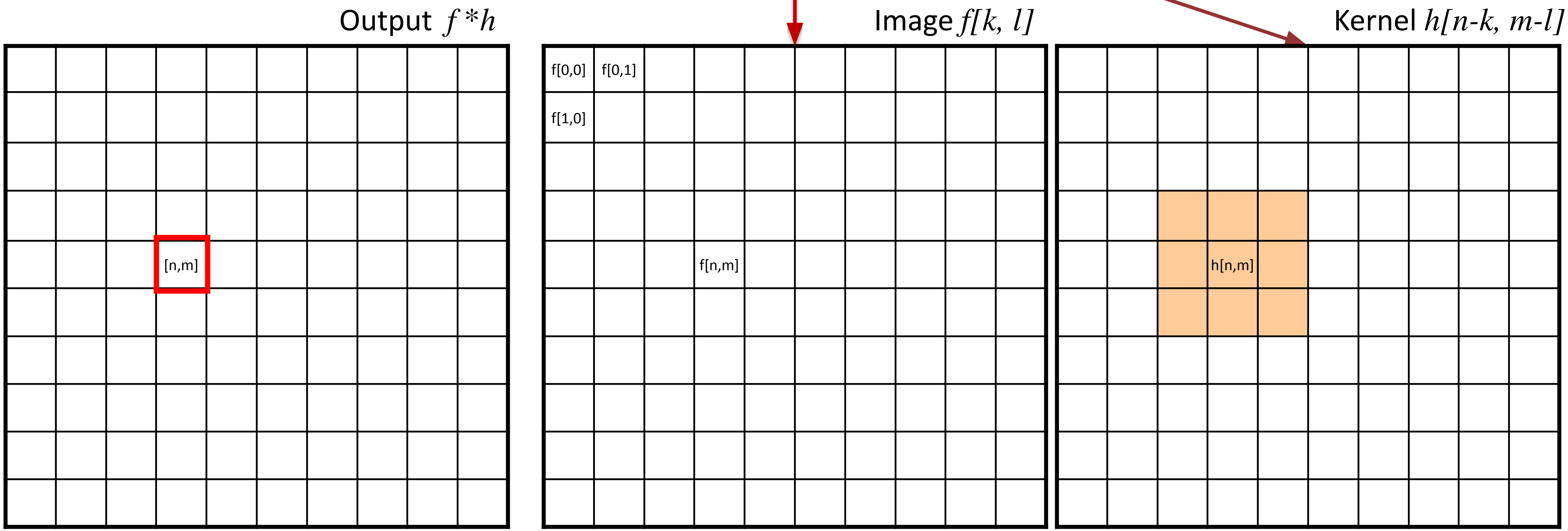
2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



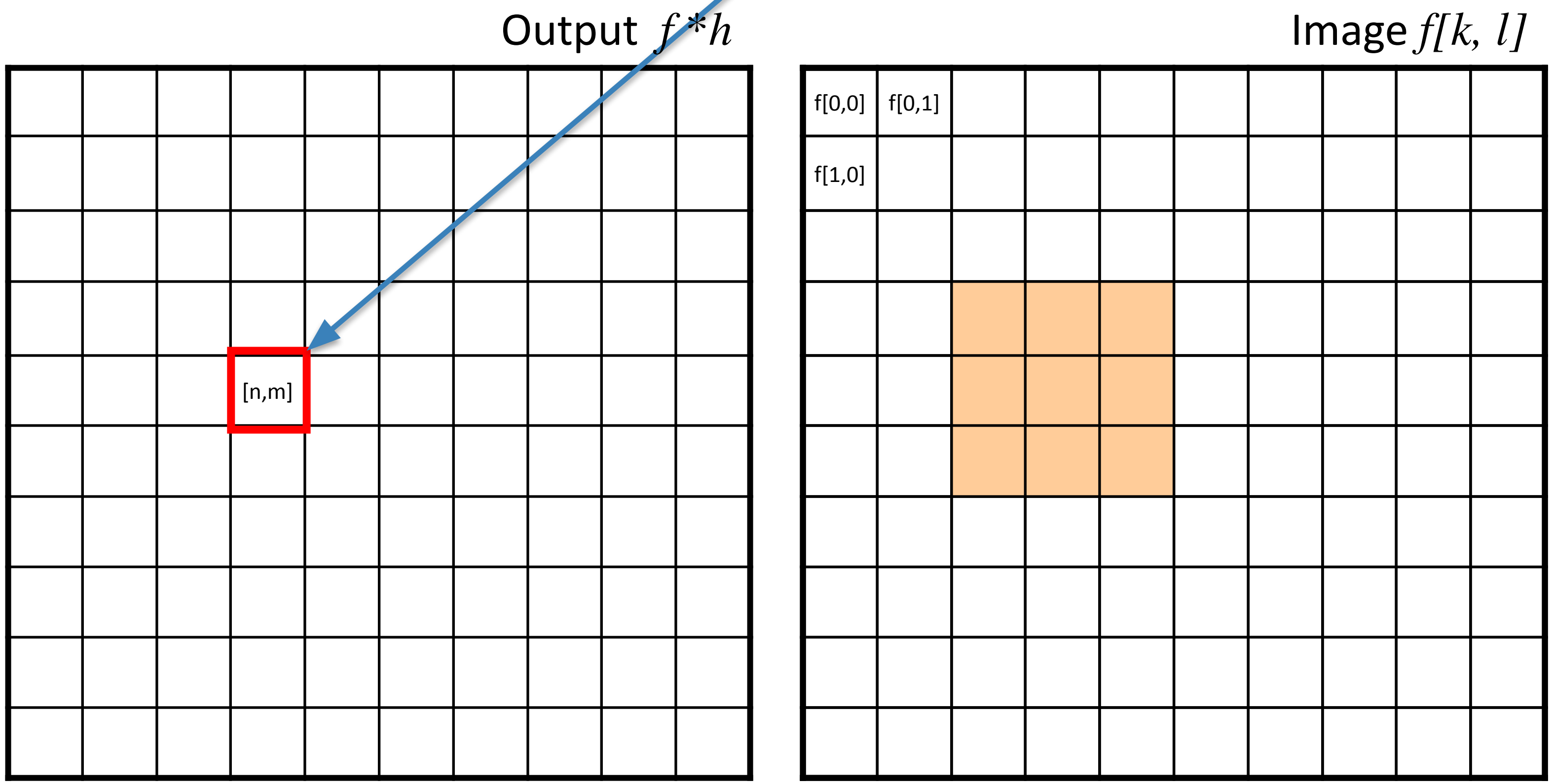
2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



2D Discrete Convolution

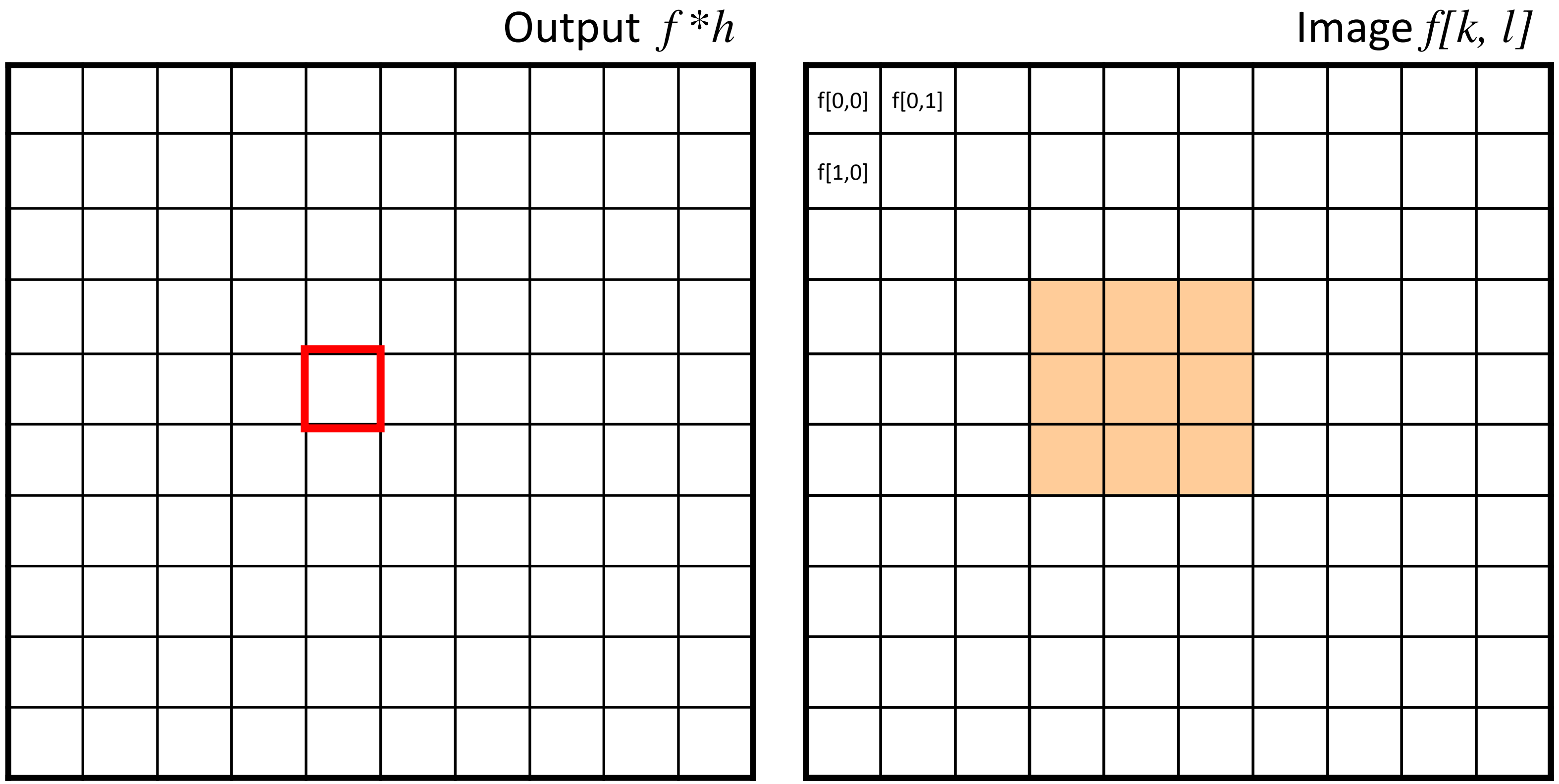
$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



Element-wise multiplication

Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

Output $f * h$

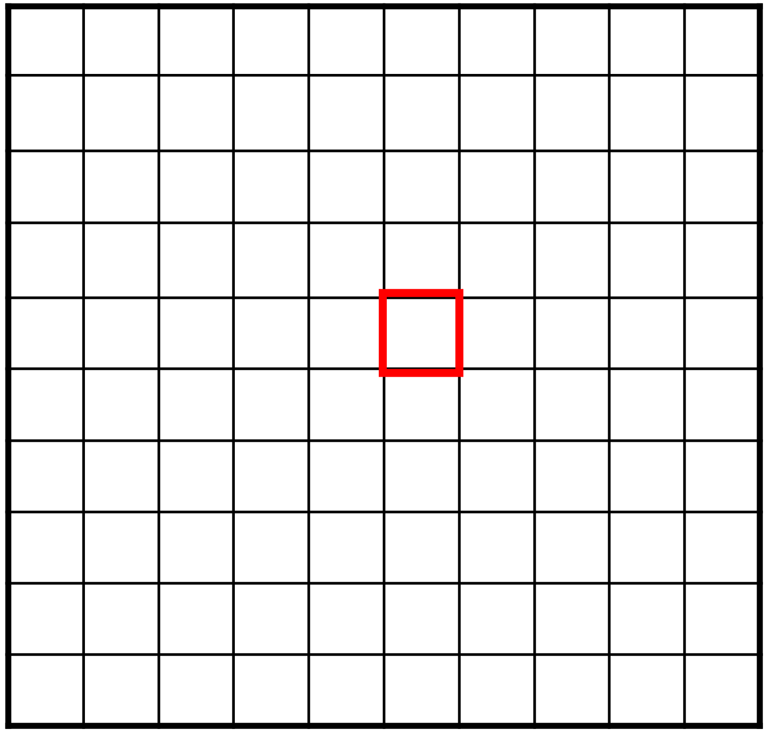
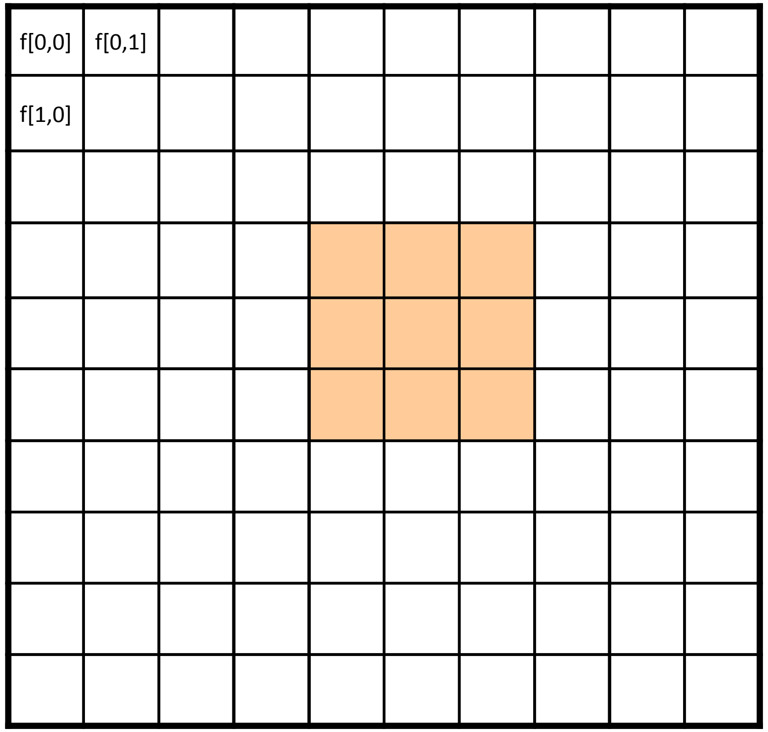


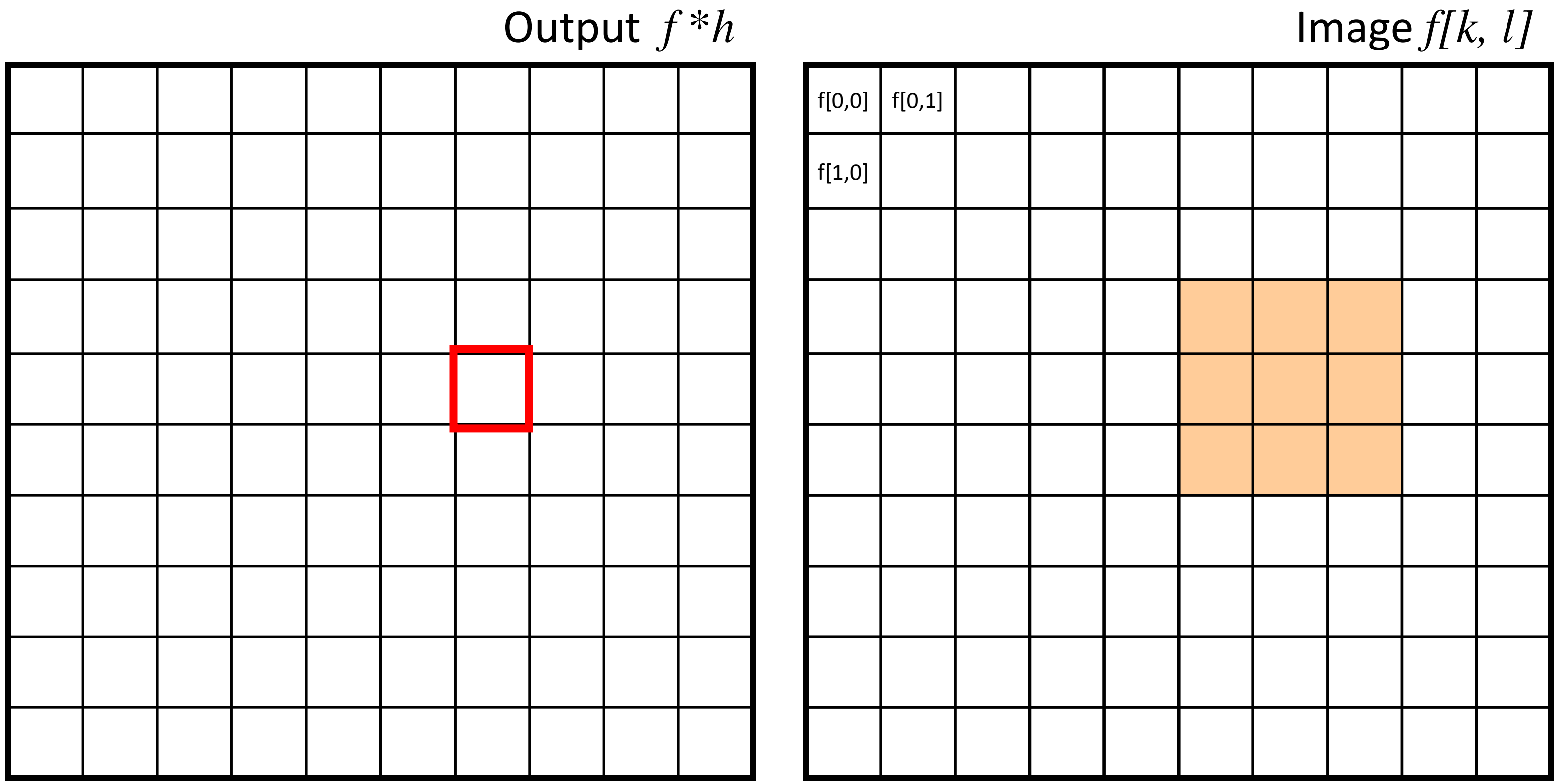
Image $f[k, l]$



Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

- $f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$

Algorithm:

- Fold $h[k,l]$ about origin to form $h[-k,-l]$
- Shift the folded results by n, m to form $h[n-k, m-l]$
- Multiply $h[n-k, m-l]$ by $f[k, l]$
- Sum over all k, l, store result in output position $[n, m]$
- Repeat for every n, m

2D convolution example

1	2	3
4	5	6
7	8	9

Input

n \ m	-1	0	1
-1	-1	-2	-1
0	0	0	0
1	1	2	1

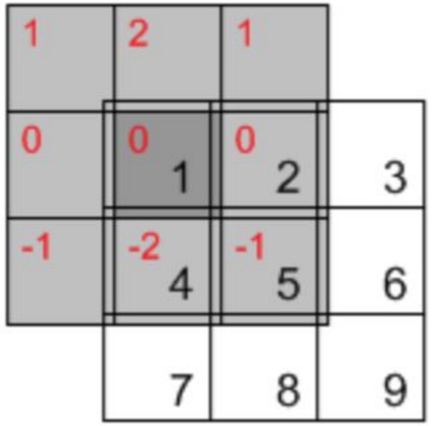
Kernel

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



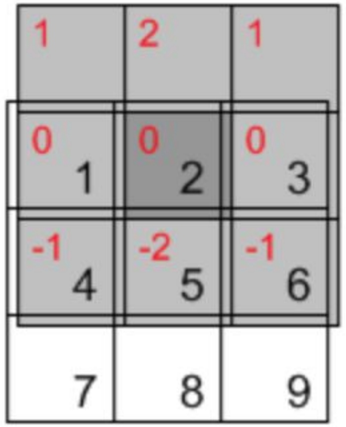
$$
\begin{aligned}
&= x[-1,-1]\cdot h[1,1] + x[0,-1]\cdot h[0,1] + x[1,-1]\cdot h[-1,1] \\
&\quad + x[-1,0]\cdot h[1,0] + x[0,0]\cdot h[0,0] + x[1,0]\cdot h[-1,0] \\
&\quad + x[-1,1]\cdot h[1,-1] + x[0,1]\cdot h[0,-1] + x[1,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 0\cdot 0 + 1\cdot 0 + 2\cdot 0 + 0\cdot(-1) + 4\cdot(-2) + 5\cdot(-1) = -13
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



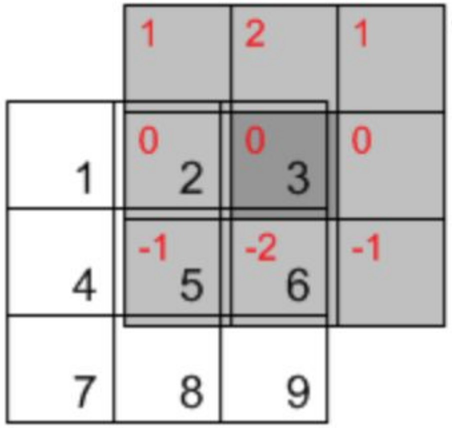
$$
\begin{aligned}
&= x[0,-1]\cdot h[1,1] + x[1,-1]\cdot h[0,1] + x[2,-1]\cdot h[-1,1] \\
&\quad + x[0,0]\cdot h[1,0] + x[1,0]\cdot h[0,0] + x[2,0]\cdot h[-1,0] \\
&\quad + x[0,1]\cdot h[1,-1] + x[1,1]\cdot h[0,-1] + x[2,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 1\cdot 0 + 2\cdot 0 + 3\cdot 0 + 4\cdot(-1) + 5\cdot(-2) + 6\cdot(-1) = -20
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



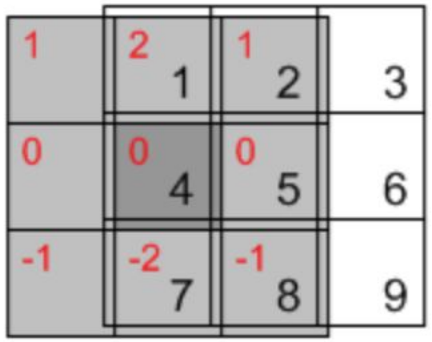
$$
\begin{aligned}
&= x[1,-1]\cdot h[1,1] + x[2,-1]\cdot h[0,1] + x[3,-1]\cdot h[-1,1] \\
&\quad + x[1,0]\cdot h[1,0] + x[2,0]\cdot h[0,0] + x[3,0]\cdot h[-1,0] \\
&\quad + x[1,1]\cdot h[1,-1] + x[2,1]\cdot h[0,-1] + x[3,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 2\cdot 0 + 3\cdot 0 + 0\cdot 0 + 5\cdot(-1) + 6\cdot(-2) + 0\cdot(-1) = -17
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



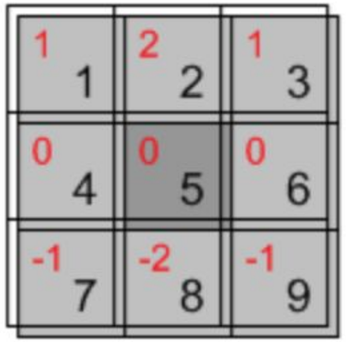
$$= x[-1,0]\cdot h[1,1] + x[0,0]\cdot h[0,1] + x[1,0]\cdot h[-1,1]$$
$$+ x[-1,1]\cdot h[1,0] + x[0,1]\cdot h[0,0] + x[1,1]\cdot h[-1,0]$$
$$+ x[-1,2]\cdot h[1,-1] + x[0,2]\cdot h[0,-1] + x[1,2]\cdot h[-1,-1]$$
$$= 0\cdot 1 + 1\cdot 2 + 2\cdot 1 + 0\cdot 0 + 4\cdot 0 + 5\cdot 0 + 0\cdot(-1) + 7\cdot(-2) + 8\cdot(-1) = -18$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



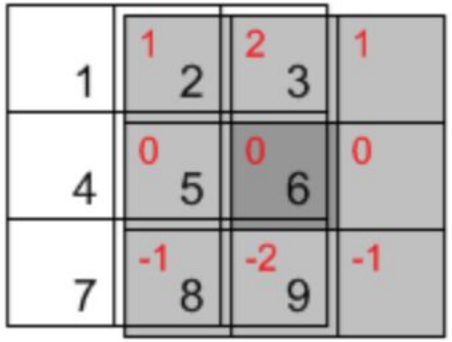
$$= x[0,0]\cdot h[1,1] + x[1,0]\cdot h[0,1] + x[2,0]\cdot h[-1,1]$$
$$+ x[0,1]\cdot h[1,0] + x[1,1]\cdot h[0,0] + x[2,1]\cdot h[-1,0]$$
$$+ x[0,2]\cdot h[1,-1] + x[1,2]\cdot h[0,-1] + x[2,2]\cdot h[-1,-1]$$
$$= 1\cdot 1 + 2\cdot 2 + 3\cdot 1 + 4\cdot 0 + 5\cdot 0 + 6\cdot 0 + 7\cdot(-1) + 8\cdot(-2) + 9\cdot(-1) = -24$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



$$\begin{aligned} &= x[1,0]\cdot h[1,1] + x[2,0]\cdot h[0,1] + x[3,0]\cdot h[-1,1] \\ &\quad + x[1,1]\cdot h[1,0] + x[2,1]\cdot h[0,0] + x[3,1]\cdot h[-1,0] \\ &\quad + x[1,2]\cdot h[1,-1] + x[2,2]\cdot h[0,-1] + x[3,2]\cdot h[-1,-1] \\ &= 2\cdot 1 + 3\cdot 2 + 0\cdot 1 + 5\cdot 0 + 6\cdot 0 + 0\cdot 0 + 8\cdot(-1) + 9\cdot(-2) + 0\cdot(-1) = -18 \end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

Practice with convolution

Original

*

0	0	0
0	1	0
0	0	0

= ?

Practice with convolution

Original

*

0	0	0
0	1	0
0	0	0

=

Filtered
(no change)

Practice with convolution

Original

*

0	0	0
0	0	1
0	0	0

= ?

Practice with convolution

Original

$*$

0	0	0
0	0	1
0	0	0

=

Shifted right
By 1 pixel

Practice with convolution

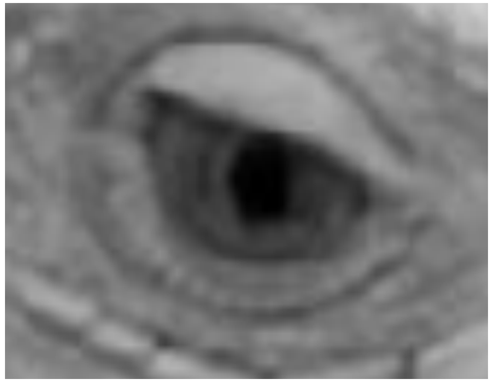
Original

$$* \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = ?$$

Practice with convolution

Original

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



Blurry output

What happens if a system contains multiple filters?

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = ?$$

(Note that filter sums to 1)

What happens if a system contains multiple filters?

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

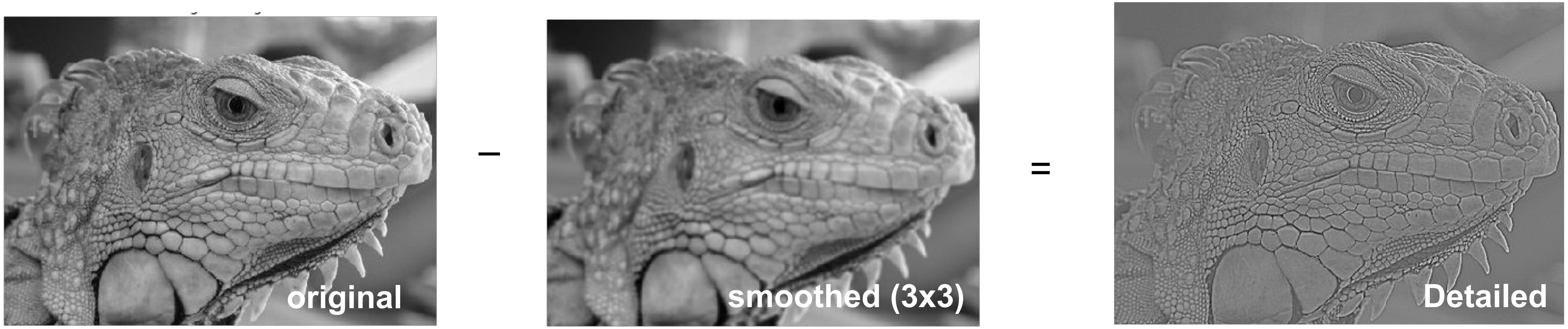
What does blurring take away?

–

=

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)$$

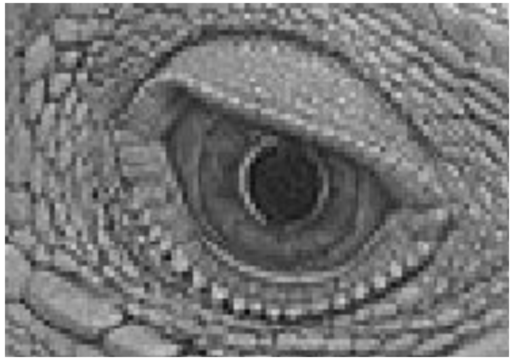
What does blurring take away?



Let's add it back to get a sharpening system:

Convolution in 2D – Sharpening filter

Original



Sharpening system: Accentuates differences with local average

Implementation detail: Image support and edge effect

- A computer will only convolve **finite support signals.**
  - That is: images that are zero for n,m outside some rectangular region

- numpy's convolution performs 2D convolution of finite-support signals.

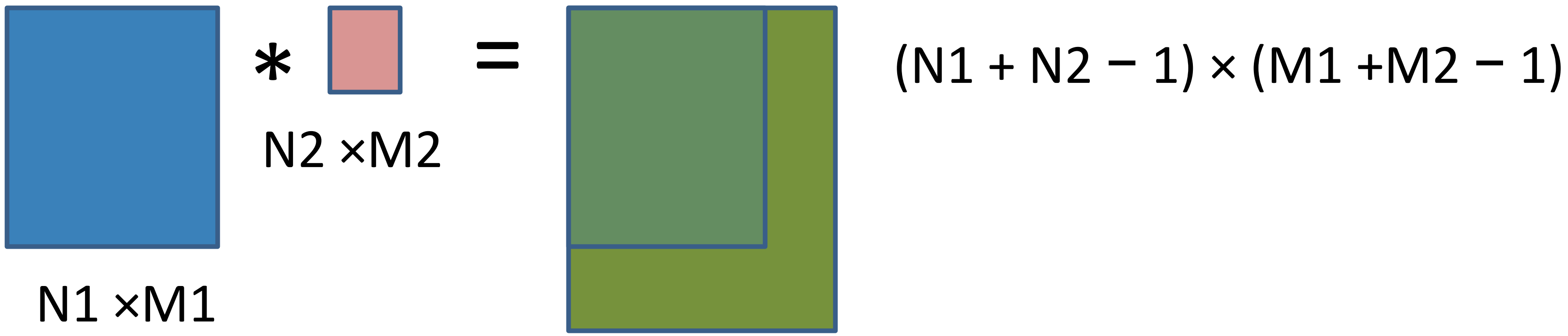
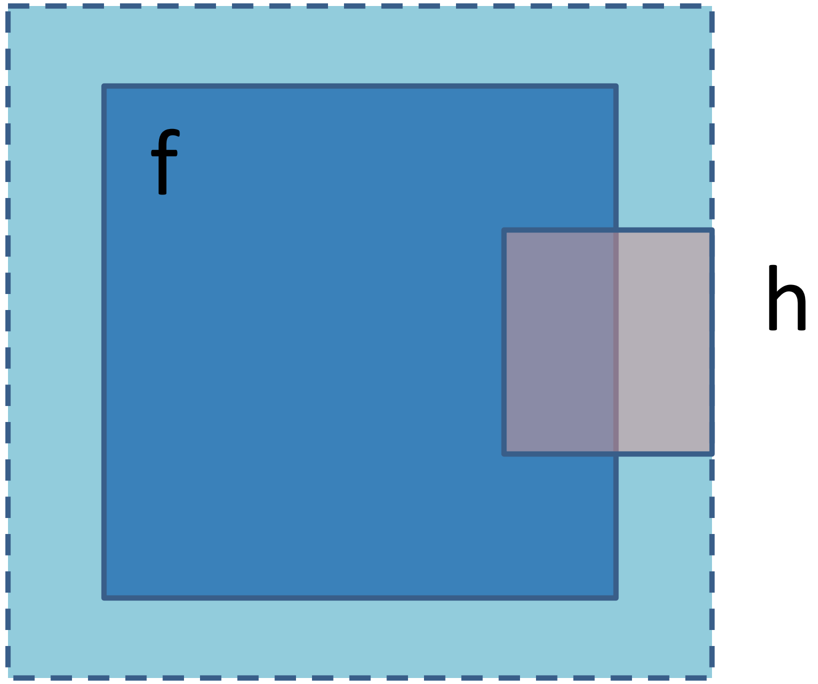


Image support and edge effect

- A computer will only convolve **finite support signals.**
- What happens at the edge?



- zero "padding"
- edge value replication
- mirror extension
- more (beyond the scope of this class)

Today's agenda

- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

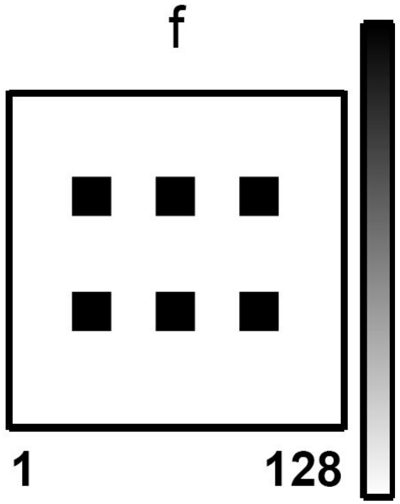
(Cross) correlation – symbol: $**$

Cross correlation of two 2D signals f[n,m] and h[n,m]

$$f[n,m] ** h[n,m] = \sum_k \sum_l f[k,l] h[n+k, m+l]$$

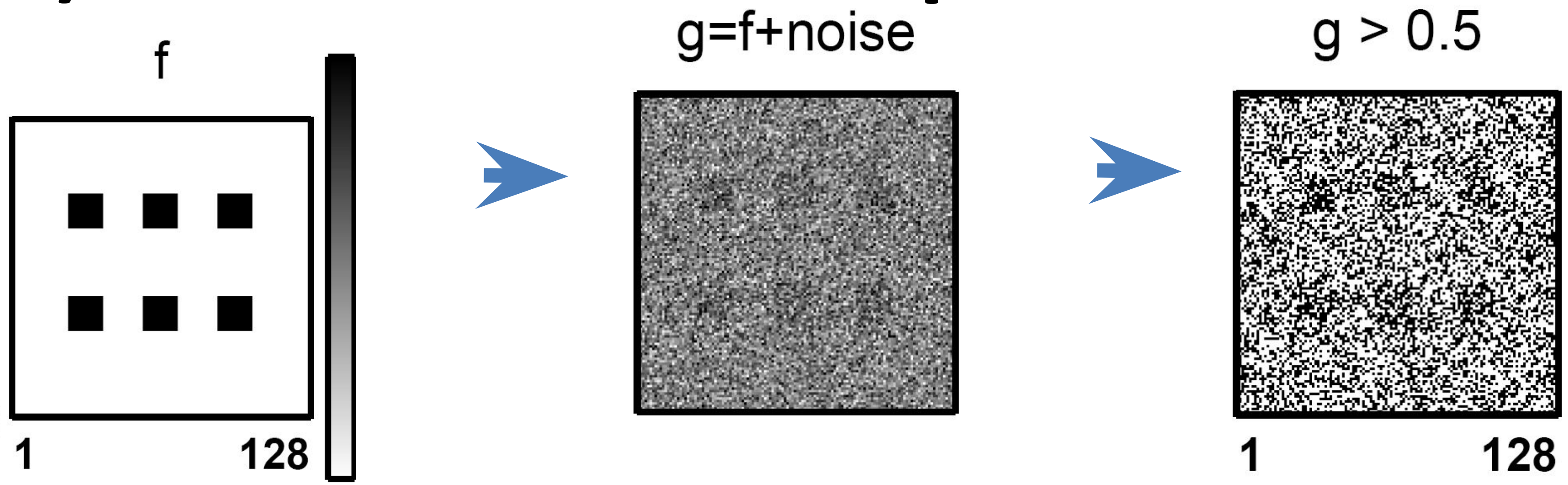
- Equivalent to a convolution without the flip
- Use it to measure 'similarity' between f and h.

(Cross) correlation – example



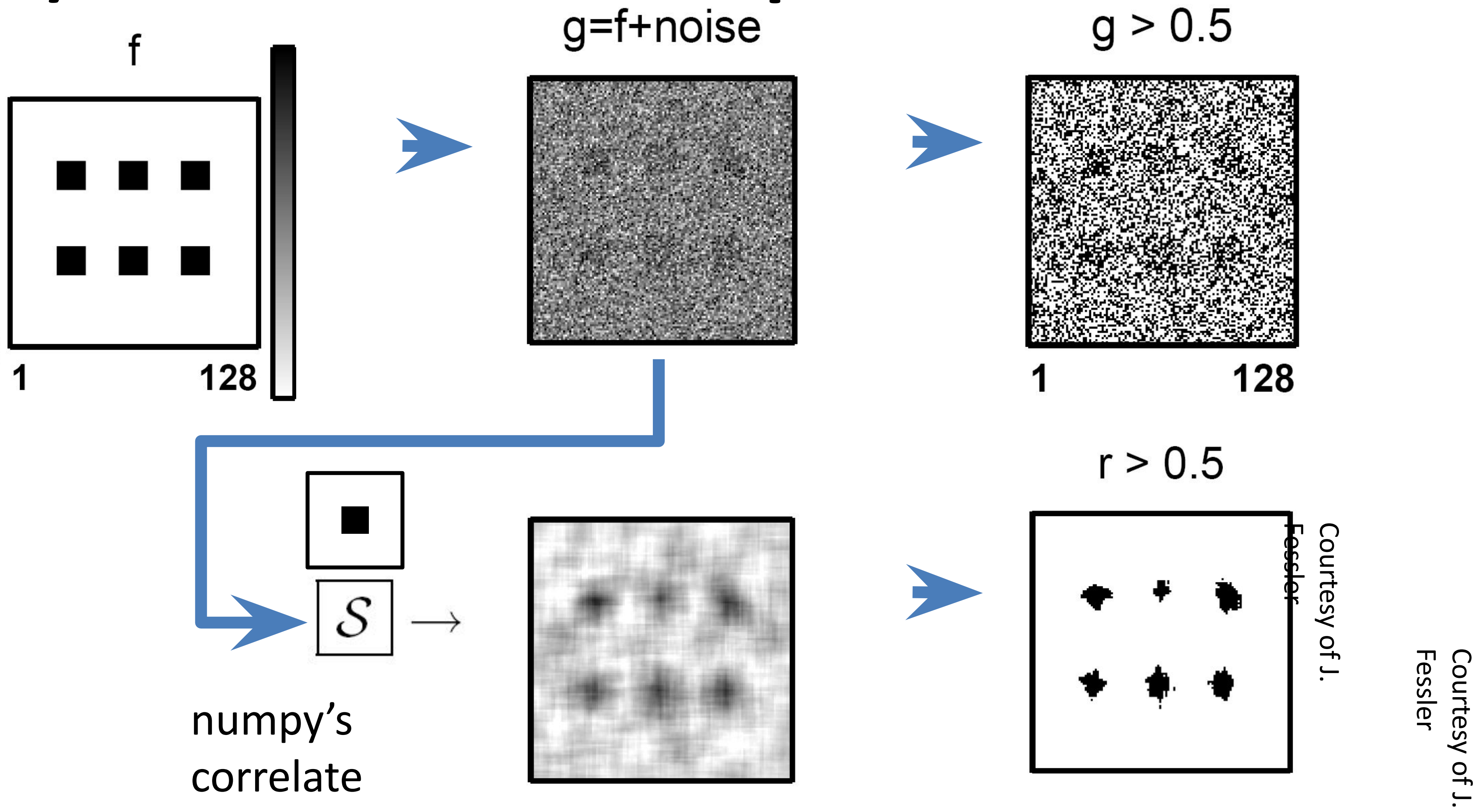
Courtesy of J. Fessler

(Cross) correlation – example

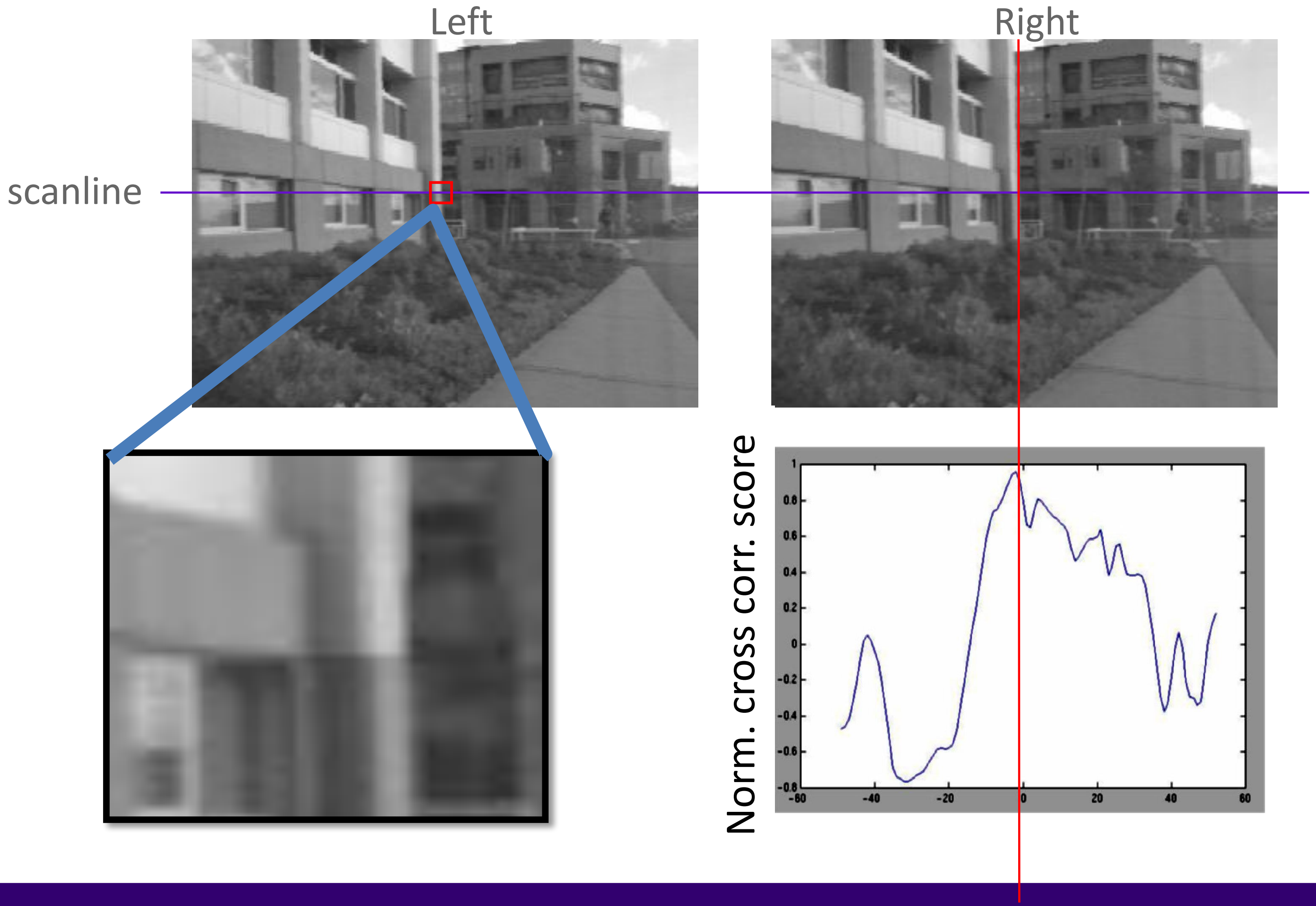


Courtesy of J. Fessler

(Cross) correlation – example



(Cross) correlation – example



Cross Correlation Application: Vision system for TV remote control

- uses template matching

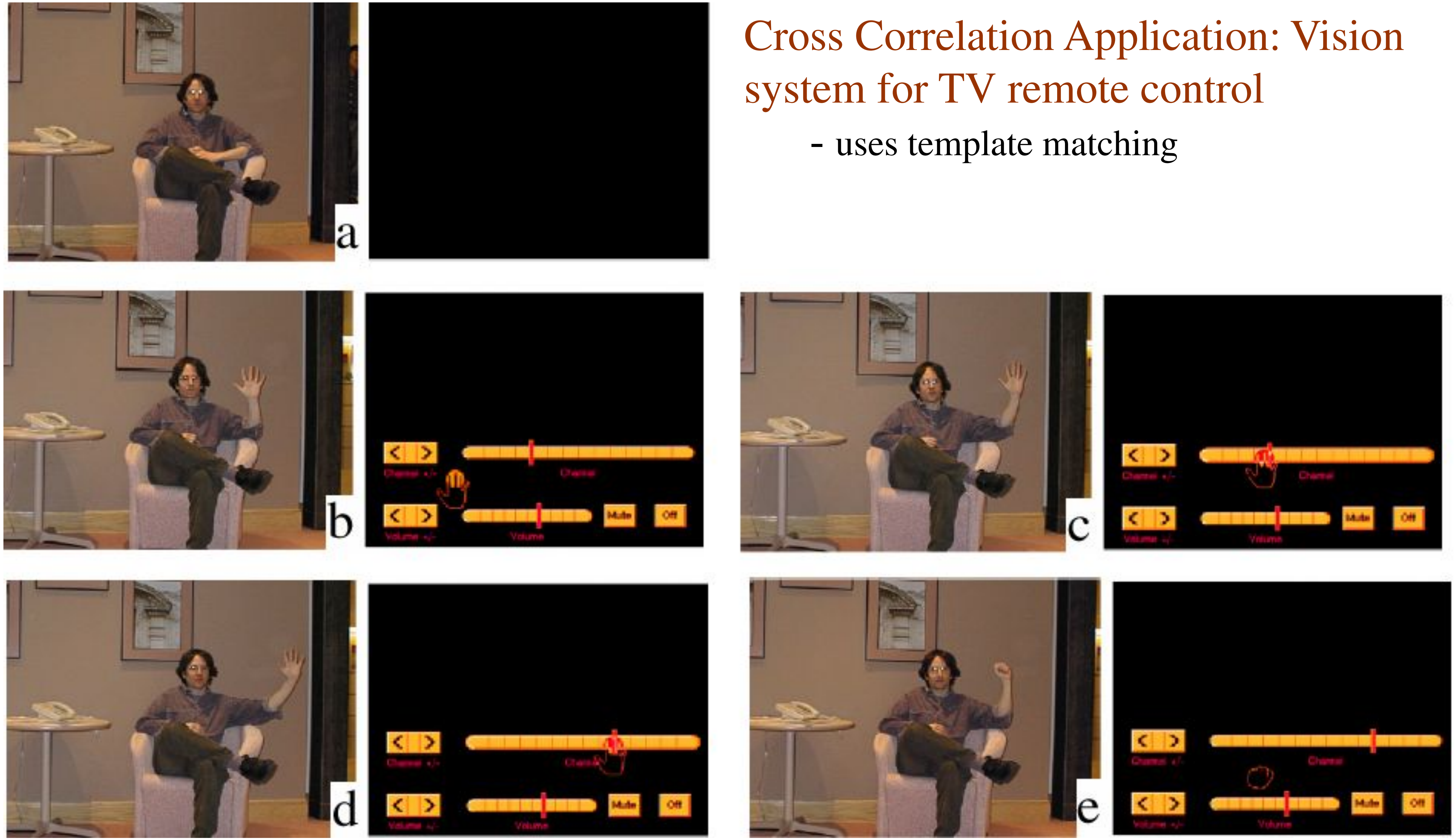


Figure from “Computer Vision for Interactive Computer Graphics,” W.Freeman et al, IEEE Computer Graphics and Applications, 1998 copyright 1998, IEEE

Properties of cross correlation

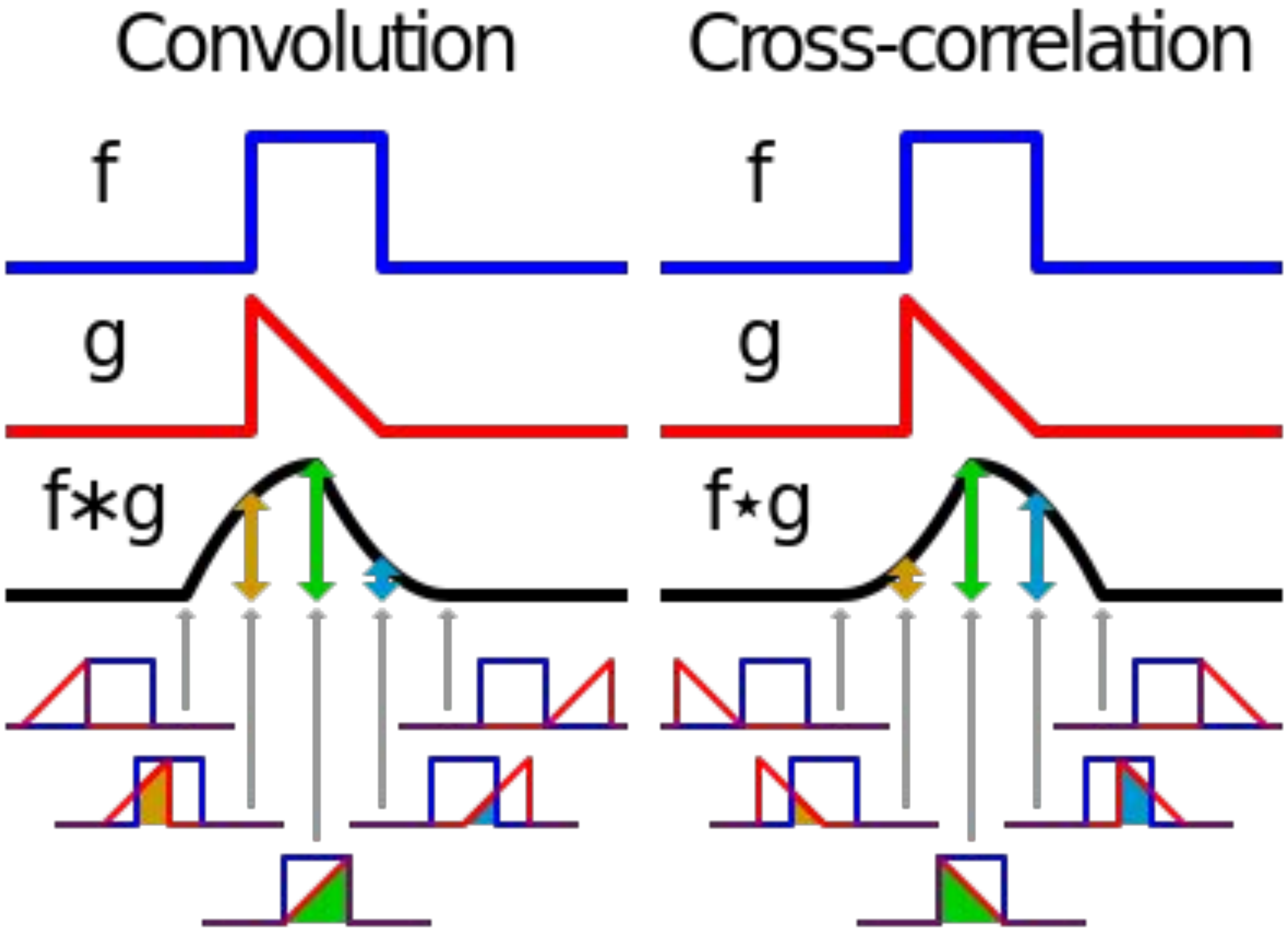
- Associative property:

$$(f ** h_1) ** h_2 = f ** (h_1 ** h_2)$$

- Distributive property:

$$f ** (h_1 + h_2) = (f ** h_1) + (f ** h_2)$$

The order doesn't matter! $h_1 ** h_2 = h_2 ** h_1$


Convolution
Cross-correlation
f
g
f∗g
f
g
f⋆g

Convolution vs. (Cross) Correlation

- When is correlation equivalent to convolution?
- In other words, Q. when is f**g = f*g?

Convolution vs. (Cross) Correlation

- A **<u>convolution</u>** is an integral that expresses the amount of overlap of one function as it is shifted over another function.
  - convolution is a **filtering** operation

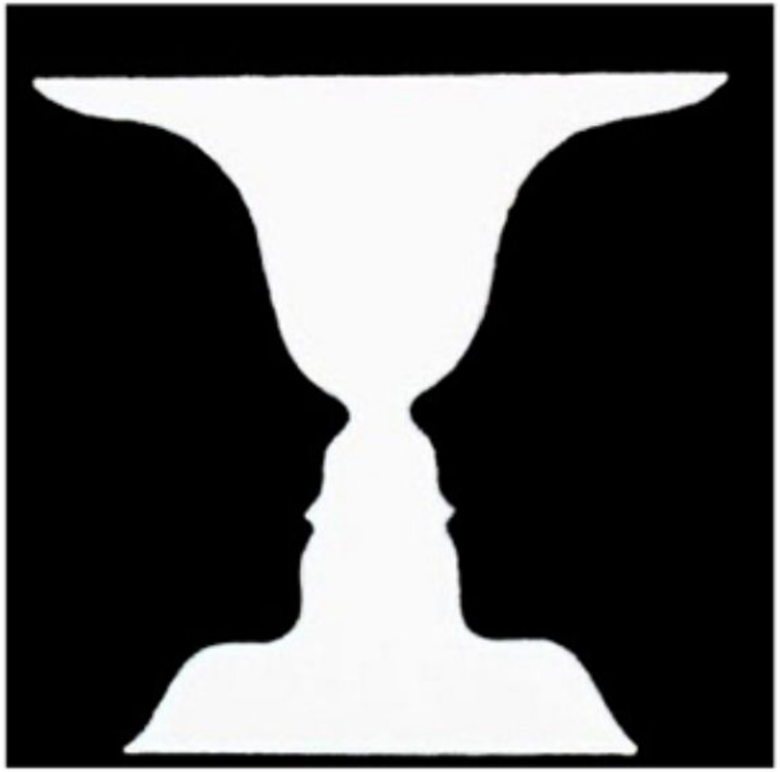
- **<u>Correlation</u>** compares the ***similarity** of **two** sets of **data.*** Correlation computes a measure of similarity of two input signals as they are shifted by one another. The correlation result reaches a maximum at the time when the two signals match best .
  - correlation is a measure of relatedness of two signals

What we will learn today

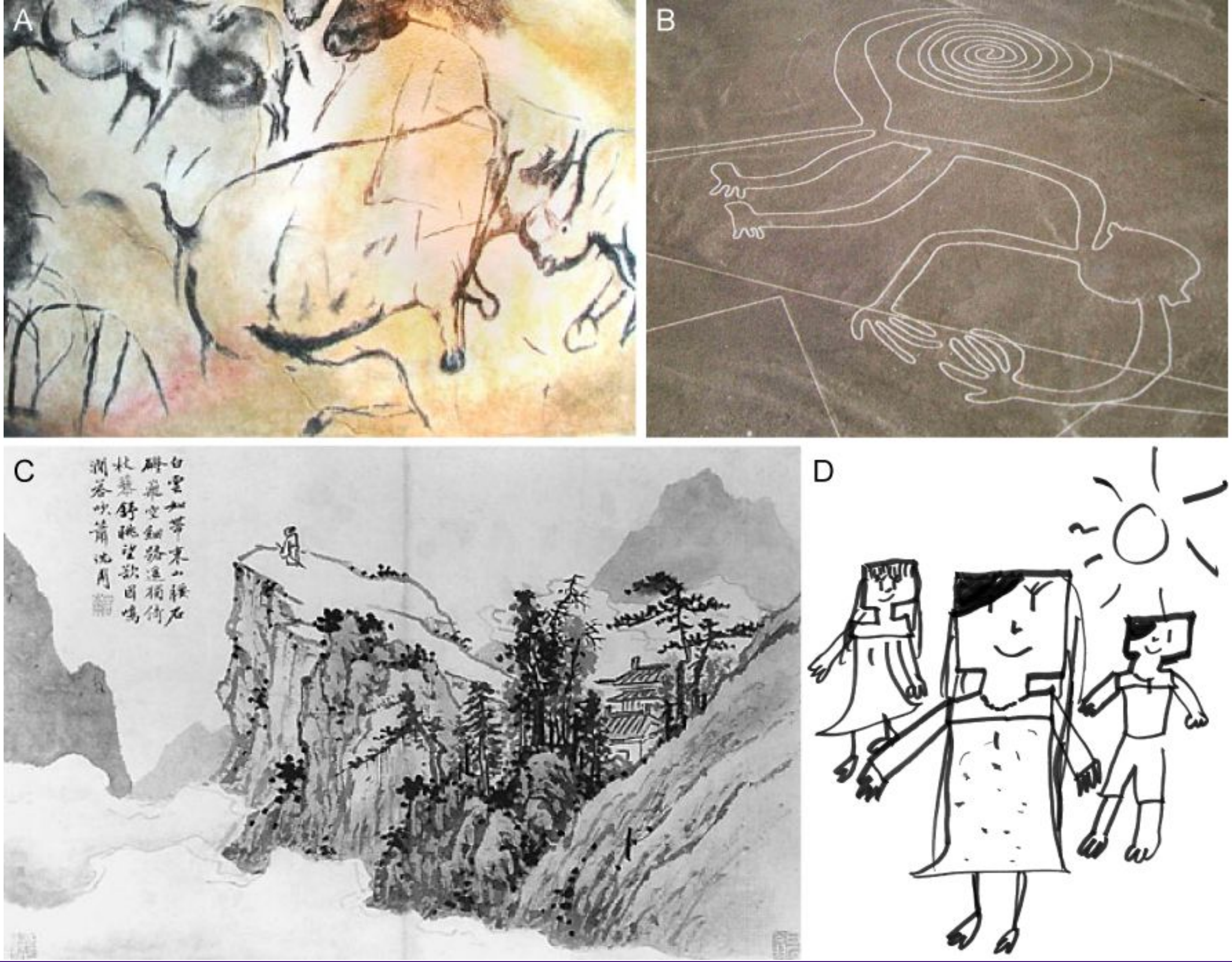
- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

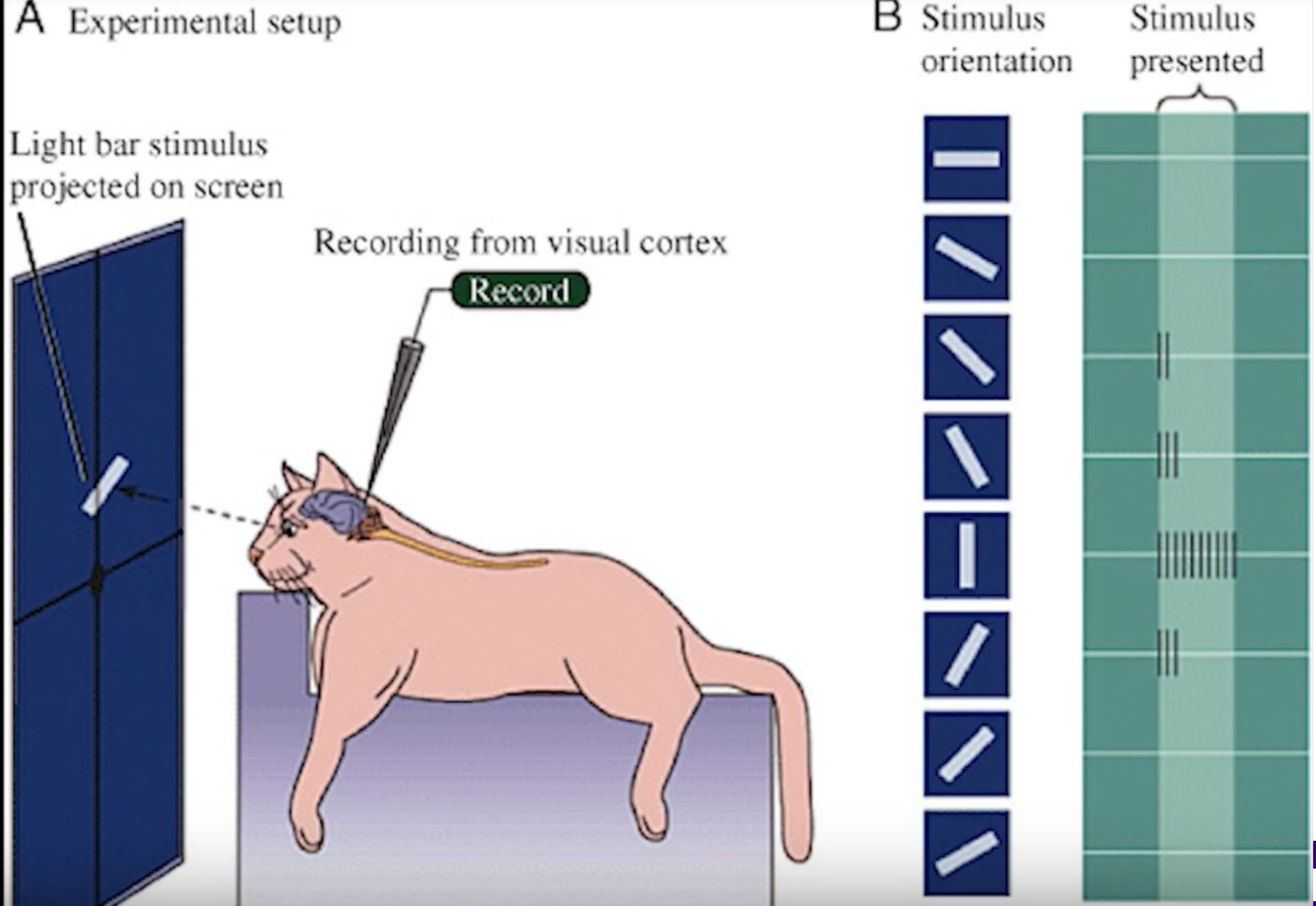
Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 8

Q. What do you see?

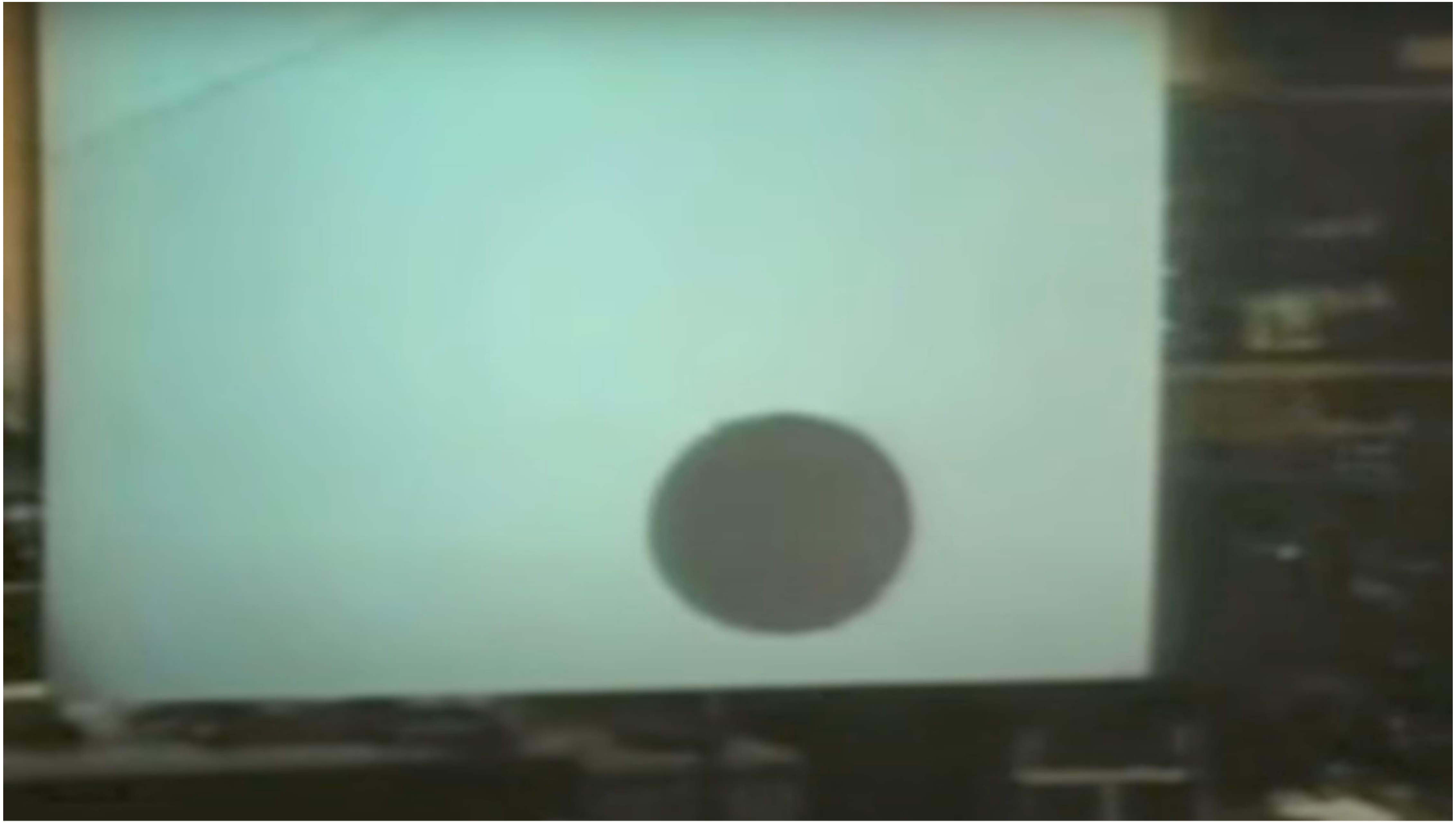


(A) Cave painting at Chauvet, France, about 30,000 B.C.;
(B) Aerial photograph of the picture of a monkey as part of the Nazca Lines geoglyphs, Peru, about 700 – 200 B.C.;
(C) Shen Zhou (1427-1509 A.D.): Poet on a mountain top, ink on paper, China;
(D) Line drawing by 7-year old I. Lleras (2010 A.D.).

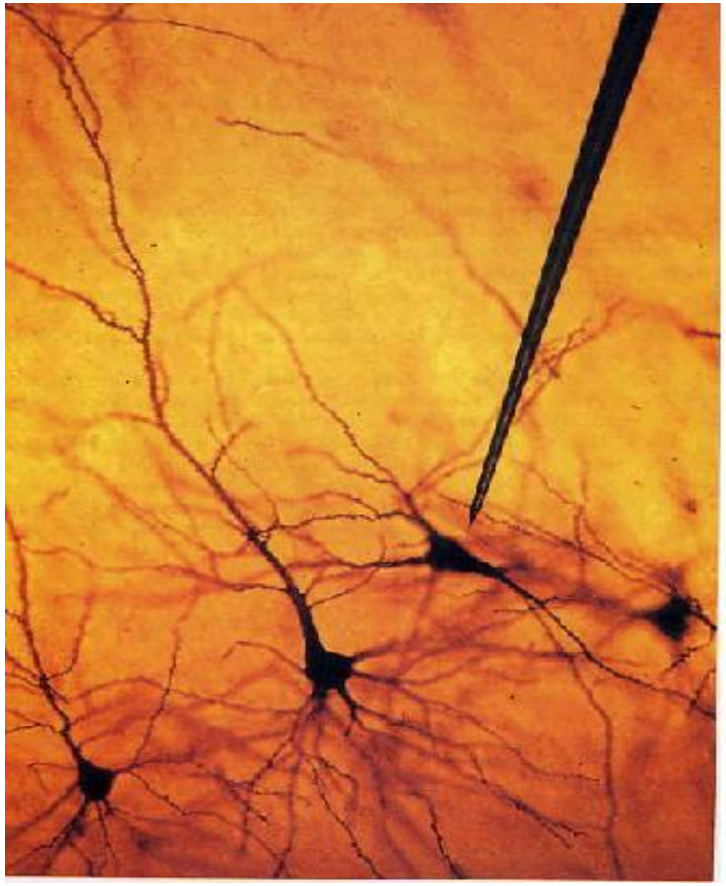


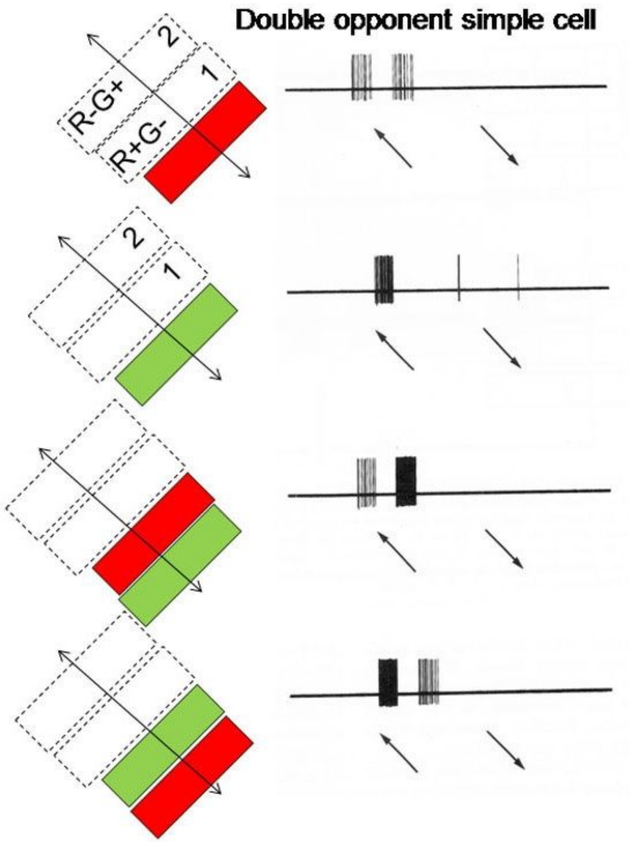


Hubel & Wiesel, 1960s



We know edges are special from human (mammalian) vision studies





Hubel & Wiesel, 1960s

We know edges are special from human (mammalian) vision studies

152 Biederman

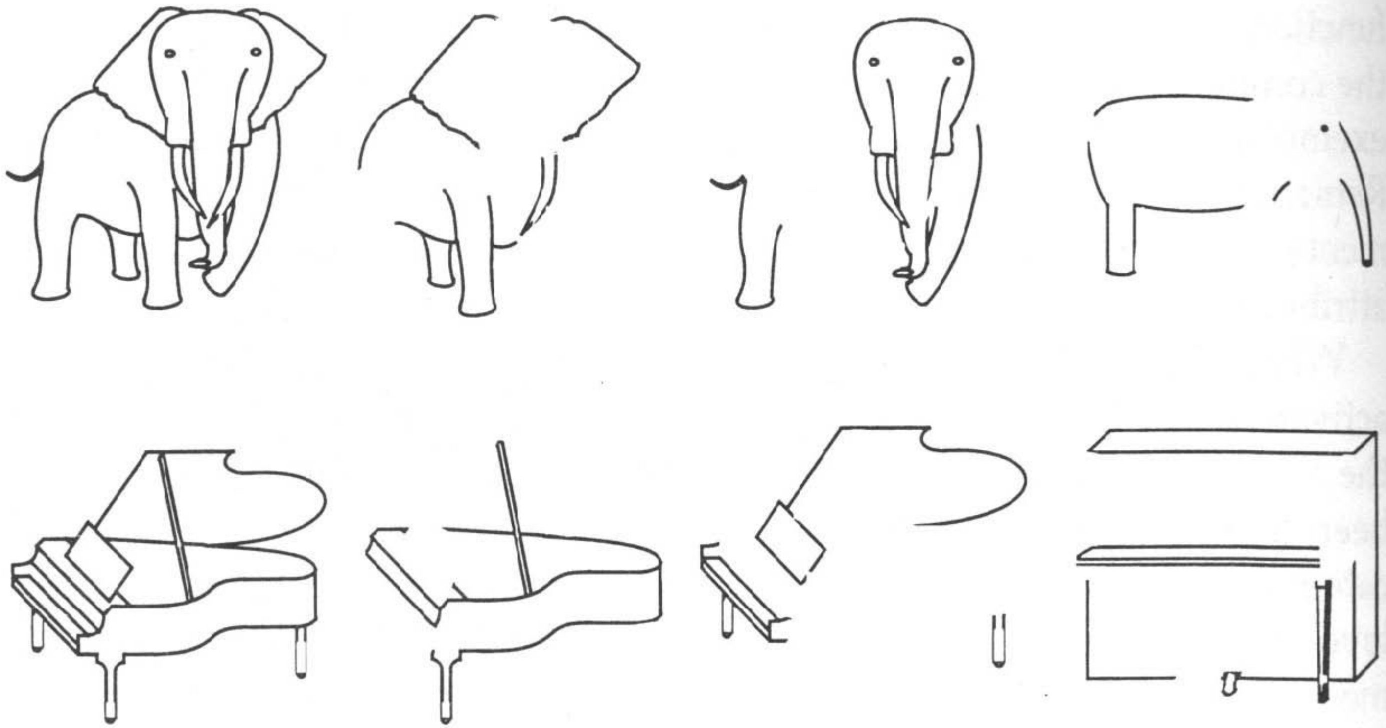
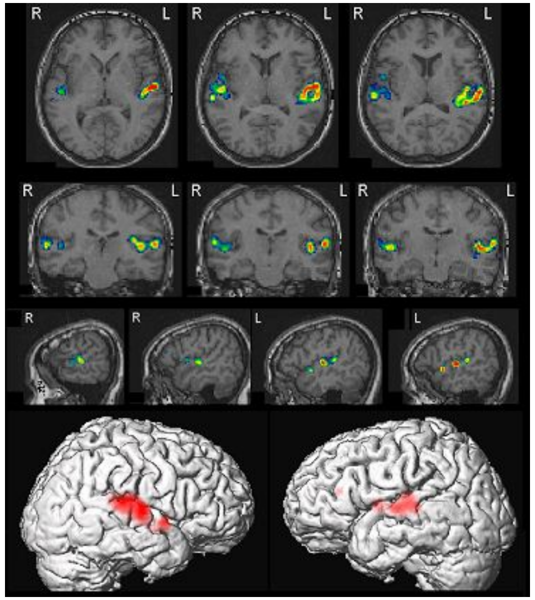
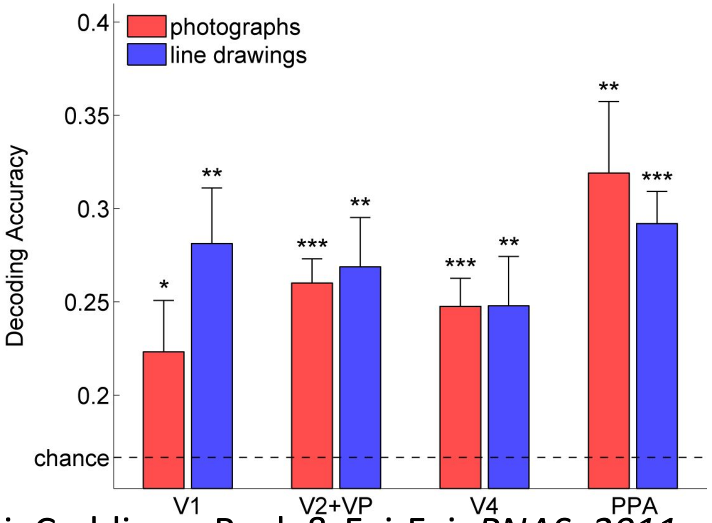


Figure 4.14
Complementary-part images. From an original intact image (left column), two complemen-



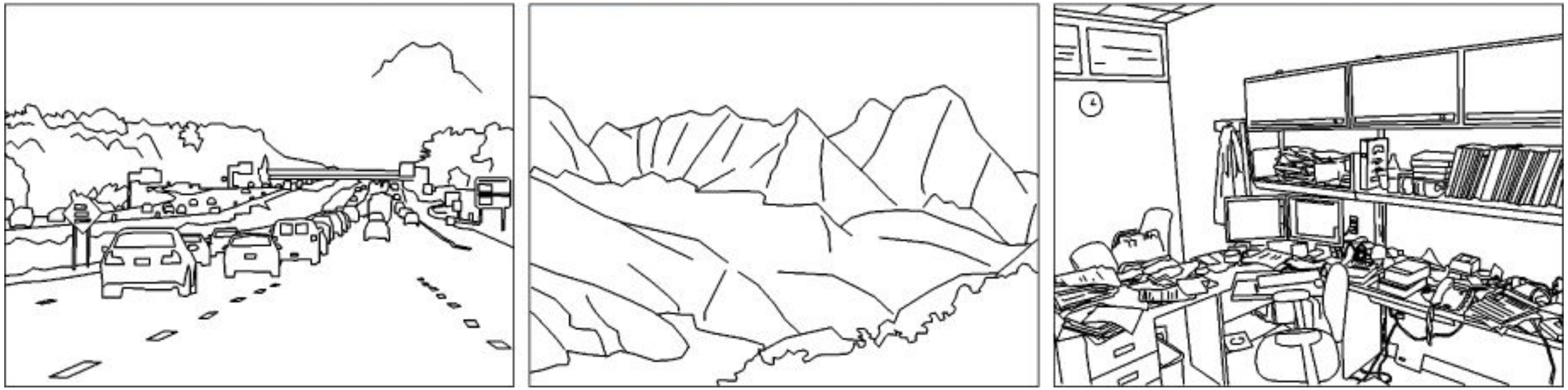


Walther, Chai, Caddigan, Beck & Fei-Fei, *PNAS, 2011*

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

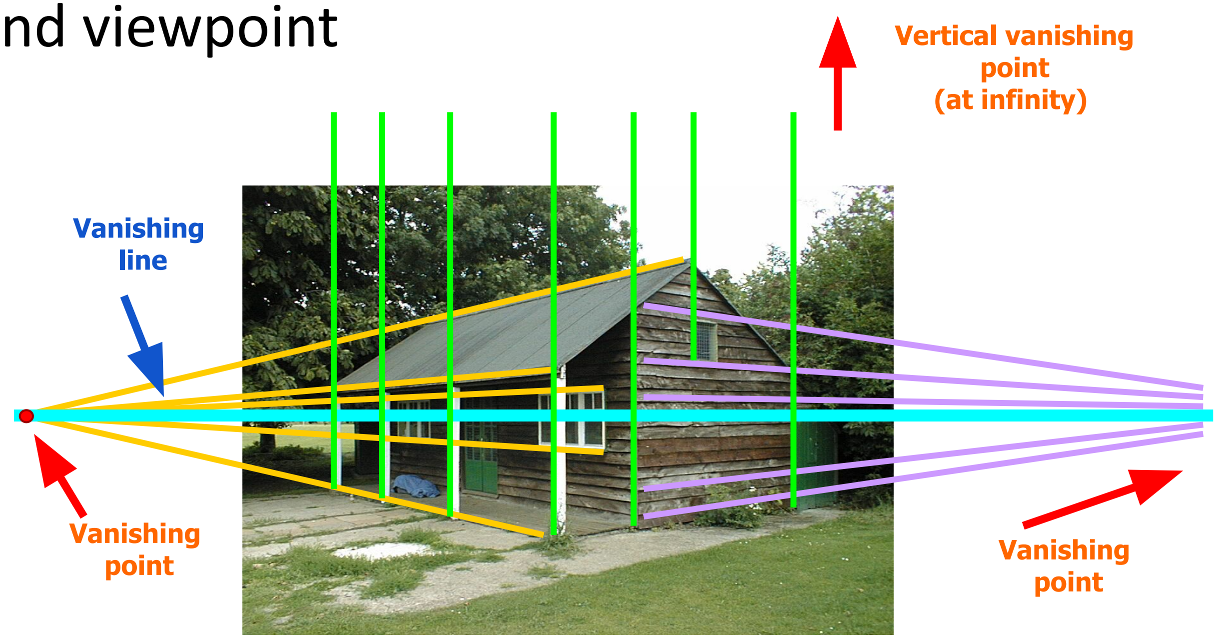
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)



Why do we care about edges?

- Extract information, recognize objects

- Recover geometry and viewpoint



Source: J. Hayes

Origins of edges

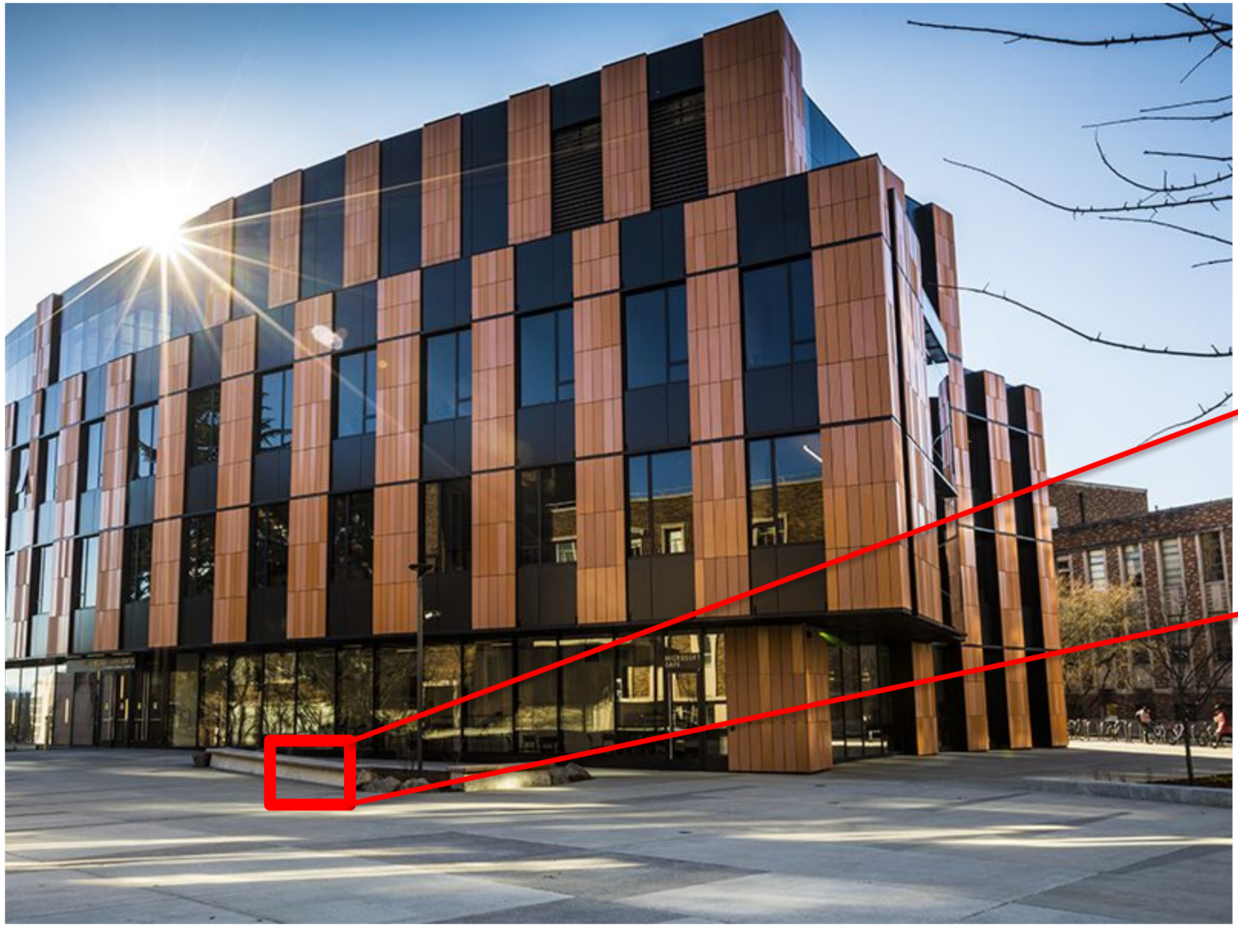
surface normal discontinuity

depth discontinuity

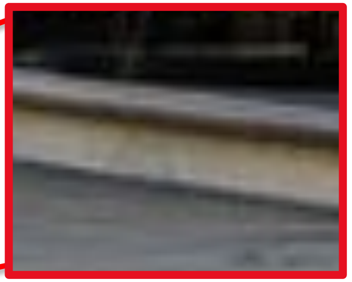
surface color discontinuity

illumination discontinuity

Closeup of edges



Surface normal discontinuity



Closeup of edges

Depth discontinuity

Closeup of edges

Surface color discontinuity

What we will learn today

- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

Review: Derivatives in 1D - example

$$y = x^2 + x^4$$

Q. What is the dy/dx?

Review: Derivatives in 1D - example

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

Derivatives in 1D - example

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

$$y = \sin x + e^{-x}$$

Q. What is the dy/dx?

Derivatives in 1D - example

$$y = x^2 + x^4 \qquad y = \sin x + e^{-x}$$

$$\frac{dy}{dx} = 2x + 4x^3 \qquad \frac{dy}{dx} = \cos x + (-1)e^{-x}$$

Approximating derivatives using numerical differentiation

$$\frac{df}{dx} = \lim_{\triangle x = 0} \frac{f[x + \triangle x] - f[x]}{\triangle x} = f'(x) = f_x$$

Approximating derivatives using numerical differentiation

Change in *f* at *x*

$$\frac{df}{dx} = \lim_{\triangle x = 0} \frac{f[x + \triangle x] - f[x]}{\triangle x} = f'(x) = f_x$$

Change in *x*

In discrete derivatives with images, smallest value of x is 1 pixel

$$\frac{df}{dx} = \lim_{\triangle x = 0} \frac{f[x + \triangle x] - f[x]}{\triangle x} = f'(x) = f_x$$

$$= \frac{f[x+1] - f[x]}{1}$$

$$= f[x+1] - f[x]$$

This is called a forward derivative

But change at x can be measured in many different ways

$$\frac{df}{dx} = f[x] - f[x-1]$$

Backward

But change at x can be measured in many different ways

$$\frac{df}{dx} = f[x] - f[x-1]$$ Backward

$$= f[x+1] - f[x]$$ Forward

But change at x can be measured in many different ways

$$\frac{df}{dx} = f[x] - f[x-1]$$ Backward

$$= f[x+1] - f[x]$$ Forward

$$= \frac{1}{2}(f[x+1] - f[x-1])$$ Central

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = ??$$

Q. What is the equation in width (2nd) dimension?

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n, m] = f[n, m] - f[n, m-1]$$

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

Remember the moving average filter:

$$h[\cdot,\cdot] = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

?	?	?
?	**?**	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

?	?	?
?	**1**	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

?	?	?
?	1	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

0	?	?
?	1	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

0	?	?
?	1	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n, m] = f[n, m] - f[n, m-1]$$

Q. Let's write this as a filter

$h[\cdot, \cdot]$

0	**0**	**0**
?	1	?
?	?	?

Remember the moving average filter:

$h[\cdot, \cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

0	0	0
?	1	?
?	?	?

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Let's write this as a filter

$h[\cdot,\cdot]$

0	0	0
?	1	?
0	**0**	**0**

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Last ones: What are these two?

$h[\cdot,\cdot]$

0	0	0
?	1	?
0	0	0

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation

$$g[n,m] = f[n,m] - f[n,m-1]$$

Q. Last ones: What are these two?

$h[\cdot,\cdot]$

0	0	0
0	1	**-1**
0	0	0

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

Remember the moving average filter:

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

- Using Forward differentiation:

Q. What is the formula?

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

- Using Forward differentiation:

?	?	?

$$g[n,m] = f[n,m+1] - f[n,m]$$

Q. What is the filter look like?

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

- Using Forward differentiation:

1	-1	0

$$g[n,m] = f[n,m+1] - f[n,m]$$

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

- Using Forward differentiation:

1	-1	0

$$g[n,m] = f[n,m+1] - f[n,m]$$

- Using Central differentiation:

Q. What is the formula?

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n, m] = f[n, m] - f[n, m-1]$$

- Using Forward differentiation:

1	-1	0

$$g[n, m] = f[n, m+1] - f[n, m]$$

- Using Central differentiation:

?	?	?

Q. What is the filter?

$$g[n, m] = f[n, m+1] - f[n, m-1]$$

Designing filters that perform differentiation

- Using Backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

- Using Forward differentiation:

1	-1	0

$$g[n,m] = f[n,m+1] - f[n,m]$$

- Using Central differentiation:

1	**0**	**-1**

$$g[n,m] = f[n,m+1] - f[n,m-1]$$

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10,\ \ 15,\ \ 10,\ \ 10,\ \ 25,\ \ 20,\ \ 20,\ \ 20]$$

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0,:] = [\ ? \qquad]$$

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10,\ \ 15,\ \ 10,\ \ 10,\ \ 25,\ \ 20,\ \ 20,\ \ 20]$$

$$\frac{df}{dm}[0,:] = [10,\ \ \textbf{?}\ \ ]$$

= 0 x -1 + 10x 1 + 15x 0

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0,:] = [10, \quad 5, \quad ? \quad]$$

= 10 x -1 + 15x 1 + 10x 0

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0,:] = [10, \quad 5, \; -5, \quad ? \quad]$$

= 15 x -1 + 10x 1 + 10x 0

Derivative in width dimension for one row

0	1	-1

Using backward differentiation:

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0,:] = [10, \quad 5, \quad -5, \quad 0, \quad ? \quad]$$

= 10 x -1 + 10x 1 + 25x 0

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n,m] = f[n,m] - f[n,m-1]$$

$$f[0,:] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0,:] = [10, \quad 5, \quad -5, \quad 0, \quad 15, \quad ? \quad ? \quad ?\ ]$$

= 0 x -1 + 10x 1 + 15x 0

Derivative in width dimension for one row

Using backward differentiation:

0	1	-1

$$g[n, m] = f[n, m] - f[n, m-1]$$

$$f[0, :] = [10, \quad 15, \quad 10, \quad 10, \quad 25, \quad 20, \quad 20, \quad 20]$$

$$\frac{df}{dm}[0, :] = [10, \quad 5, \ -5, \quad 0, \quad 15, \ -5, \ 0, \quad 0]$$

Discrete derivation in 2D:

$$\text{Given function } f[n,m]$$

$$\text{Gradient filter } \nabla f[n,m] = \begin{bmatrix} \frac{df}{dn} \\ \frac{df}{dm} \end{bmatrix} = \begin{bmatrix} f_n \\ f_m \end{bmatrix}$$

Discrete derivation in 2D:

$$\text{Given function } f[n,m]$$

$$\text{Gradient filter } \nabla f[n,m] = \begin{bmatrix} \frac{df}{dn} \\ \frac{df}{dm} \end{bmatrix} = \begin{bmatrix} f_n \\ f_m \end{bmatrix}$$

$$\text{Gradient magnitude } |\nabla f[n,m]| = \sqrt{f_n^2 + f_m^2}$$

Discrete derivation in 2D:

$$\text{Given function } f[n,m]$$

$$\text{Gradient filter } \nabla f[n,m] = \begin{bmatrix} \frac{df}{dn} \\ \frac{df}{dm} \end{bmatrix} = \begin{bmatrix} f_n \\ f_m \end{bmatrix}$$

$$\text{Gradient magnitude } |\nabla f[n,m]| = \sqrt{f_n^2 + f_m^2}$$

$$\text{Gradient direction } \theta = \tan^{-1}\left(\frac{f_m}{f_n}\right)$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} ? & ? & ? & ? & ? \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

= 0 x -1
+ 10x 0
+ 10x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ ? & ? & ? & ? & ? \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

= 10x -1
+ 10x 0
+ 10x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ ? & ? & ? & ? & ? \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

= 10x -1
+ 10x 0
+ 10x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ & & & & \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ ? & ? & ? & ? & ? \\ & & & & \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

= 10x -1
+ 10x 0
+ 10x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ ? & ? & ? & ? & ? \end{bmatrix}$$

2D discrete derivative - example

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad h[n,m] = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

= 10x -1
+ 10x 0
+ 0x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -10 & -10 & -20 & -20 & -20 \end{bmatrix}$$

Let’s do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

= 0 x -1 +10x 0 +10x 1

$$g[n,m] = \begin{bmatrix} 10 & & & & \\ 10 & & & & \\ 10 & & & & \\ 10 & & & & \\ 10 & & & & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & ? & & & \\ 10 & ? & & & \\ 10 & ? & & & \\ 10 & ? & & & \\ 10 & ? & & & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

=10x -1 +10x 0 +20x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & & & \\ 10 & 10 & & & \\ 10 & 10 & & & \\ 10 & 10 & & & \\ 10 & 10 & & & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & ? & & \\ 10 & 10 & ? & & \\ 10 & 10 & ? & & \\ 10 & 10 & ? & & \\ 10 & 10 & ? & & \end{bmatrix}$$

Let’s do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

=10x -1 +20x 0 +20x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 10 & & \\ 10 & 10 & 10 & & \\ 10 & 10 & 10 & & \\ 10 & 10 & 10 & & \\ 10 & 10 & 10 & & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 10 & ? & \\ 10 & 10 & 10 & ? & \\ 10 & 10 & 10 & ? & \\ 10 & 10 & 10 & ? & \\ 10 & 10 & 10 & ? & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

=20x -1 +20x 0 +20x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 10 & 0 & \\ 10 & 10 & 10 & 0 & \\ 10 & 10 & 10 & 0 & \\ 10 & 10 & 10 & 0 & \\ 10 & 10 & 10 & 0 & \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$g[n,m] = \begin{bmatrix} 10 & 10 & 10 & 0 & ? \\ 10 & 10 & 10 & 0 & ? \\ 10 & 10 & 10 & 0 & ? \\ 10 & 10 & 10 & 0 & ? \\ 10 & 10 & 10 & 0 & ? \end{bmatrix}$$

Let's do the other one

$$f[n,m] = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

=20x -1 +20x 0 + 0x 1

$$g[n,m] = \begin{bmatrix} 10 & 10 & 10 & 0 & -20 \\ 10 & 10 & 10 & 0 & -20 \\ 10 & 10 & 10 & 0 & -20 \\ 10 & 10 & 10 & 0 & -20 \\ 10 & 10 & 10 & 0 & -20 \end{bmatrix}$$

2D discrete derivative filters

Q. What does this filter do?

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

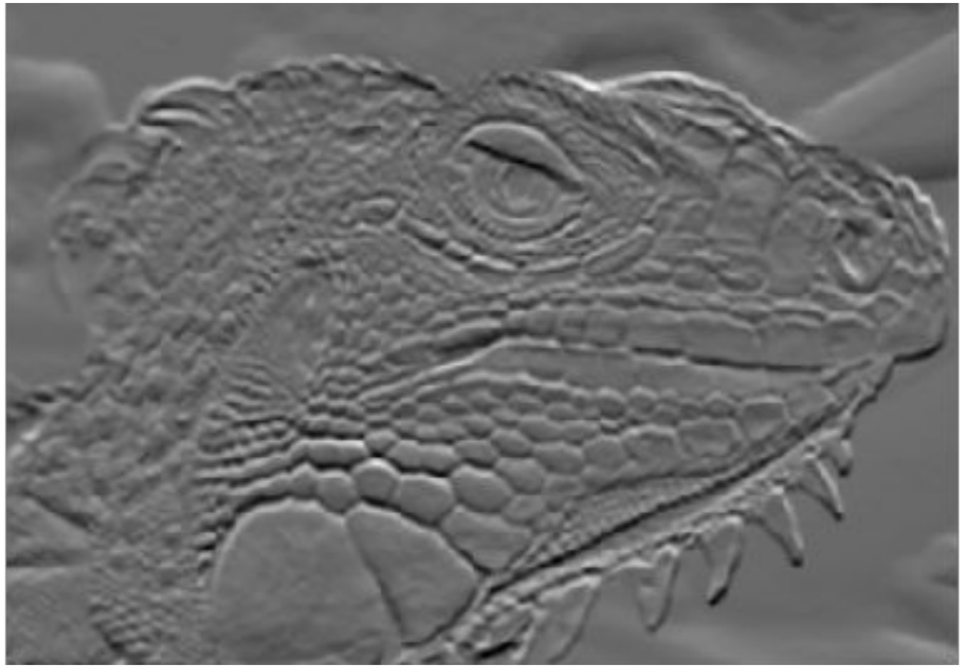
2D discrete derivative filters

$$h[n,m] = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Q. What does this filter do?

$$h[n,m] = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Q. Which filter was applied?



$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

A **B**

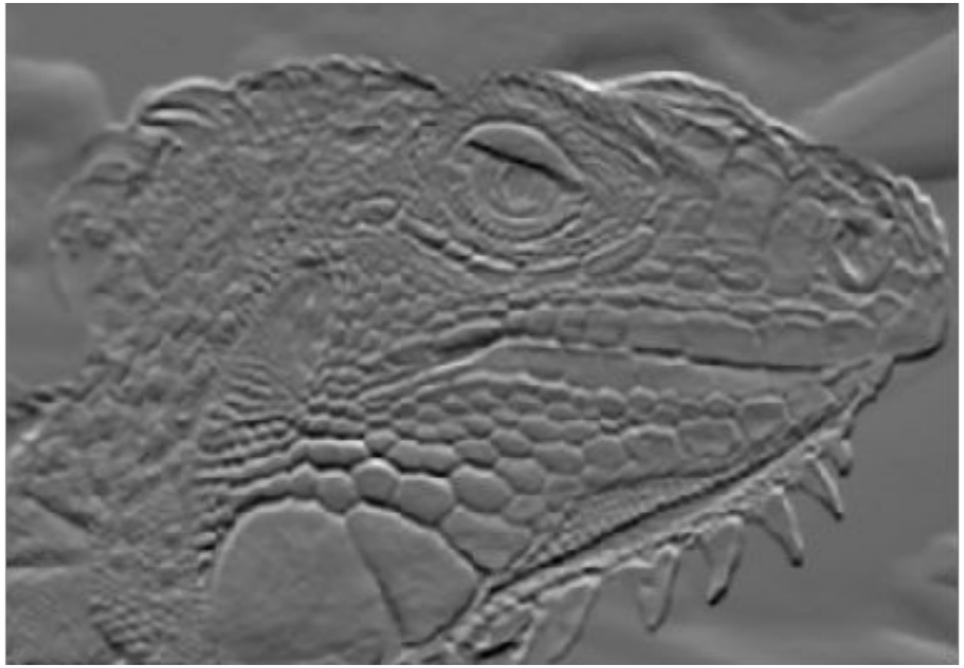
Q. Which filter was applied?

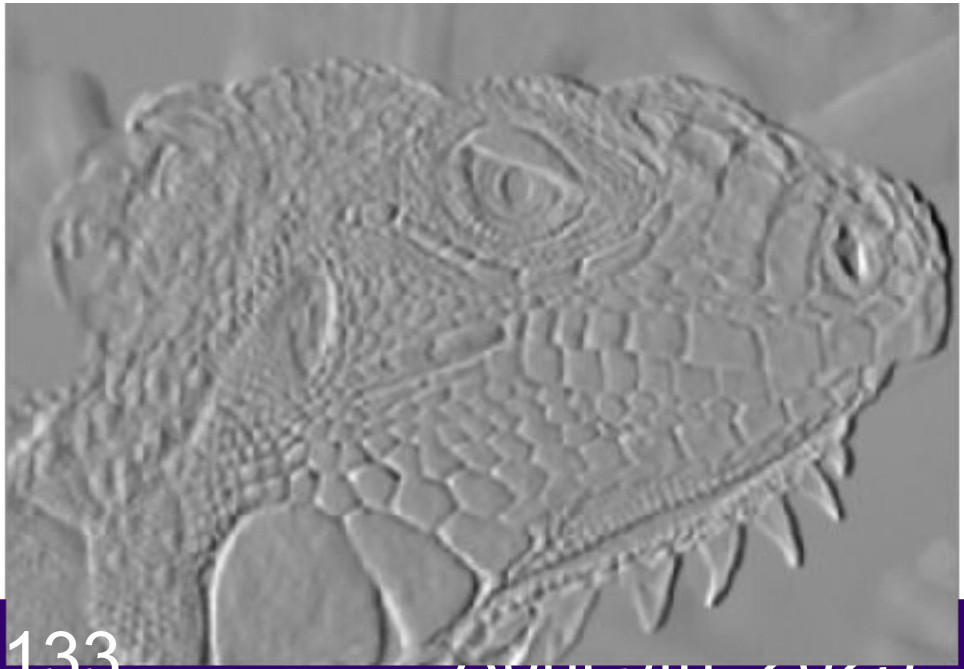
$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

A

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

B





What we will learn today

- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

Characterizing edges

An edge is a place of rapid change in the image intensity function

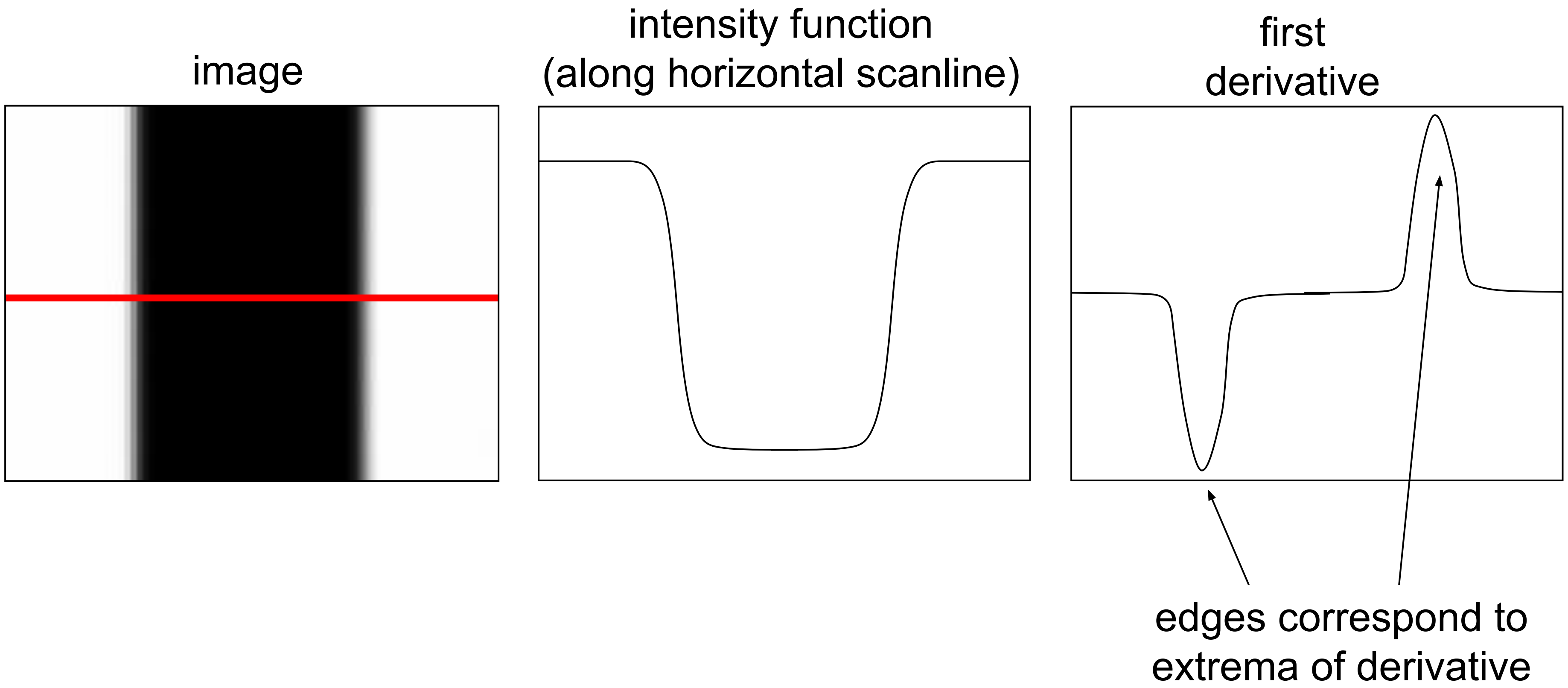
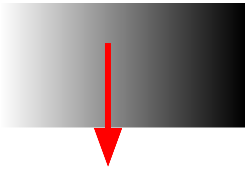


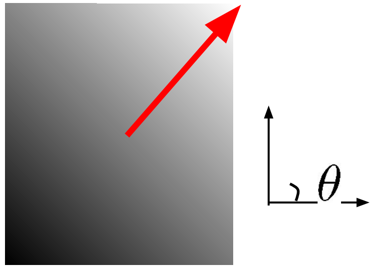
Image gradient

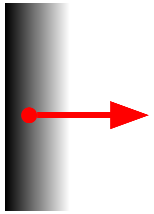
The gradient of an image:



$$\nabla_n f[n,m] = \left[\frac{df}{dn} \quad 0\right]$$

$$\nabla f[n,m] = \left[\frac{df}{dn} \quad \frac{df}{dm}\right]$$





$$\nabla_m f[n,m] = \left[0 \quad \frac{df}{dm}\right]$$

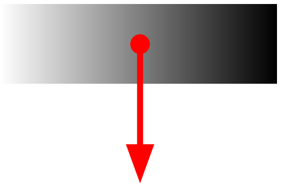
The gradient vector points in the direction of most rapid increase in intensity

$$\theta = \tan^{-1}\left(\frac{f_m}{f_n}\right)$$

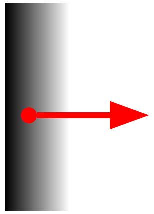
Source: Steve Seitz

Image gradient

The gradient of an image:

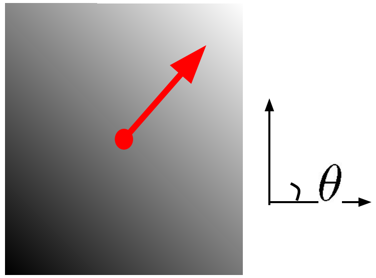


$$\nabla_n f[n,m] = \left[\frac{df}{dn} \quad 0\right]$$



$$\nabla_m f[n,m] = \left[0 \quad \frac{df}{dm}\right]$$

$$\nabla f[n,m] = \left[\frac{df}{dn} \quad \frac{df}{dm}\right]$$



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

$$\theta = \tan^{-1}\left(\frac{f_m}{f_n}\right)$$

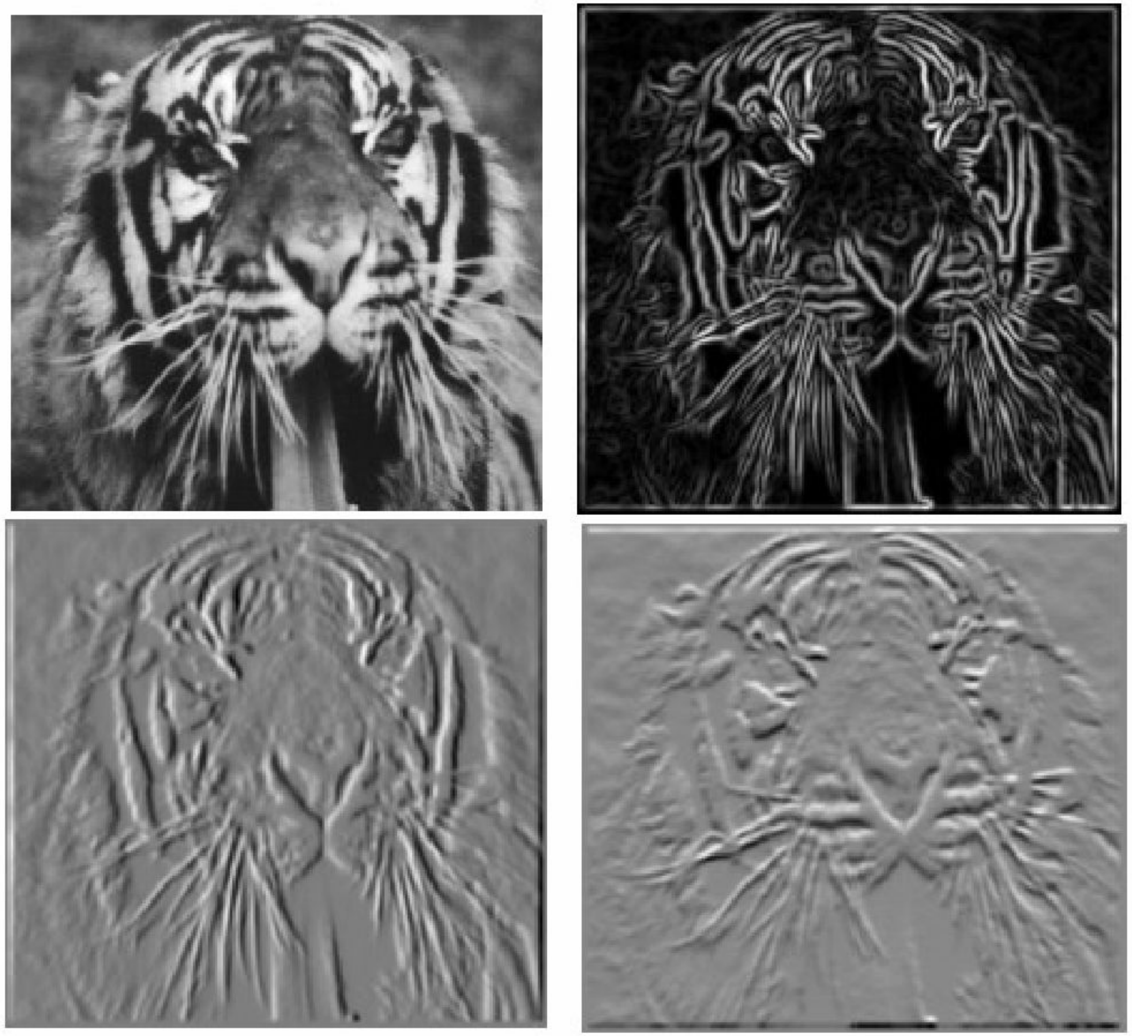
The *edge strength* is given by the gradient magnitude

$$|\nabla f[n,m]| = \sqrt{f_n^2 + f_m^2}$$

Finite differences: example

Original Image

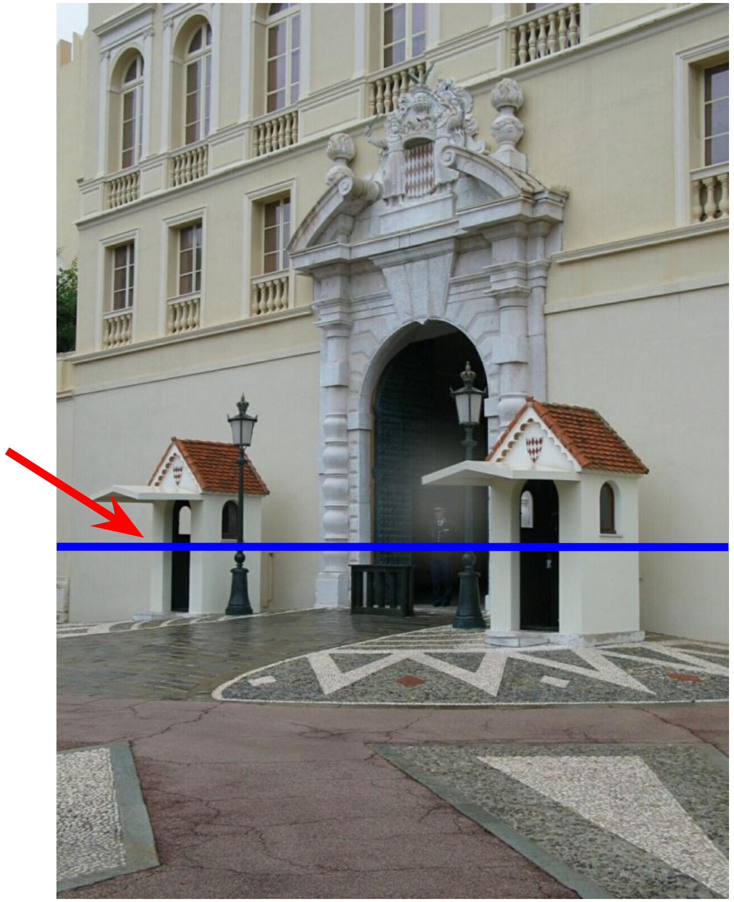
Gradient magnitude

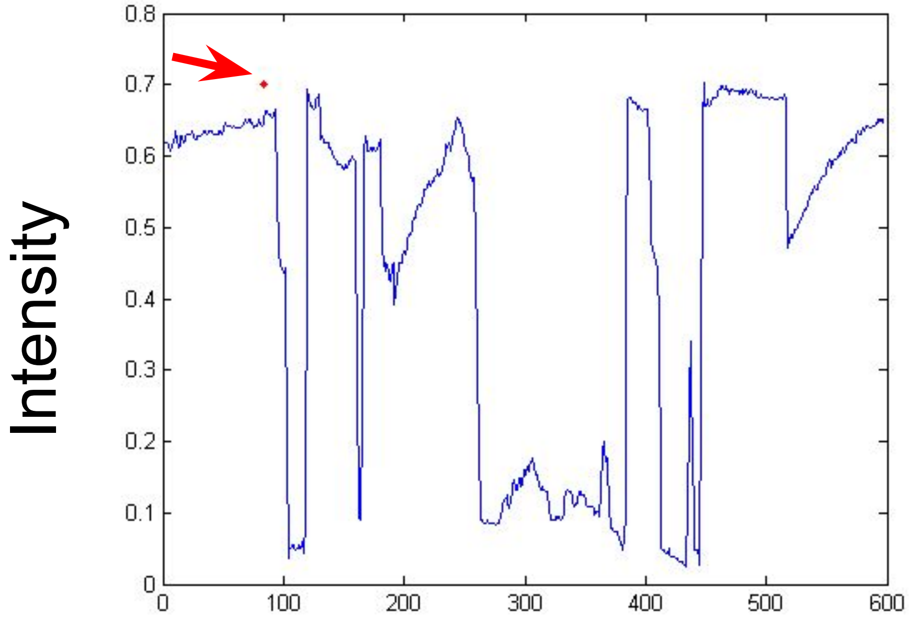


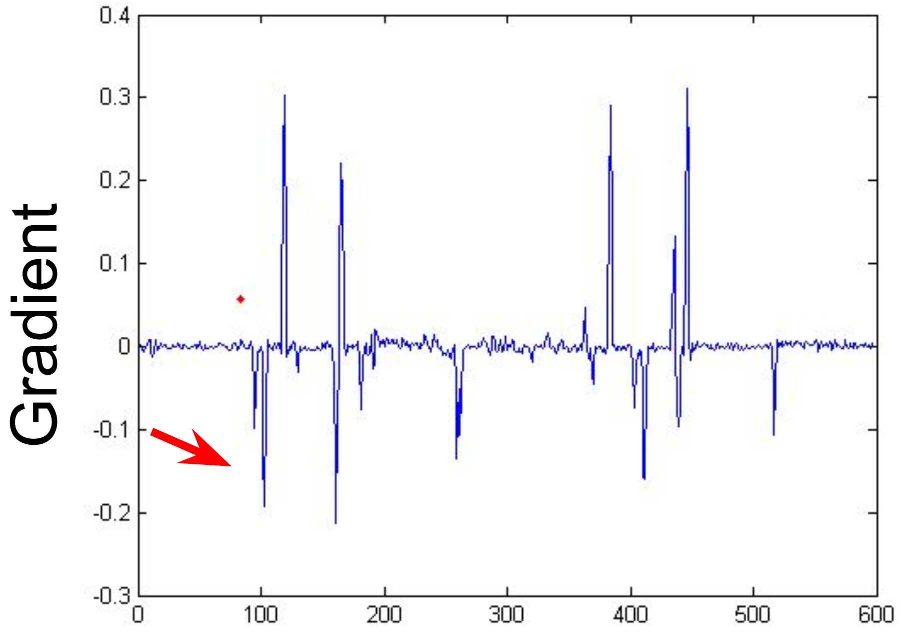
width-direction

height-direction

Intensity profile







Summary

- Convolutions and Cross-Correlation
- Edge detection
- Image Gradients
- A simple edge detector

Next time: Detecting lines