Image Processing - II

SAUMYA JETLEY [March 2022]

- Previously...
  1. Space-frequency duality
  2. Fourier Transform
  3. LTI and ‘convolution’
  4. Extracting and manipulating frequency information

❑ Computer Vision via Human Vision - What is salient to us ?

Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Computer Vision via Human Vision - What is salient to us ?

Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

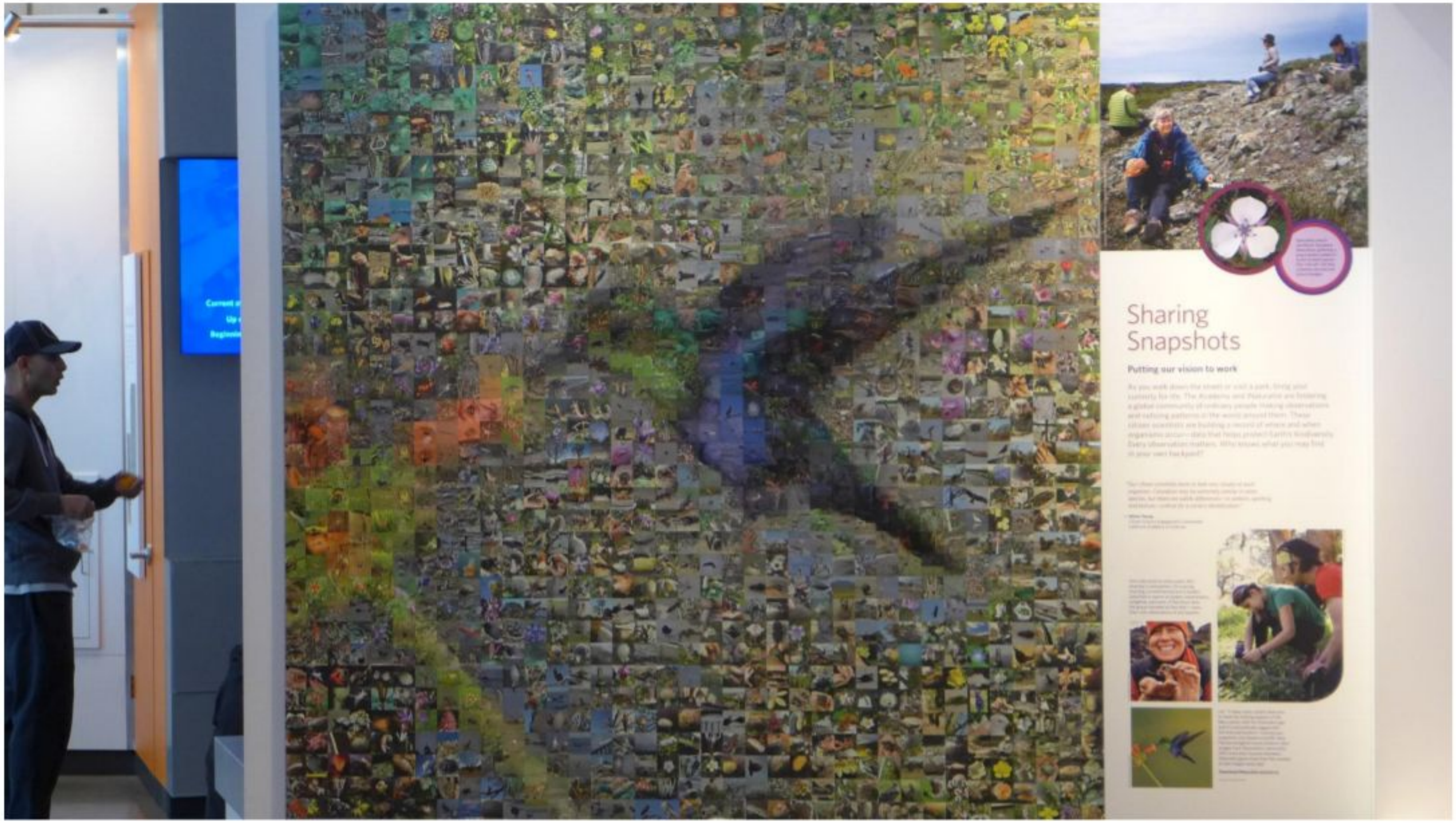
❑ Computer Vision via Human Vision - What is salient to us ?

Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Computer Vision via Human Vision - What is salient to us ?

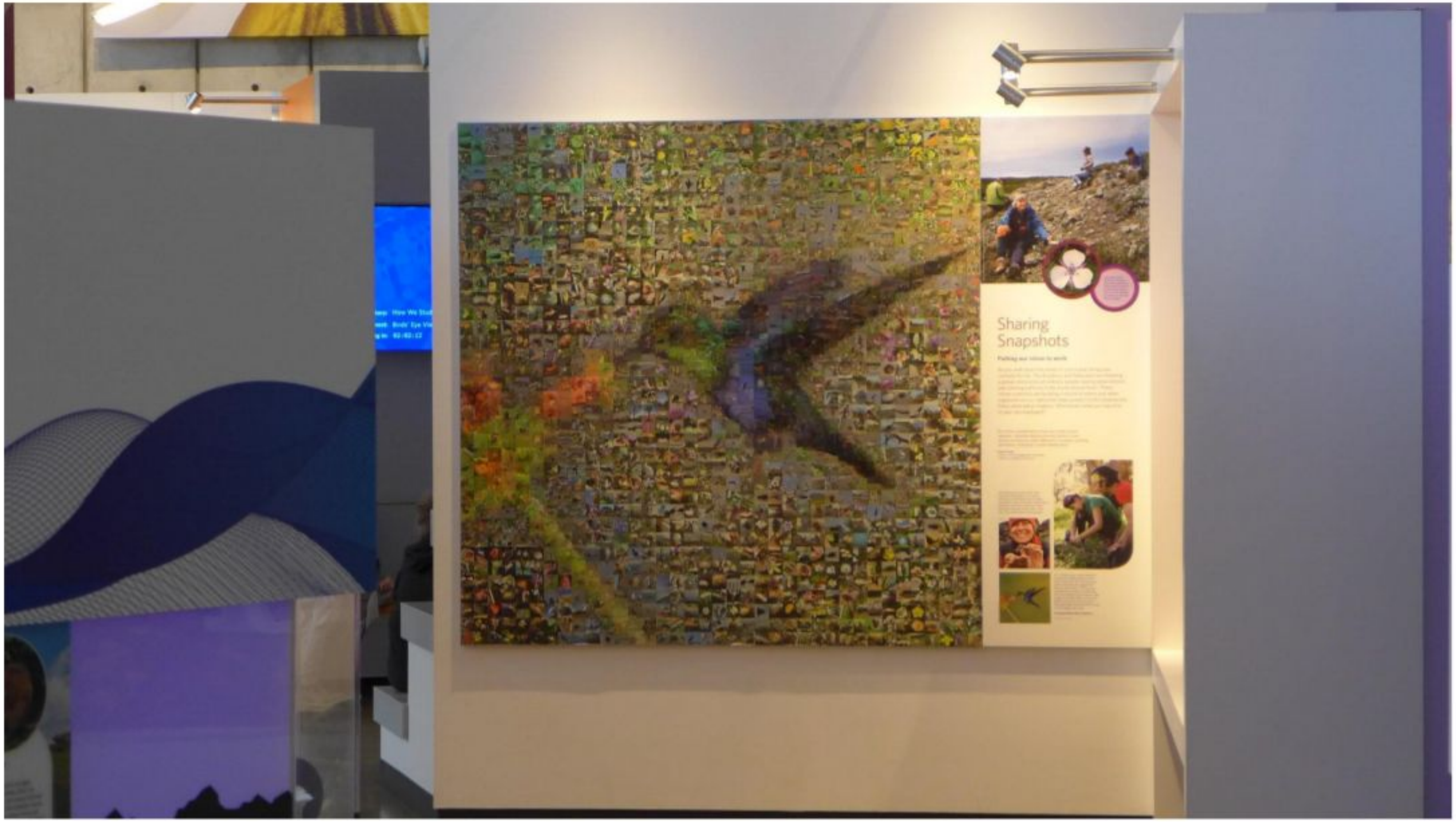
Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Computer Vision via Human Vision - What is salient to us ?



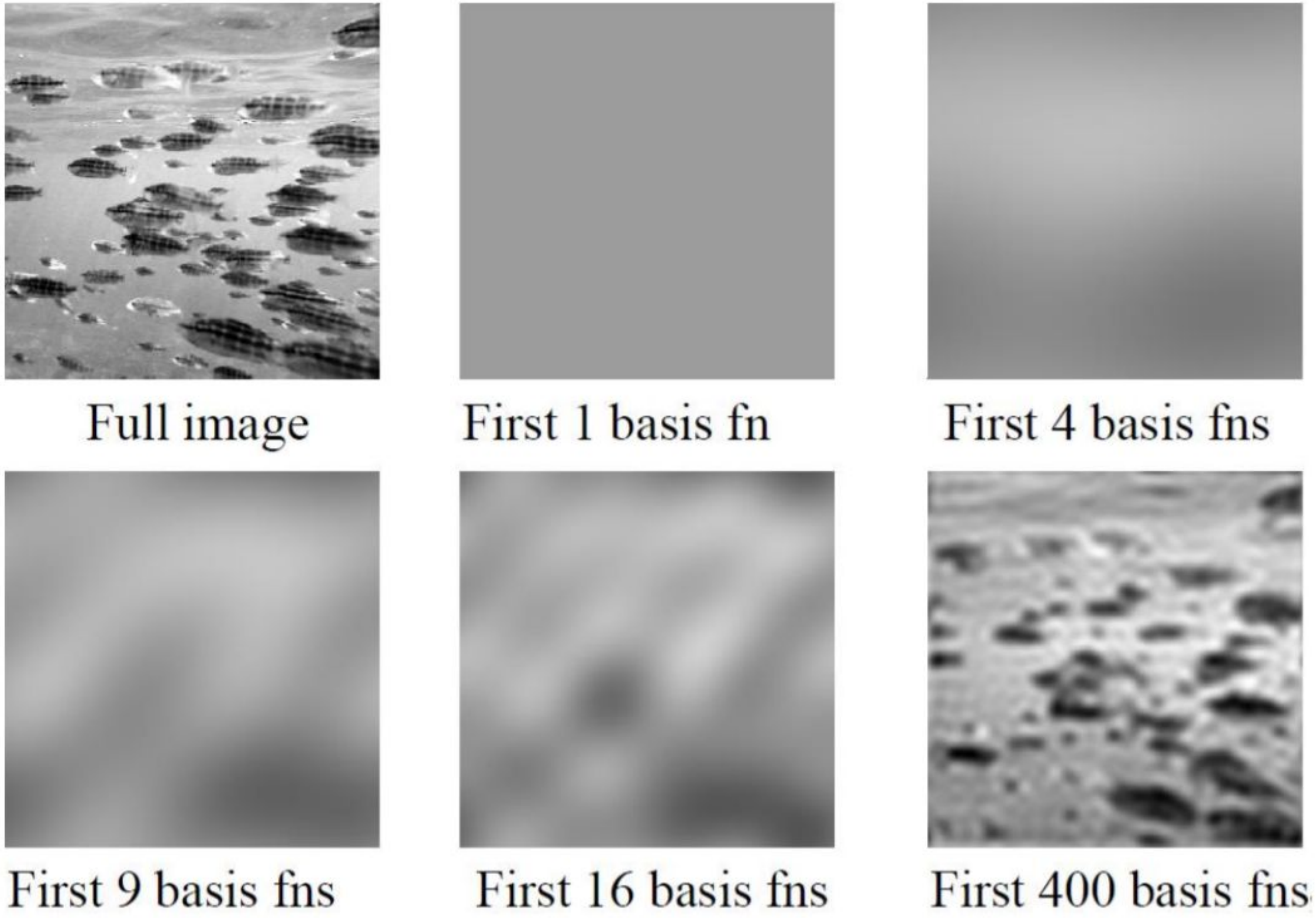
Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Computer Vision via Human Vision - What is salient to us ?



Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Fourier Basis and Reconstruction



Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Properties of Fourier Transform - I

- Linearity $$\mathrm{F}[ax(t)+by(t)] = a\,\mathrm{F}[x(t)] + b\,\mathrm{F}[y(t)]$$

- Fourier transform of a real signal is symmetric about the origin

- The energy of the signal is the same as the energy of its Fourier transform

Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Properties of Fourier Transform - II

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

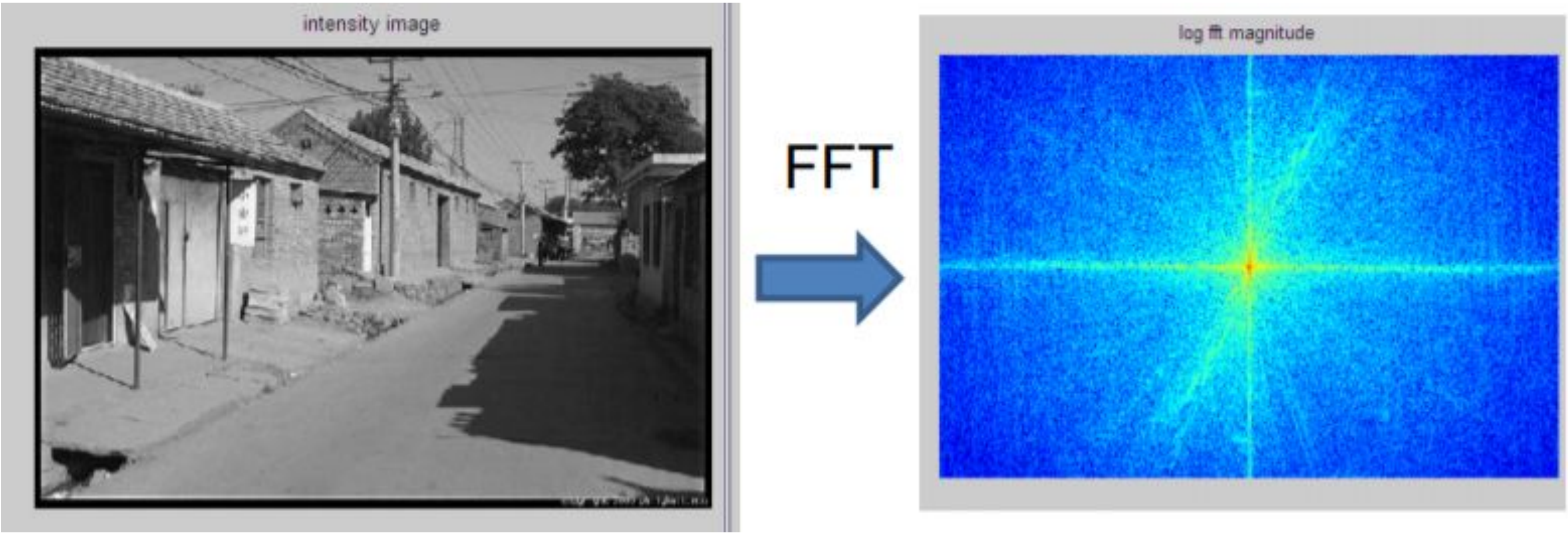
$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

$$g * h = \mathrm{F}^{-1}[\mathrm{F}[g]\,\mathrm{F}[h]]$$

Courtesy of https://cs.brown.edu/courses/cs143/2017_Fall/lectures_Fall2017/06_Fall2017_ThinkingInFrequencyPart3.pdf

❑ Reading an FT: Frequency Range



Extreme cases along X-axis:

o Same value throughout	[1 1 1 1 1 1]	Frequency = 0
o Value alternates every other pixel	[1 0 1 0 1 0]	Frequency = N/2

[Note the discussion on euler formula to interpret negative frequency of -N/2]
https://www.youtube.com/watch?v=Nupda1rm01Y&t=13s

Same for Y-axis!

- Intuition Building: Subsampling images and frequency adjustment

Courtesy of Hybrid Images, Oliva et al, SIGGRAPH 2006

❑ Intuition Building: Subsampling images and frequency adjustment

o When we reduce the size of an image, we subsample the image.

o This results in MULTIPLYING the existing image frequencies by the sampling factor

o For example, halving an image = doubling the frequencies

- f_1 becomes 2*f_1
- f_2 becomes 2*f_2
- All frequencies above N/2 are lost!

o Aliasing is the fact of high frequencies (above sampling rate) appearing as low frequency components

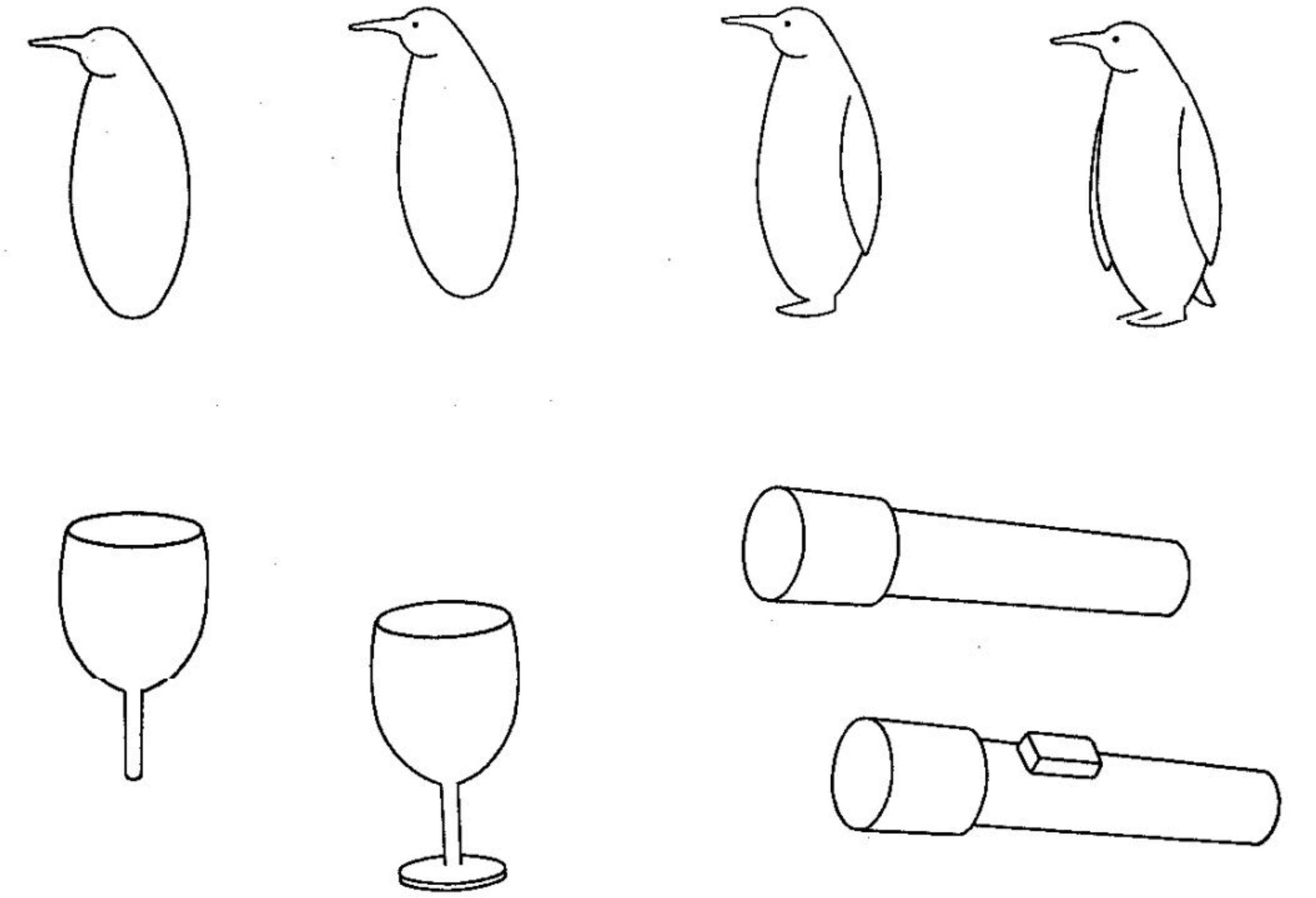
Courtesy of Hybrid Images, Oliva et al, SIGGRAPH 2006

❑ Intuition Building: Subsampling images and frequency adjustment

Notice anything new in their expressions ?

Courtesy of Hybrid Images, Oliva et al, SIGGRAPH 2006

- ❑ What does it say about human perception ?



High frequency information is salient !

Courtesy of Biederman (1987) : Recognition-by-components, a theory of human image understanding

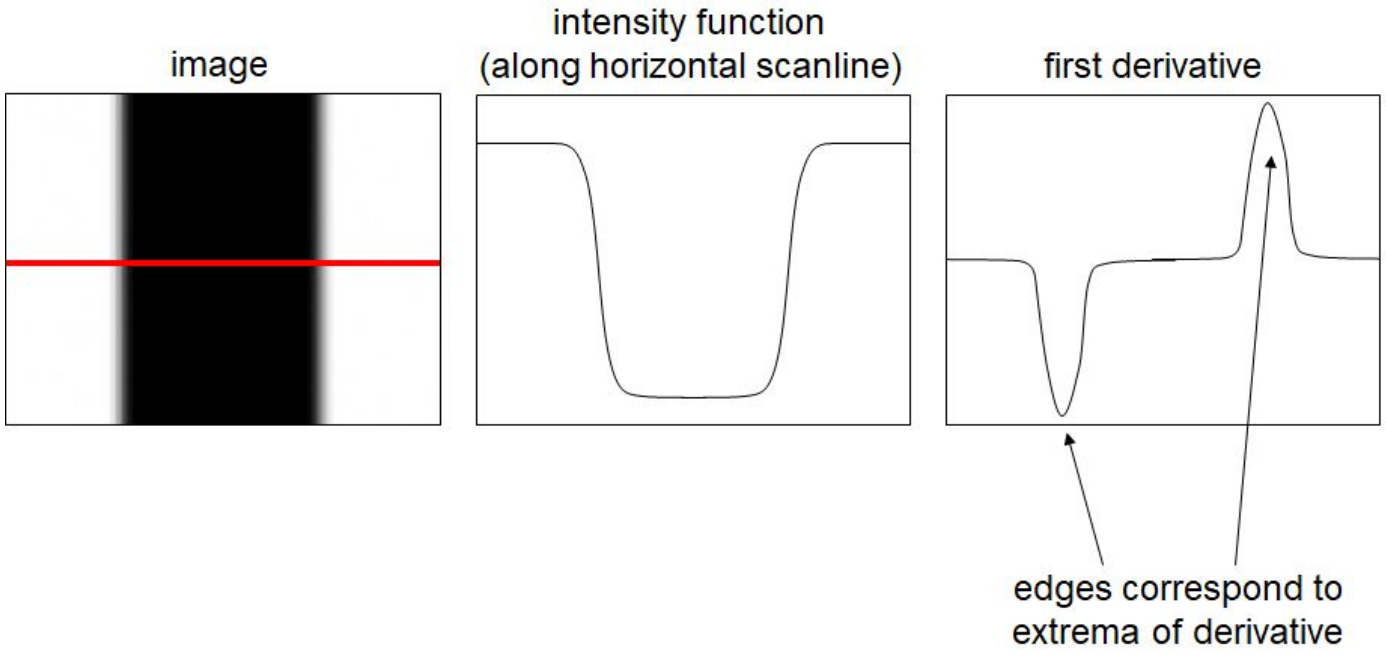
- What does it say about human perception ?

High frequency information can be recovered/filled-in by the human brain upto a certain extent !

Courtesy of Biederman (1987) : Recognition-by-components, a theory of human image understanding

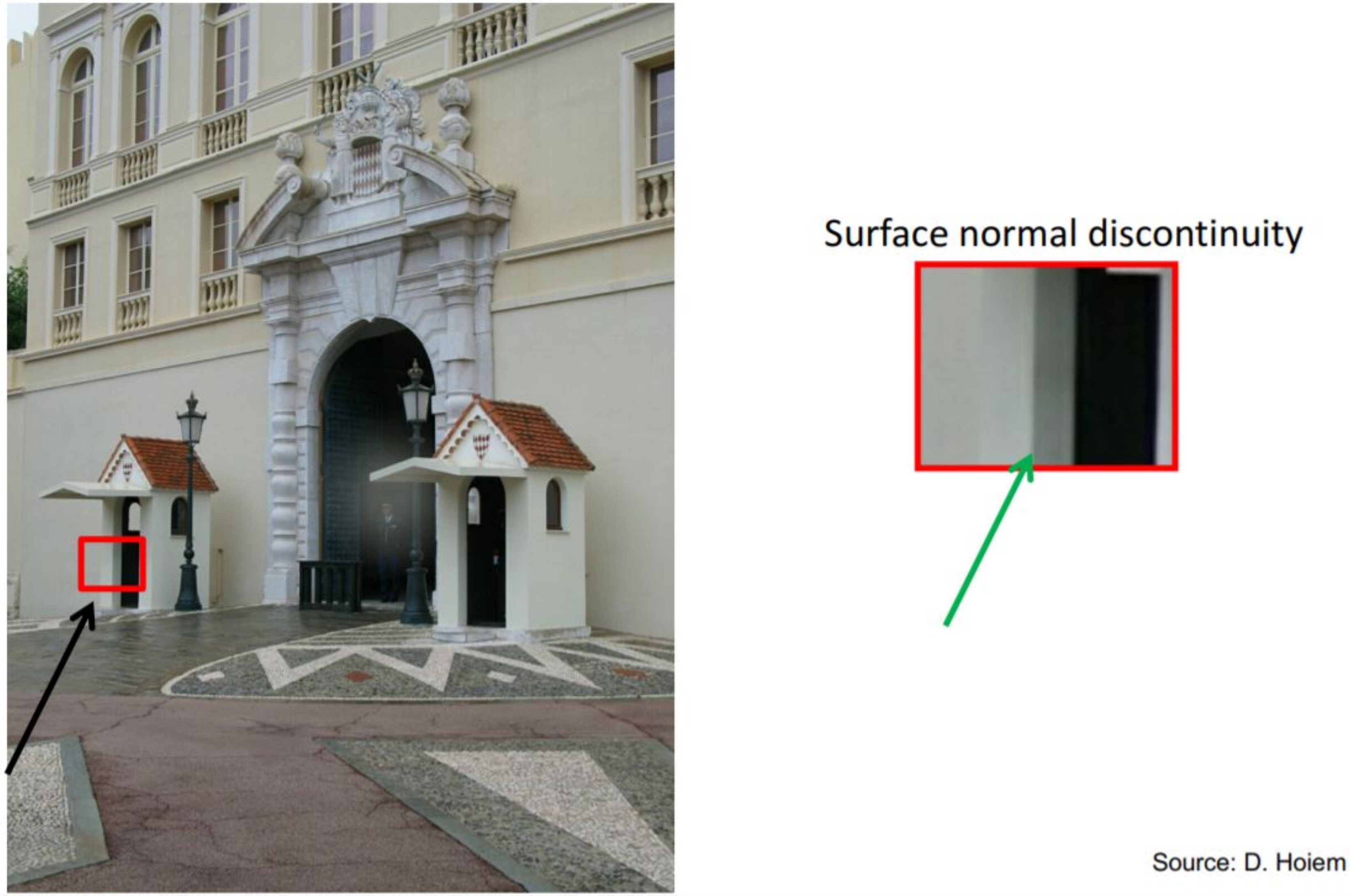
❑ EDGES contribute to high frequency information

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



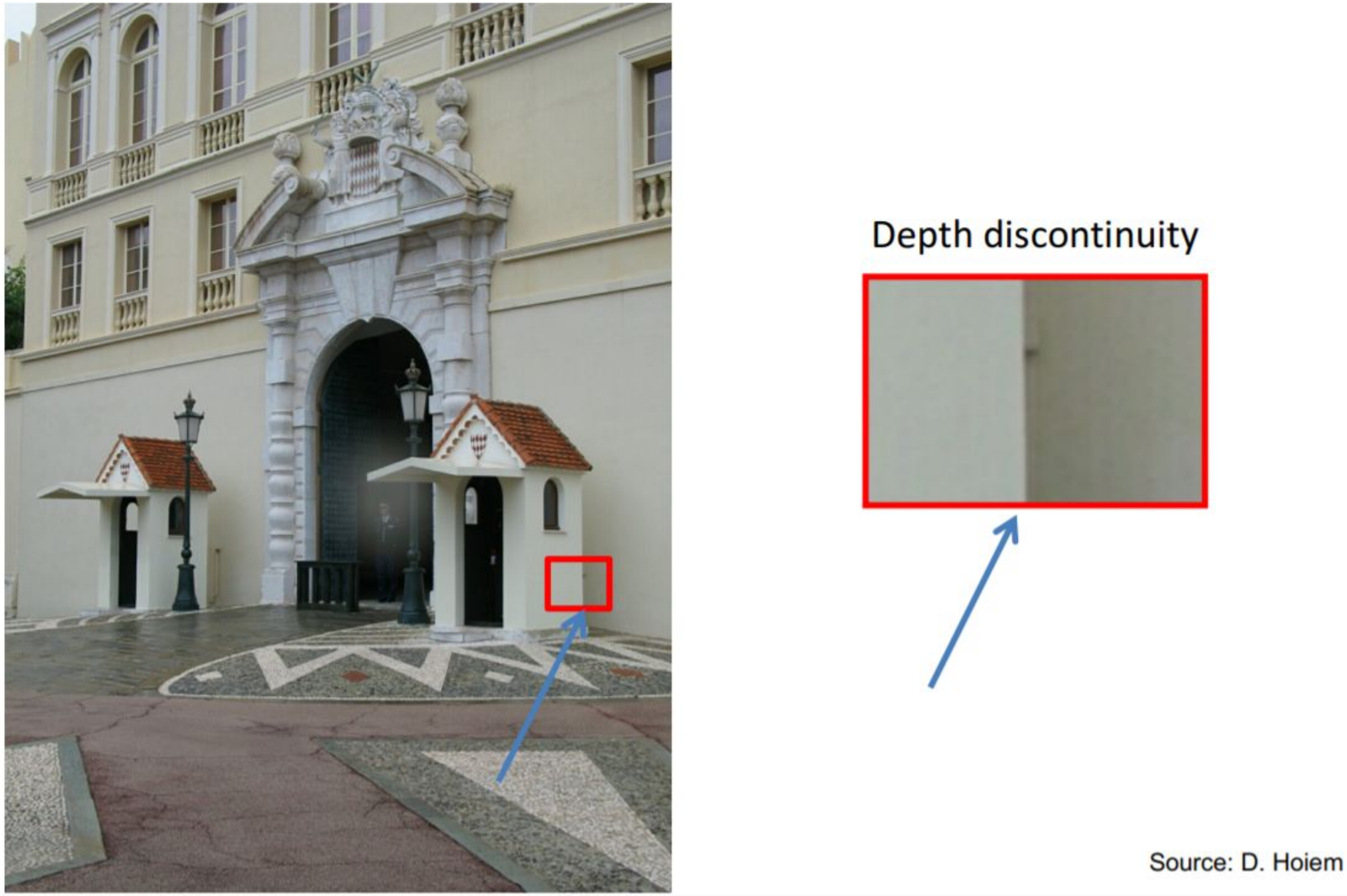
Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Where do they occur ?



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Where do they occur ?



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Where do they occur ?

Surface color discontinuity

Source: D. Hoiem

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟷ Estimating Image Gradients

Derivatives in 1D:

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

Derivatives in 1D: Solve these!

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

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

Derivatives in 1D: Solve these!

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

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

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES $\longleftrightarrow$ Estimating Image Gradients

Derivatives in 1D:

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

Types of discrete derivatives in 1D:

Backward $$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Forward $$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Central $$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

1D discrete derivative filters:

- Backward filter: [0 1 -1]

$$f(x) - f(x-1) = f'(x)$$

- Forward: [-1 1 0]

$$f(x) - f(x+1) = f'(x)$$

- Central: [1 0 -1]

$$f(x+1) - f(x-1) = f'(x)$$

- Backward filter: [0 1 -1]

$$f(x) - f(x-1) = f'(x)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES $\longleftrightarrow$ Estimating Image Gradients

Revisit the formula for convolution:

$$y(t) = u(t) \otimes h(t) = \int_{-\infty}^{t} u(\tau) h(t-\tau) d\tau.$$

- Extracting EDGES ⟵⟶ Estimating Image Gradients

1D discrete derivative filters:

- Backward filter: [0 1 -1]

$$f(x) - f(x-1) = f'(x)$$

- Forward: [-1 1 0]

$$f(x) - f(x+1) = f'(x)$$

- Central: [1 0 -1]

$$f(x+1) - f(x-1) = f'(x)$$

- Backward filter: [0 1 -1]

$$f(x) - f(x-1) = f'(x)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

1D discrete derivative filters:

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

Which filter has been applied here?

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟷ Estimating Image Gradients

1D discrete derivative filters:

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

Backward filter.

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES $\longleftrightarrow$ Estimating Image Gradients

2D discrete derivative filters:

Given function $f(x, y)$

Gradient vector
$$\nabla f(x, y) = \begin{bmatrix} \dfrac{\partial f(x, y)}{\partial x} \\ \dfrac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude
$$|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$$

Gradient direction
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

2D discrete derivative filters:

o What does this filter do ?

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

o What does this filter do ?

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Extracting EDGES ⟵⟶ Estimating Image Gradients

2D discrete derivative filters:

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$\boxed{\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}}$$

$$\longrightarrow$$

$$I_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

2D discrete derivative filters:

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$\boxed{\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}}$$

$$\longrightarrow \quad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients

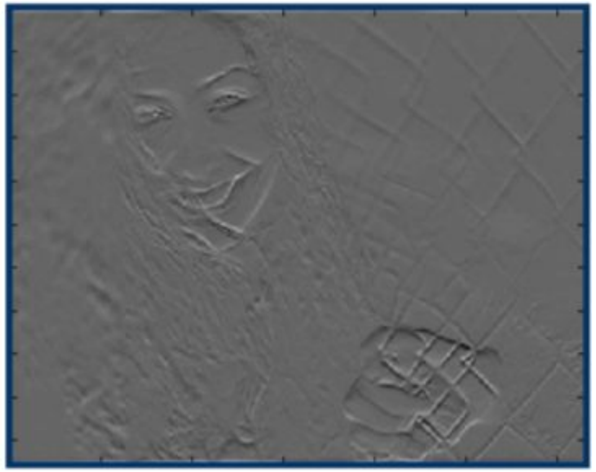
2D discrete derivative filters on natural images:

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

x-direction

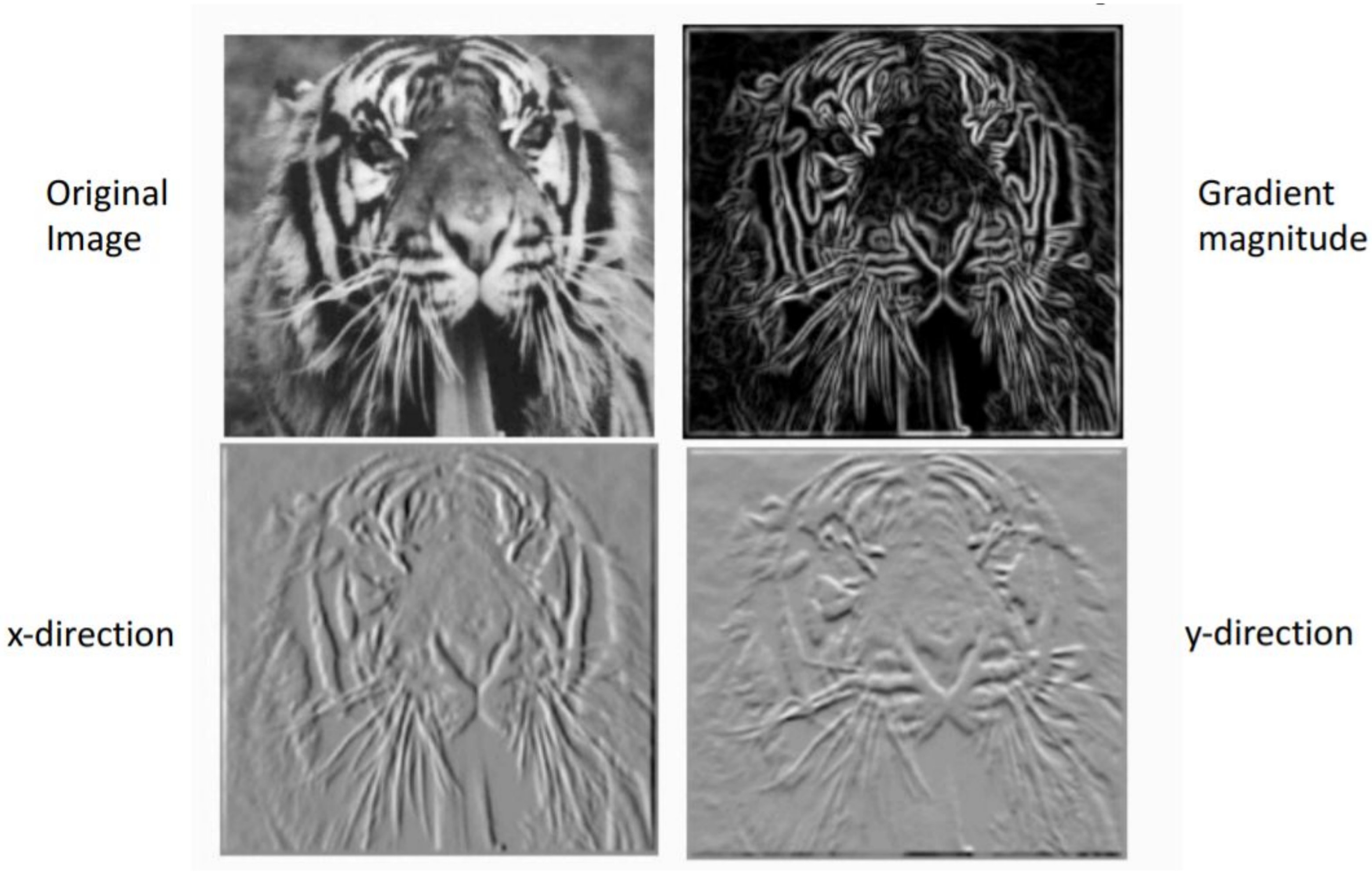
$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

y-direction



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟵⟶ Estimating Image Gradients



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES ⟷ Estimating Image Gradients

DEMYSTIFYING: 2D discrete derivative filters:

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

o Why is the filter on the left **x-direction filter**, and on the right **y-direction filter** ?

o How are they estimated ? Go from 1D to 2D ?

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Extracting EDGES $\longleftrightarrow$ Estimating Image Gradients

DEMYSTIFYING: 2D discrete derivative filters:

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [-1\ 0\ 1]$$

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * [1\ 1\ 1]$$

KEY is in filter decomposition!

- Extracting EDGES $\longleftrightarrow$ Estimating Image Gradients

DEMYSTIFYING: 2D discrete derivative filters:

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [-1\ 0\ 1]$$

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * [1\ 1\ 1]$$

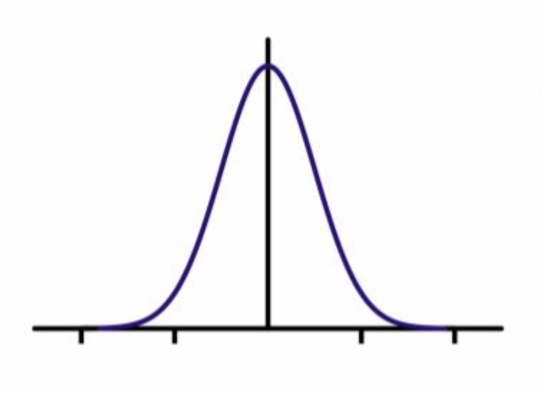
KEY is in filter decomposition!

❑ Extracting EDGES with smoothing filters

- Mean smoothing

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

- Gaussian (smoothing * derivative)



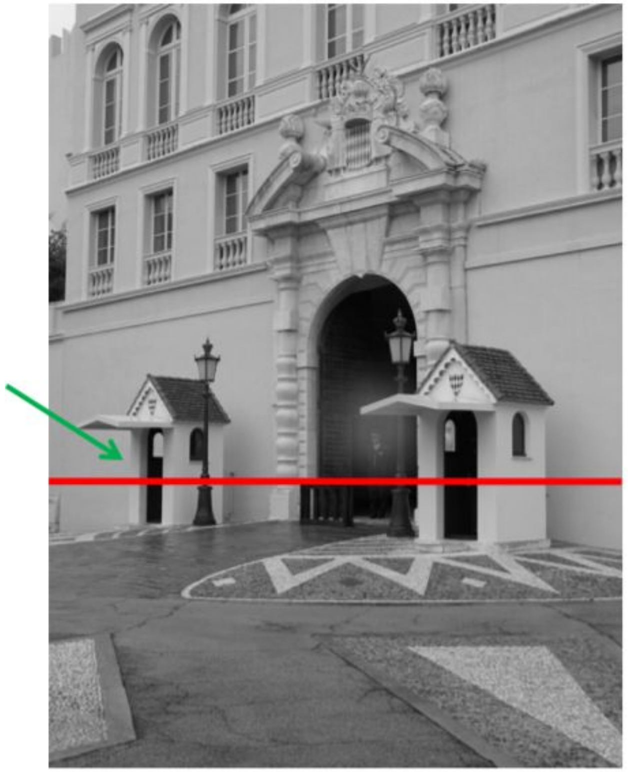
$$\begin{bmatrix}1\\2\\1\end{bmatrix} \qquad \begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

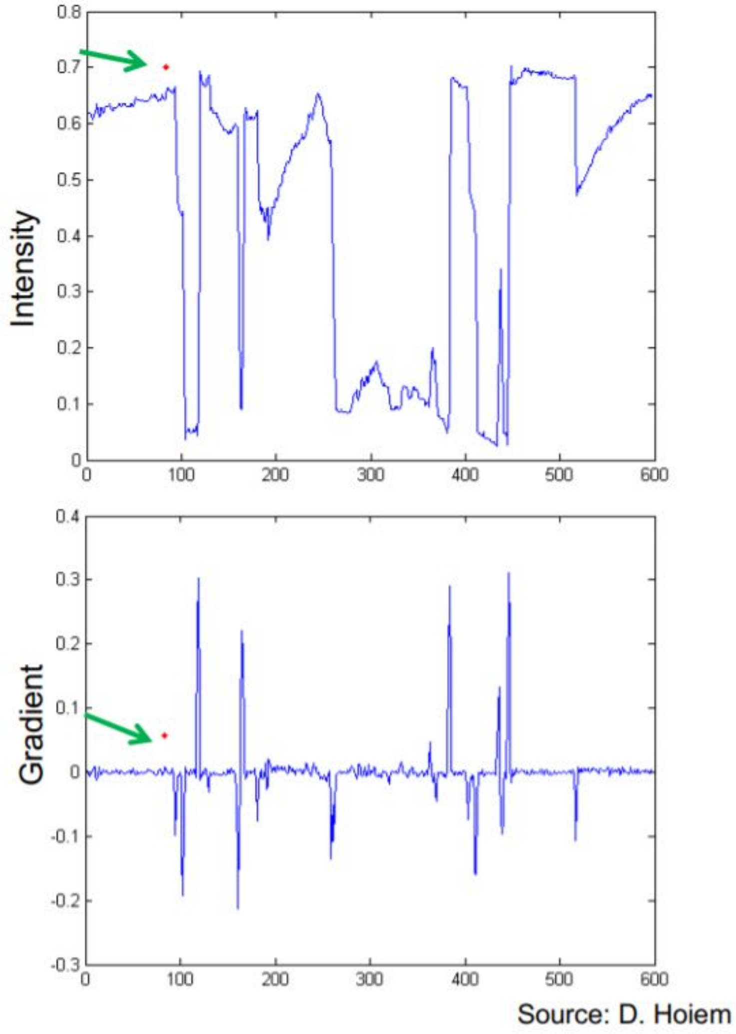
Slide credit: Steve Seitz

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- ❑ How does an image look at the pixel level ?

Intensity profile



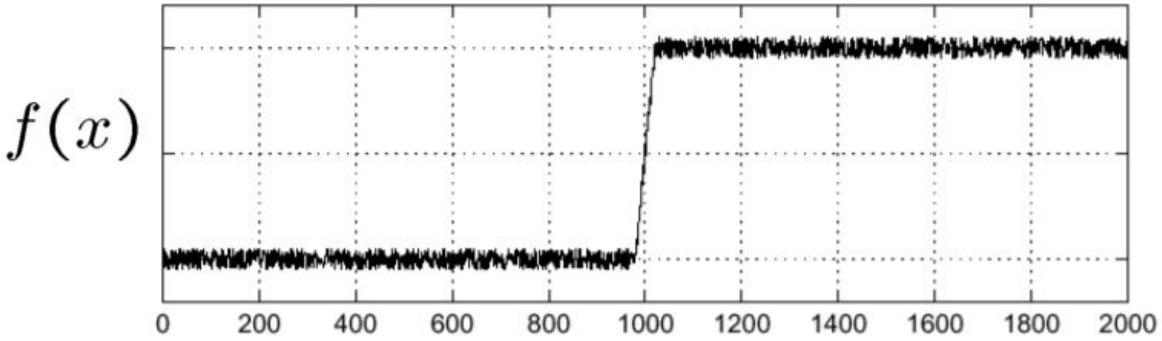


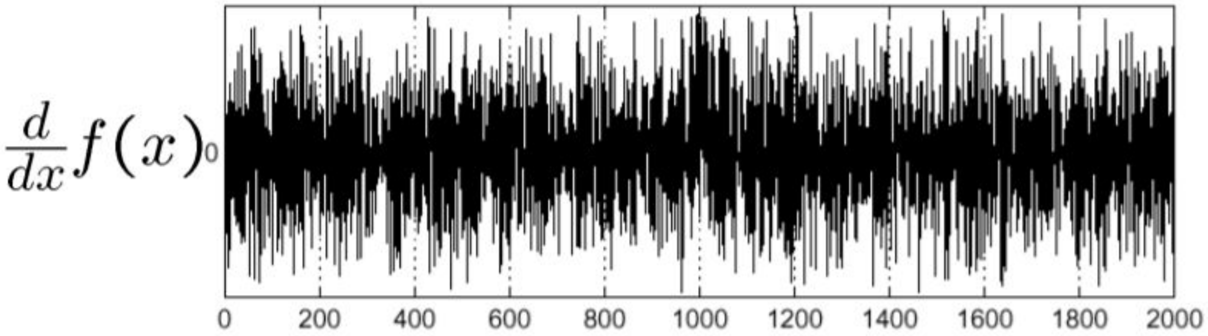
NOISY!

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Finding the edge at the pixel level !

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



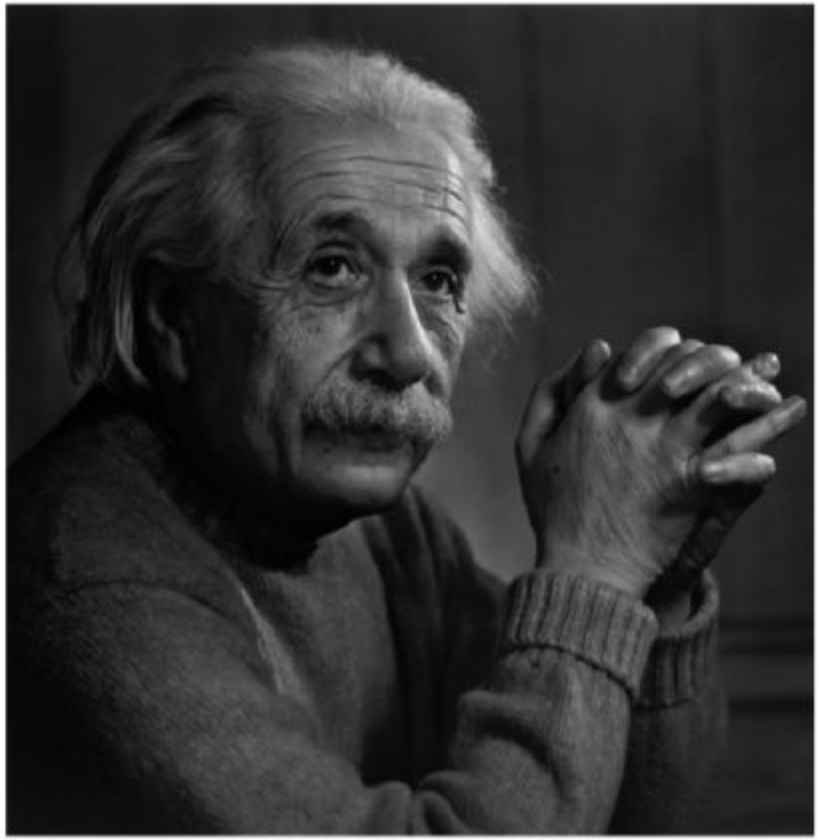


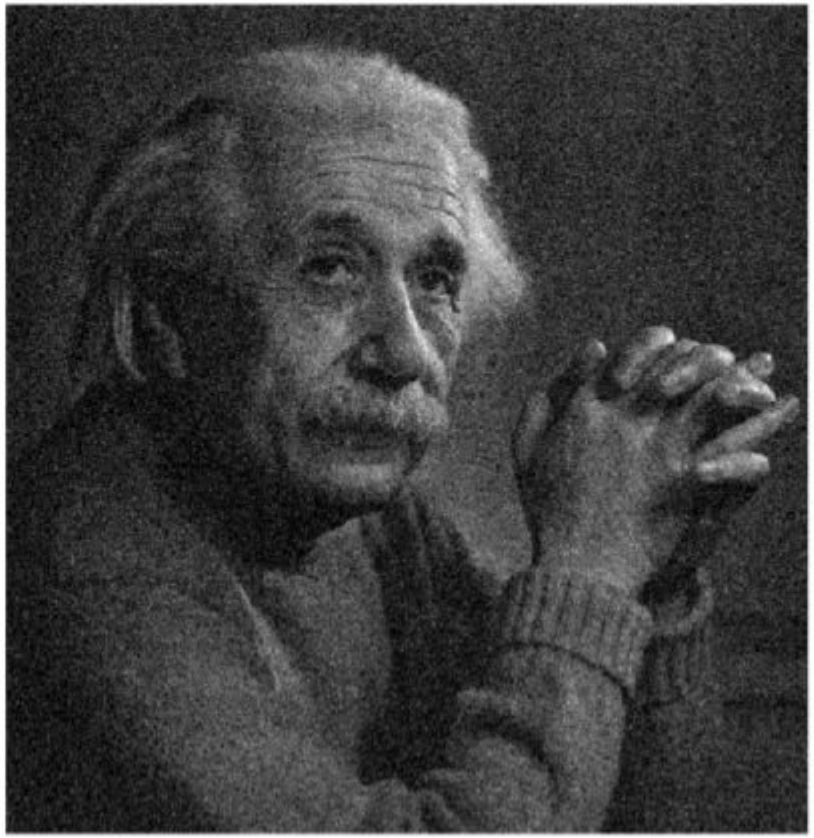
Where is the edge?

Source: S. Seitz

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Overcoming NOISE





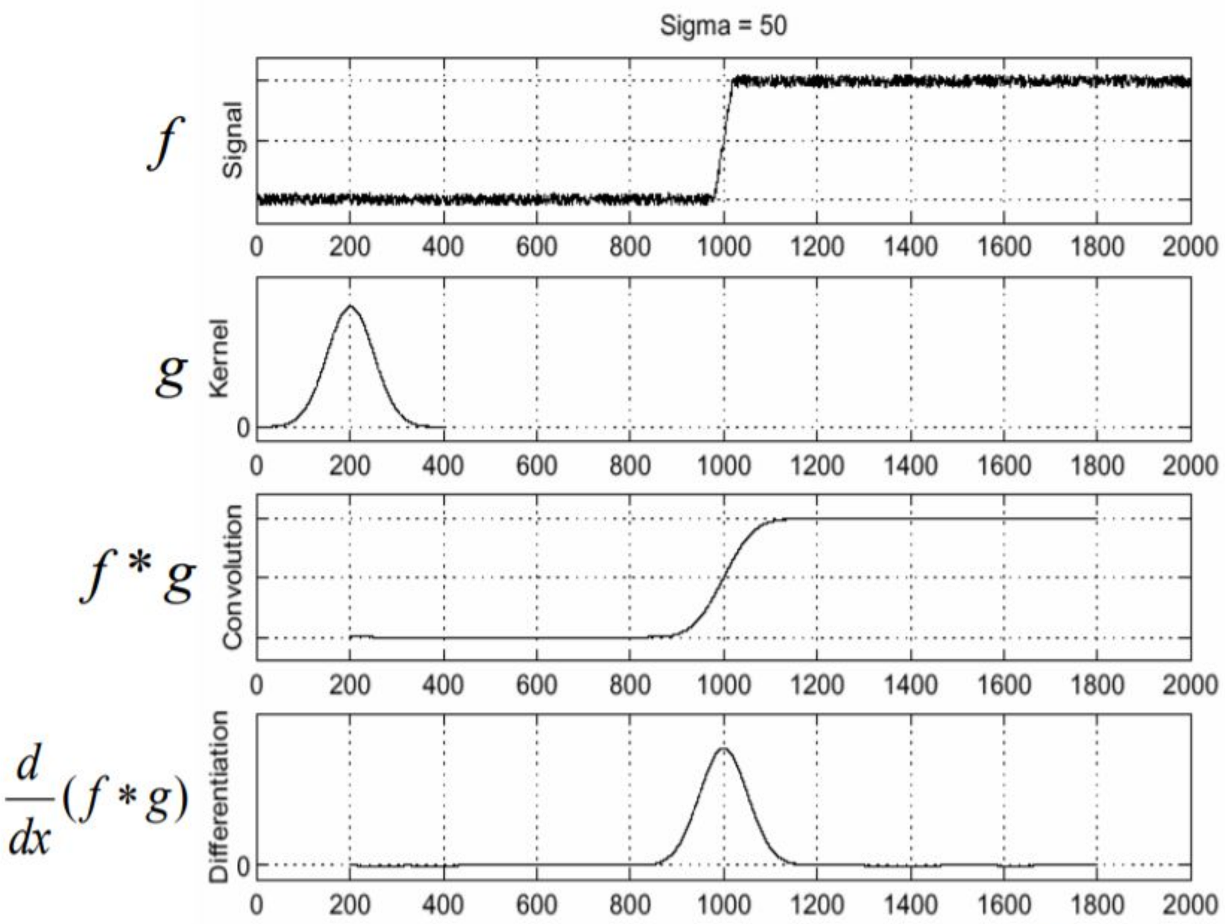
Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Overcoming NOISE: Different filters at different sizes

Slide credit: Steve Seitz

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Extracting EDGES with smoothing filters



- To find edges, look for peaks in $\frac{d}{dx}(f * g)$

Source: S. Seitz

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Extracting EDGES with smoothing filters: A smarter way!

The derivative theorem of CONVOLUTION:

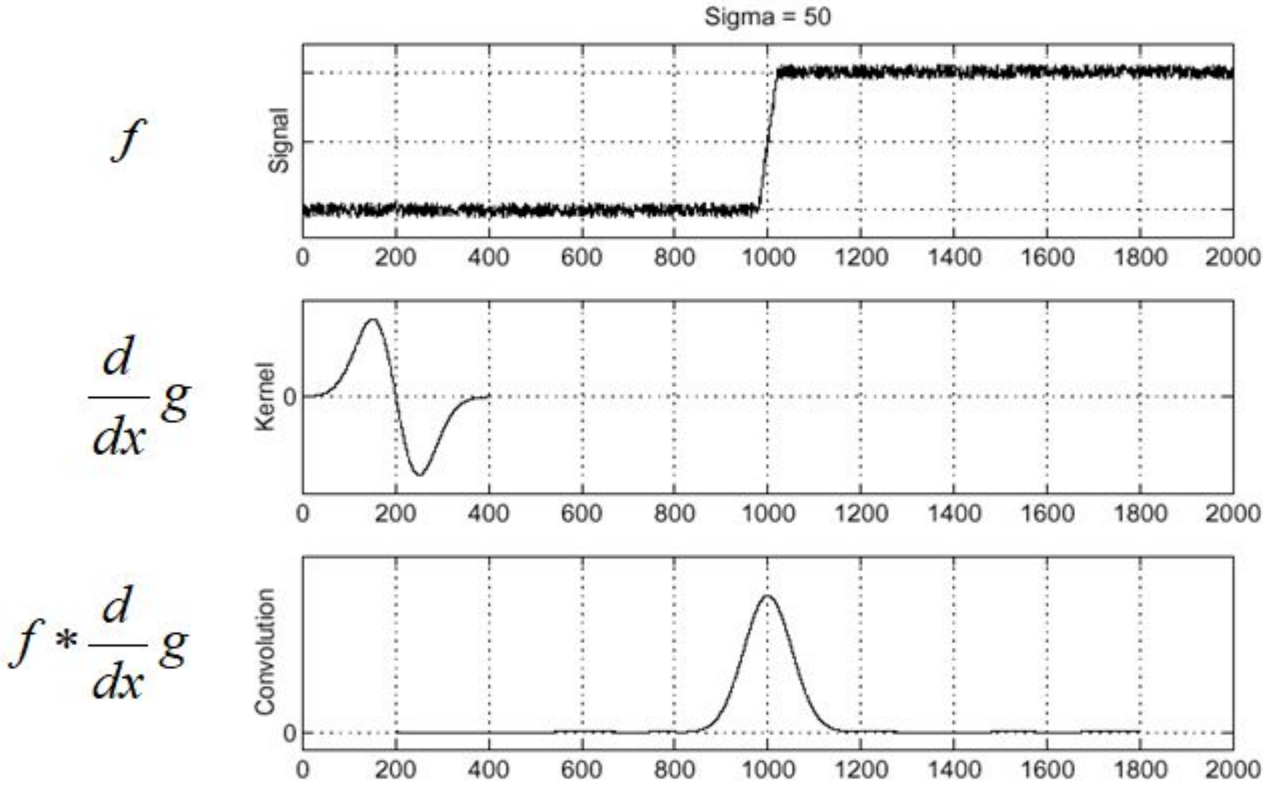
- This theorem gives us a very useful property:

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Extracting EDGES with smoothing filters: A smarter way!

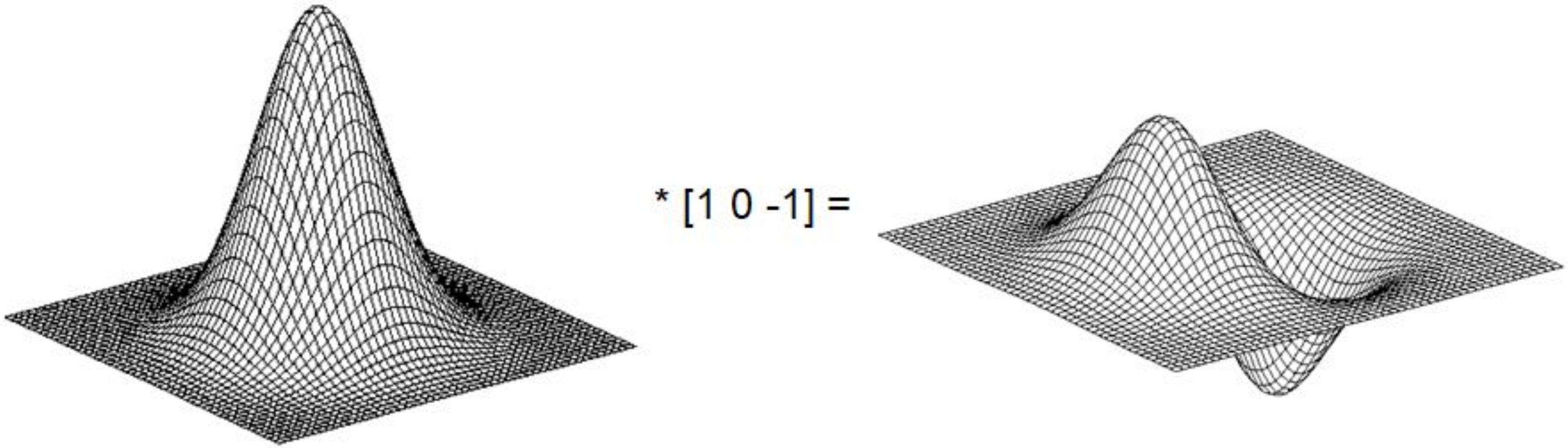
The derivative theorem of CONVOLUTION saves us one operation:



FIXED MASK/FILTER!
[One time operation]

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Visualising the SINGLE STEP: Derivative of Gaussian



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Visualising ‘Derivative of Gaussian’ in numbers!

- Smoothing + differentiation

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}}_{\text{Gaussian smoothing}} \underbrace{\begin{bmatrix} +1 & 0 & -1 \end{bmatrix}}_{\text{differentiation}}$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

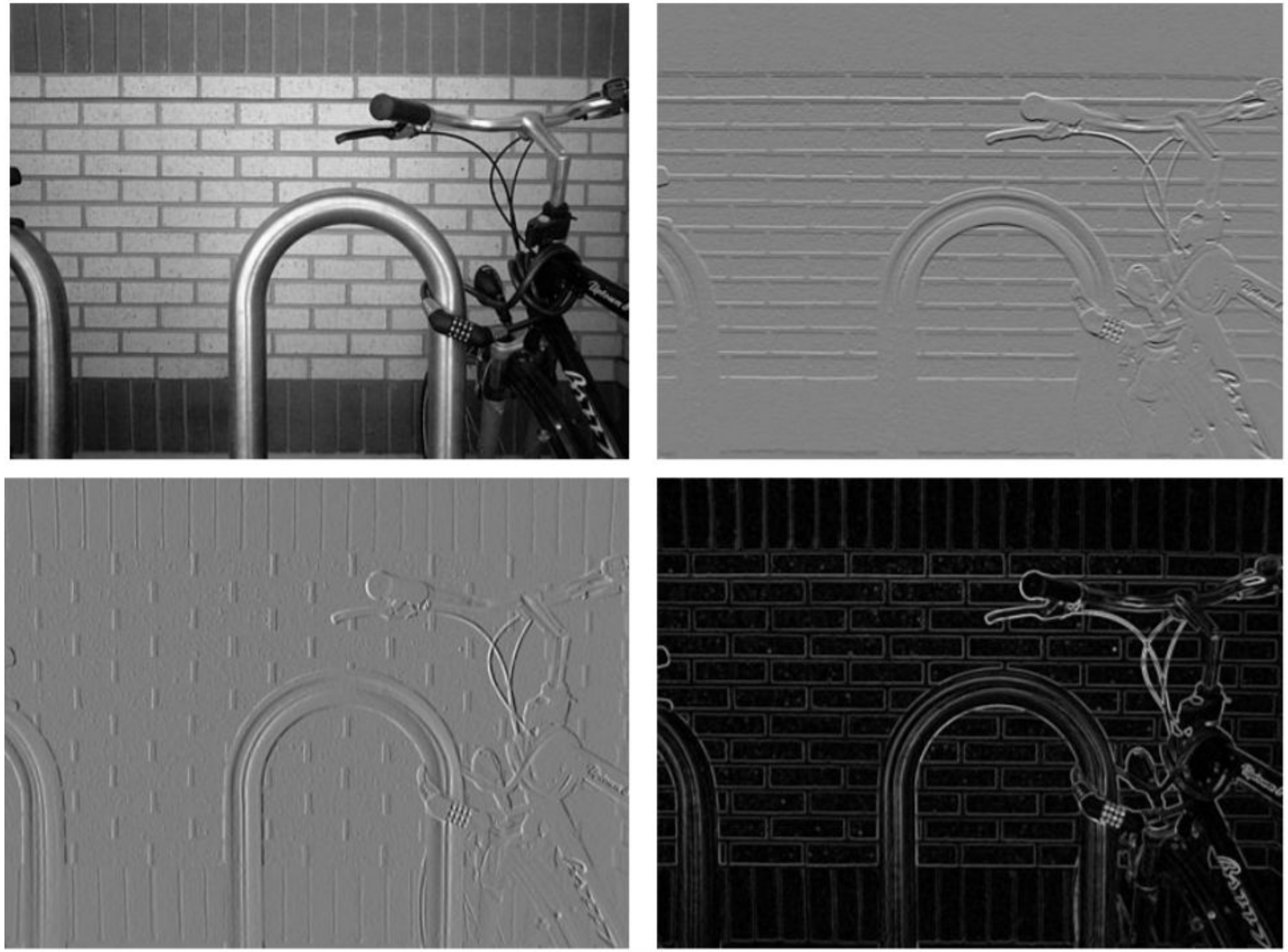
- Visualising ‘Derivative of Gaussian’ in numbers!

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \qquad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel Operator

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Sobel in Operation



- Magnitude:

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

- Angle or direction of the gradient:

$$\mathbf{\Theta} = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

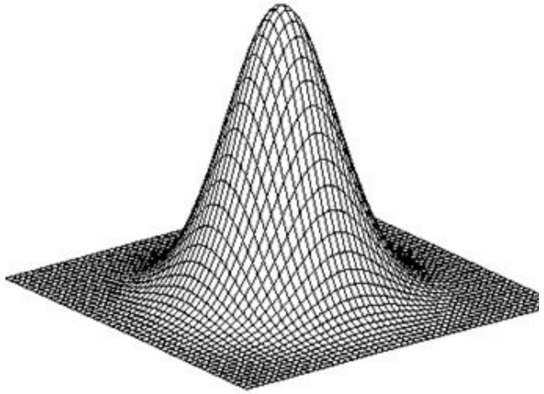
❑ Problems with Sobel Operator

1. Poor Localisation: Triggers response in several adjacent pixels

Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Problems with Sobel Operator

1. Poor Localisation: Triggers response in several adjacent pixels
   Gaussian smoothing

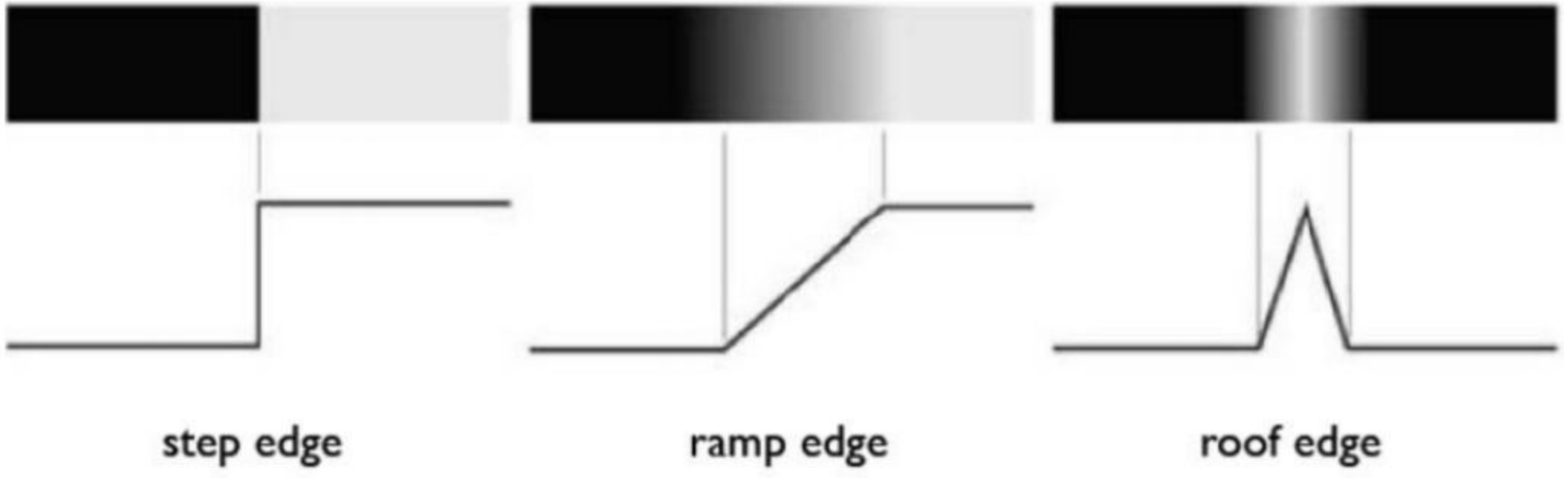


Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Problems with Sobel Operator

1. Poor Localisation: Triggers response in several adjacent pixels
   Gaussian smoothing

2. Thresholding of magnitude favours certain directions
   - Can **miss oblique edges**
   - **Favours step edge** over others
   - **False negatives**



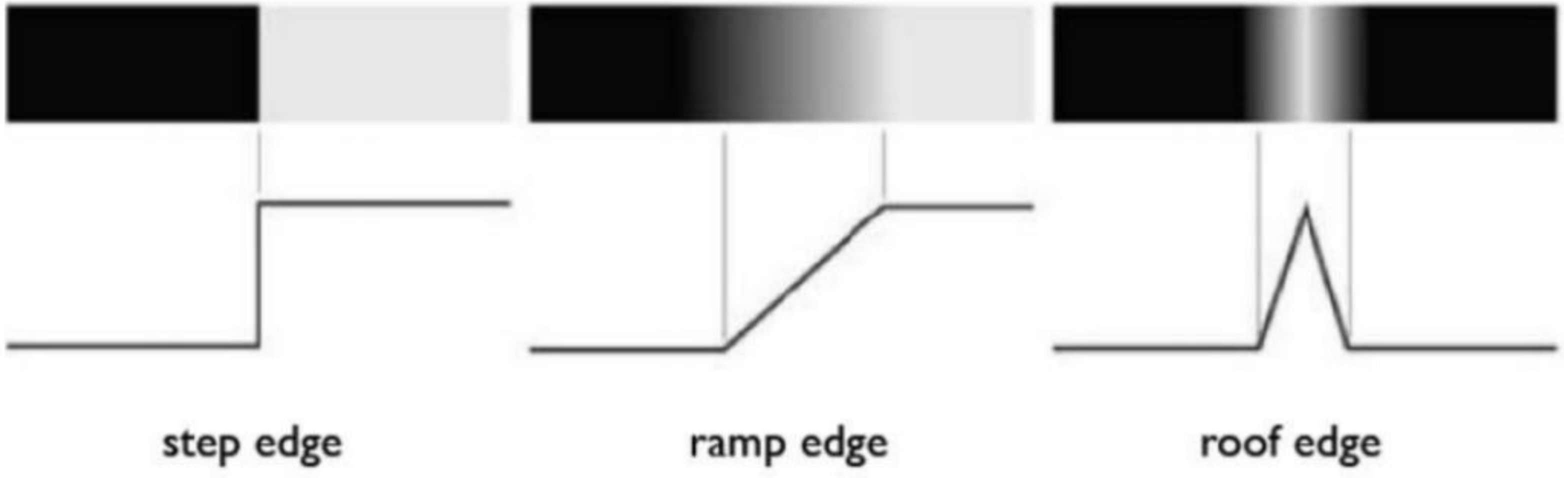
Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Problems with Sobel Operator

1. Poor Localisation: Triggers response in several adjacent pixels
   Gaussian smoothing

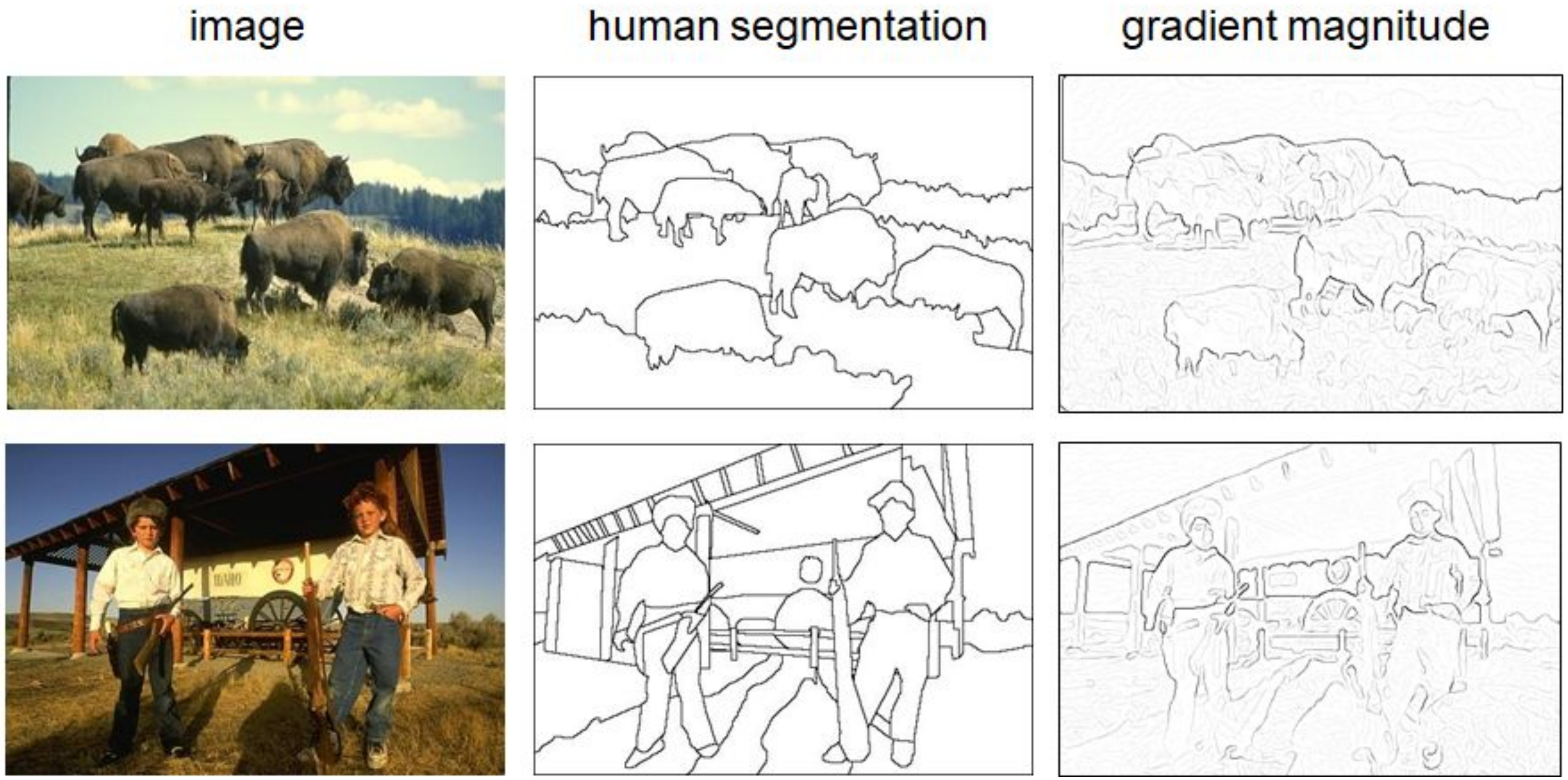
2. Thresholding of magnitude favours certain directions
- Can **miss oblique edges**
- **Favours step edge** over others
- **False negatives**

How could you overcome ?



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

❑ Machines versus Humans at edge detection ?



Courtesy of http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf

- Homework

  Canny edge detector: Nuances beyond vanilla edge detectors

  Hints:

  - Weak and strong edges
  - Non-maximal suppression

Thank you!