GRAPHICS GROUP

Intelligent Vision Systems New Zealand

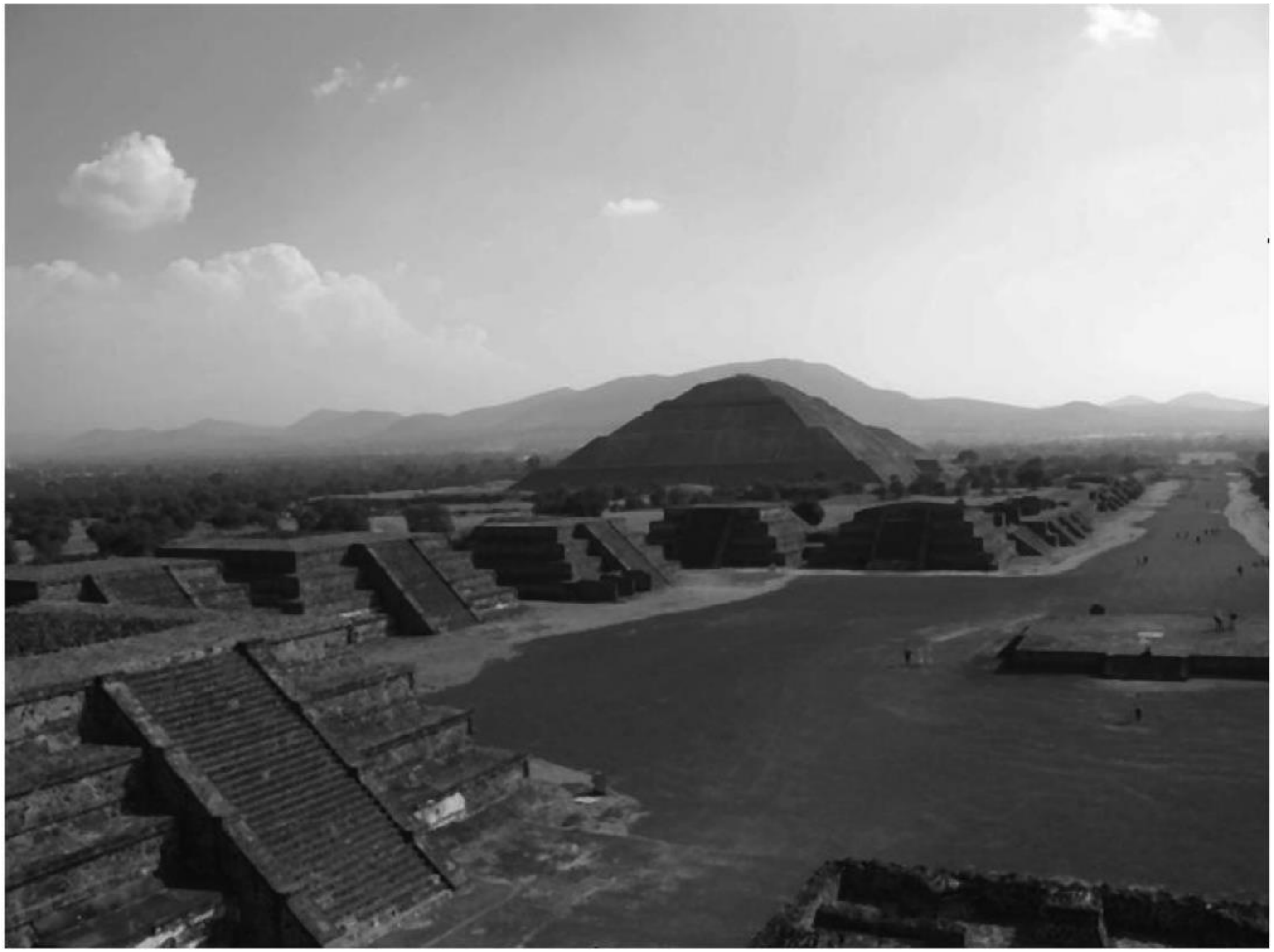
# Computer Graphics and Image Processing

Part 3: Image Processing

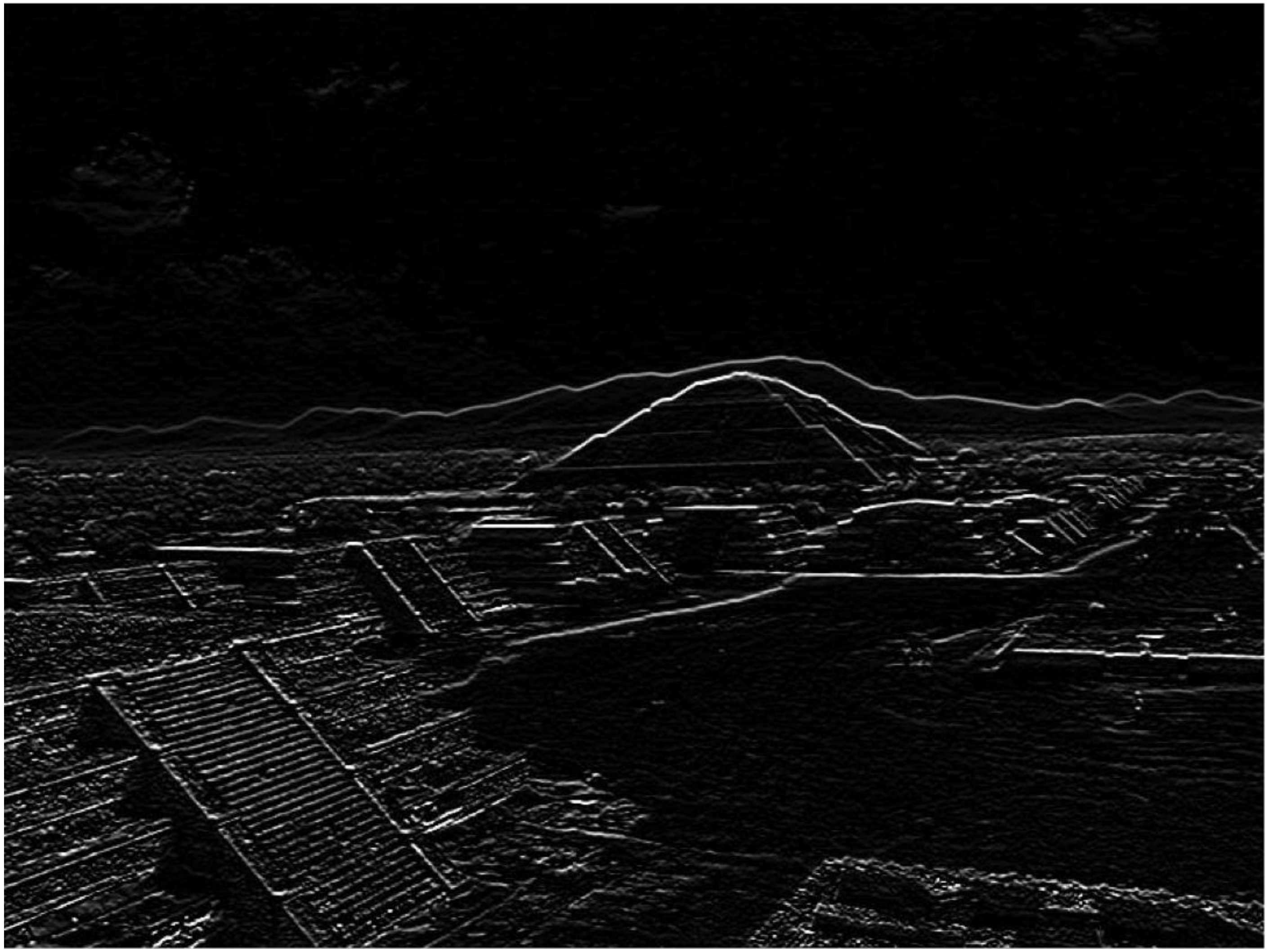
5 – Edges

*Martin Urschler, PhD*

## Edges are important image features



# Edges are important image features

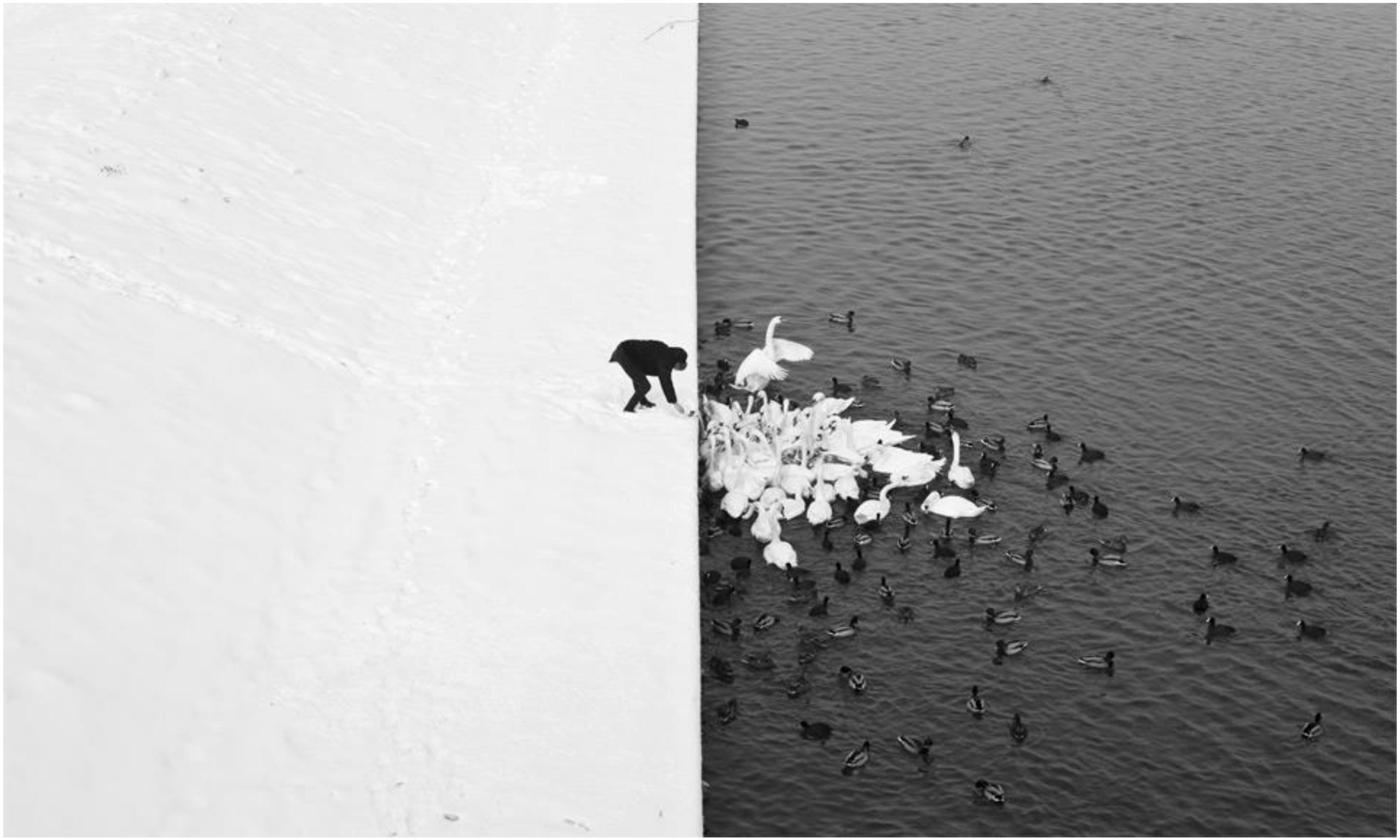


Semantic & shape information!

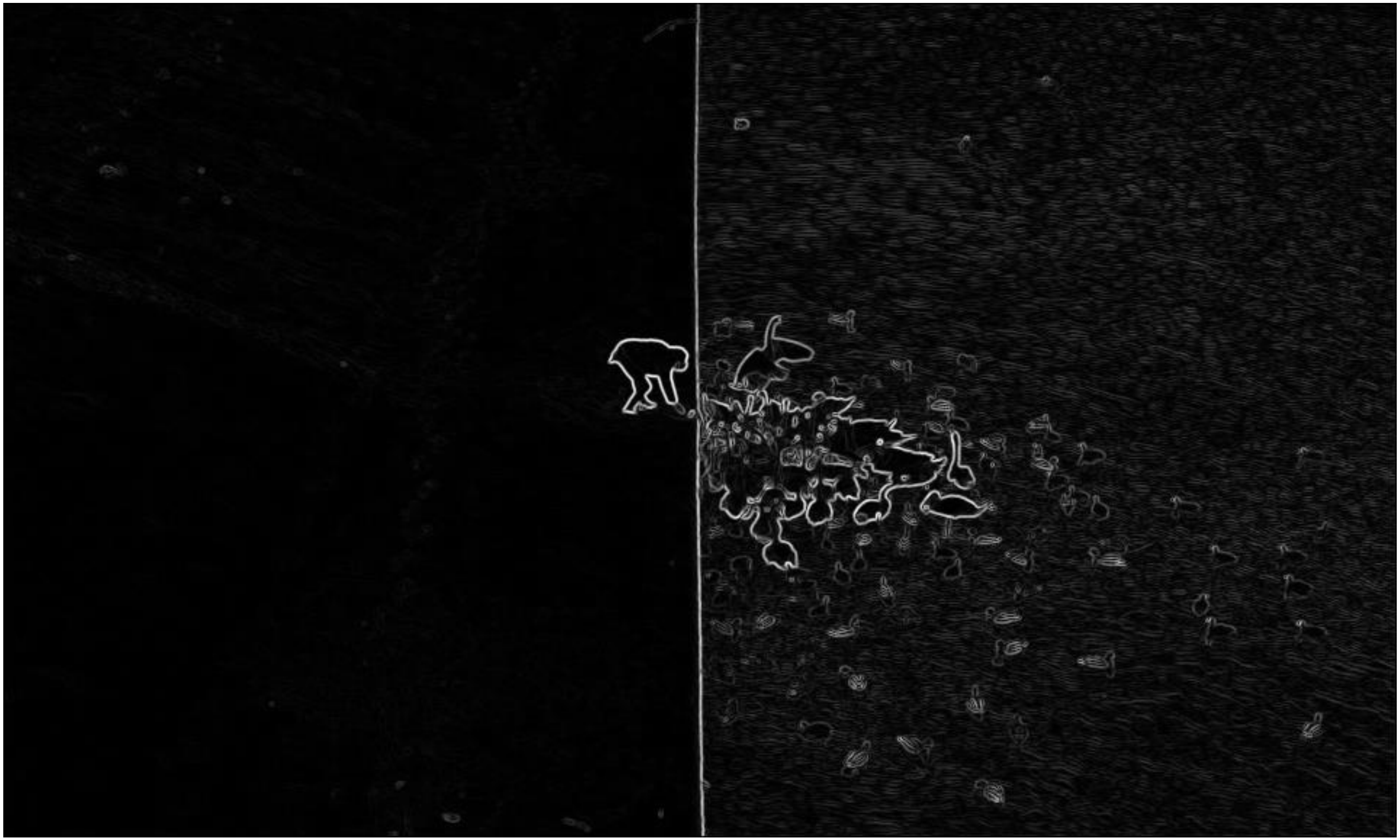
# Moving window transform for edges

- Edge detection is a major application of image filtering
- Informal definition of an edge:
  - Locations of sudden grey level / color changes
  - Transition between objects or between object and background
  - Locations that attract visual attention (salient regions)
- Problem: Noise in the image has similar properties!
- Conventional 3-step edge detection approach:
  1. Noise reduction (preserving edges as much as possible)
  2. Edge enhancement
  3. Edge localization (single pixel wide edges)

## Example image

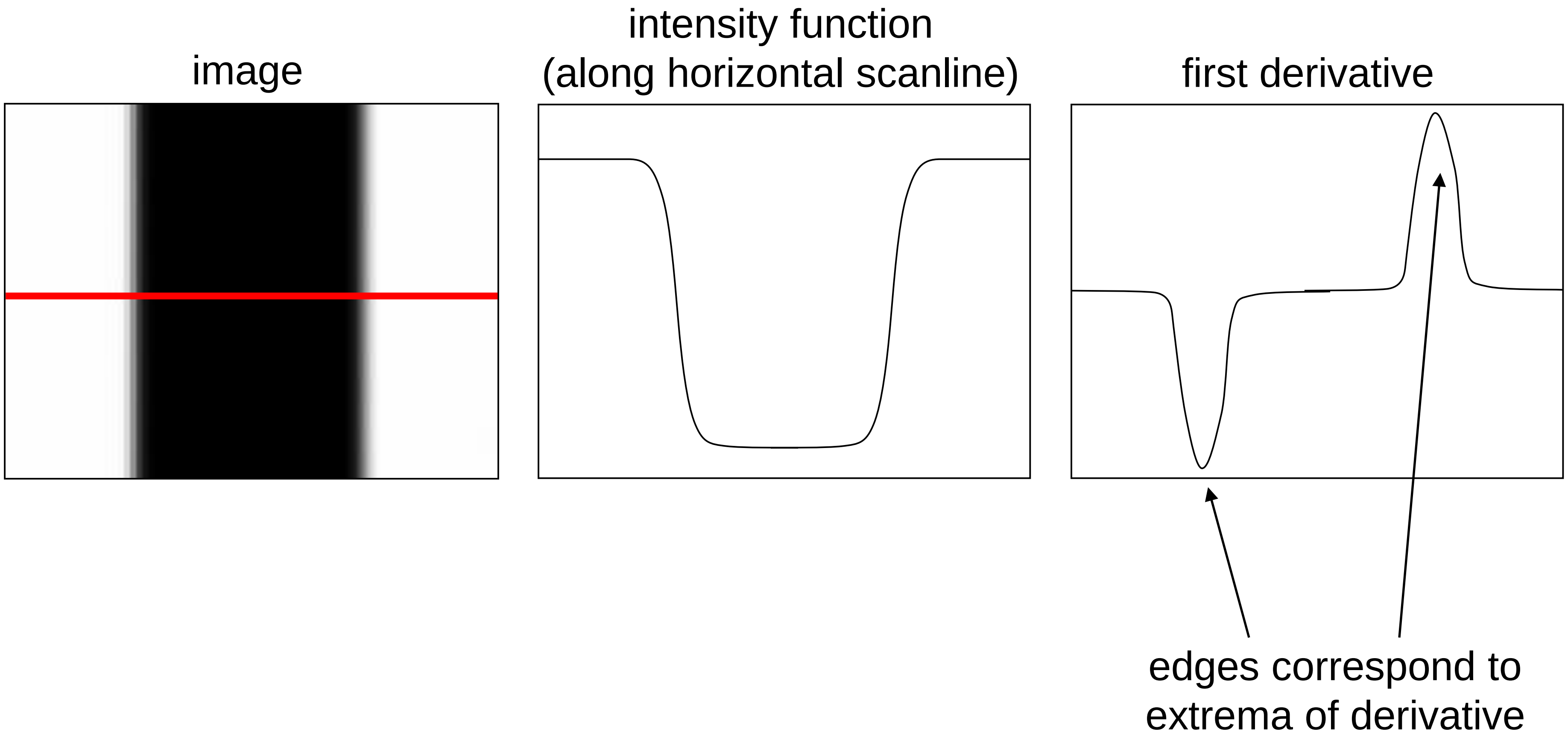


# Edges



# Edge detection

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



## Moving window transform for edges

- Estimation of the grey level gradient at pixels:

$$g_x(x,y) = \frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x-1,y)}{2}$$

$$g_y(x,y) = \frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1) - f(x,y-1)}{2}$$

Finite difference approximations

# Derivative filter in x (vertical edge)

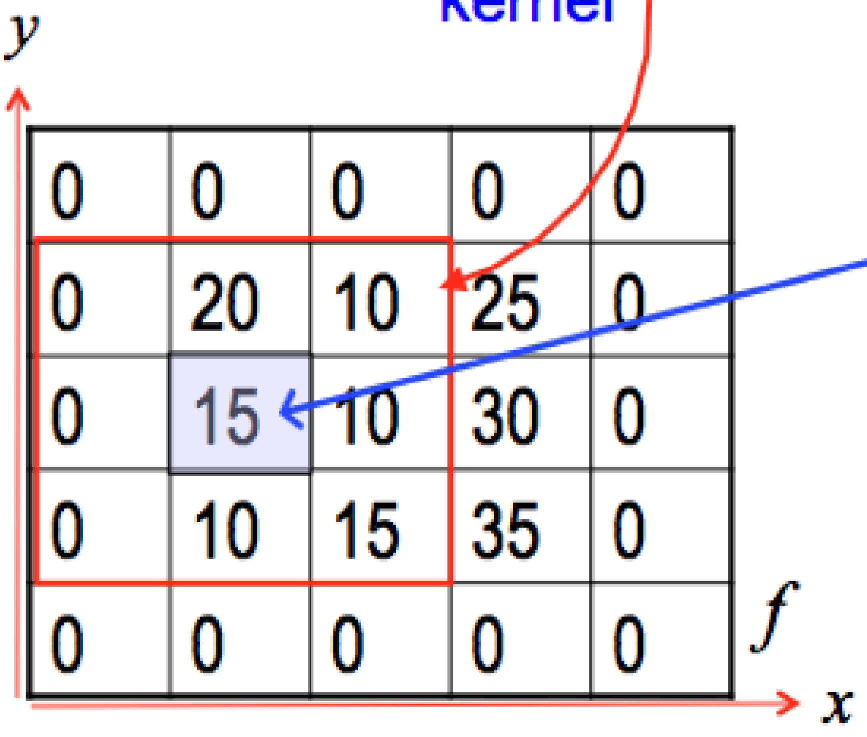
Prewitt operator

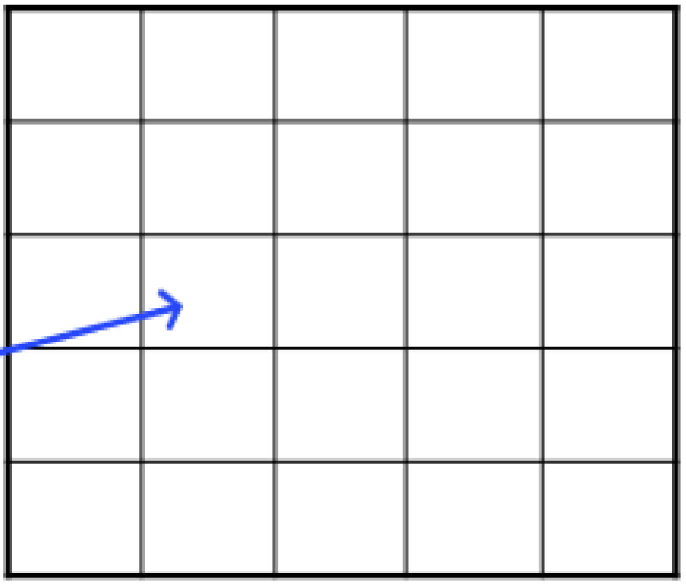
| | | |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

kernel

Prewitt combines derivative and Smoothing in a 3x3 kernel!

$g = \mathrm{MWT}(f) \Rightarrow$





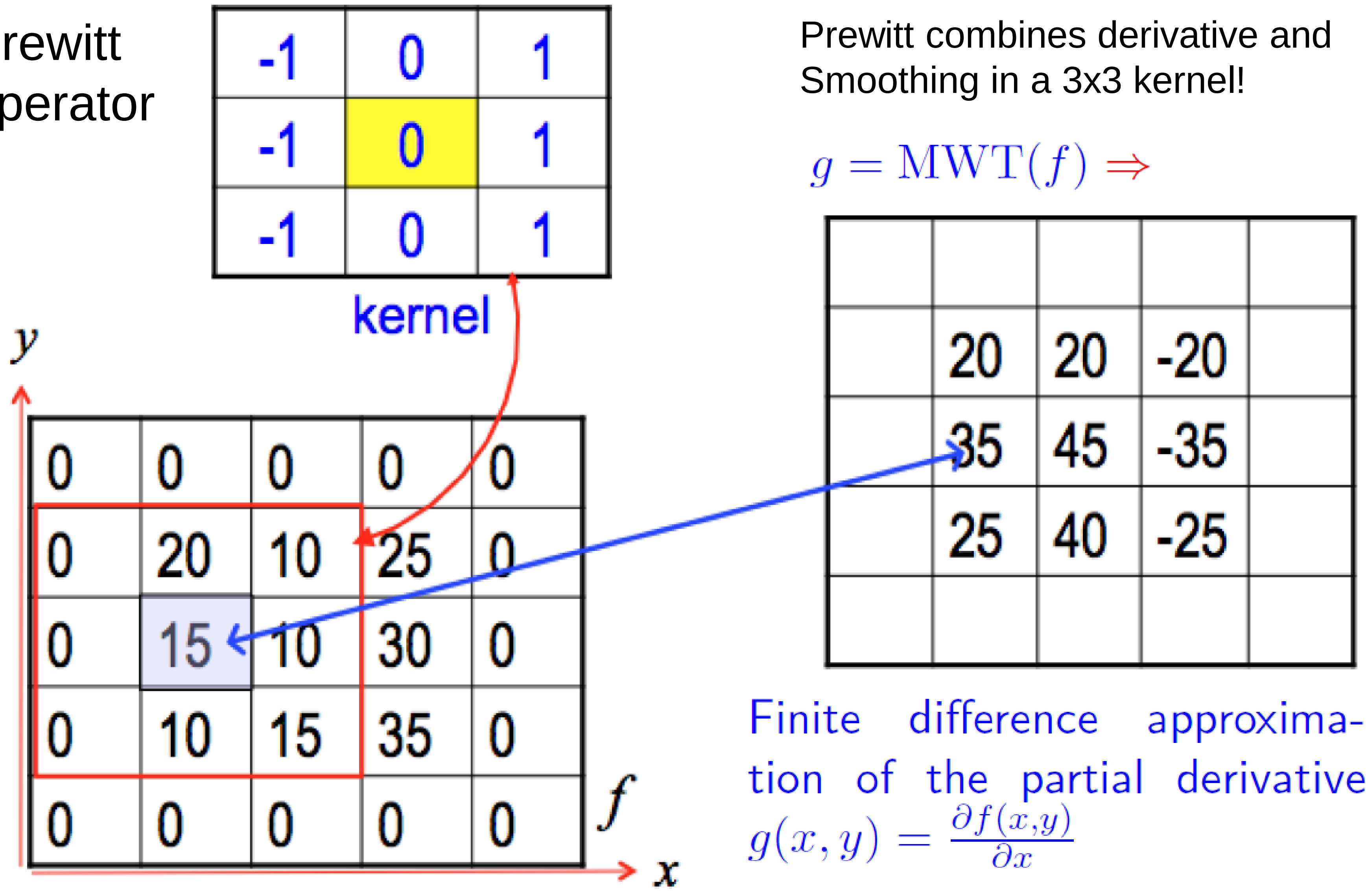
Finite difference approximation of the partial derivative $g(x,y) = \frac{\partial f(x,y)}{\partial x}$

# Derivative filter in x (vertical edge)

Prewitt operator

Prewitt combines derivative and Smoothing in a 3x3 kernel!

$g = \mathrm{MWT}(f) \Rightarrow$



|  |  |  |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 20 | 10 | 25 | 0 |
| 0 | 15 | 10 | 30 | 0 |
| 0 | 10 | 15 | 35 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  |  |
|  | 20 | 20 | -20 |  |
|  | 35 | 45 | -35 |  |
|  | 25 | 40 | -25 |  |
|  |  |  |  |  |

Finite difference approximation of the partial derivative $g(x, y) = \frac{\partial f(x,y)}{\partial x}$

# Moving window transform for edges

- Noise smoothing using a low-pass filter (mean, Gaussian, etc.)
  - Separable Prewitt kernels (includes $3x3$ mean):

$$\frac{1}{6}\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix} \qquad \frac{1}{6}\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{bmatrix}$$

vertical edge

horizontal edge

  - Separable Sobel kernels (includes $3x3$ weighted mean, as an approximation to Gaussian kernel):

$$\frac{1}{8}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \qquad \frac{1}{8}\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

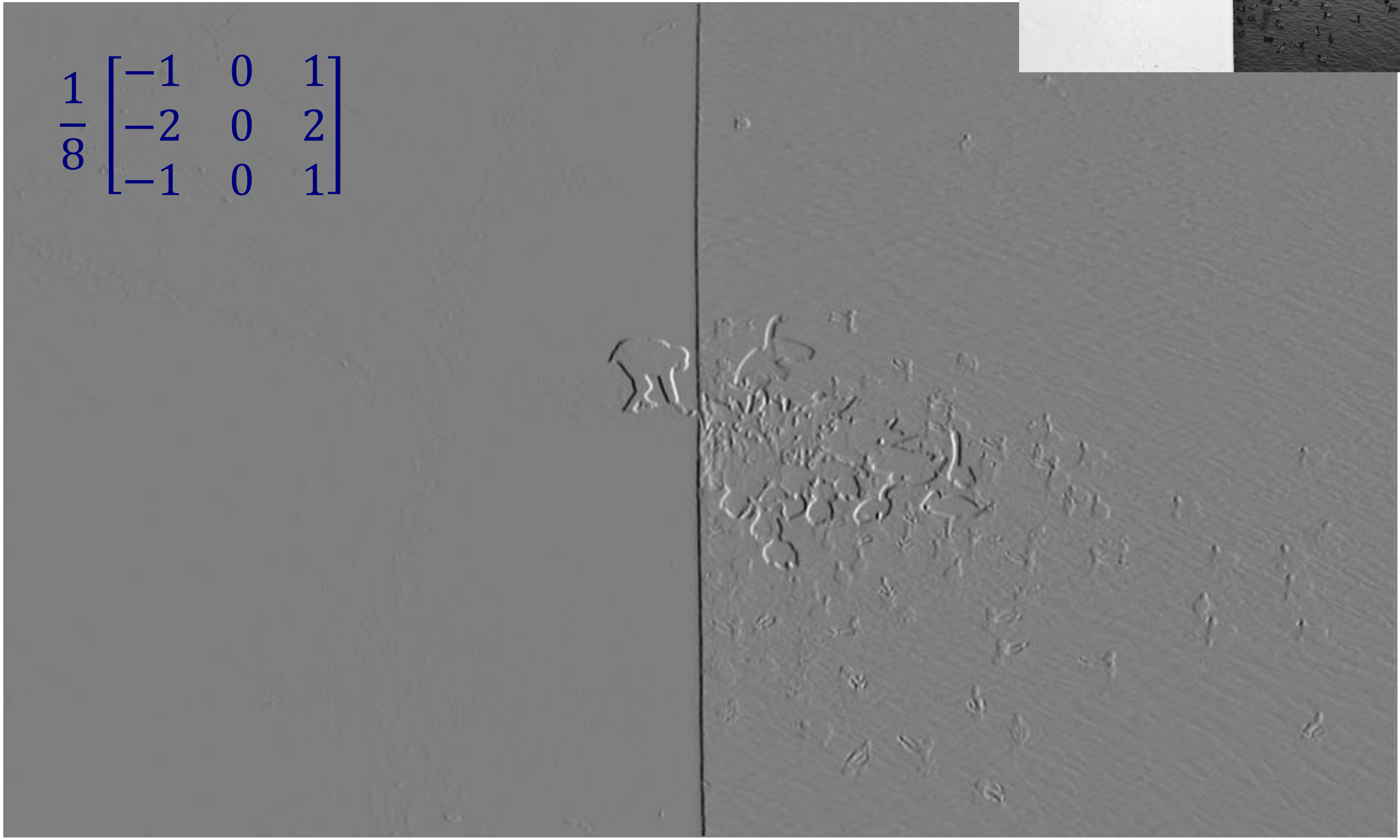
## Moving window transform for edges

- Separable Sobel kernel by (outer) matrix product

$$\frac{1}{8}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} = \frac{1}{8}\begin{bmatrix}1\\ 2\\ 1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

$$\frac{1}{8}\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} = \frac{1}{8}\begin{bmatrix}1\\ 0\\ -1\end{bmatrix}\begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

# Gradient in x: vertical edges

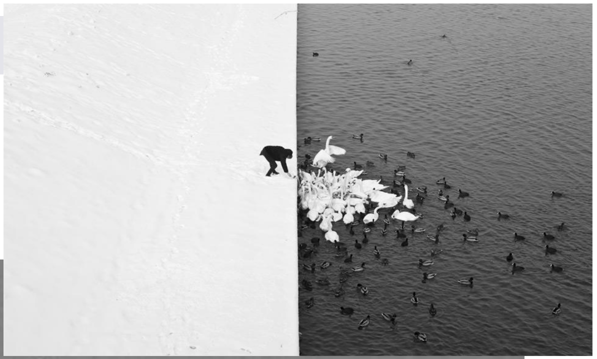


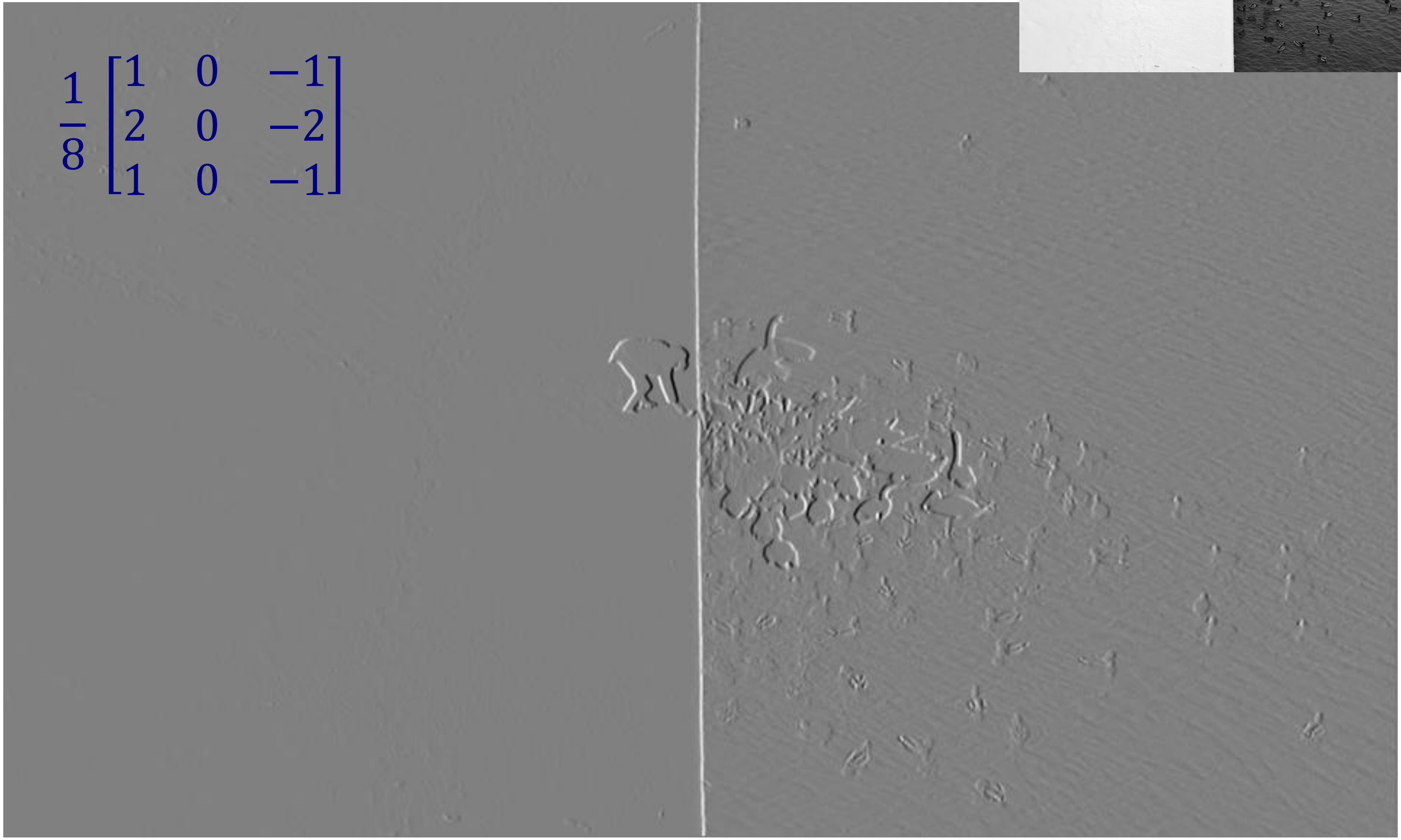
$$\frac{1}{8}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$$

## Edge detection using gradients

- Attention: Sign-alternating kernels!
  - Output images with positive and negative values
  - Display: by mapping zero gradient to mid-grey level
    - Positive gradient values appear brighter
    - Negative gradient values appear darker

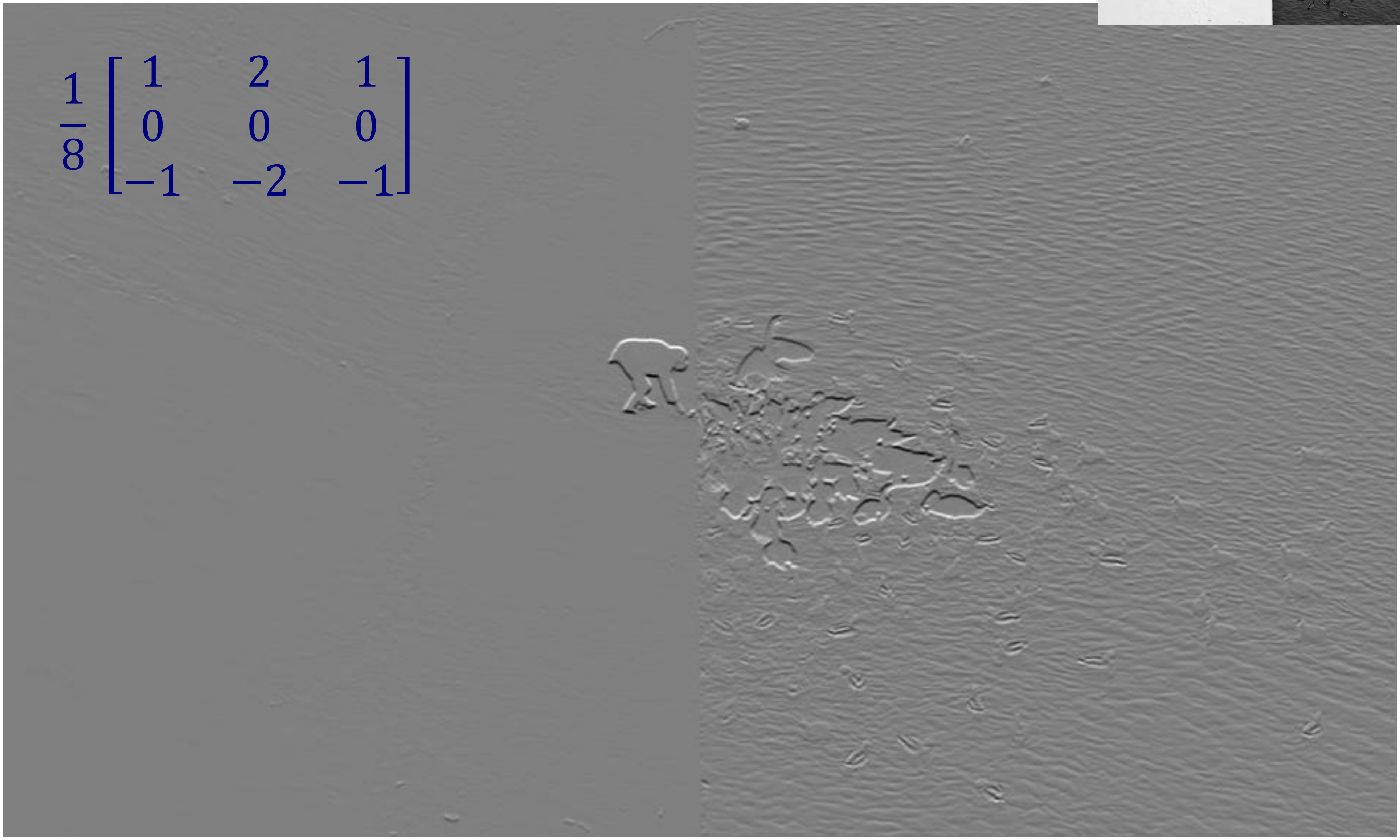
## Gradient in x: vertical edges





$$\frac{1}{8}\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}$$

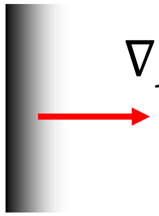
# Gradient in y: horizontal edges



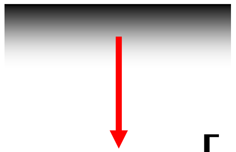
$$\frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

# Image gradient vector

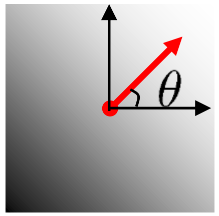
- The gradient vector of image pixels: $\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$



$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ 0 \end{bmatrix}$



$\nabla f = \begin{bmatrix} 0 \\ \frac{\partial f}{\partial y} \end{bmatrix}$

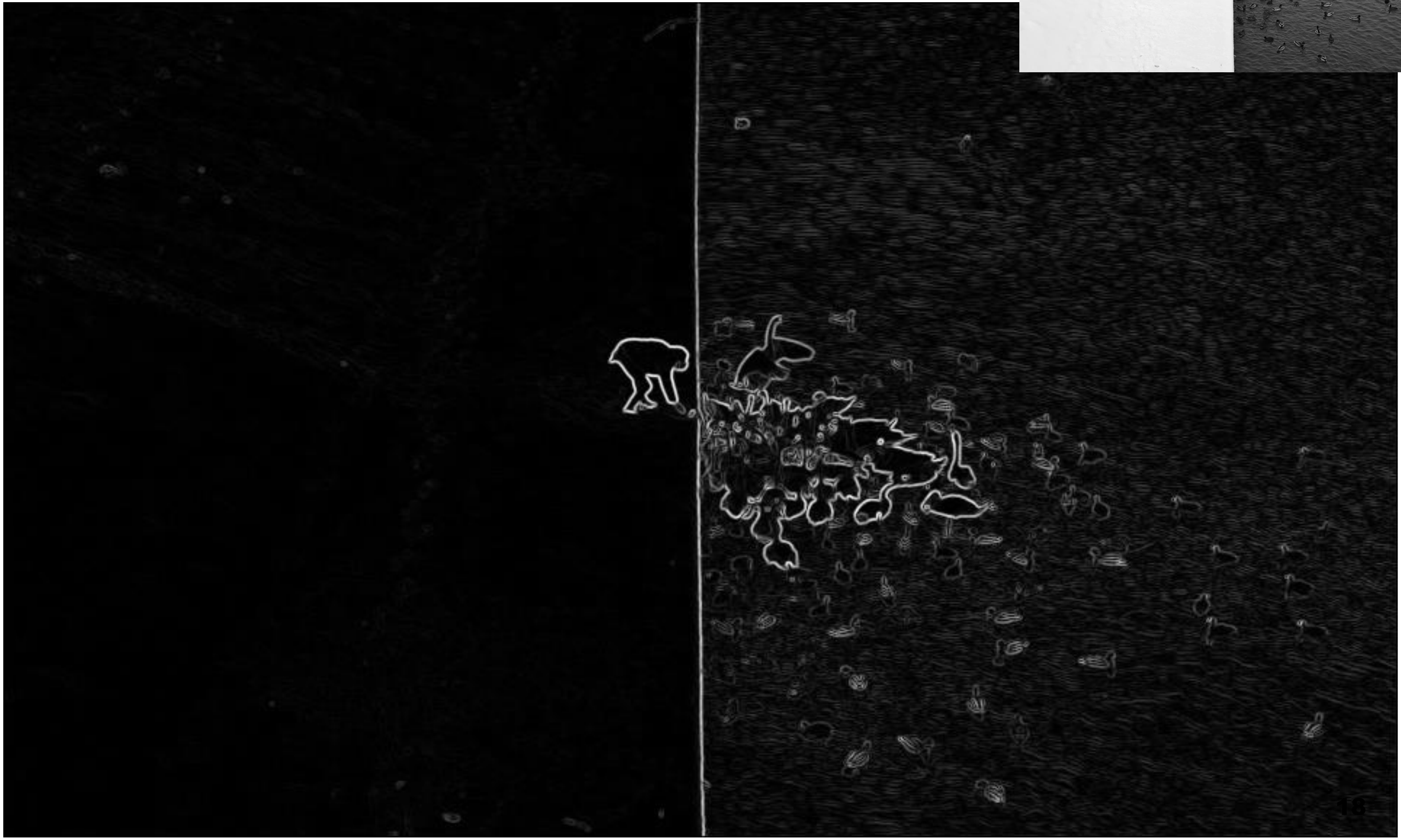


$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

  - The gradient vector points in the direction of most rapid increase in intensity (perpendicular to the edge)
  - The gradient direction is given by $\theta(x,y) = tan^{-1}\left(\frac{g_y(x,y)}{g_x(x,y)}\right)$
  - The edge strength is given by the gradient magnitude

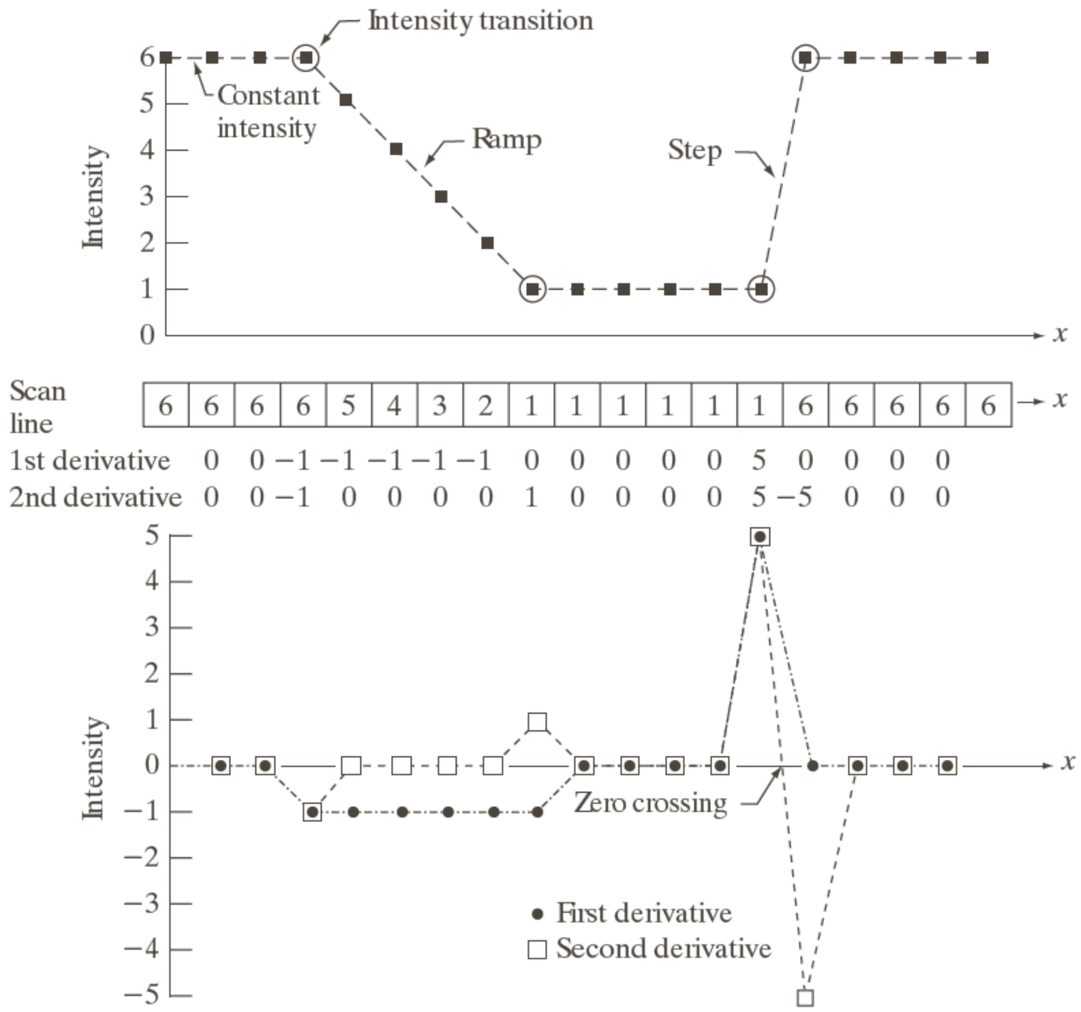
$$g_m(x,y) = \sqrt{g_x^2(x,y) + g_y^2(x,y)} \approx |g_x(x,y)| + |g_y(x,y)|$$

## Gradient magnitude



# More fun with derivatives…

- 2nd derivative is where the gradient changes



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scan line | 6 | 6 | 6 | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 |
| 1st derivative |  | 0 | 0 | −1 | −1 | −1 | −1 | −1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 |  |
| 2nd derivative |  | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 5 | −5 | 0 | 0 | 0 |  |

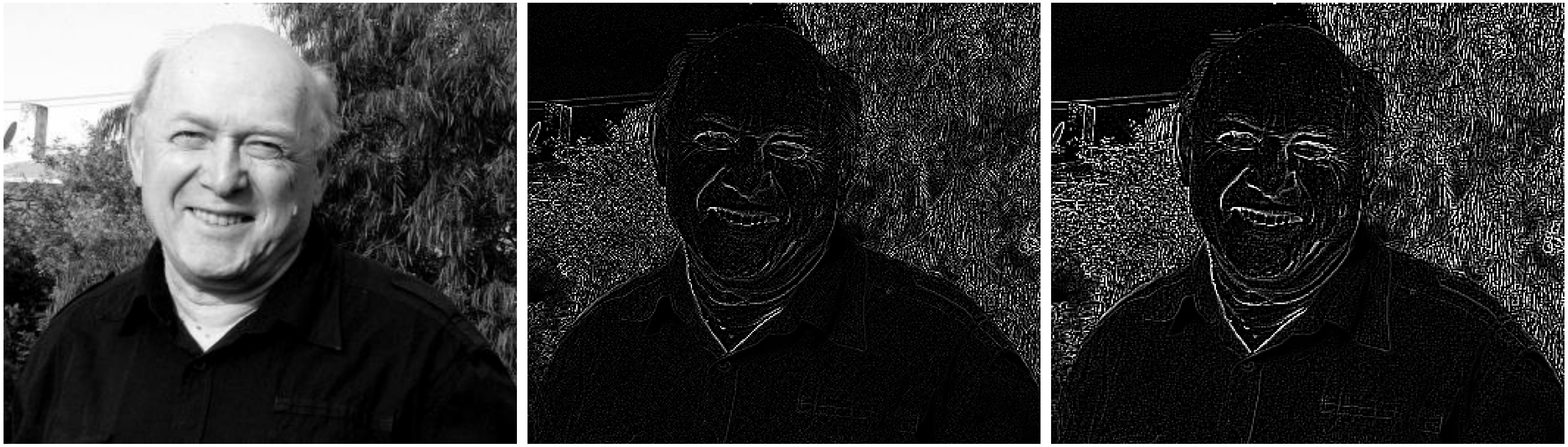
a
b
c

**FIGURE 3.36** Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

## Laplacian filter

- We can approximate 2nd derivative using either of these Laplacian filter kernels
  - Correspond to sum of 2nd derivatives in x and y

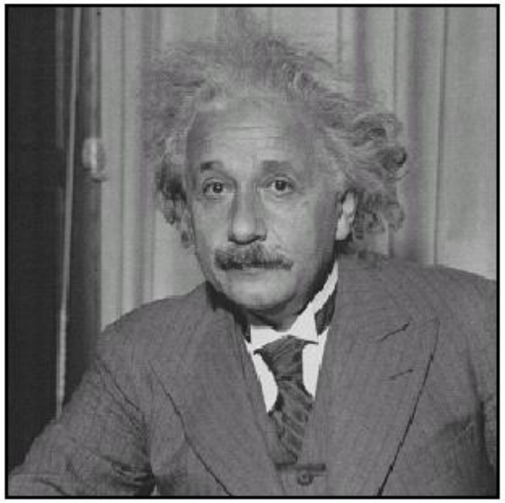
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

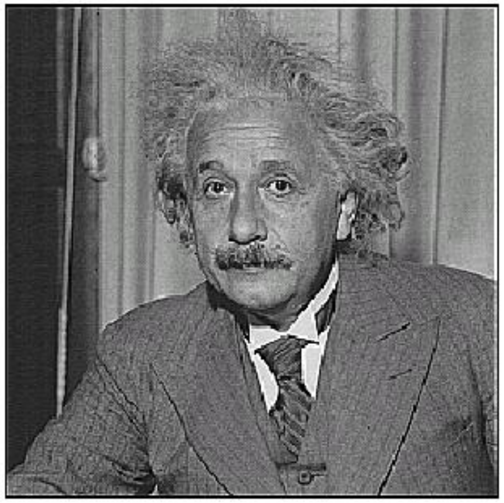


Enhances image details

# Image sharpening

- Add the Laplacian filter result to the original image
- Note that Laplacian has positive and negative values (careful scaling!)
- Drawback: 2nd derivative also enhances noise!


before


after

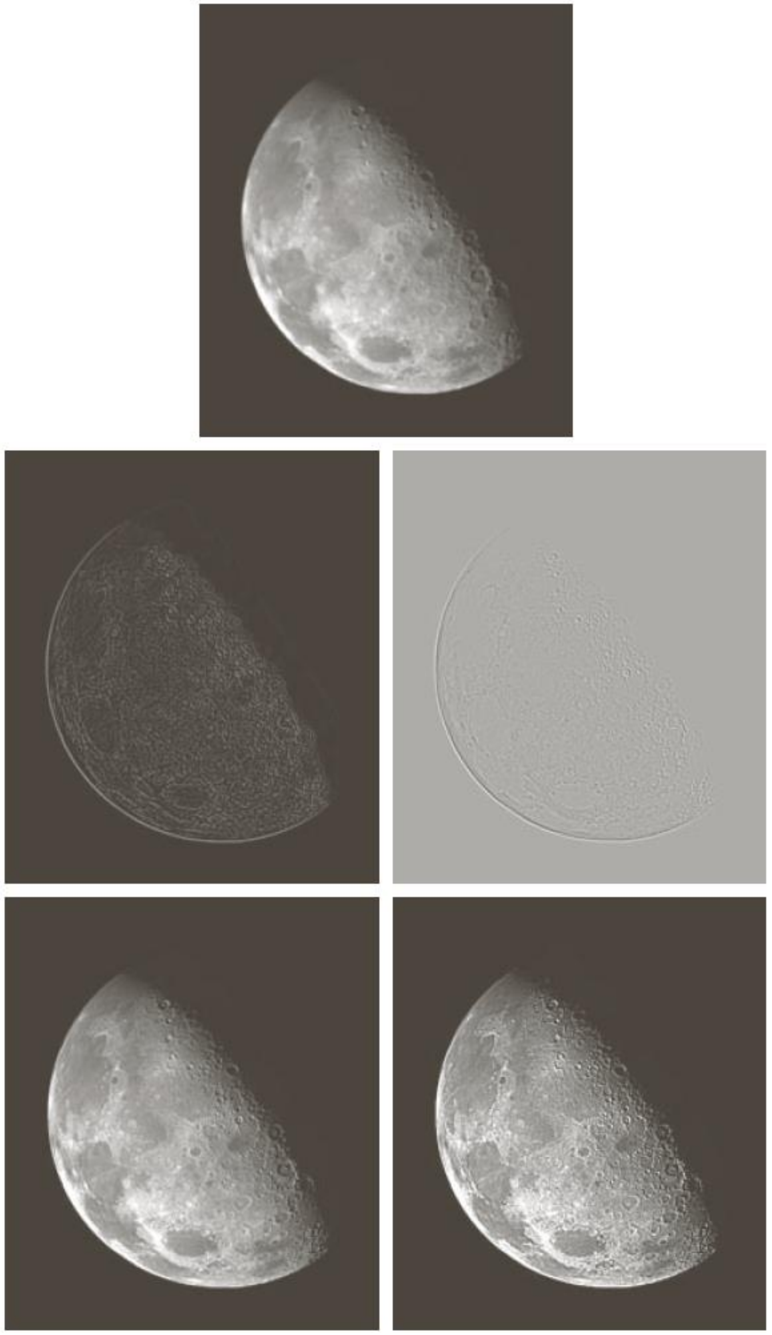
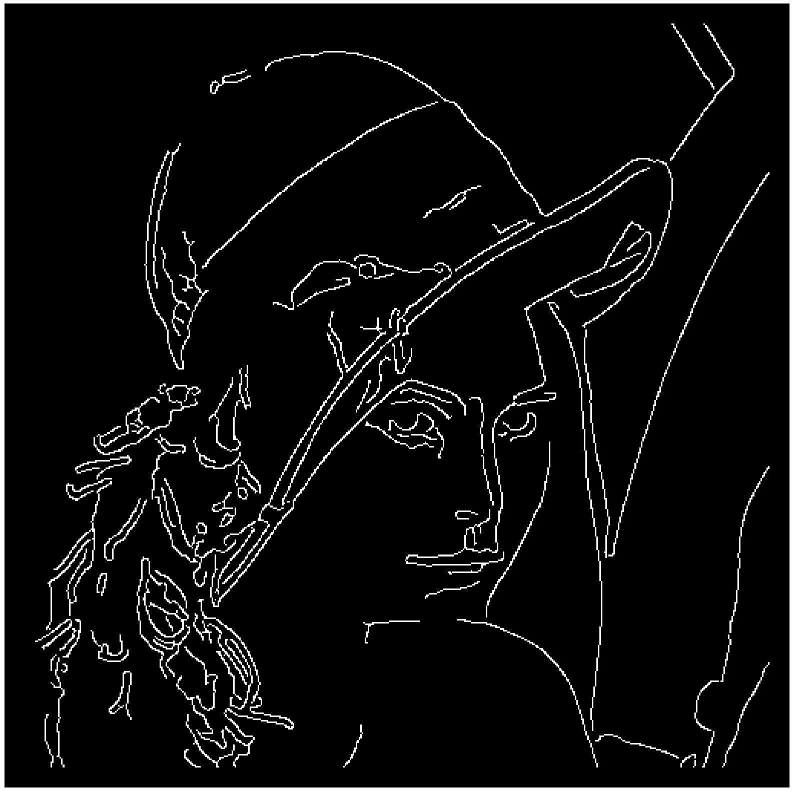


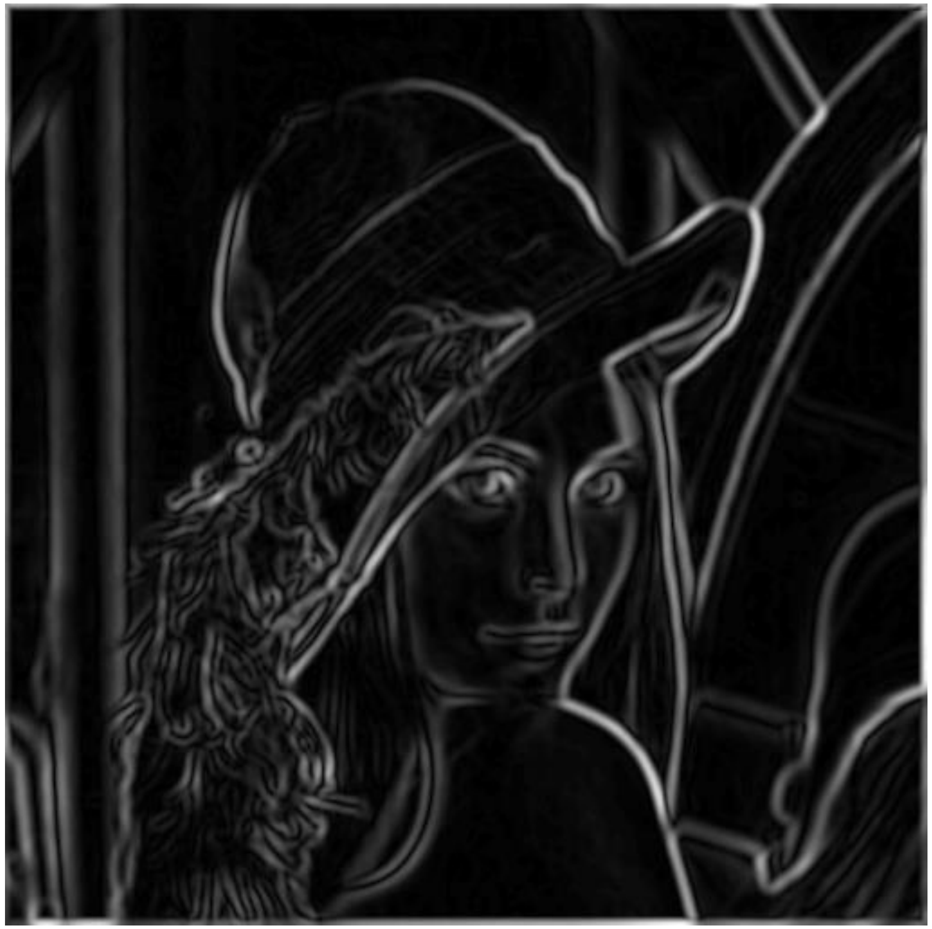
**FIGURE 3.38** (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

# Back to building an edge detector

original image

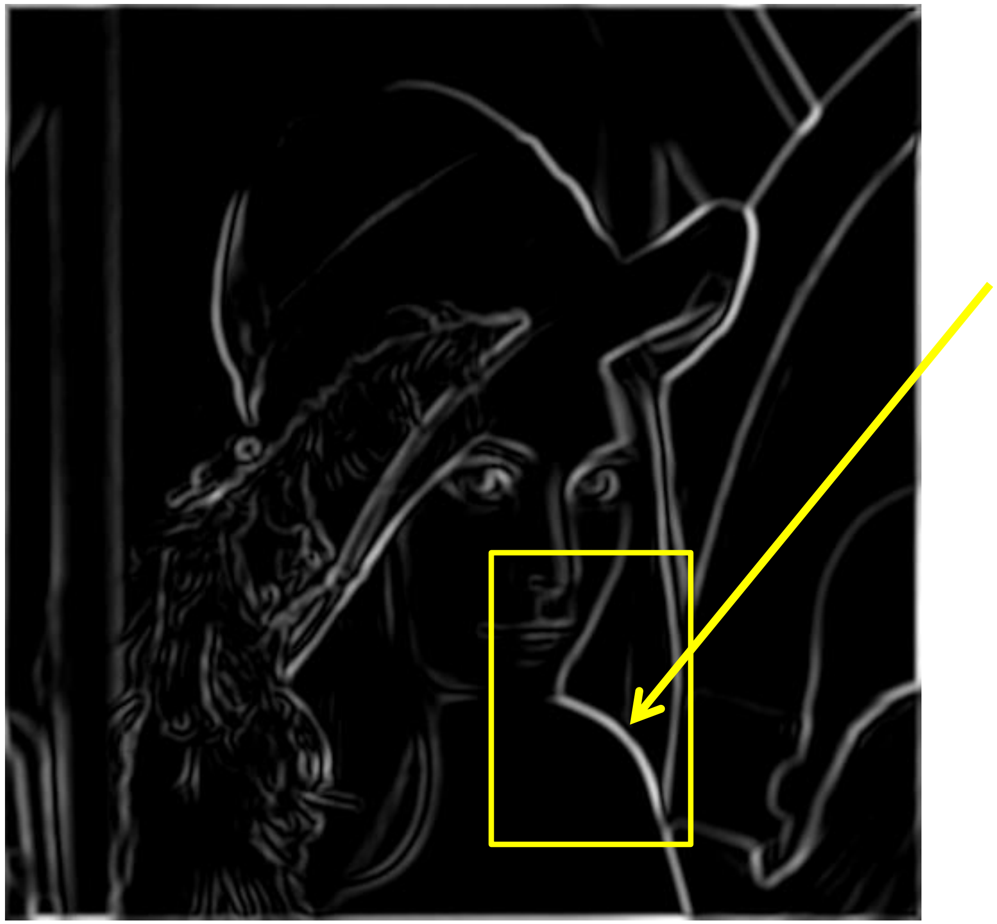

final output

## Building an edge detector



gradient magnitude of the pre-smoothed image

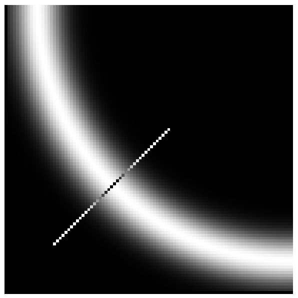
## Building an edge detector

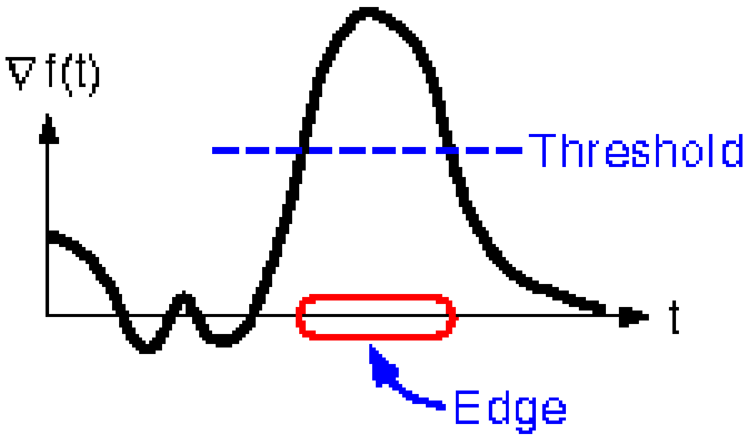


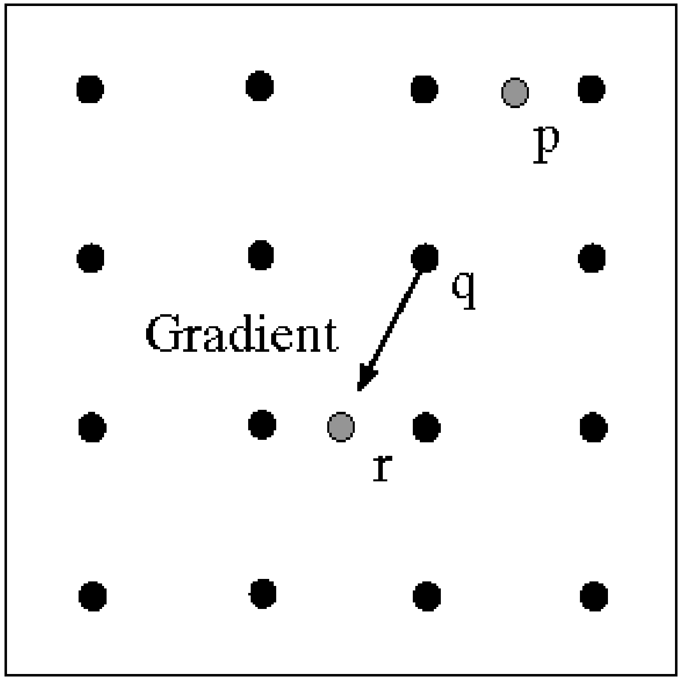
How to turn these thick regions of the gradient into curves?

Thresholded gradient magnitude

# Non-maximum suppression

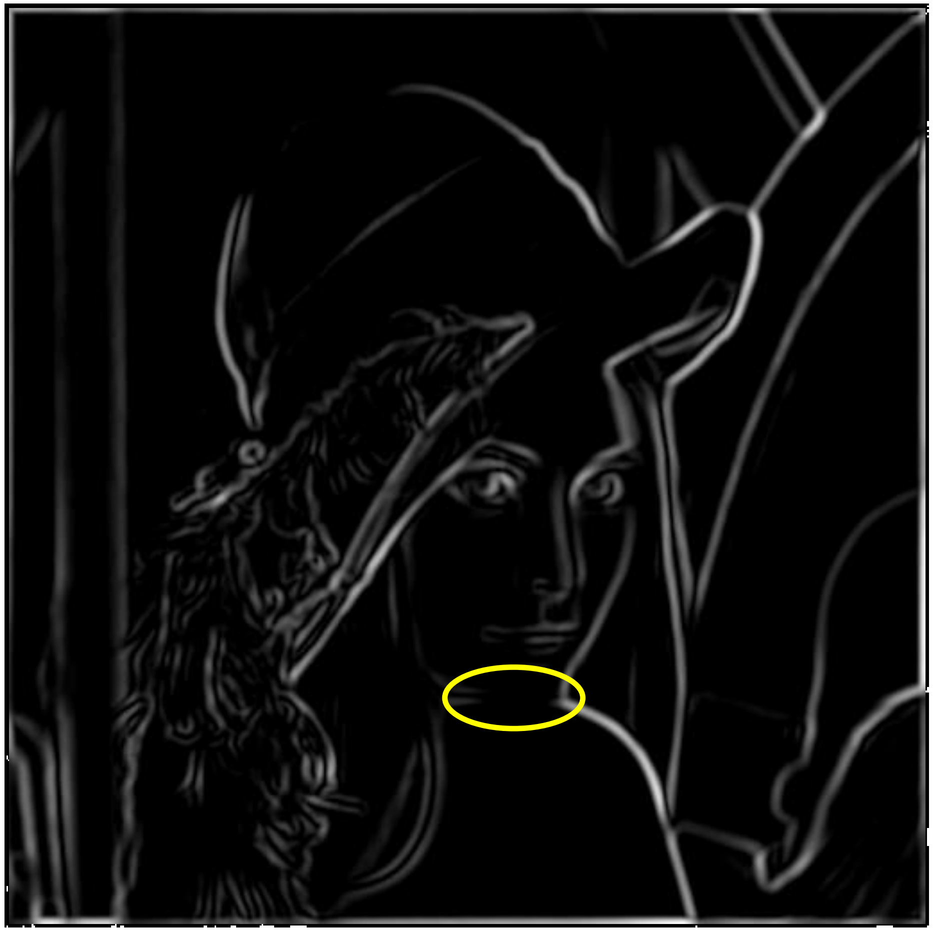






- For each location q above threshold, check that the gradient magnitude is **higher than at neighbors** p and r **along the direction of the gradient**
    - May need to interpolate to get the magnitudes at p and r

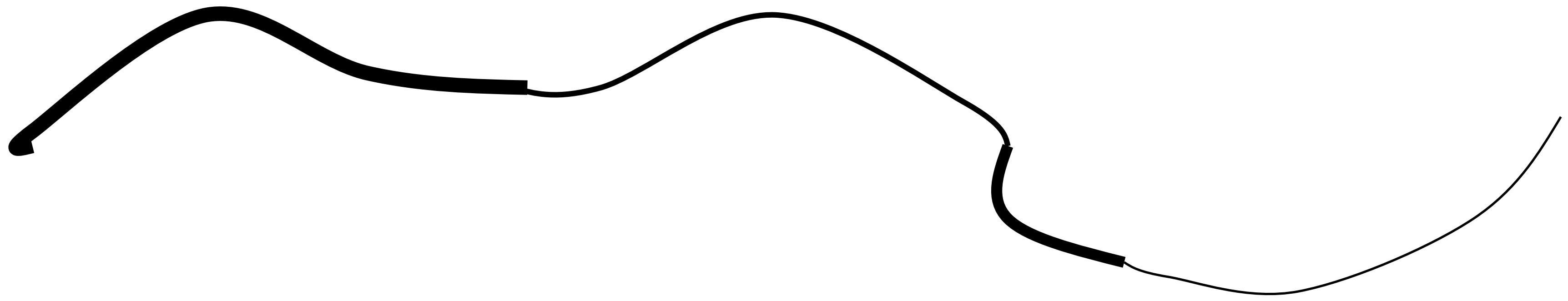
## Non-maximum suppression



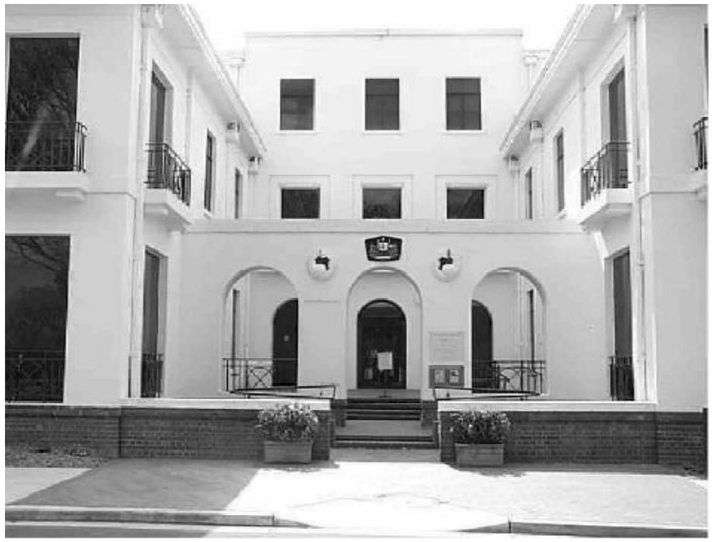
Another problem: pixels along this edge didn’t survive the thresholding

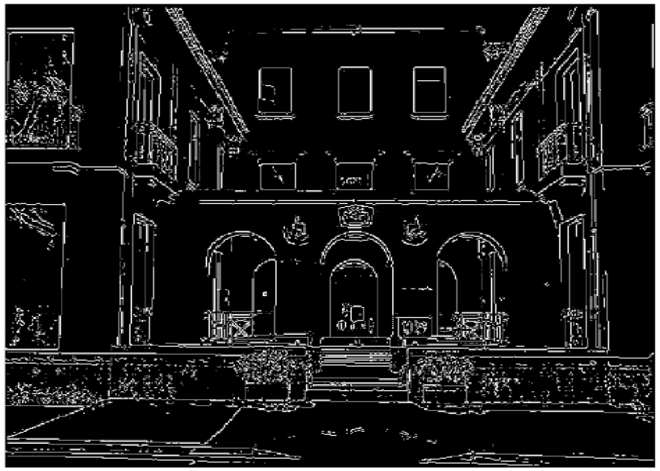
# Hysteresis thresholding

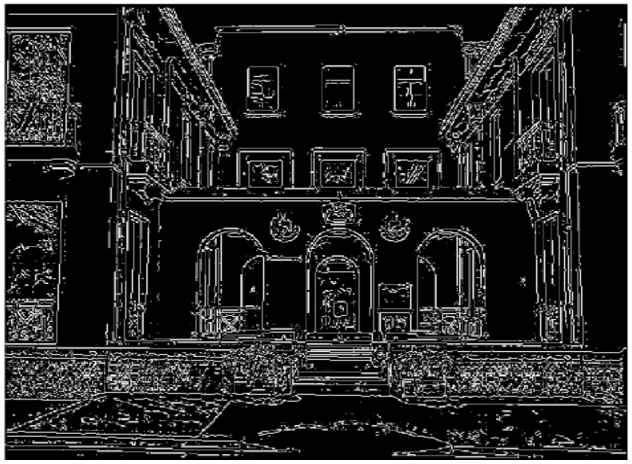
- Use a high threshold to start edge curves, and a low threshold to continue them

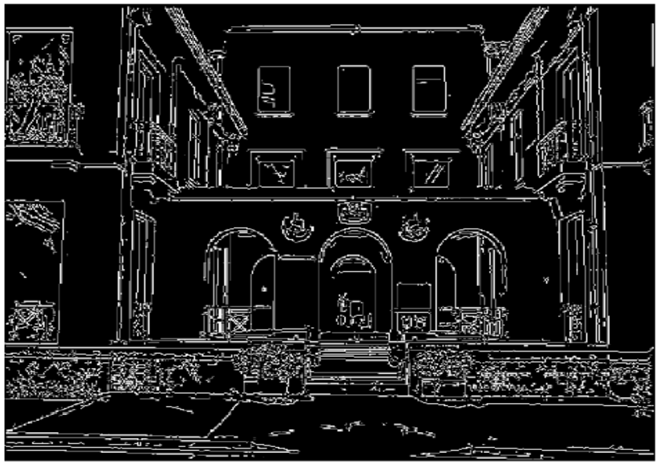


## Hysteresis thresholding


original image


high threshold
(strong edges)


low threshold
(weak edges)

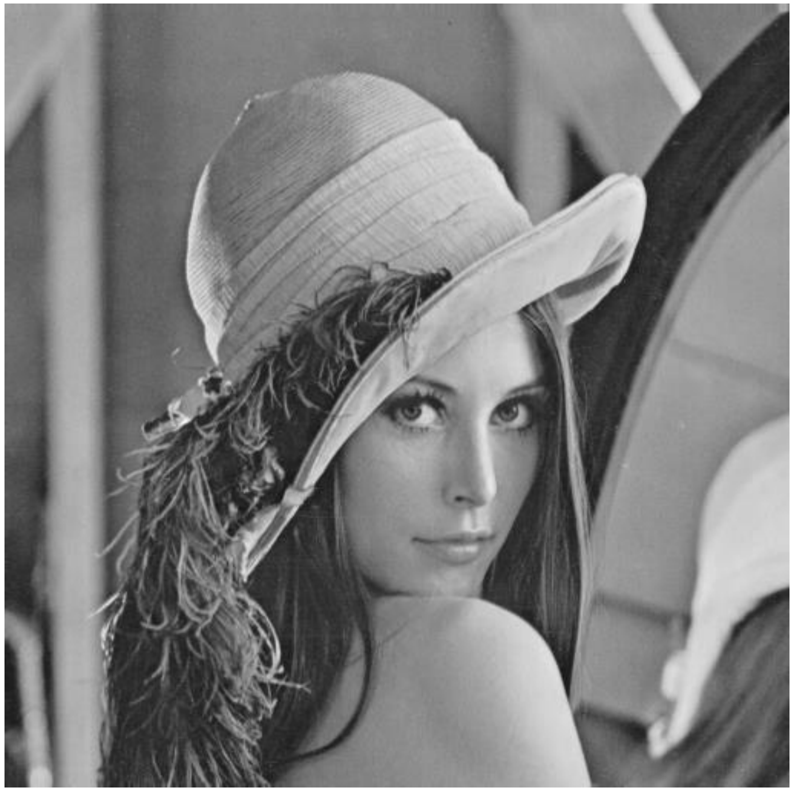

hysteresis threshold

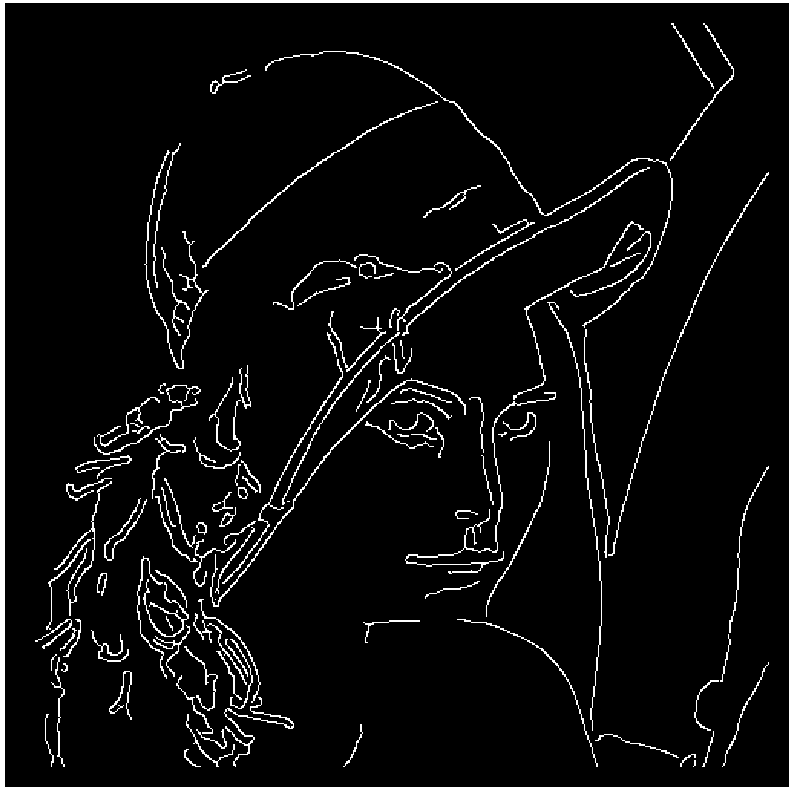
# Canny edge detector

- Compute x and y gradient images
- Find magnitude and orientation of gradient
- Non-maximum suppression:
  - Thin wide “ridges” down to single pixel width
- Linking and thresholding (hysteresis):
  - Define two thresholds on gradient magnitude: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

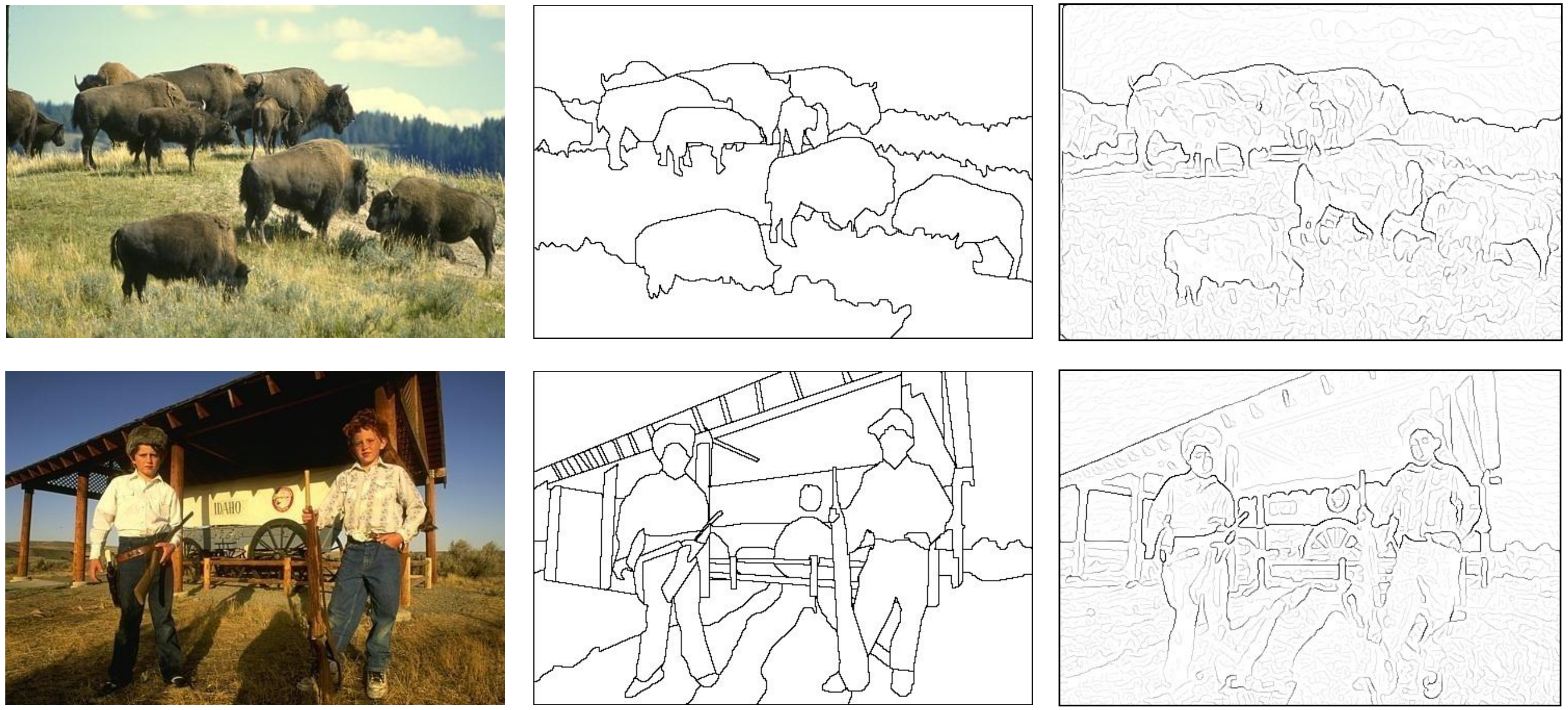
J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

## Canny edge detector result


original image

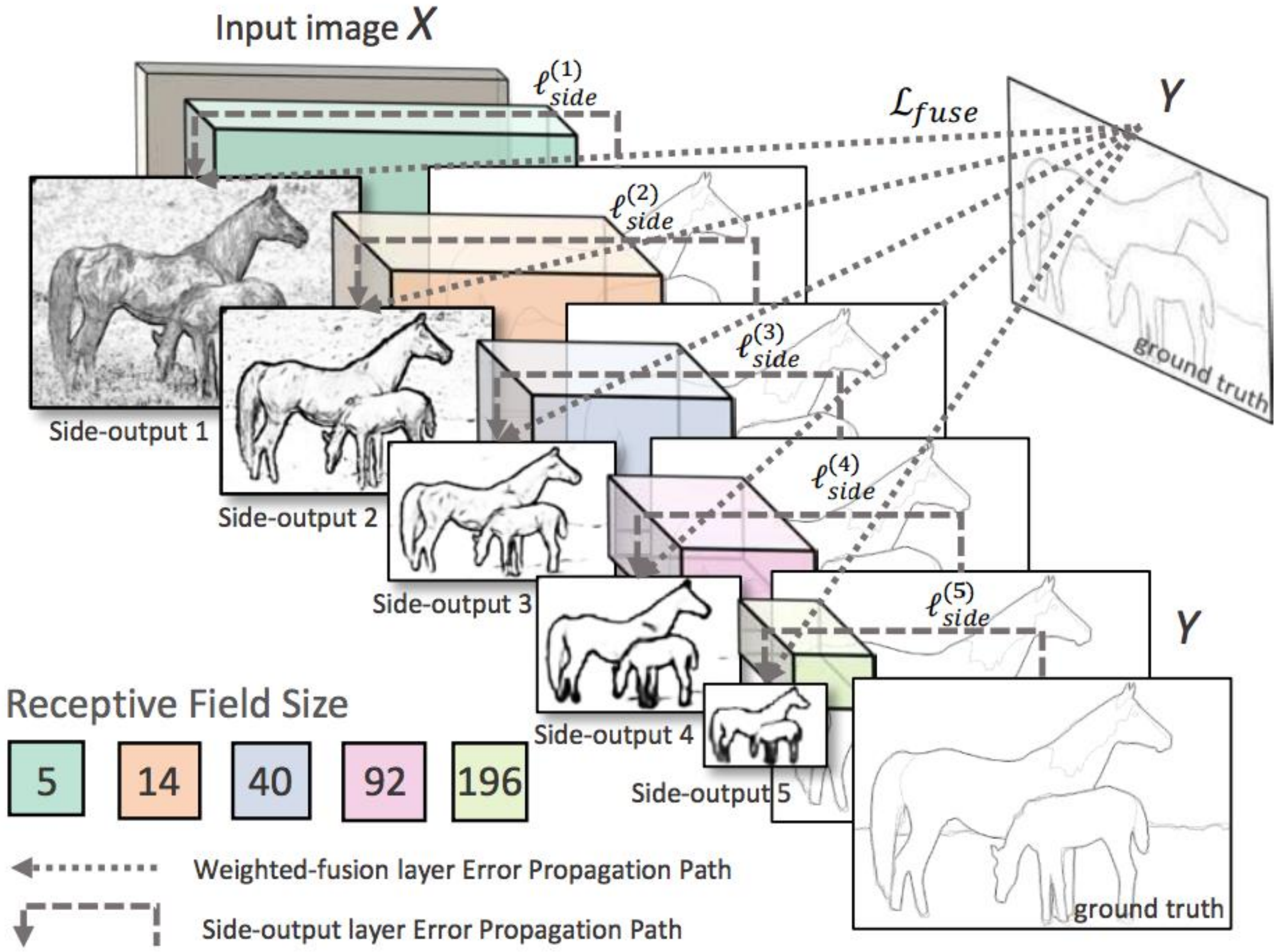

final output

# What are meaningful contours?



- Berkeley segmentation database

# Outlook Data-driven edge detection



S. Xie and Z. Tu, Holistically-nested edge detection, ICCV 2015