Spatial Linear Filtering

- Correlation, as introduced, is a form of spatial linear filtering
- An image of size MxN is filtered with a filter mask of size mxn

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)$$

- w is the kernel filter (e.g. 3x3 box matrix)
- Assume odd matrix size, and 0,0 is the centre
- Therefore the above expression loops over the filter

Image Credit: All images (unless otherwise noted) in these 2 lectures are by Jason Xie (Swansea)

Spatial Linear Filtering

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)f(x+s,y+t)$$

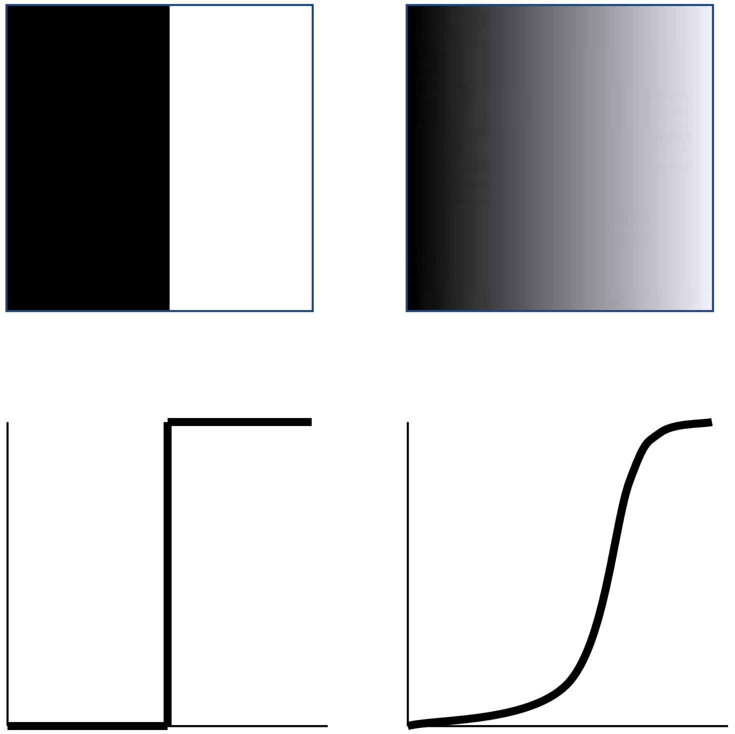
- to describe the equation: e.g. in pseudo code

```
for (s=-a; s<=a; s++) {
        for (t=-b; t<=b; t++) {
                sum=sum+w[s][t]*f[x+s][y+t];
        }
}
```

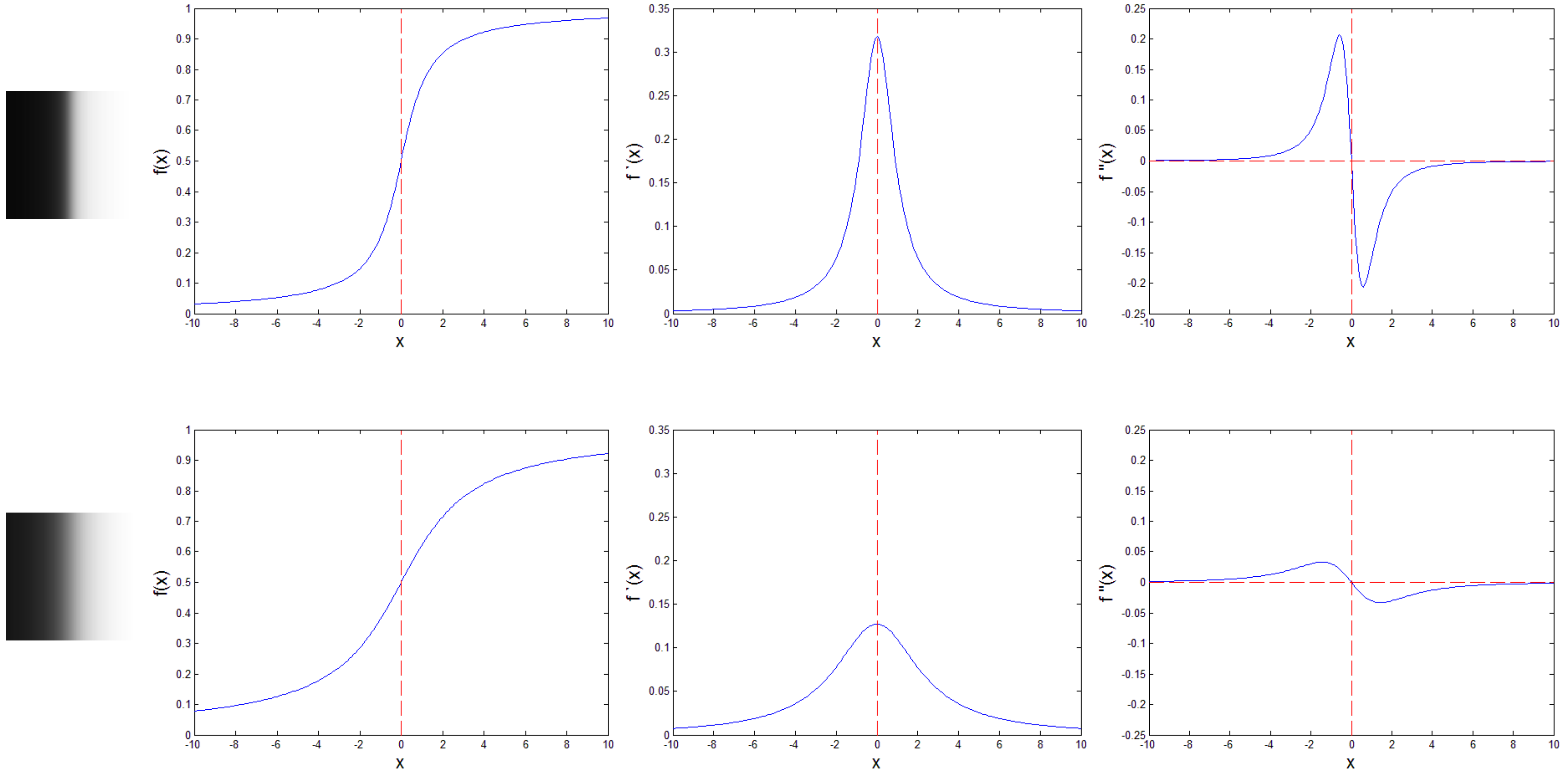
- where w is a 2D filter array, and f is a 2D image (does not take into account colour)

Edge Detection

- What are edges?
- Usually a perceivable change in intensity
- e.g. see the abrupt change on the left (graphed bottom left)
- Compared to the change on the right (graphed bottom right)
- Can we create edge detectors?
- Edges can be spotted where the "rate of change" of intensity is high.
- Recall "rate of change" is the derivative of a function



Edge Detection: Derivatives



Edge profile

1st order derivative

2nd order derivative

Edge Detection: Derivatives

- In the previous images, the vertical dotted line is the perceived edge
- This is difficult to compute on the edge profile
- Its where the rate of change of intensity is highest
- We can see this in the 1st derivative where the function has a maximum value
- This is harder to spot for smoother edges
- The maximum of a function can be found where its derivative equals zero
- Therefore see the 2nd derivative, where the edge is obvious as the function crosses from +ve to -ve

Edge Detection: Derivatives

- Edges can be detected by considering the first derivative (2nd column of graphs)
- Notation, given a function f its derivative is denoted f’
- Defining, using limits,

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

- that is, as h tends toward zero (limit definition), the right hand side defines the derivative.

Edge Detection: Derivatives

- The derivative can be approximated by evaluating the function with a small h
- But, if h does not tend to zero, the derivative will have some error (that is why its approximating)
- One discrete approximation is central differences:

$$f'(x) \approx f(x+\frac{1}{2}h) - f(x-\frac{1}{2}h)$$

- where h is small

Edge Detection: Derivatives

$$f'(x) \approx f(x+\frac{1}{2}h) - f(x-\frac{1}{2}h)$$

- If we make h twice the distance between pixels
- The above equation simple states that the derivative of the image gradient at a pixel, is the next (right) pixel’s value minus the previous (left) pixel’s value
- The *Prewitt* operator takes this idea and creates a correlation filter which calculates the discrete derivative

Edge Detection: Finite Differences

- Prewitt operator

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

- This is sensitive to noise, so it was combined with a Gaussian smoothing filter to give the:
- Sobel operator

$$M_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Edge Detection: Finite Differences

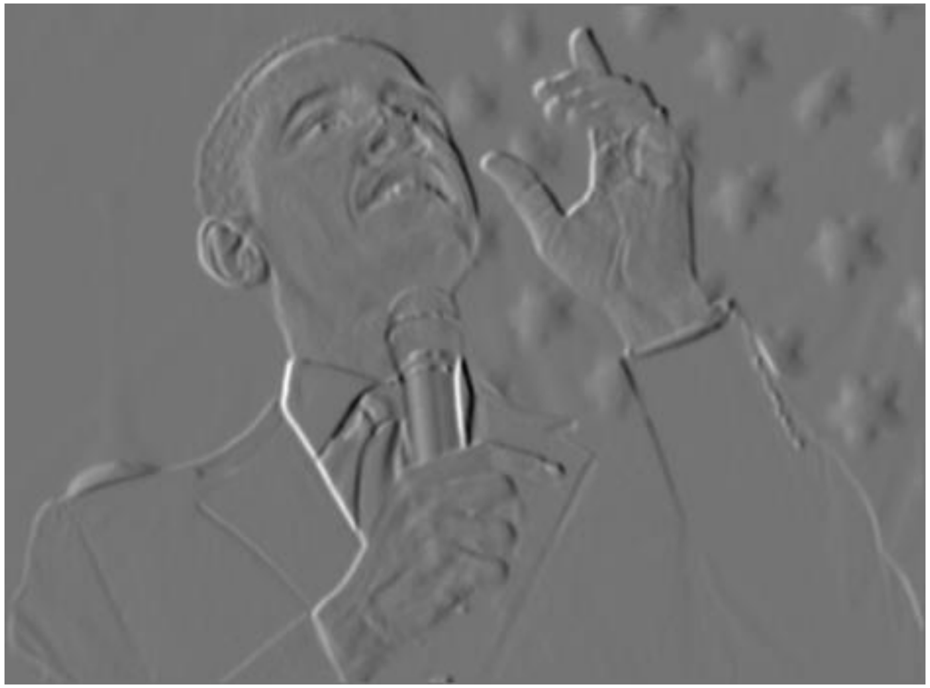
- The Sobel operator uses the first filter to give image Ix, and the second to give Iy
- Then the magnitude of the gradient is calculated along with its direction, thus:

$$\text{magnitude} = \sqrt{I_x^{\;2} + I_y^{\;2}} \qquad \text{gradient direction} = \arctan\left(\frac{I_y}{I_x}\right)$$

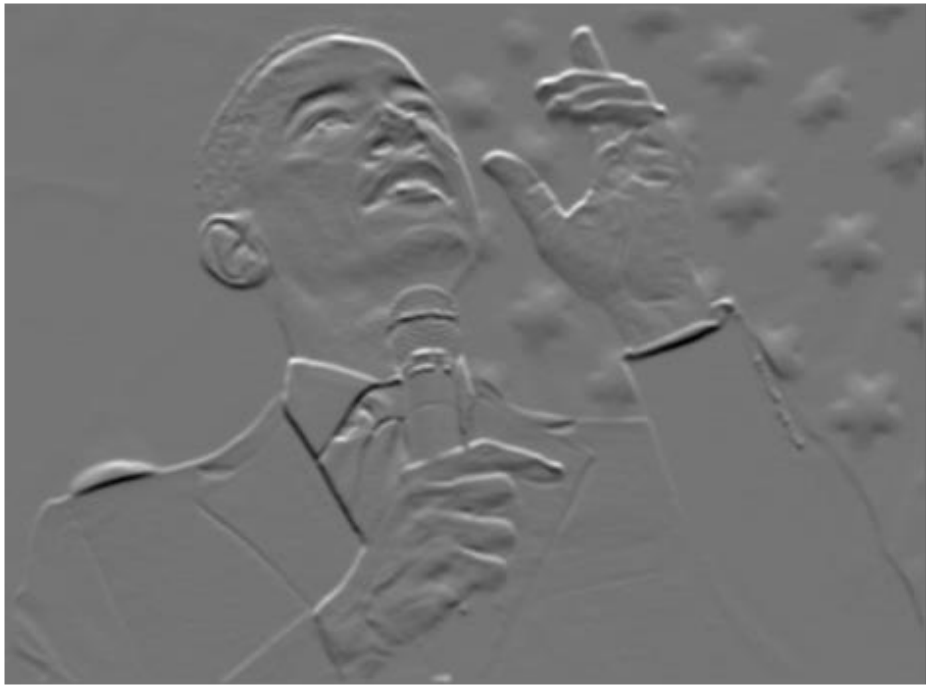
- think of any operation on image Ix as applying the operation to every pixel in Ix
- e.g. the magnitude of pixel (i,j) is

$$magnitude(i, j) = \sqrt{I_x(i, j)^2 + I_y(i, j)^2}$$

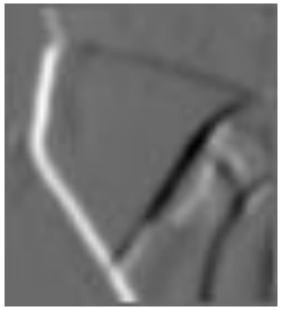
Edge Detection: Sobel

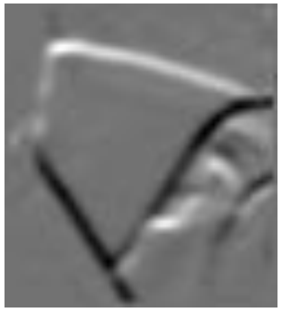


I_x



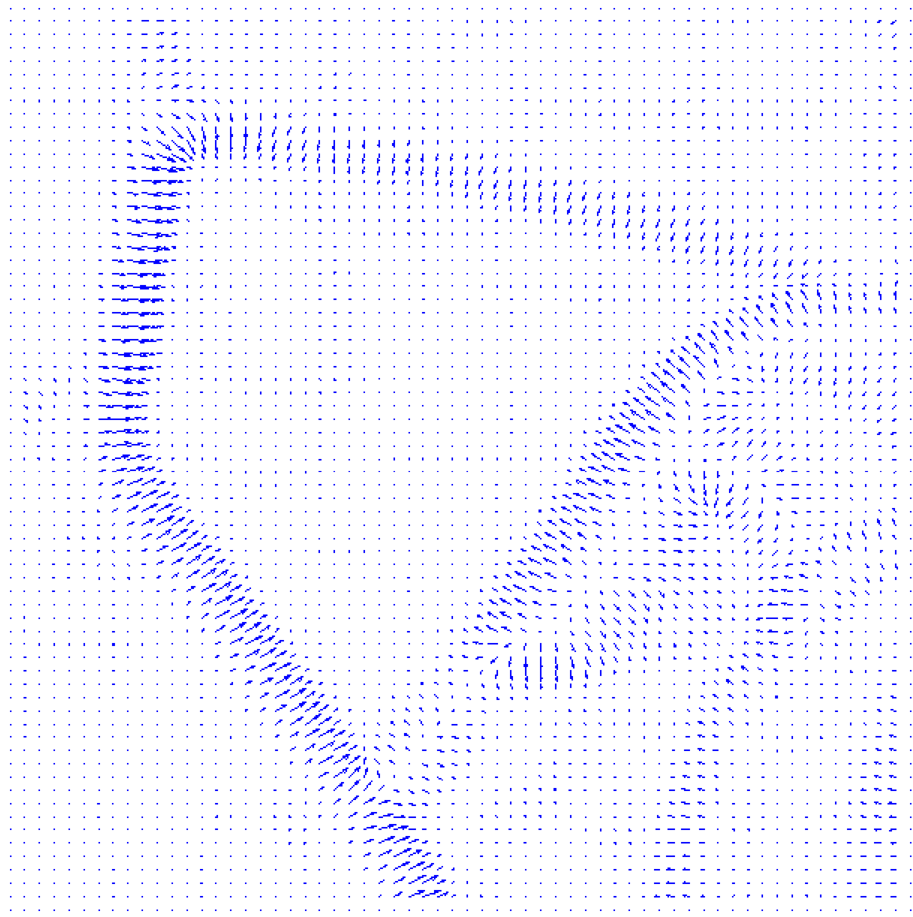
I_y





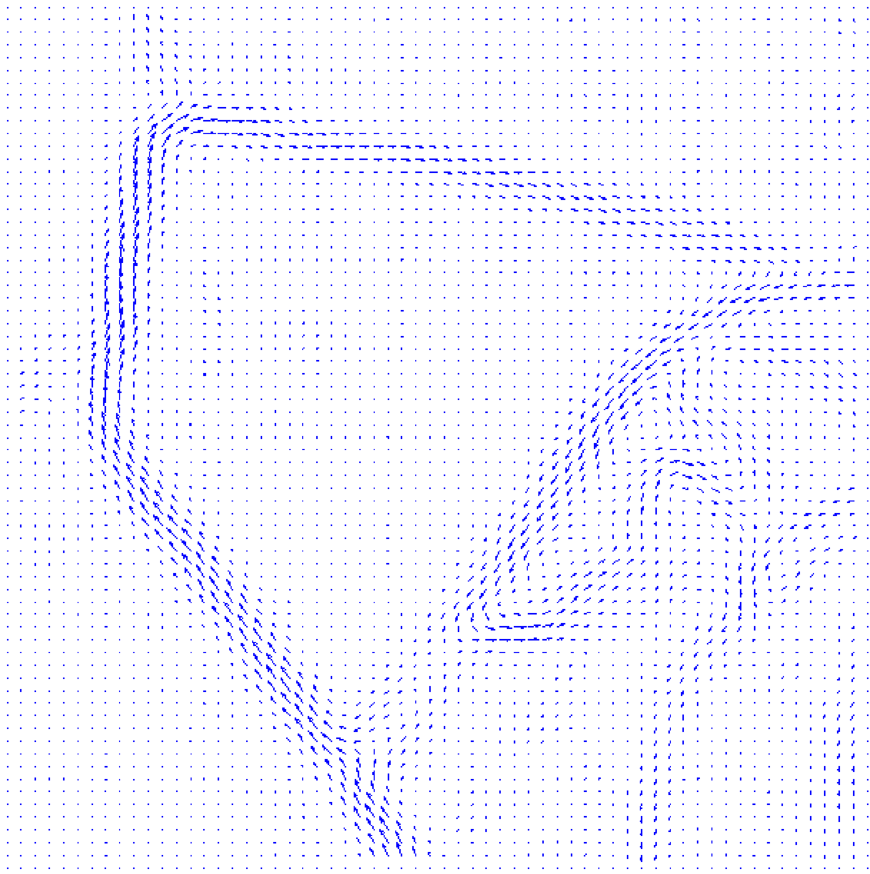
Edge Detection: Sobel – Gradient and Magnitude

- Each arrow direction is gradient direction, and the length is the magnitude
- Note how the unvarying regions have small arrows, and the edges have long arrows pointing across the edge



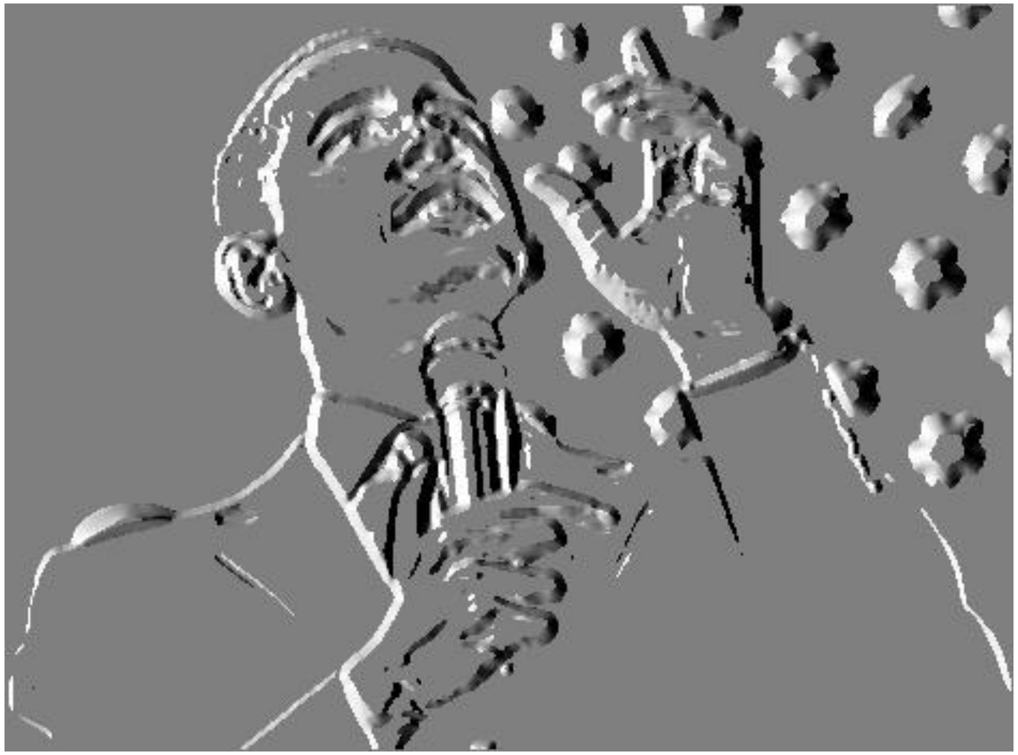
Edge Detection: Sobel
Gradient Normal and Magnitude

- Each arrow direction is at right angles to the gradient direction, and the length is the magnitude
- Note how the unvarying regions have small arrows, and the edges now have long arrows pointing in the direction of the edge



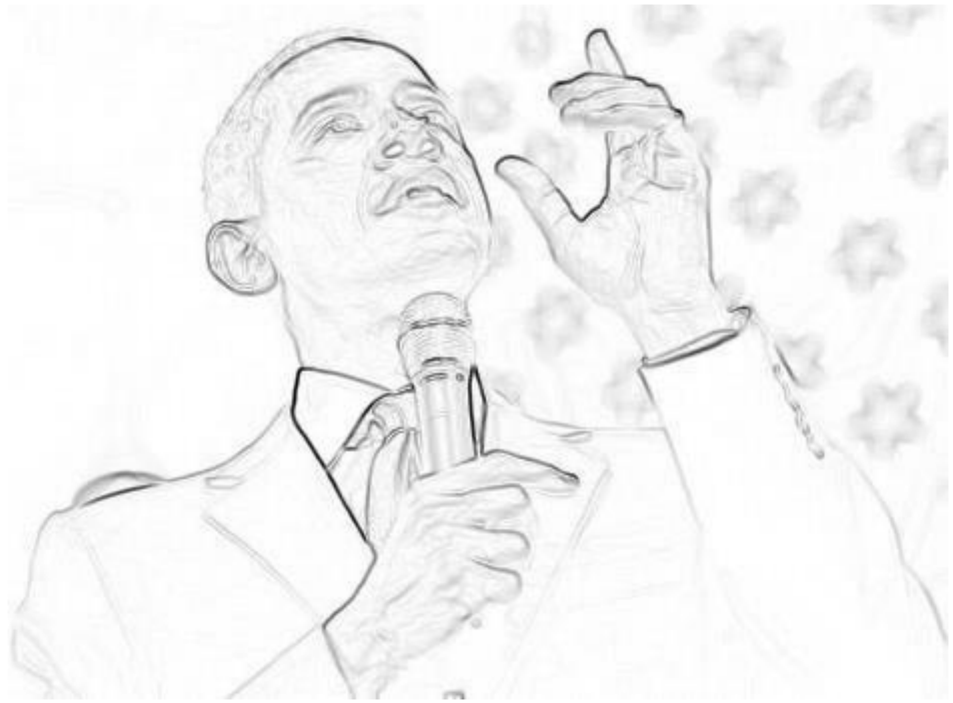
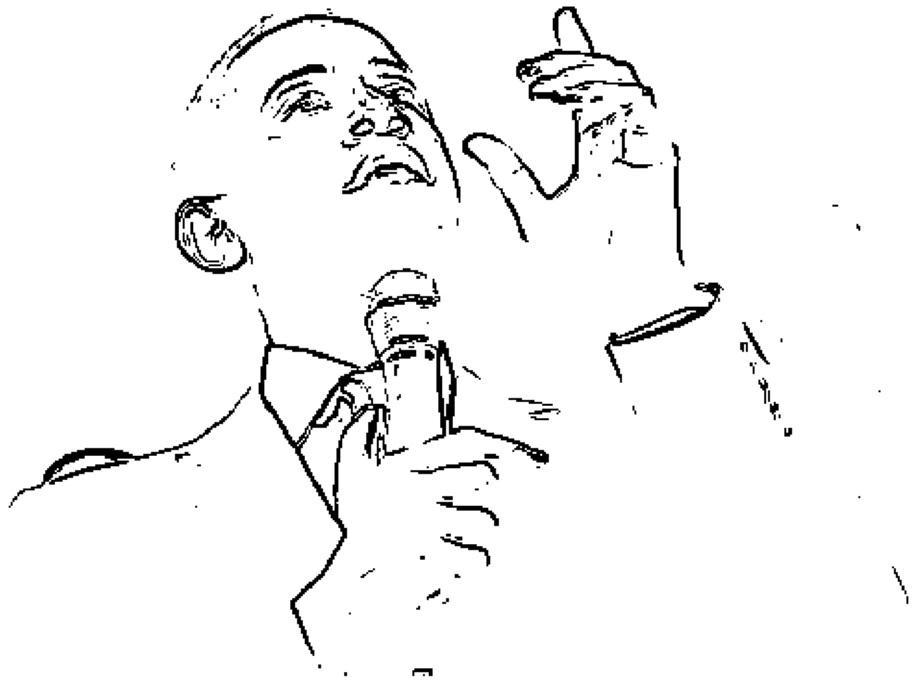
Sobel: Finding edges

- Taking the previous information
- If the gradient is greater than some value, map the angle onto a grey value (e.g. 0 deg (up)=white, 180 deg (down)=black

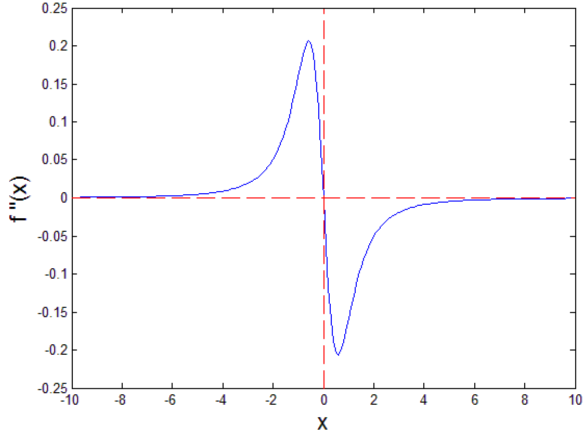


Sobel: Finding edges

- Or, high magnitude=black, low=white (left image), then all pixel values above a certain magnitude are set to black (thresholding – right image)
- Note, not all edges are present – poor connectivity

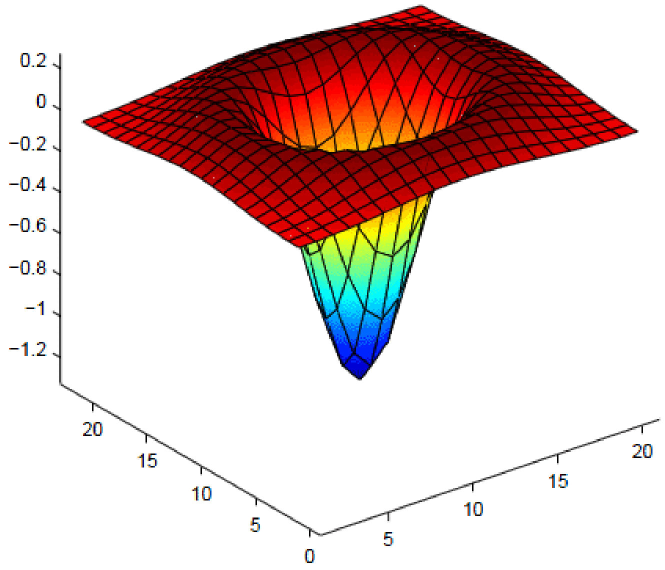
2nd order operator



- Sobel was the first derivative of a Gaussian (blur)
- The 2nd derivative can be used, and we then look for any pixel that is above zero that has a neighbour below zero – this is the crossing, and therefore the edge
- The following 5x5 (high pass) filter can be used

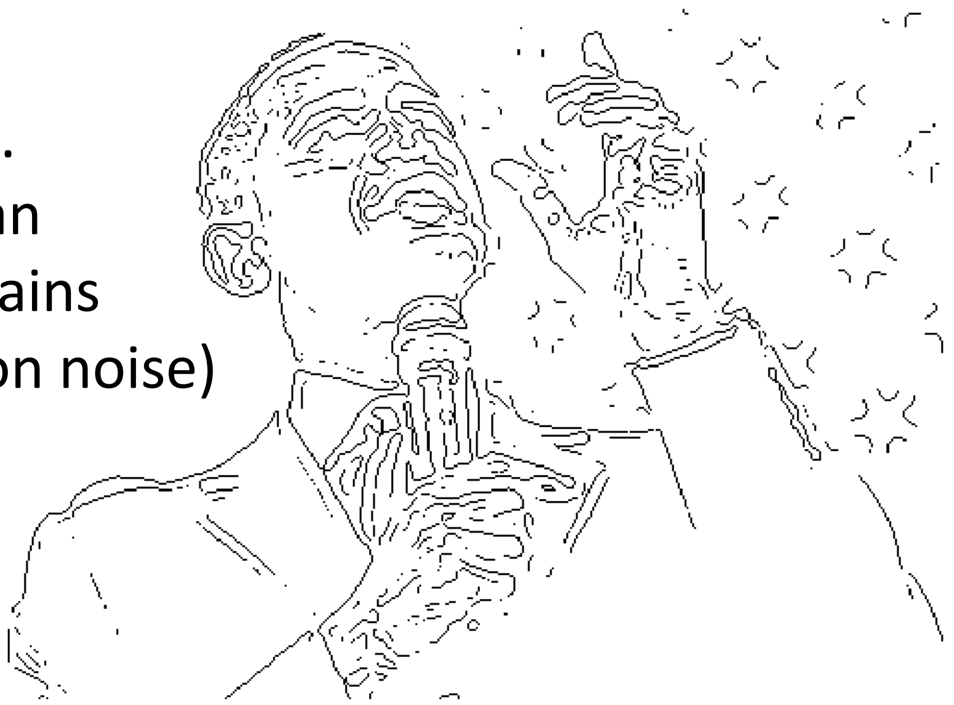
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

- This is the 2nd derivative of the Gaussian and when plotted looks like:



2nd order operator

- Edges are found by looking for zero crossings – pixels above zero next to pixels below zero
- It produces good connectivity at low processing cost
- Finding edges using zero crossing is more accurate than the 1st order operators
- It is sensitive to noise (i.e. produces more edges than needed if the image contains noise – see next lecture on noise)

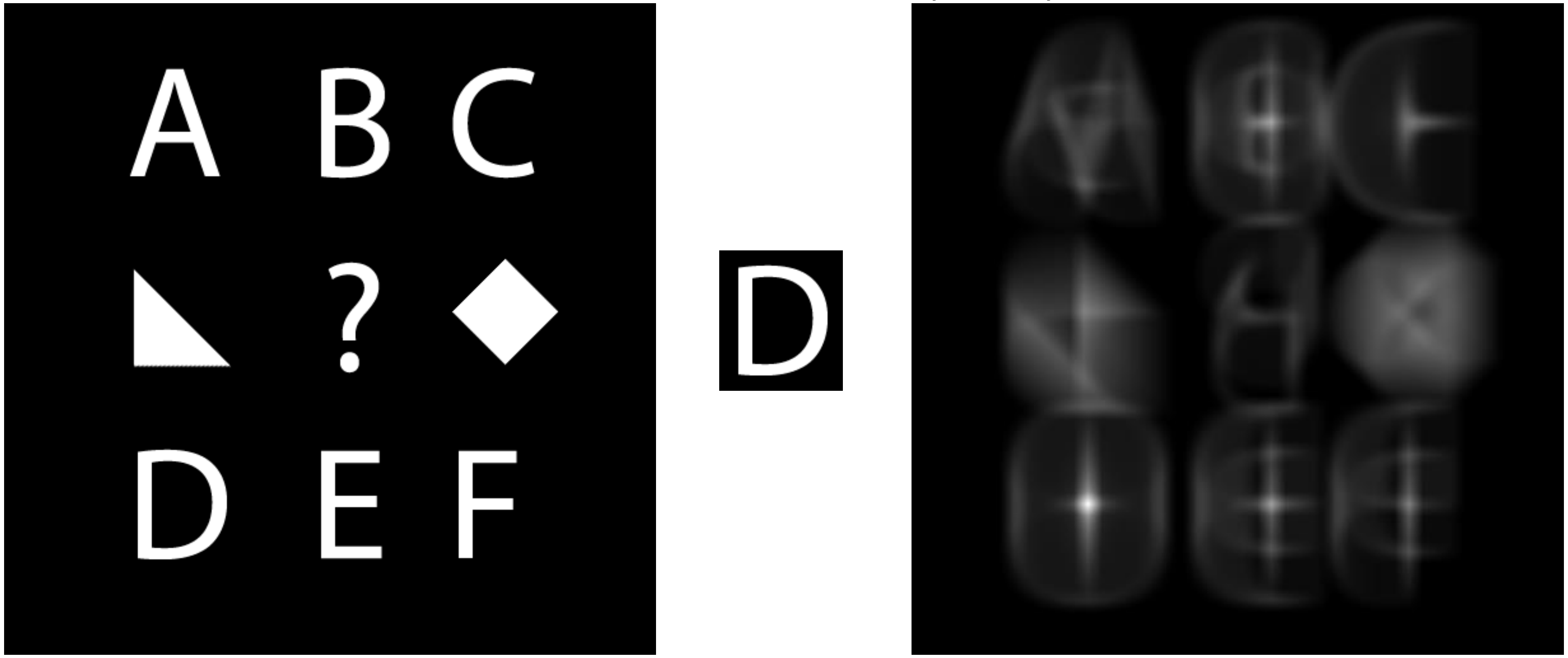


Summary

- Linear spatial filtering described (equation)
- Edges as high changes of intensity
- Rate of change of intensity (1st derivative) – edge spotted as maximum
- 2nd derivative shows edge at zero crossings
- Use correlation (Prewitt / Sobel) to find 1st derivative
- Use correlation (2nd order operator) to find 2nd derivative (then look for zero crossings)

Template Matching

- Detection and recognition by matching patterns
- Simple example
  - Filtering based upon correlation of template with image
  - Brightest spot indicates best match
- But what if its rotated? Different font, size, ...?

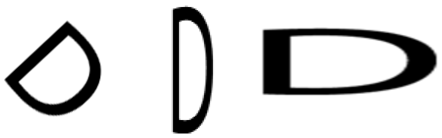


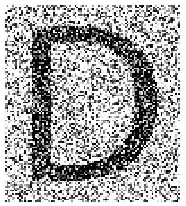
Template Matching

- Do we use templates for all possible differences?

Scale changes

Intensity variations


Transformations

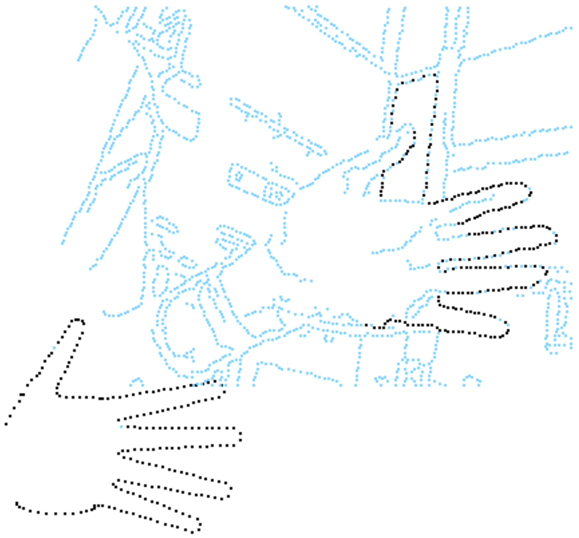

Noise

Partial occlusion

Computer Vision

- This introduces the topic of computer vision
- Real world problems are hard
- Identifying objects/people in images and video is a big research problem

[1]



[2]

[1] Schneiderman and Kanade, CVPR 2000

[2] Stenger et al. ICCV 2003