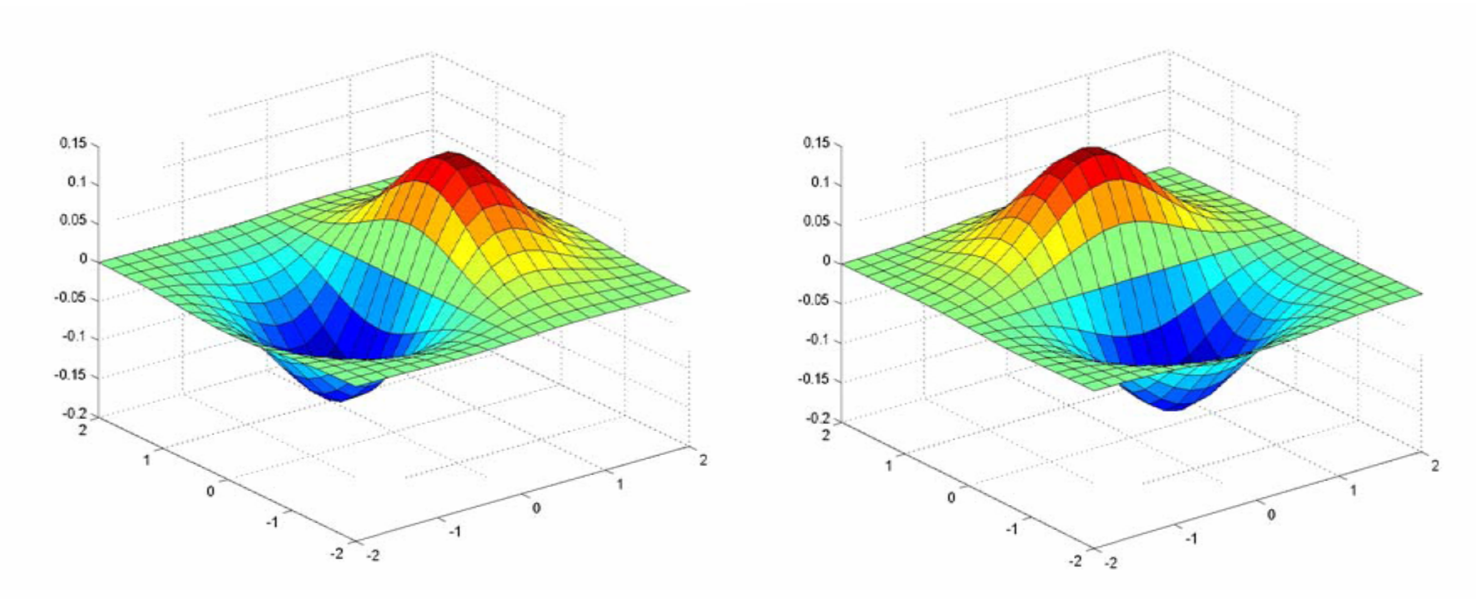
CSCI 497P/597P: Computer Vision



Lecture 4: Edges, Gradients, Frequency Content

Goals

- Understand the limitations of linear filtering
- Know how to compute image derivatives using convolution filters
- Understand how the Sobel filter works to detect edges.
- Have an intuitive understanding of what constitutes high frequency and low frequency image content.
- Know how to make images smaller:
  - The naive way via subsampling (and why this is bad)
  - The better way by prefiltering (and why this is better)

next time!

Stepping back:

- Filtering:
  output pixel depends on input neighborhood

- Linear filtering:
  output pixel is a weighted average of input neighborhood
  (must always the same weights to be linear)

- Cross-correlation is a kind of linear filtering:
  output pixel = weighted average(neighborhood)
  must have same weights for all pixels

- Convolution: cross-correlation, but first flip the kernel horizontally and vertically

Linear Filtering: Questions

- What happens at the edges?

  padding modes   output sizes

- What properties does this operator have?

  shift invariance   associativity (parens)

  linearity   commutativity (order)

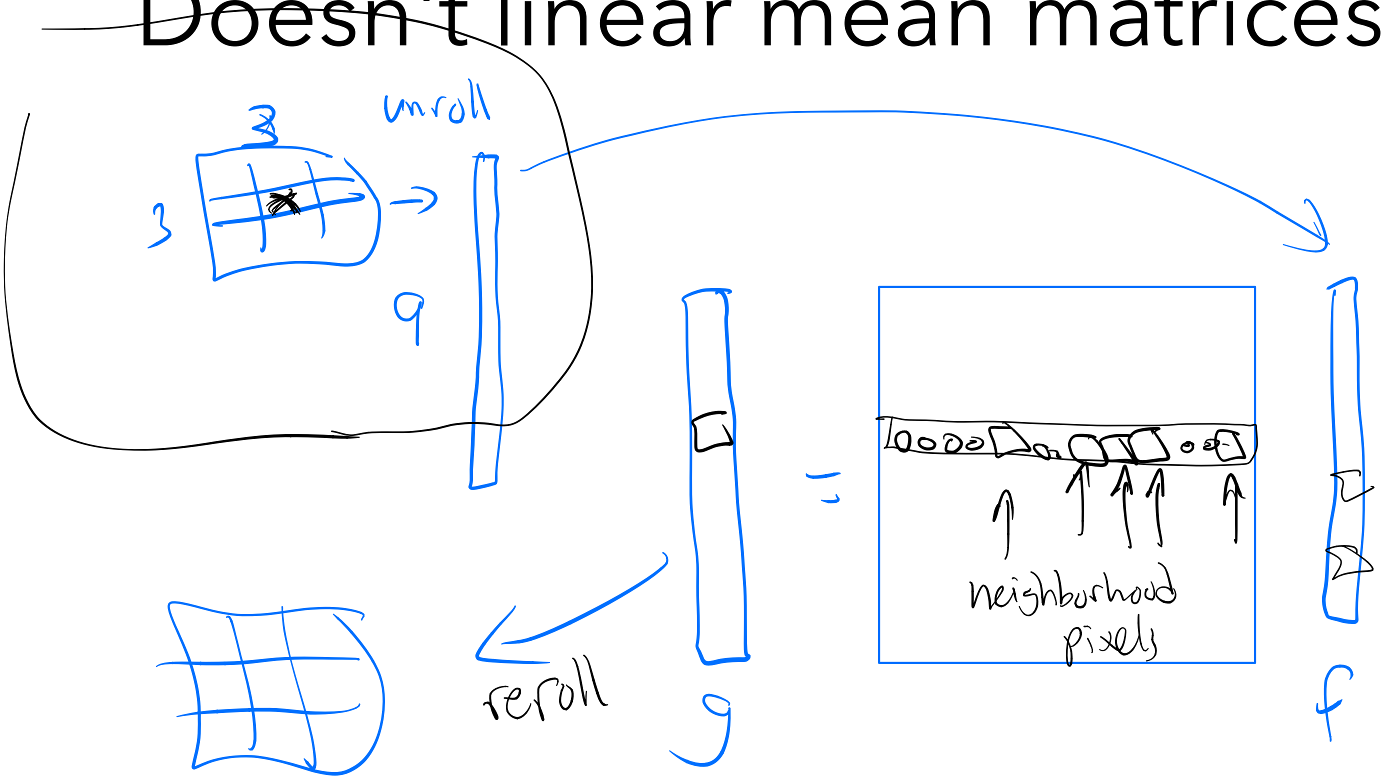
- What can and can't this operator do?

  can: blur   shift   can't: ?

  sharpen

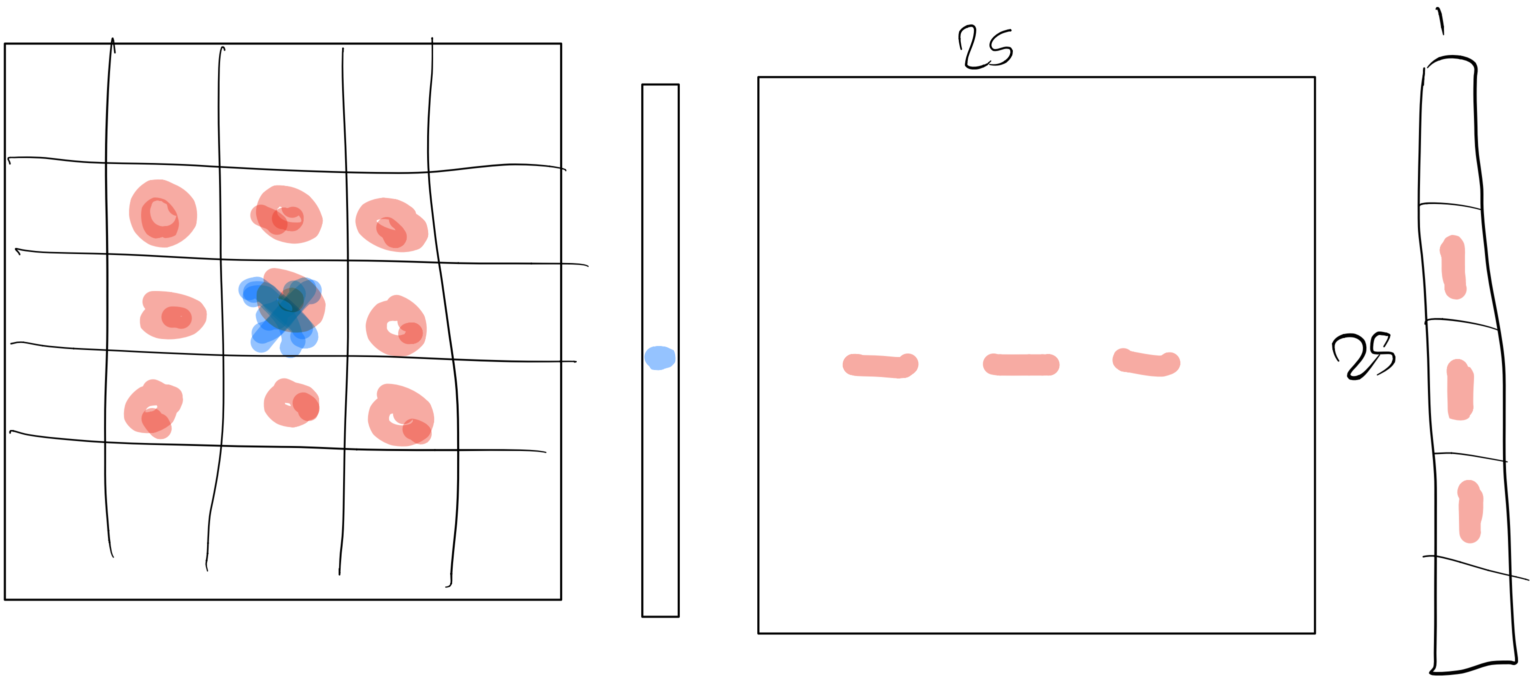
Aside:

Doesn't linear mean matrices?



Aside:

Doesn't linear mean matrices?



output pixel is the result of a dot product (i.e., weighted average)

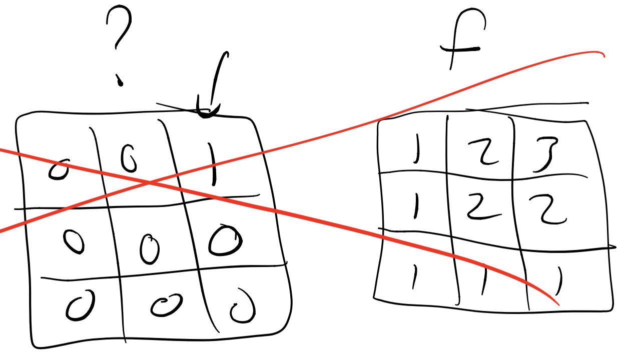
Can a linear filter do this?

Discuss in breakout rooms, answer on Socrative:

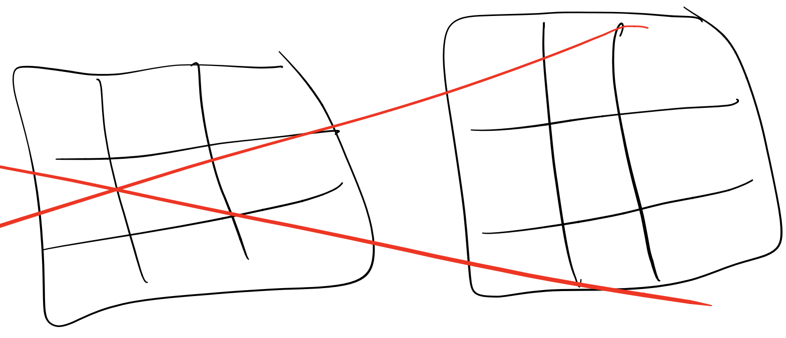
- Output pixel = max value in the neighborhood
- Output pixel is
  - 255 if input pixel is > 127
  - 0 otherwise

- Compute a finite-difference approximation of the derivative of the image function?

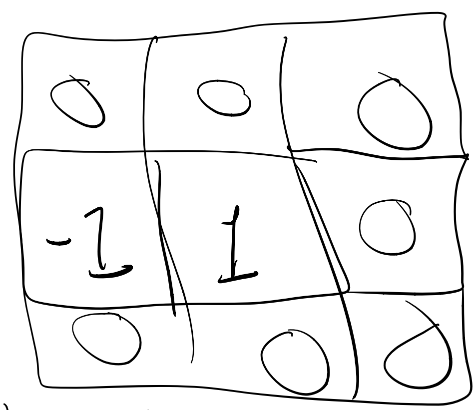
Maximum:



Threshold:

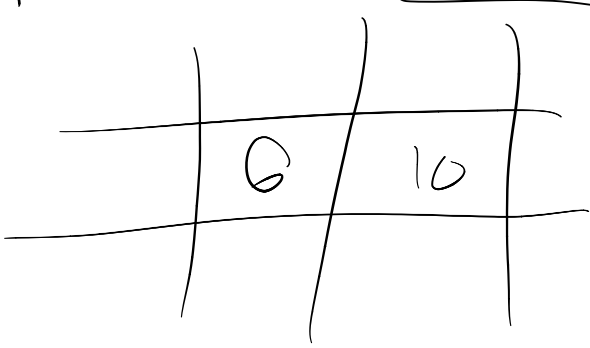


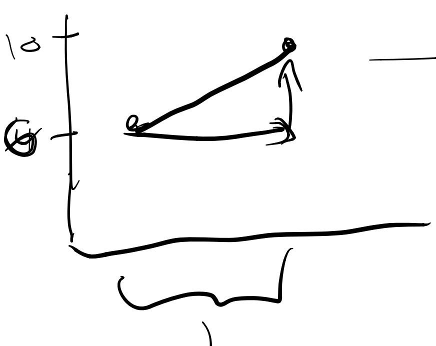
Derivative



$-1 \cdot 6 + 1 \cdot 10$

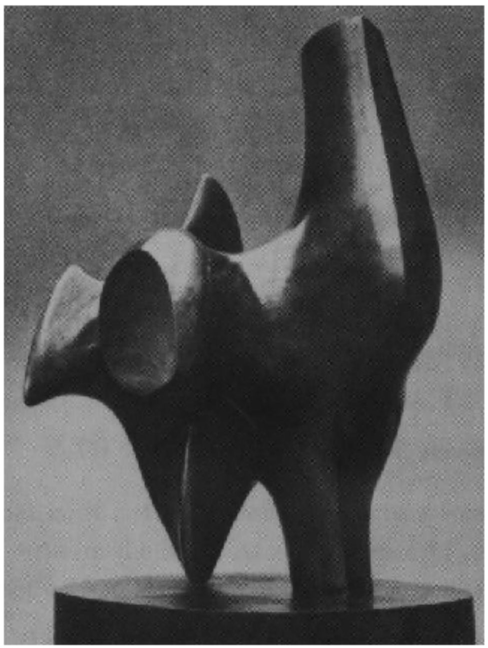
$= 4$

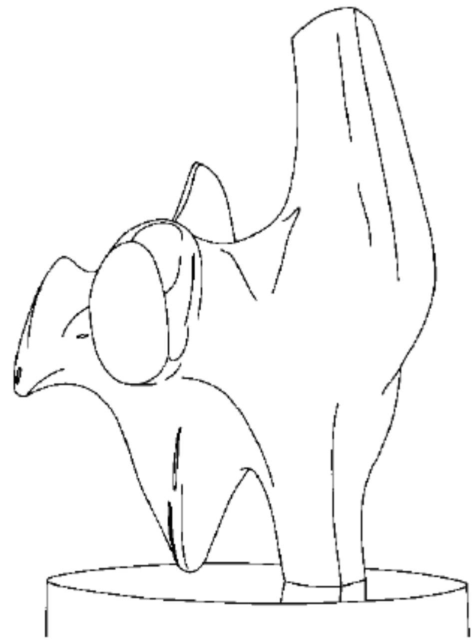




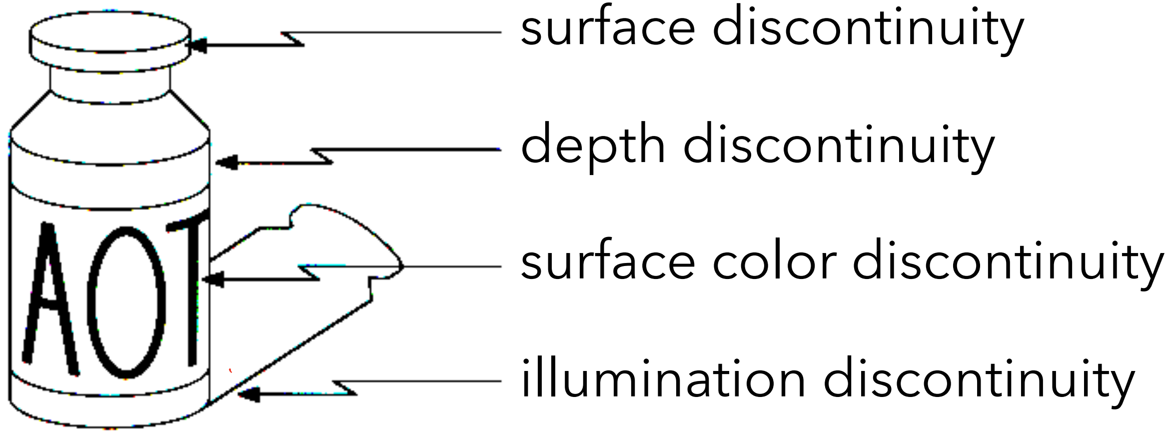
Calculus!?

- Edge detection: a classic vision problem.



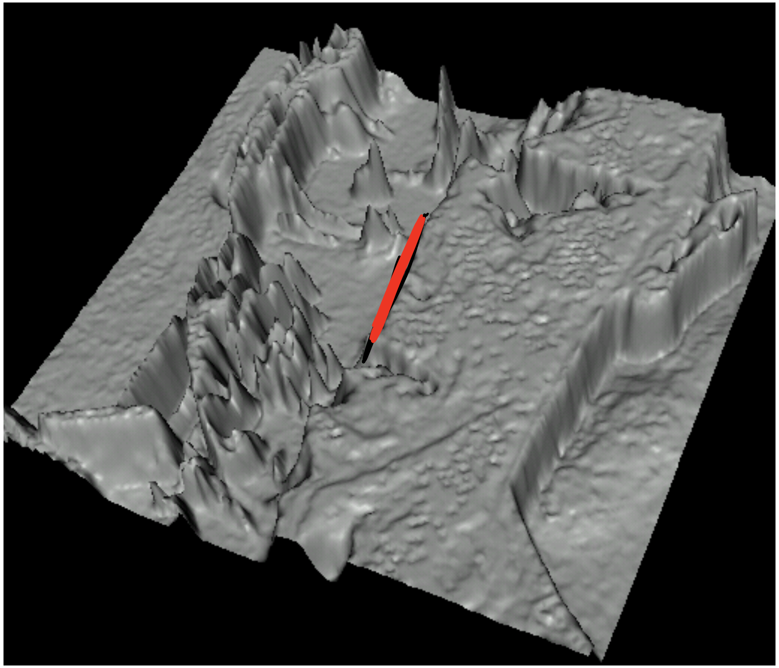


What is an edge?



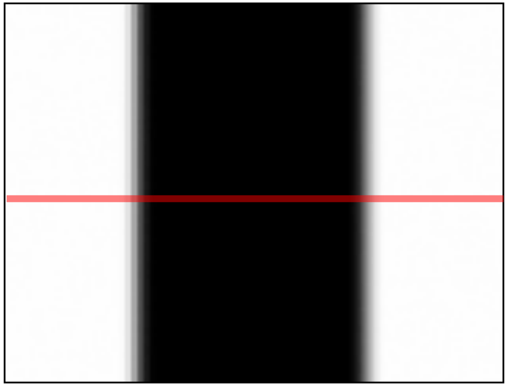
How do we find them?

f(x,y) as brightness

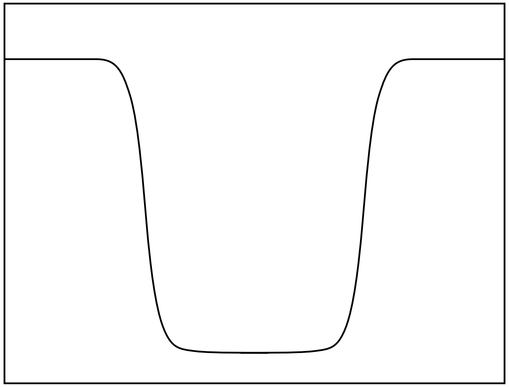


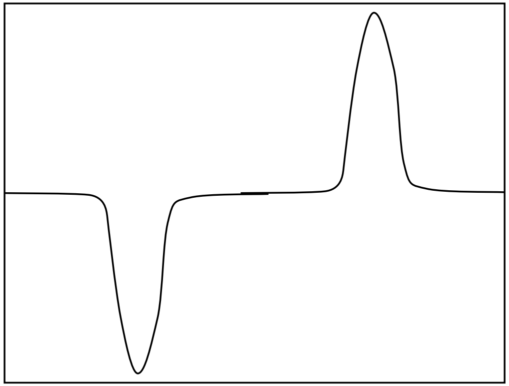
f(x,y) as height

Characterizing edges



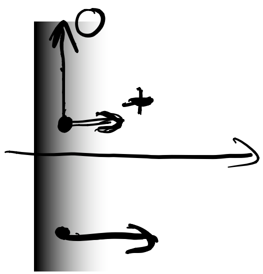
image


1D scanline


first derivative

Partial Derivatives

- Images are 2D - have x and y partial derivatives



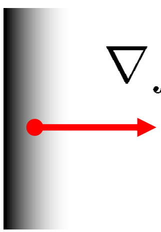
$$f(x,y) \quad \frac{\partial f}{\partial x}$$

- The partial derivatives together in a 2-vector are the image gradient:

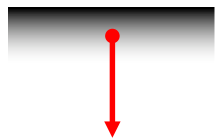
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Image Gradient as Edge Detector

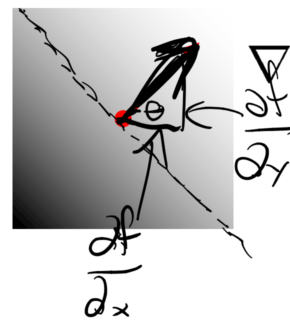
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



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

What is the edge **strength**?

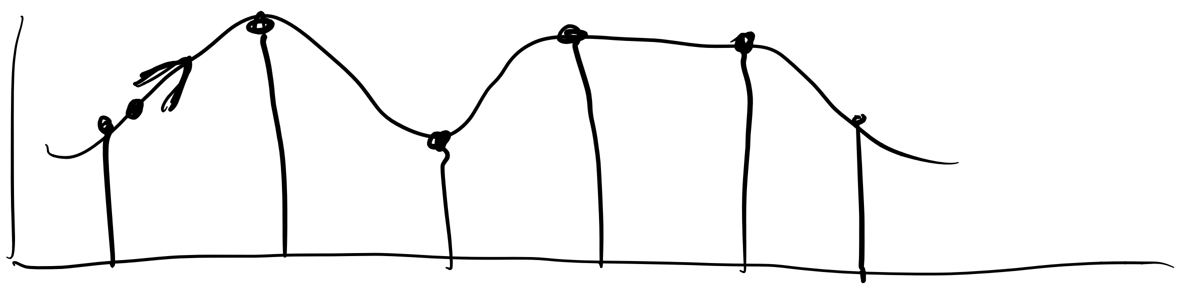
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

What is the edge **direction**?

$$\tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Image Derivatives

- How do we differentiate a discrete (sampled) image?
  - Reconstruct a continuous function and compute the derivative



  - Use finite differences

Derivative Filters

- How do we differentiate a discrete digital image?
  - Use finite differences

Derivative Filters

- How do we differentiate a discrete digital image?
  - Use finite differences

1	1	0
1	1	0
1	0	0

not centered

0	0	0
-1	1	0
0	0	0

0	0	0
0	1	0
0	-1	0

centered

0	0	0
-1	0	1
0	0	0

0	1	0
0	0	0
0	-1	0

Derivative Filters

- How do we differentiate a discrete digital image?
  - Use finite differences

0	0	0	1	1
0	0	0	1	1
0	0	0	1	1
0	0	1	1	1
0	0	1	1	1

*

0	0	0
-1	0	1
0	0	0

=

0	1	1
0	1	1
1	1	0

Exercise: Compute the horizontal (x) derivative using the centered finite difference filter. Assume "repeat" padding mode, "same" output size.

Image Gradient: Visually