

CSSE463: Image Recognition

Day 6

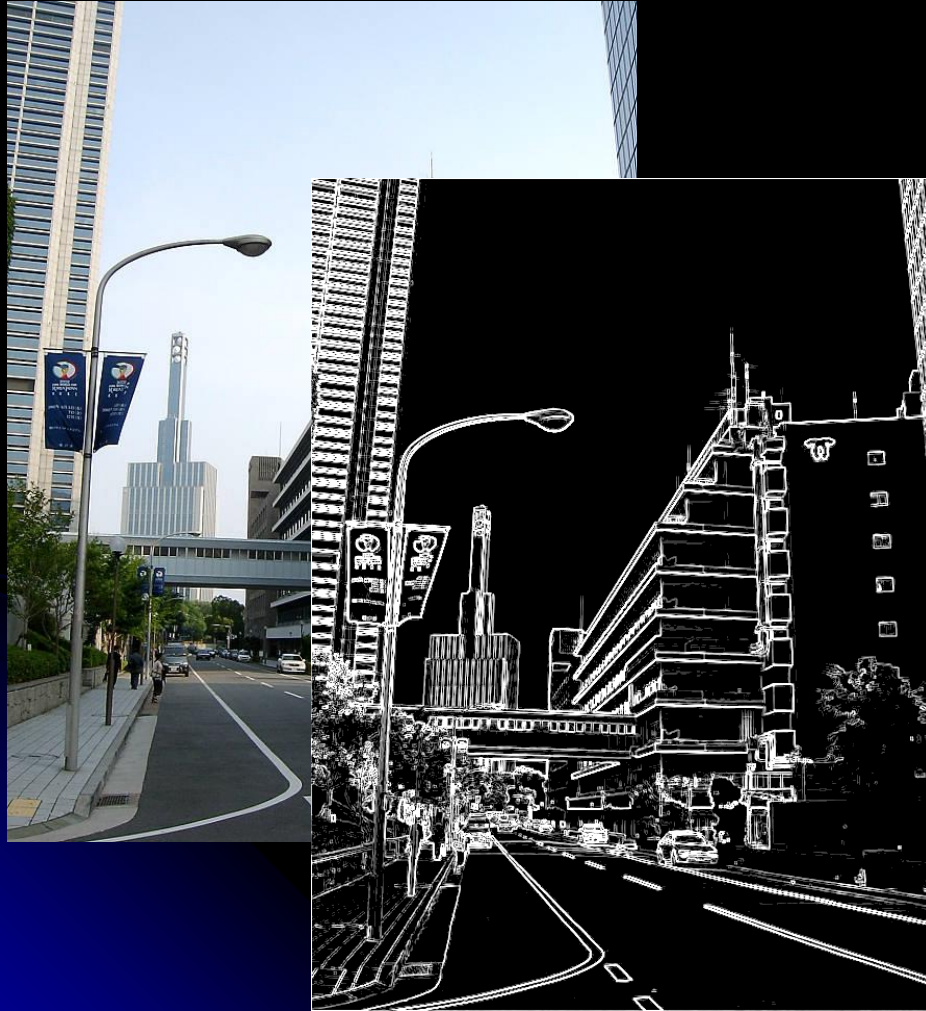
- Yesterday: Local, global, and point operators use different context, but **all**
 - operate on entire image,
 - changing one pixel at a time!!
- Lab due tomorrow 9:00 am
- Fruit-finder deadline **Friday**, 11:59pm
 - Please leave time for a solid write-up
 - Nice example of how to show results:
 - See rubric online for other standards
 - Questions?
- Today: edge features (another local operator)
 - Sonka 5.3



- There are 2 types of programmers in the world:
 1. Those who prefer 1-based indexing
 1. Those who prefer 0-based indexing

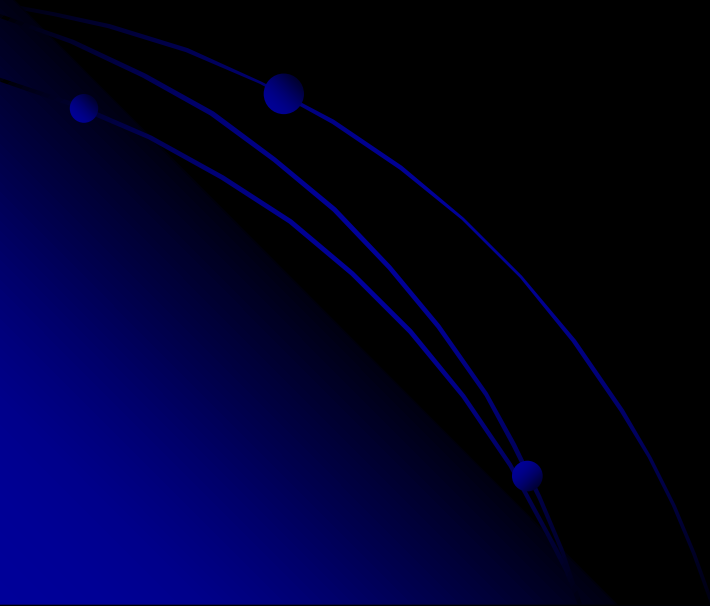
Thanks to 463 student John Krasich for clarifying this for us.

Edge Features – Why?

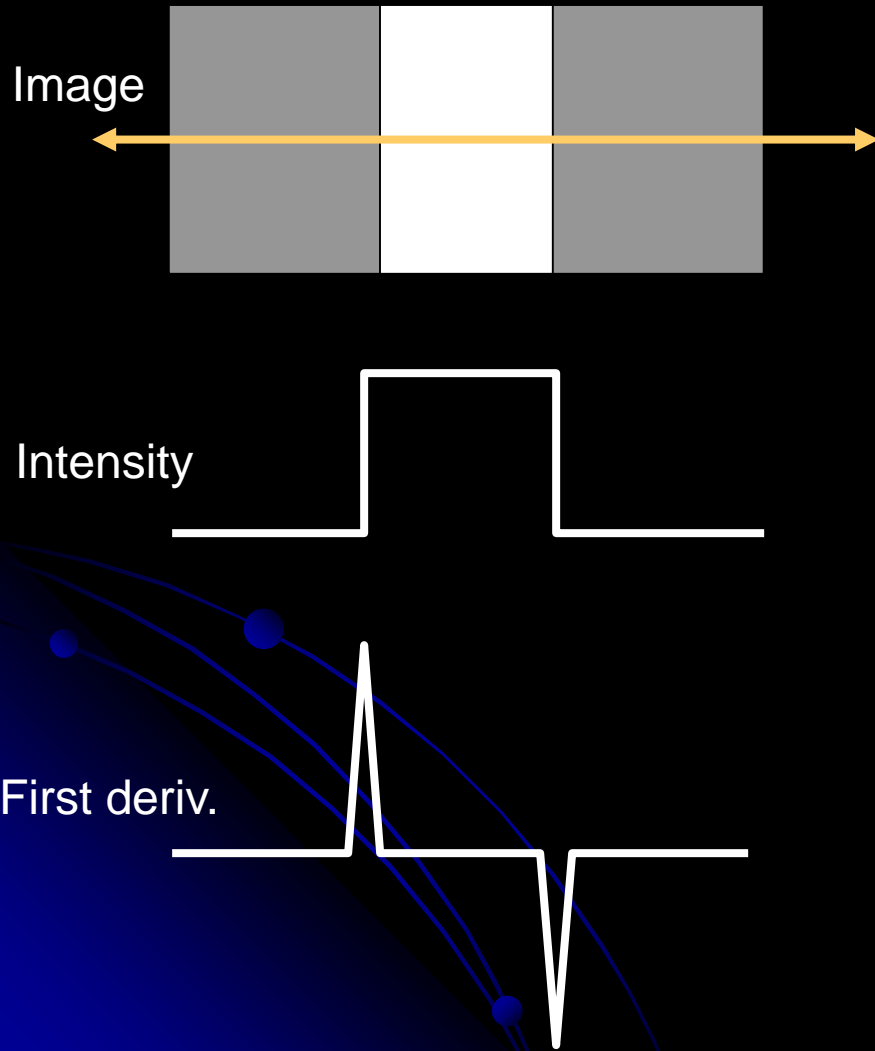


- “Edginess” (# edges) and their directions can give you info about the scene content
 - Orientation of the image
 - Natural vs. manmade images
- Edges can be used to segment the image.
 - Color information is usually used as well.
 - Specifically, boundaries occur where the chroma and/or luminance change (drastically).
- We could use to enhance the fruit-finder in a later assignment (*not* now).

Outline for next 2 sessions

- Concept: How to find “edges” in 1D signal
 - Edges in 2D images
 - Limitations
 - Edges vs edgels, Canny edge detector
- 

Intuition: Finding edges



- What's an edge?
- How to find changes in intensity?
- How to find first derivative?

Finding derivatives (1D)

- Let y be intensity of point at location x

- Def:
$$\frac{\partial y}{\partial x} \approx \frac{\Delta y}{\Delta x}$$

- Fix $\Delta x = 1$ pixel

- $dy/dx = y_2 - y_1$

f : [0 0 0 0 0 50 50 50 50 0 0 0 0 0] ;

f' : [0 0 0 0 0 50 0 0 0 -50 0 0 0 0] ;

- Correlate image with filter $[-1, 1]$ to find positions of change.
 - Edges “between” pixels.
 - What is significance of magnitude of first deriv. ?

Applying Filters

- *Example for differential with $\Delta x = 2$ pixels:
(Better; no output “between” pixels)*

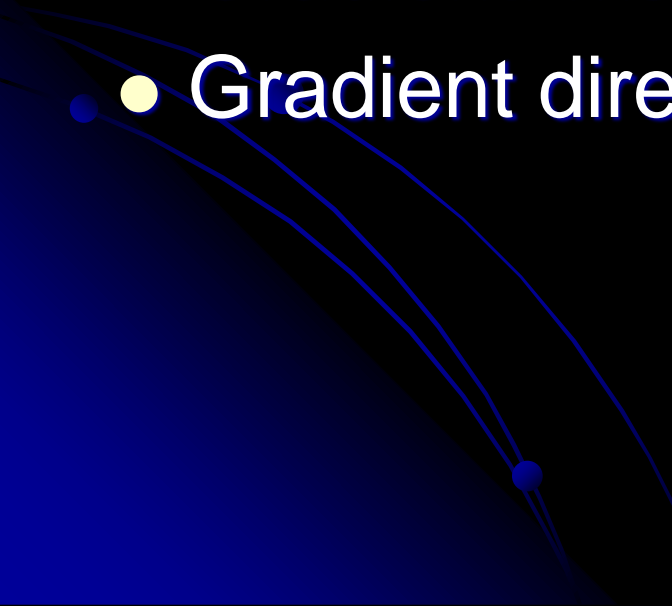
Image	5	8	9	1	2	2	1	2	1	3	1	3
Mask	$-\frac{1}{2}$	$-\frac{1}{2}$	$0\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$							
Output		2	-3.5	-3.5	...							

- Could you do
 - Ramps? Impulse? Step edges? (on quiz)
- Properties
 - If no contrast?

Why should the values in an edge filter sum to 0?

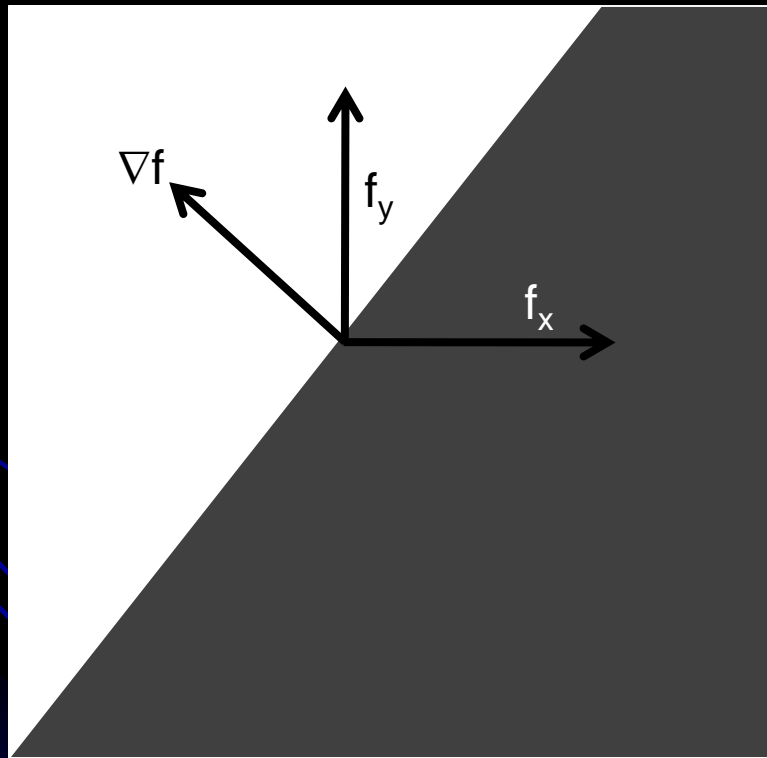
- What if they didn't?
- Consider running it on a homogeneous region: 40, 40, 40, 40, 40, 40

2D Edges

- Local operators
 - Prewitt operators
 - Sobel masks
 - Roberts 2x2 cross-operators
 - Gradient: magnitude
 - Gradient direction
- 

Edge gradient

Vector pointing in direction of greatest positive change.
We want its magnitude and direction



1. Find partials using filters

To find $\frac{\partial f}{\partial x}$, use Prewitt: $\frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ or Sobel: $\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ filter

To find $\frac{\partial f}{\partial y}$, use Prewitt: $\frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ or Sobel: $\frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ filter

Note that this is 1D filter, but averaged over 3 rows (for df/dx) or 3 cols (for df/dy) and with $1/6$ factored out to allow integer multiplication

Roberts 2x2 cross operators $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ are more sensitive to noise

Demo

- My homemade edgefinder
 - Finds vertical and horizontal edges using filters
 - Combines to find edge magnitude
 - Combines to find edge direction
 - Re-scale for display
- Similar to part of Lab 3.
 - So I can't post code

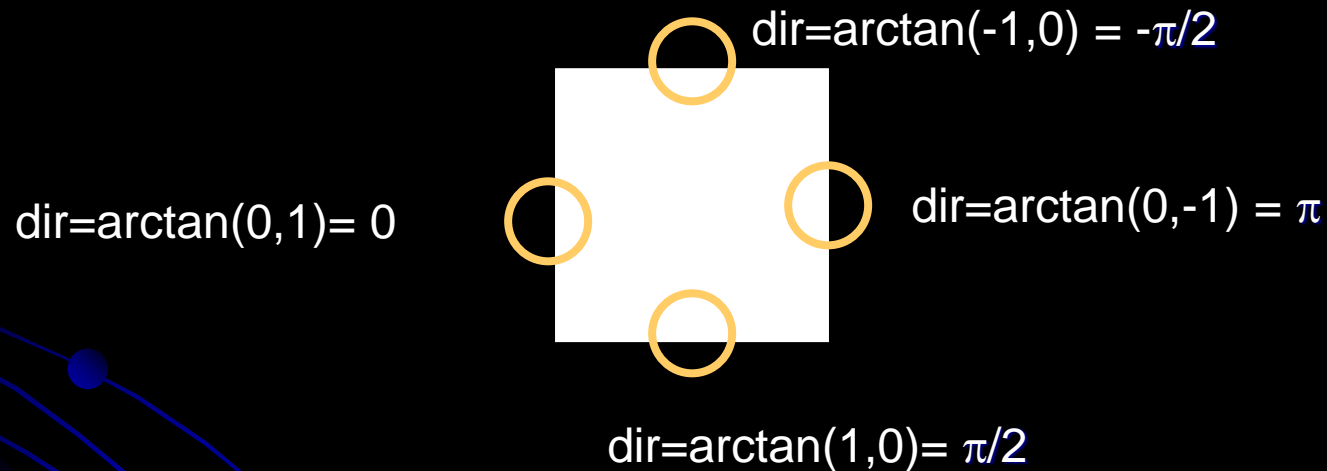
2. Find edge gradient magnitude

- Definition: the gradient, ∇f , is the vector pointing in the direction of greatest change.
- To find its magnitude:

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

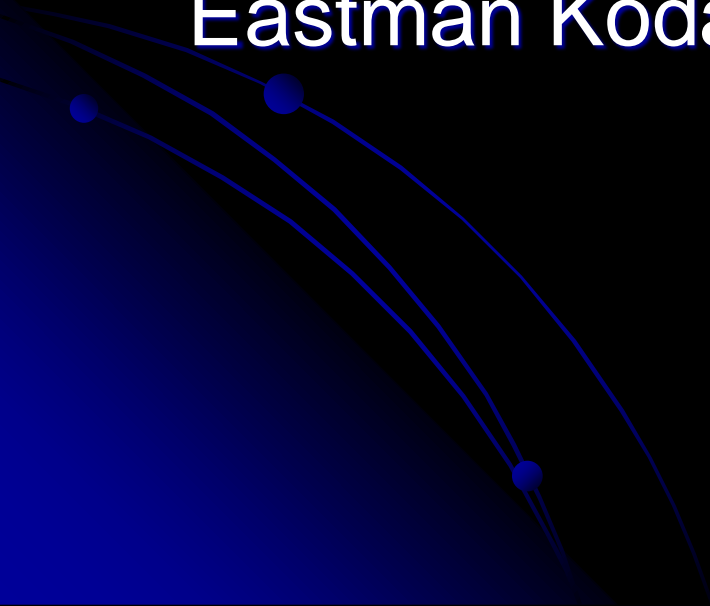
3. Find edge gradient direction

- $\tan^{-1}(y,x)$
- Matlab's $\text{atan2}(y,x)$ gives full range, $[-\pi, \pi]$



- Direction is thus the angle formed by the x-axis and the line “pointing towards” light region.

Color edges

- Rarely used historically
 - Intuition: edges occur between regions of different hue but same intensity.
 - One technique patented by David Cok, Eastman Kodak Co.
- 

Limitations of edgel-finders

- Natural variation
 - Shadows and highlights can obscure edges
- Internal vs. external edges
 - We might want the outline of an article of clothing, but the stripes in our shirt are edges too.
- Noise!
 - Signal-to-noise ratio important in determining how hard it is to find edges.

Edgels vs. Edges



- *Edgels* are unconnected groups of pixels detected by a mask
- *Edges* are longer segments found by grouping edgels
 - Intuitively, we think of edges
- Different data structure
- How might you process a “raw” edge image?

From mask output to edgels: ideas

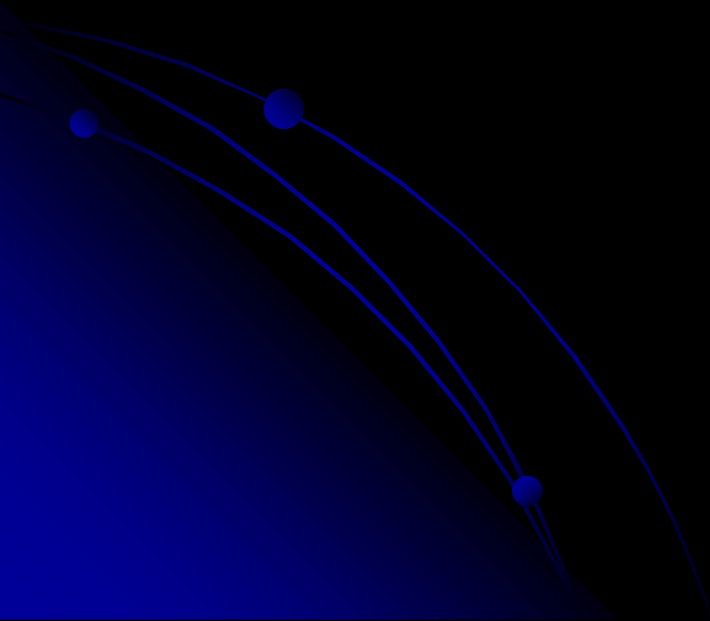
- Threshold away “weak” output
 - What threshold to use?
 - Always fixed or should it vary?
- “Thin” edges by nonmaximum suppression.
 - Idea: If an edge is 5 pixels wide, we can replace it with only the innermost segment.
 - Remove the edge response of a pixel not greater than its 2 neighbors in the direction of the gradient.

Canny edge detection

- First smooths the intensity image
 - Parameter σ controls how many edges found
- Non-maximal suppression
- Uses **two thresholds**:
 - High: to initiate contour following
 - Low: to follow along a contour
 - Result: segments from noise are less likely to be found (unless the noise is too strong)
- Aggregates neighboring edgels into curves (“edges”)

Canny edge detection

- You'll get to play with various edgefinders in Lab 3 using Matlab's built-in *edgedemo*



- Some neat image rec/computer vision demos using the kinect:

- http://www.youtube.com/watch?v=7QrnwoO1-8A&feature=mfu_in_order&list=UL
- <http://www.engadget.com/2010/12/09/kinect-finally-fulfills-its-minority-report-destiny-video/>