Computer Vision for HCI

Edge Detection

1

Edges in Images

- <u>Edge pixels</u> are pixels in an image where brightness changes sharply
  - Sharp changes in image brightness
- Interesting things happen at an edge
  - Object boundaries (light object on dark background)
  - Reflectance changes/patterns (zebra stripes, leopard spots)
  - Sharp changes in surface orientation
- Look at derivatives in image to detect edges
  - Gradients in 2-D
- Primary problem in edge detection is dealing with image noise

2

2

Gradients and Edges

- High-contrast image pixels can be detected by computing intensity differences in local image regions
- Detect high-contrasts using neighborhood templates or masks
  - Similar to approach for noise removal
- Begin study with 1-D signals, then move onto 2-D
  - A 1-D signal could be a row or column of an image

3

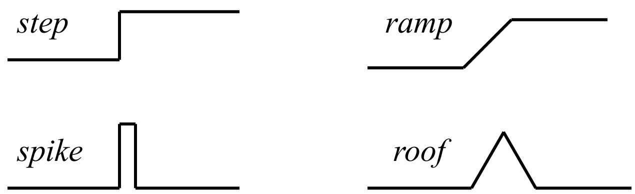
3

1-D Signals

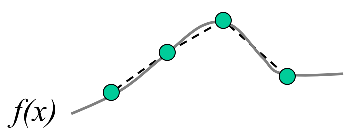
4

Differencing 1-D Signals

- A few idealized types of 1-D edges



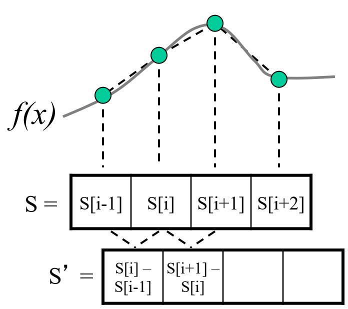
- Consider 1-D <u>sampled</u> signal (roof):



5

5

Difference/Derivative Mask



$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \approx f(x_i) - f(x_{i-1})$$

<u>Difference Masks</u>

M' = | **-1** | **+1** |

6

6

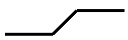
First Derivative Examples

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 \otimes M'$	-	0	0	0	0	**12**	0	0	0	0

S_1 is upward step edge

S_2	24	24	24	24	24	12	12	12	12	12
$S_2 \otimes M'$	-	0	0	0	0	**-12**	0	0	0	0

S_2 is downward step edge



S_3	12	12	12	12	15	18	21	24	24	24
$S_3 \otimes M'$	-	0	0	0	**3**	**3**	**3**	**3**	0	0

S_3 is upward ramp

S_4	12	12	12	12	24	12	12	12	12	12
$S_4 \otimes M'$	-	0	0	0	**12**	**-12**	0	0	0	0

S_4 is bright impulse

7

7

Smoothing

- Simple differences tend to give strong (unwanted) responses to **noise**
  - Poor way to estimate derivatives in real signals/images
- In practice, signal/image almost always smoothed before taking derivate
- Typically, <u>Gaussian</u> smoothing is used

8

8

Derivative of Gaussian

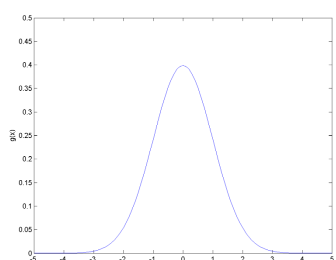
- **Smoothing then differentiating is same as convolving with the derivative of a smoothing kernel**
- Thus need only to convolve with a "derivative of the Gaussian" filter
  - Use **equation** for "derivative of Gaussian"!
- Results in smaller noise responses from derivative estimates

9

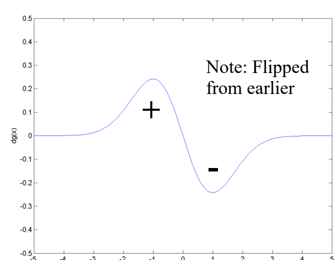
9

Derivative of Gaussian

$$g(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_{\circ})^2}{2\sigma^2}} \longrightarrow$$



$$\frac{dg(x;\sigma)}{dx} = \frac{-(x-x_{\circ})}{\sqrt{2\pi}\sigma^3} e^{-\frac{(x-x_{\circ})^2}{2\sigma^2}} \longrightarrow$$



(σ controls the scale/spread)

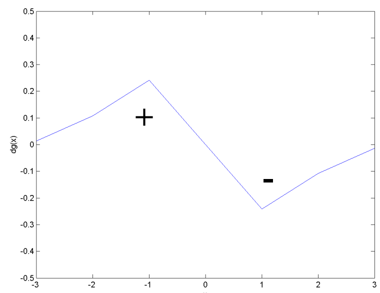
10

10

Discrete Gaussian Derivative Mask

- Set mask size: ceil(3σ)*2+1
  - Examine values for x = [-ceil(3σ) : ceil(3σ)]
- For σ = 1, yields a "7-tap" filter

  mask = [.01, .11, .24, 0, -.24, -.11, -.01]



11

11

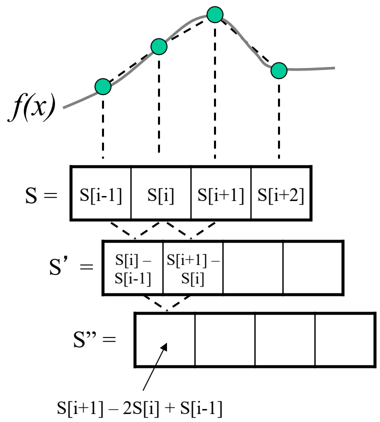
Second Derivative Option

- Second derivative is zero when derivative magnitude is extremal (at a peak/valley)
- To find large changes (edges), another good place to look is where second derivative makes "zero-crossings"
- Look for a change from "+ to –" or "– to +"
  - Can also look for "0 to +/–" or "+/– to 0"
- Produces double-sided edges

12

12

Difference/Derivative Masks



$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \approx f(x_i) - f(x_{i-1})$$

<u>Difference Masks</u>

M' =

-1	**+1**

M" =

+1	**-2**	**+1**

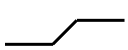
Second Derivative Examples

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 \otimes$ M"	-	0	0	0	**12**	**-12**	0	0	0	0

S_1 is upward step edge

S_2	24	24	24	24	24	12	12	12	12	12
$S_2 \otimes$ M"	-	0	0	0	**-12**	**12**	0	0	0	0

S_2 is downward step edge



S_3	12	12	12	12	15	18	21	24	24	24
$S_3 \otimes$ M"	-	0	0	**3**	0	0	0	**-3**	0	0

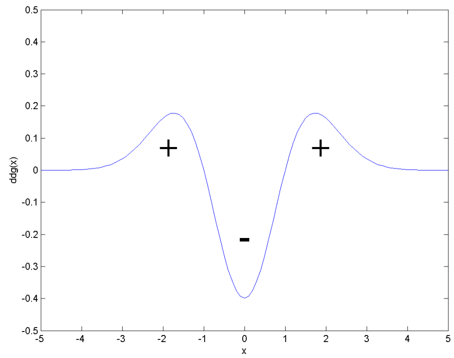
S_3 is upward ramp

S_4	12	12	12	12	24	12	12	12	12	12
$S_4 \otimes$ M"	-	0	0	**12**	**-24**	**12**	0	0	0	0

S_4 is bright impulse

Gaussian Second Derivative

$$\frac{\partial^2 g(x;\sigma)}{\partial x^2} = \left(\frac{(x - x_\circ)^2}{\sqrt{2\pi}\sigma^5} - \frac{1}{\sqrt{2\pi}\sigma^3} \right) \cdot e^{-\frac{(x - x_\circ)^2}{2\sigma^2}}$$

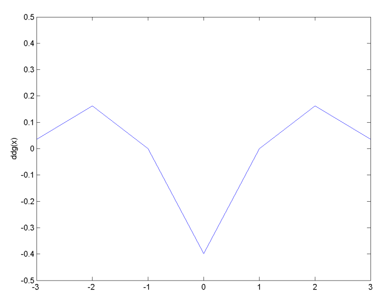


15

15

Gaussian 2nd Derivative Mask

- Set mask size: ceil(3σ)*2+1
  - Examine values for x = [-ceil(3σ) : ceil(3σ)]
- For σ = 1, yields a 7-tap filter
  - mask = [.04, .16, 0, -.40, 0, .16, .04]



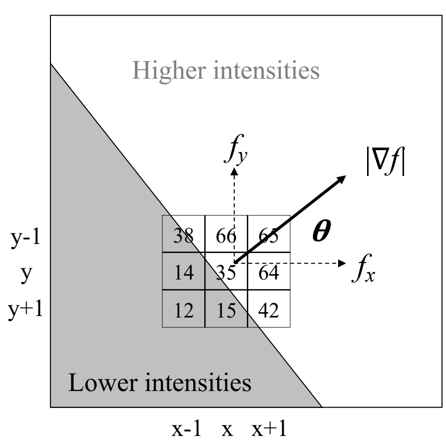
16

16

2-D Signals (Images)

17

2-D Difference "Gradient" Operators



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

$$\begin{aligned}\frac{\partial f}{\partial x} \equiv f_x \approx\ & \frac{1}{3}[(I[x+1,y]-I[x-1,y])/2 \\ & +(I[x+1,y-1]-I[x-1,y-1])/2 \\ & +(I[x+1,y+1]-I[x-1,y+1])/2]\end{aligned}$$

$$\begin{aligned}\frac{\partial f}{\partial y} \equiv f_y \approx\ & \frac{1}{3}[(I[x,y+1]-I[x,y-1])/2 \\ & +(I[x-1,y+1]-I[x-1,y-1])/2 \\ & +(I[x+1,y+1]-I[x+1,y-1])/2]\end{aligned}$$

$$\theta = \text{atan}(f_y, f_x) \qquad |\nabla f| = \sqrt{f_x^2 + f_y^2}$$

18

18

Classic Gradient Masks

Prewitt:

$$F_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} /6 \qquad F_y = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} /6$$

<u>Sobel</u>:

$$F_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} /8 \qquad F_y = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} /8$$

19

19

Separability

Prewitt:

$$F_x = 1/6 \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline \end{array} \qquad F_y = 1/6 \begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array} \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}$$

<u>Sobel</u>:

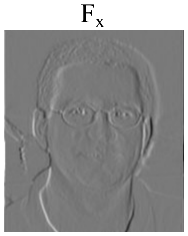
$$F_x = 1/8 \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline \end{array} \qquad F_y = 1/8 \begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array} \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline \end{array}$$

20

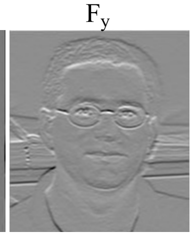
20

Prewitt mask results

F_x



F_y



Sobel mask results

F_x

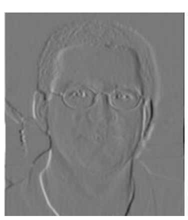
F_y

21

21

Gradient Magnitude

$\sqrt{\quad}$



2

F_x

+

2

F_y

=

22

22

Different Gradient Magnitude Strengths

(using increasing threshold to remove weaker gradient magnitudes)

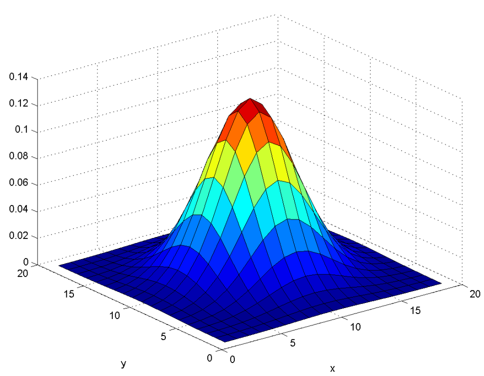
(Gradient magnitude obtained with Sobel masks)

23

23

2-D Gaussian

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_\circ)^2+(y-y_\circ)^2}{2\sigma^2}}$$



24

24

Gaussian Derivatives

$$\frac{\partial g(x,y;\sigma)}{\partial x} = \frac{-(x-x_{\circ})}{2\pi\sigma^4} e^{-\frac{(x-x_{\circ})^2+(y-y_{\circ})^2}{2\sigma^2}}$$

$$\frac{\partial g(x,y;\sigma)}{\partial y} = \frac{-(y-y_{\circ})}{2\pi\sigma^4} e^{-\frac{(x-x_{\circ})^2+(y-y_{\circ})^2}{2\sigma^2}}$$

25

25

Creating Masks

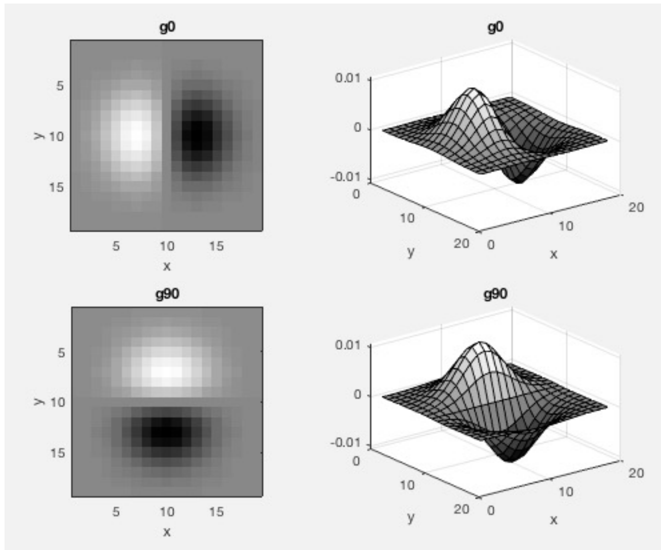
- Gx mask
  - Fill mask values with $g_x(x,y;\sigma)$ **EQUATION**
    - x-range: [-ceil(3σ) : ceil(3σ)] (could use 2σ)
    - y-range: [-ceil(3σ) : ceil(3σ)]
- Gy mask
  - Fill mask values with $g_y(x,y;\sigma)$ **EQUATION**
    - x-range: [-ceil(3σ) : ceil(3σ)] (could use 2σ)
    - y-range: [-ceil(3σ) : ceil(3σ)]

**** Do **NOT** make a Gaussian mask and take difference operator over the mask!*

26

26

Results



27

27

Convention: Alter to Point in <u>Increasing</u> Axis Dimension

$$\frac{\partial g(x,y;\sigma)}{\partial x} = \frac{\cancel{-}(x-x_{\circ})}{2\pi\sigma^4} e^{-\frac{(x-x_{\circ})^2+(y-y_{\circ})^2}{2\sigma^2}}$$

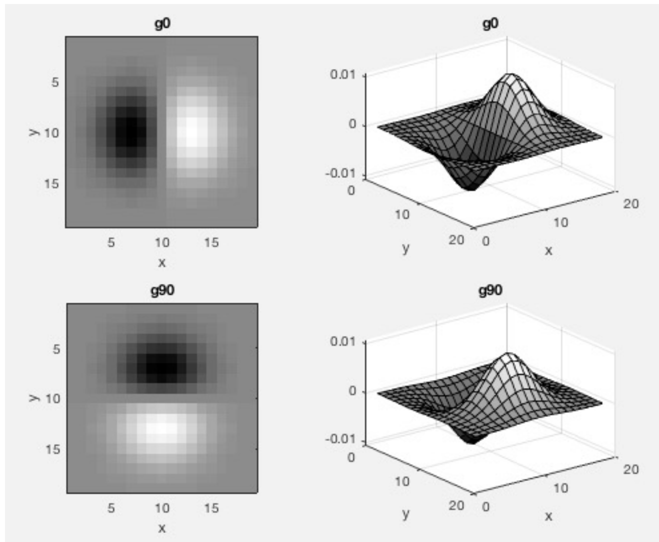
Remove negative (-) sign!

$$\frac{\partial g(x,y;\sigma)}{\partial y} = \frac{\cancel{-}(y-y_{\circ})}{2\pi\sigma^4} e^{-\frac{(x-x_{\circ})^2+(y-y_{\circ})^2}{2\sigma^2}}$$

28

28

Results



29

29

Effect of Scale

(Magnitude shown)

$\sigma = 1$ $\sigma = 3$ $\sigma = 5$

Thicker gradient ridges →

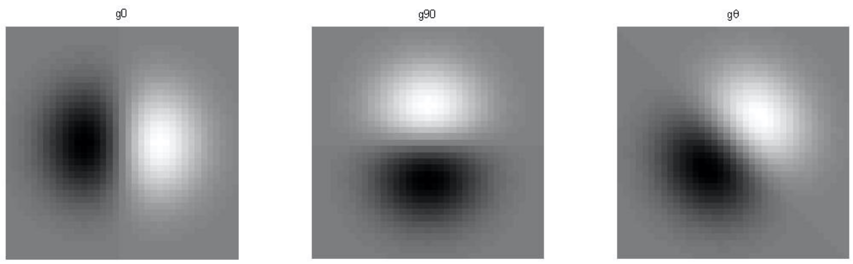
30

30

“Steerable” Filters

- Simple version: rotate by angle θ
  - Use G_x and G_y as basis set, and synthesize filter by linear combination of G_x and G_y

$$\cos(\theta)\cdot G_x + \sin(\theta)\cdot G_y$$



“The design and use of steerable filters”, Freeman and Adelson, *IEEE PAMI*, Sept. 1991.

31

31

Laplacian of Gaussian

- Recall second derivative function of Gaussian
  - Zero-crossings are found at edge locations
  - Smoothing used to combat noise
- Now combine two orientations into <u>one</u> circular filter
  - Simply sum Gaussian second derivatives in x direction and y direction
  - Non-oriented, 2^{nd} derivative filter
- Laplacian of Gaussian operator:

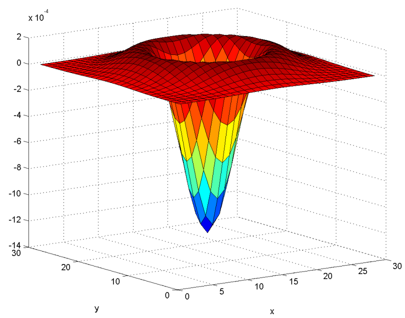
$$\nabla^2 g(x,y;\sigma) = \frac{\partial^2 g(x,y;\sigma)}{\partial x^2} + \frac{\partial^2 g(x,y;\sigma)}{\partial y^2}$$

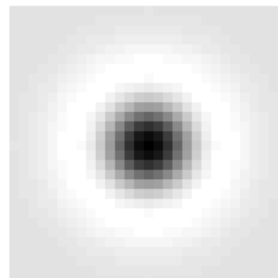
32

32

Laplacian of Gaussian

$$\nabla^2 g(x,y;\sigma) = \frac{\partial^2 g(x,y;\sigma)}{\partial x^2} + \frac{\partial^2 g(x,y;\sigma)}{\partial y^2}$$





33

33

Zero-Crossings

$\sigma = 2$

Selected
zero-crossings

Overlay

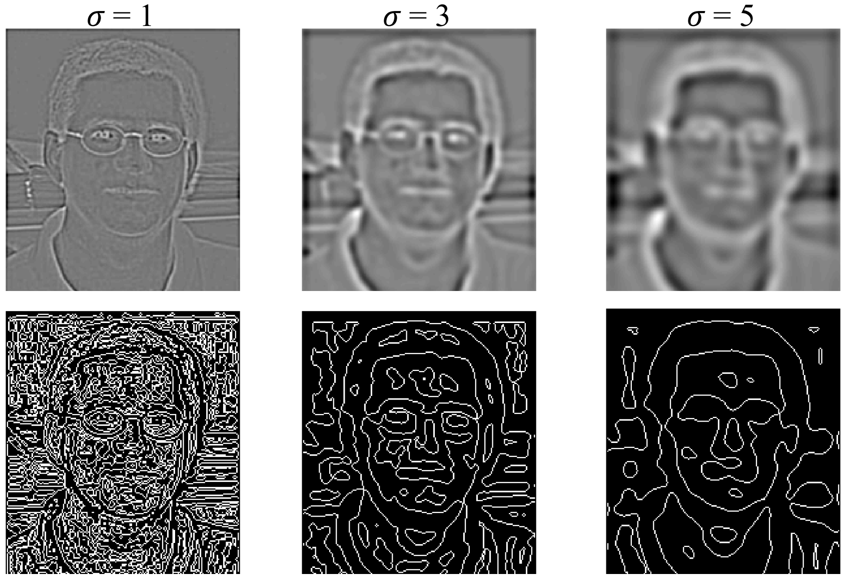
34

34

Effect of Scale

(<u>all</u> zero-crossings shown)

$\sigma = 1$ $\sigma = 3$ $\sigma = 5$



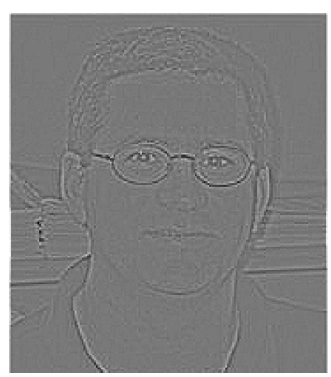
35

Simple Laplacian Mask

- Small 3 x 3 mask approximation of LOG

LOG =

0	-1	0
-1	4	-1
0	-1	0



36

36

Canny Edge Detector

- Very/Most popular and effective method
- Produces extended contour/edge segments by “following” high gradient magnitudes within the smoothed image

37

37

Canny Edge Detection Approach

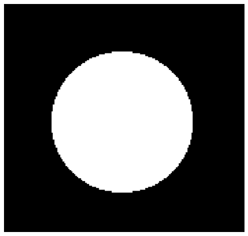
- Step 1: Smooth the image
  - Gaussian
- Step 2: Compute the gradients
  - Calculate magnitude and orientation at each pixel
- Step 3: Suppress non-maximal gradients
  - Keep points where gradient magnitude is maximal along the direction of the gradient (look for “hills”)
- Step 4: Follow edge contours (edge linking)
  - Use upper and lower thresholds (hysteresis thresholding)

Combine into one!

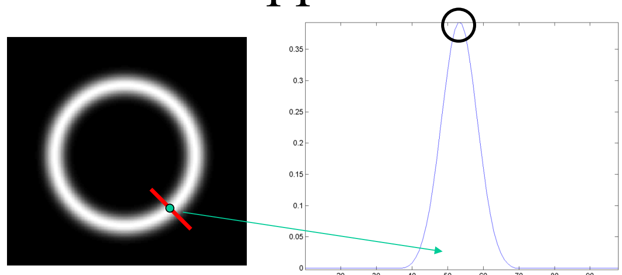
38

38

Non-Maximal Suppression



Input image

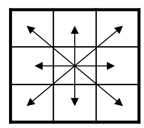


Gradient magnitude

Along gradient orientation at point

Do for all points/pixels in gradient magnitude image (results in a thin circle)

- Detection
  - Slice the gradient magnitude along gradient direction (perpendicular to edge)
    - Quantize gradient orientation by 45 degrees
    - Could also interpolate values
  - Mark points along slice that are maximal



39

39

Edge Linking

- Sequentially follow continuous contour segments
- Initiate only on edge pixels where gradient magnitude meets high threshold (T_1)
- A single threshold can cause many broken edge segments
- Once started, follow through connected pixels whose gradient magnitude meet a lower threshold (T_2)
  - In Matlab, the default is $T_2 = .4 * T_1$
- Referred to as "**hysteresis thresholding**"

40

40

Canny Results

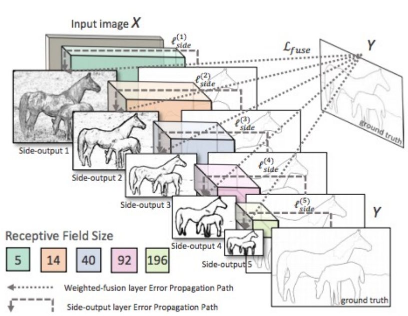
Original　　Canny edges　　Overlay

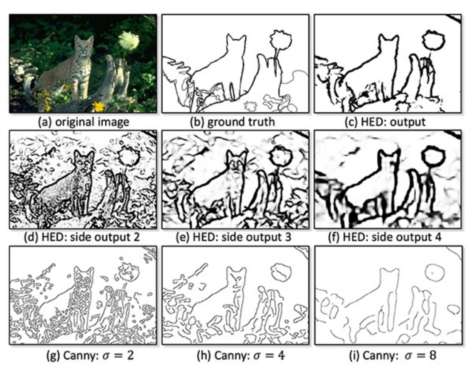
41

41

Deep Learning Approach

Holistically-Nested Edge Detection - HED (2015)





Note: **Ground-truth** of examples required for training.
There is no “training” with Canny

42

42

Perceptual Test: What is this???

43

43

Summary

- Interesting things happen at an edge
  - Object boundaries
- Look for derivatives/gradients in image
  - First and second derivatives/gradients
- Classic gradient operators
  - Sobel, Prewitt
  - Gaussian derivatives
- Primary problem in edge detection is dealing with image noise
  - Smoothing and hysteresis thresholding
- Canny edge detector

44

44