Lecture 13
Canny Edge Detection
ECEN 5283 Computer Vision

Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goals

- To review the LoG edge detector.

- To implement the gradient-based Canny edge detector.

John F. Canny
Professor
EECS, UC-Berkeley

Laplacian-of-Gaussian (LoG) for Edge Detection

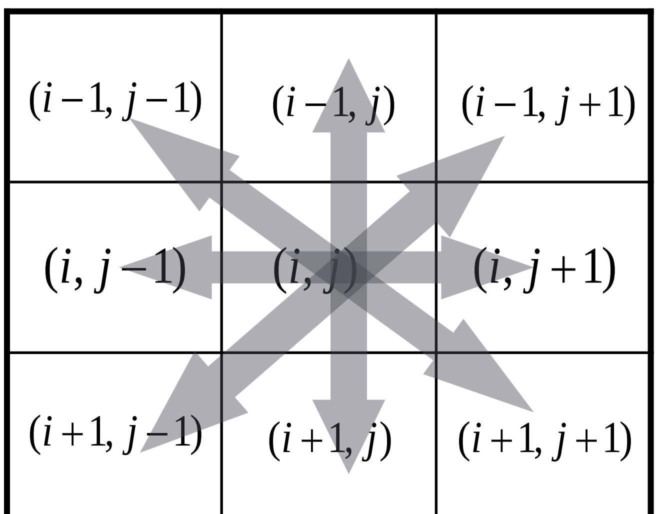
- If an image is pre-smoothed by a Gaussian filter, then we have the Laplacian-of-Gaussian (LoG) operation that is defined as

$$(K_{\nabla^2} ** (G_\sigma ** I)) = (K_{\nabla^2} ** G_\sigma) ** I = \left(\frac{\partial^2 G_\sigma}{\partial x^2} + \frac{\partial^2 G_\sigma}{\partial y^2}\right) ** I.$$

$$= (\nabla^2 G_\sigma) ** I \quad \text{where } \nabla^2 G_\sigma(x,y) = \left(\frac{1}{2\pi\sigma^4}\right)\left[\frac{x^2+y^2}{\sigma^2} - 2\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

(Discrete LoG)

$$K[i,j] = \left(\frac{1}{2\pi\sigma^4}\right)\left[\frac{(i-k-1)^2 + (j-k-1)^2}{\sigma^2} - 2\right] e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}}$$



Checking ZCs

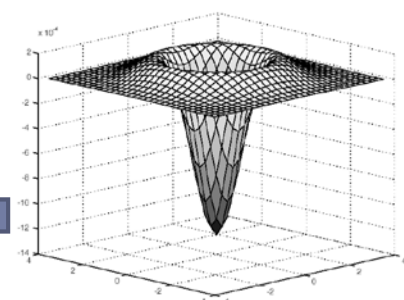


Figure 9.7. The Laplacian of Gaussian filter kernel, shown here for σ one pixel, can be thought of as subtracting the center pixel from a weighted average of the surround (hence the analogy with unsharp masking, described in the text). It is quite common to replace this kernel with the difference of two Gaussians, one with a small value of σ and the other with a large value of σ.

Advantages/disadvantages of LoG

- **Advantages**
  - **Location:** ZCs are easier to find compared with peaks. Small variances have high precision while large ones are more robust.
  - **Robustness:** the second derivative is much less noise-sensitive when Gaussian smoothing is applied first.
- **Disadvantages**
  - The LoG filter is not oriented, its response is composed of an average across an edge (normal) and one along the edge (tangent).
  - The ZCs are slightly displaced when the LoG is applied to objects having corners and thin line structures.

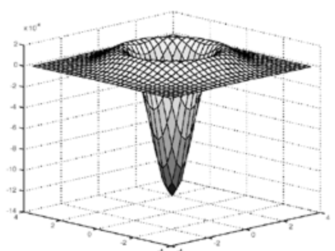
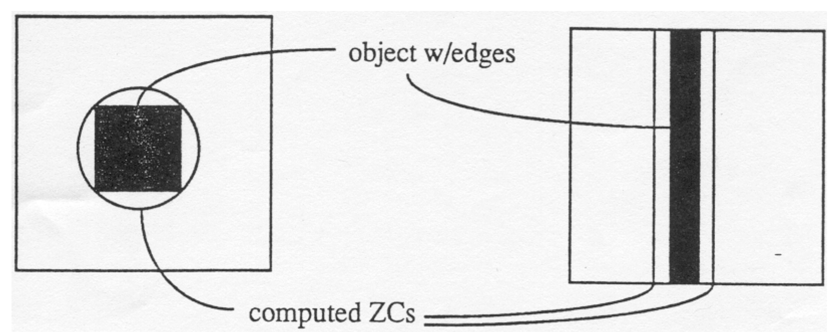


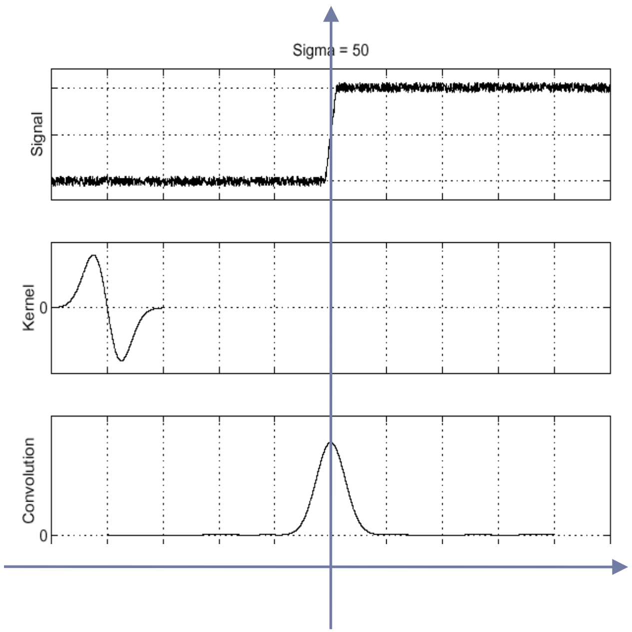
Figure 9.7. The Laplacian of Gaussian filter kernel, shown here for σ one pixel, can be thought of as subtracting the center pixel from a weighted average of the surround (hence the analogy with unsharp masking, described in the text). It is quite common to replace this kernel with the difference of two Gaussians, one with a small value of σ and the other with a large value of σ.



Canny Edge Detection

- Canny (1986) established the practice of choosing a derivative estimation filter by optimizing three criteria:
  - Signal to noise ratio: the filter should respond more strongly to the edge at x=0 than to noise.
  - Edge Localization: the filter response should reach a maximum close to x=0.
  - Low false positives: there should be only one maximum of the response in a reasonable neighborhood x=0.
- It is a remarkable fact that the optimal smoothing filters tend to look a great deal like Gaussians.

$$edge(x) = AU(x) + B + n(x).$$



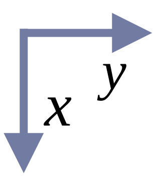
Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

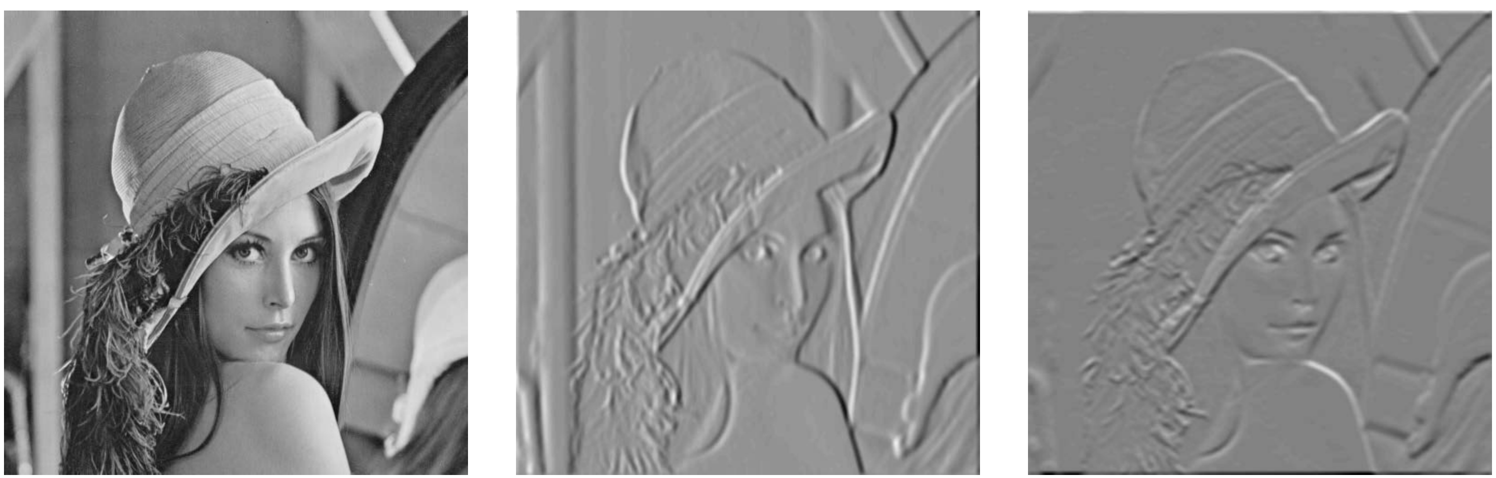
Gradient Maps

- We estimate the gradient magnitude, and use this estimate to determine the edge positions and directions.

$$f_x = \frac{\partial f}{\partial x} = K_{\nabla x} ** (G_\sigma ** I) = (\nabla_x G_\sigma) ** I. \qquad \nabla_x G_\sigma = \frac{-x}{2\pi\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$f_y = \frac{\partial f}{\partial y} = K_{\nabla y} ** (G_\sigma ** I) = (\nabla_y G_\sigma) ** I. \qquad \nabla_y G_\sigma = \frac{-y}{2\pi\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$





http://robotics.eecs.berkeley.edu/~sastry/ee20/cademo.html

Canny Edge Detector

- We can compute the magnitude and orientation of the gradient for each pixel based two filtered images.

$$|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2} = \text{rate of change of } f(x,y)$$

$$\angle \nabla f(x,y) = \tan^{-1}\left(\frac{f_y}{f_x}\right) = \text{orientation of change of } f(x,y) \qquad (\text{with possible} \pm \pi)$$

Non-maximum Suppression

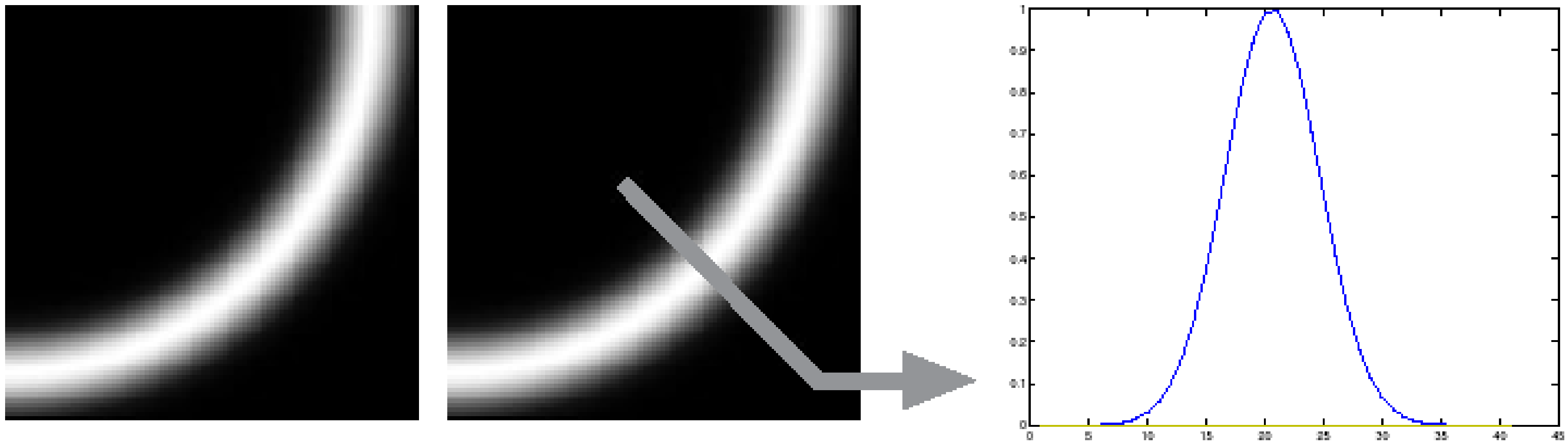
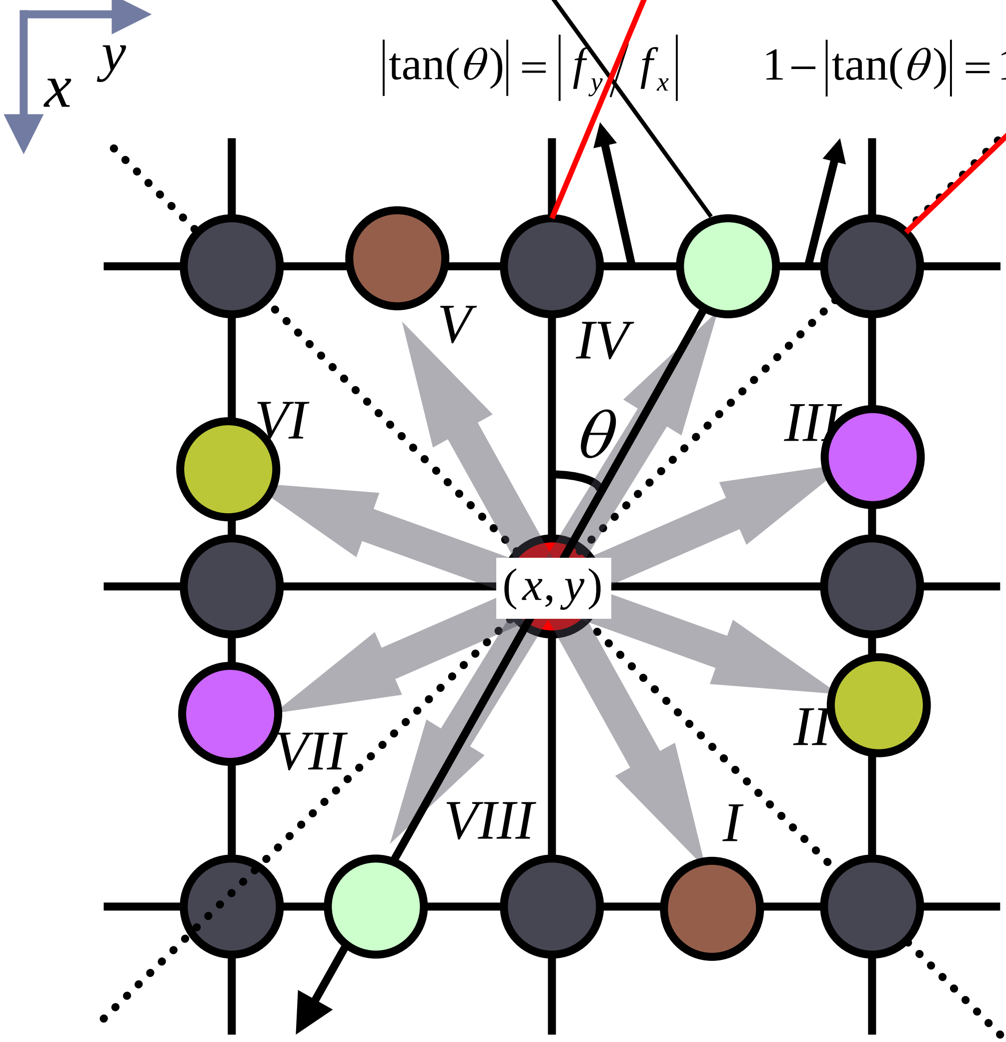


Figure 9.11. The gradient magnitude tends to be large along thick trails in an image. Typically, we would like to condense these trails into curves of representative edge points. A natural way to do this is to cut the trail perpendicular to its direction and look for a peak. We will use the gradient direction as an estimate of the direction in which to cut. The top left figure shows a trail of large gradient magnitude; the figure on the top right shows an appropriate cutting direction; and below, we show the peak in this direction.

$$\nabla f(x-1, y') = \nabla f(x-1, y)(1-|f_y / f_x|) + \nabla f(x-1, y+1)|f_y / f_x|$$

Non-maximum Suppression (Cont'd)

$$|\tan(\theta)| = |f_y / f_x| \qquad 1-|\tan(\theta)| = 1-|f_y / f_x|$$



An edge point is declared if it is a larger than the two neighboring pixels along the gradient direction (by linear interpolation)

$\text{Case 1}: f_x = 0 \text{ or } f_y = 0$

$\text{Case 2}: f_x f_y > 0, |f_x| \geq |f_y|$

$\text{Case 3}: f_x f_y > 0, |f_x| < |f_y|$

$\text{Case 4}: f_x f_y < 0, |f_x| \geq |f_y|$

$\text{Case 5}: f_x f_y < 0, |f_x| < |f_y|$

A simplified interpolation that uses the average of two neighboring pixels on the grid can be used.

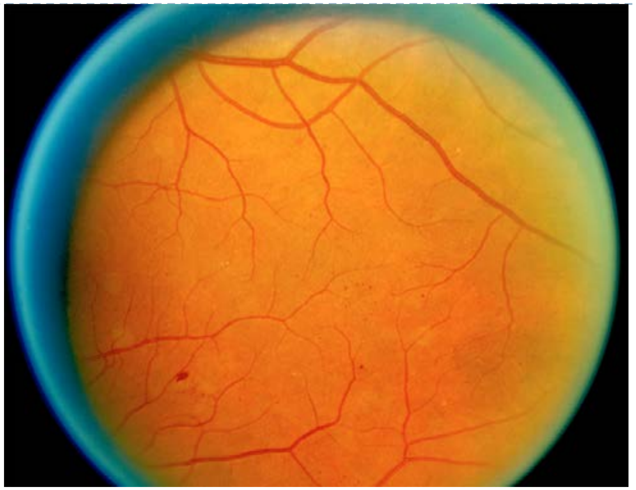
Canny Edge Detection Algorithm

```
form an estimate of the image gradient

obtain the gradient magnitude from this estimate

identify image points where the value
  of the gradient magnitude is maximal
  in the direction perpendicular to the edge
  and also large; these points are edge points
```

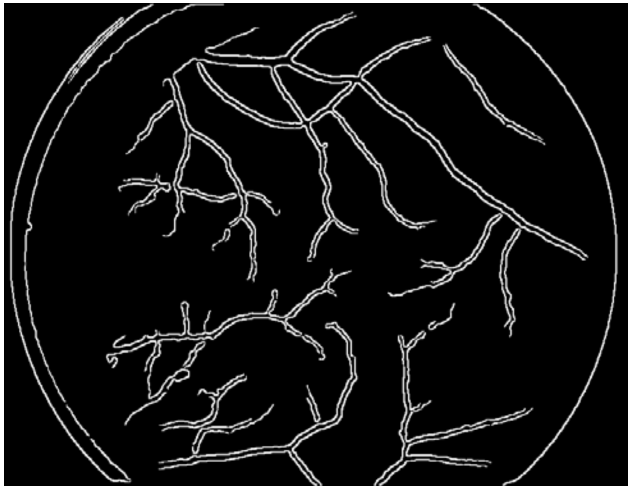
Algorithm 9.1: *Gradient based edge detection.*

Note: Non-maximum suppression is very time-consuming, It is a good idea to do it only for those pixels that have sufficiently large gradient magnitudes.

Project 2 (Due Feb. 25, 2015)



Canny detection



LoG detection