Image processing
(part 2)

Comparison: Gaussian noise

Mean Gaussian Median

3x3

5x5

7x7

Comparison: salt and pepper noise

Mean Gaussian Median

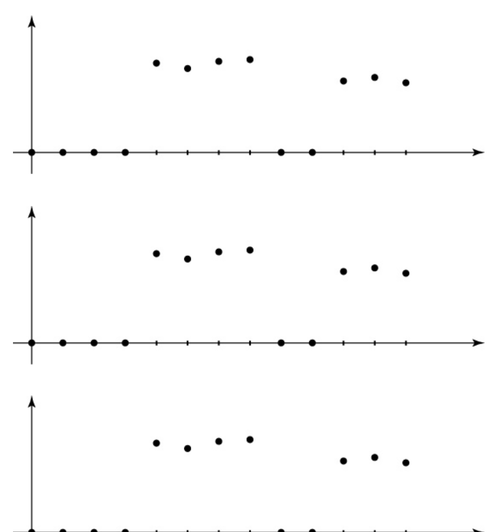
3x3

5x5

7x7

Bilateral filtering

Bilateral filtering is a method to average together nearby samples only if they are similar in value.



$$g[n] = 1/C \sum_{n'} f[n'] h_{\sigma_s}[n-n']\ h_{\sigma_r}(f[n]-f[n'])$$

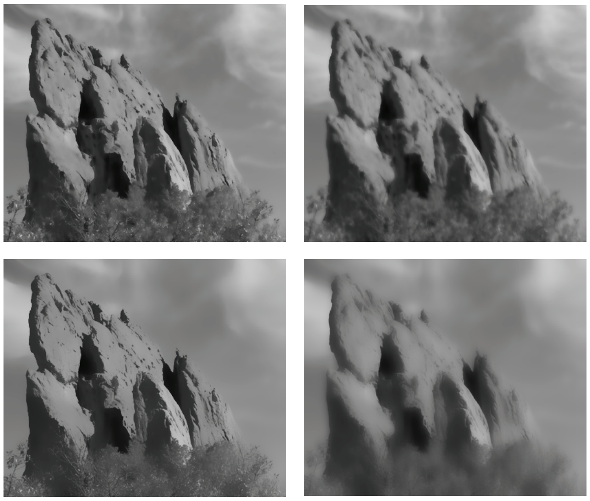
$$C = \sum_{n'} h_{\sigma_s}[n-n']\ h_{\sigma_r}(f[n]-f[n'])$$

Input

$\sigma_r = 0.1$ $\sigma_r = 0.25$

$\sigma_s = 2$

$\sigma_s = 6$



Paris, et al. SIGGRAPH course notes 2007

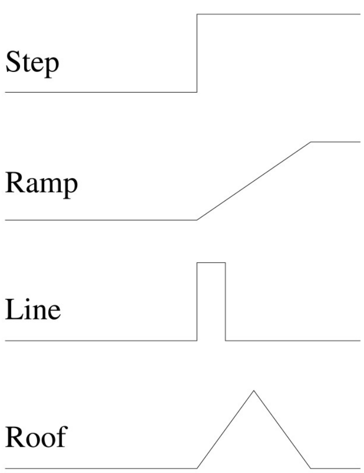
Edge detection

One of the most important uses of image processing is **edge detection:**

- Really easy for humans
- Really difficult for computers

- Fundamental in computer vision
- Important in many graphics applications

What is an edge?



Q: How might you detect an edge in 1D?

Gradients

The **gradient** is the 2D equivalent of the derivative:

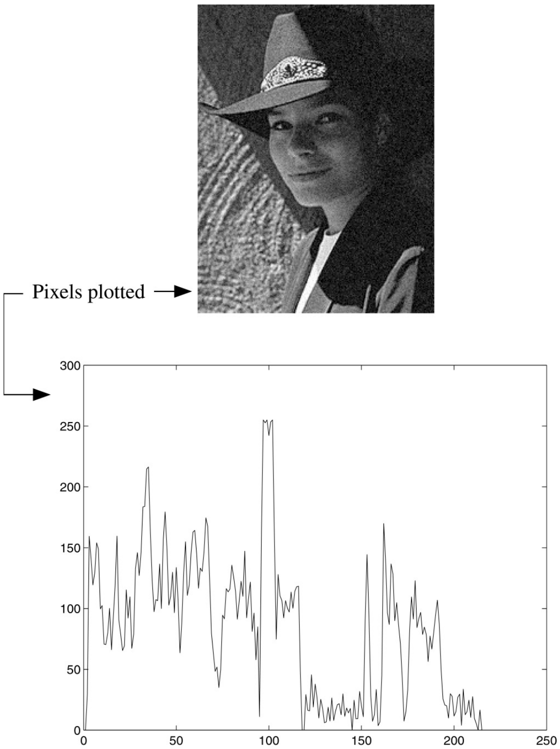
$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Properties of the gradient

- It's a vector
- Points in the direction of maximum increase of f
- Magnitude is rate of increase

How can we approximate the gradient in a discrete image?

Less than ideal edges



Steps in edge detection

Edge detection algorithms typically proceed in three or four steps:

- **Filtering**: cut down on noise
- **Enhancement**: amplify the difference between edges and non-edges
- **Detection**: use a threshold operation
- **Localization** (optional): estimate geometry of edges beyond pixels

Edge enhancement

A popular gradient magnitude computation is the **Sobel operator**:

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

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

We can then compute the magnitude of the vector (s_x, s_y).

Results of Sobel edge detection


Original


Smoothed


Sx + 128

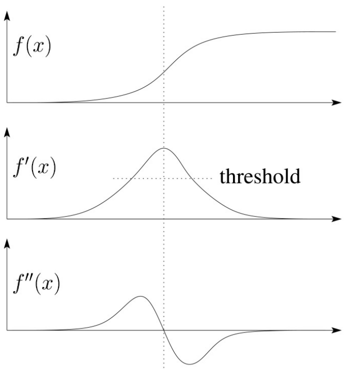

Sy + 128


Magnitude


Threshold = 64


Threshold = 128

Second derivative operators



The Sobel operator can produce thick edges. Ideally, we're looking for infinitely thin boundaries.

An alternative approach is to look for local extrema in the first derivative: places where the change in the gradient is highest.

Q: A peak in the first derivative corresponds to what in the second derivative?

Q: How might we write this as a convolution filter?

Localization with the Laplacian

An equivalent measure of the second derivative in 2D is the **Laplacian**:

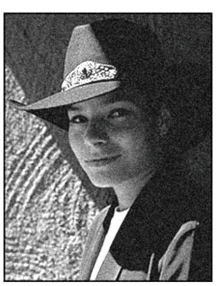
$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Using the same arguments we used to compute the gradient filters, we can derive a Laplacian filter to be:

$$\Delta^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Zero crossings of this filter correspond to positions of maximum gradient. These zero crossings can be used to localize edges.

Localization with the Laplacian



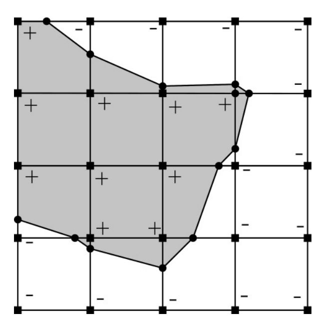
Original

Smoothed

Laplacian (+128)

Marching squares

We can convert these signed values into edge contours using a "marching squares" technique:



Sharpening with the Laplacian

Original

Laplacian (+128)

Original + Laplacian

Original - Laplacian

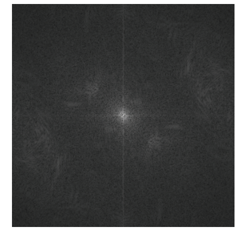
Why does the sign make a difference?

How can you write each filter that makes each bottom image?

Spectral impact of sharpening

We can look at the impact of sharpening on the Fourier spectrum:

Spatial domain | Frequency domain



$$\delta - \Delta^2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$