Edge detection

1

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

2

Edge detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

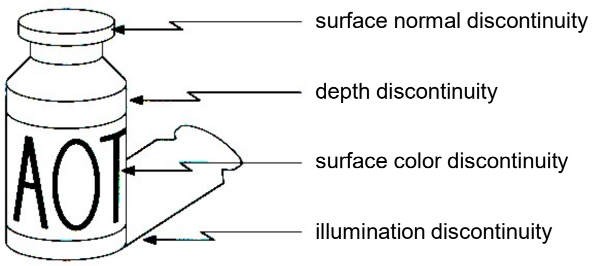
不连续的东西就是边

白
黑
白

Source: D. Lowe

Origin of edges

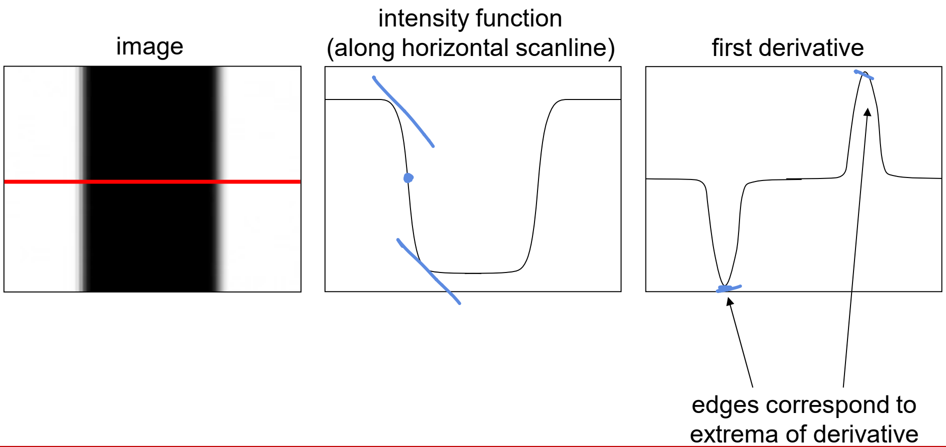
Edges are caused by a variety of factors:



Source： Steve Seitz

Characterizing edges

- An edge is a place of rapid change in the image intensity function



Source： S. Lazebnik

Derivatives with convolution

For 2D function f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

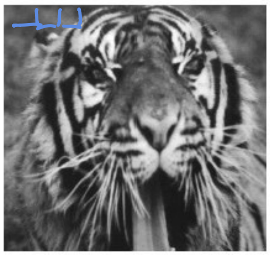
$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

ε取1，便用卷积运算求导

To implement above as convolution, what would be the associated filter?

Source: K. Grauman

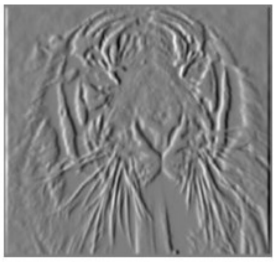
Partial derivatives of an image

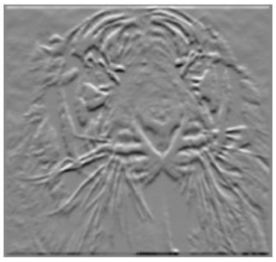


$$\frac{\partial f(x,y)}{\partial x}$$

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

-1	1





-1
1

or

1
-1

Which shows changes with respect to x?

Source： S. Lazebnik

Finite difference filters

Other approximations of derivative filters exist:

高斯平滑 边缘提取

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

$= \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \cdot (-1\ 0\ 1)$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

检测的边与它垂直

Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

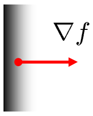
-1 1

135°方向的边

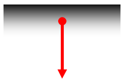
Source: K. Grauman

Image gradient

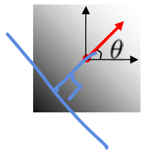
The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



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



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



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

梯度方向与边垂直

The gradient points in the direction of most rapid increase in intensity

- How does this direction relate to the direction of the edge?

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

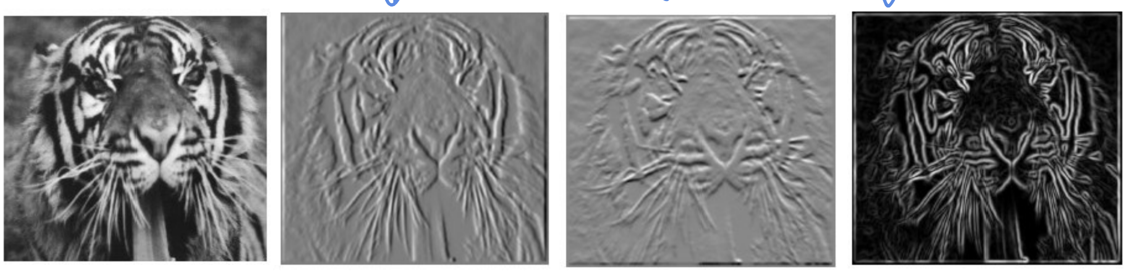
The *edge strength* is given by the gradient magnitude

幅值 $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$ 取模 => 强度 (越大,是边可能性越大)

Source: S. Seitz

Gradient Magnitude

$f * (-1, 1)$ $f * \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ 使用 $\|\nabla f\|$ 的值来输出一张图

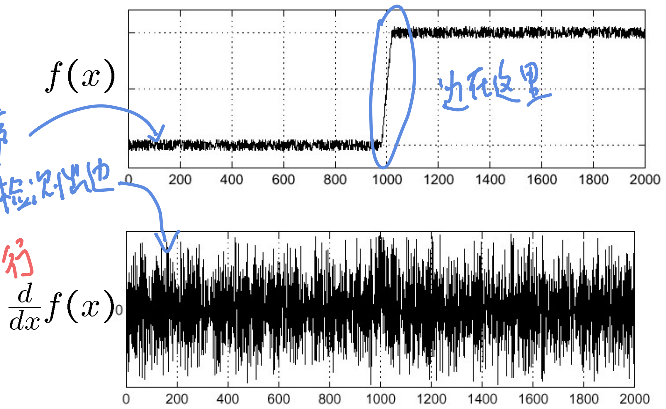


X-Derivative Y-Derivative Gradient Magnitude

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal



边在这里

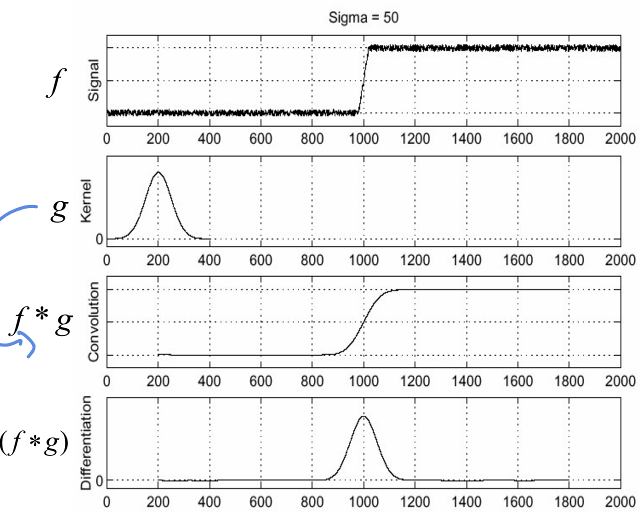
①有噪声
②也能检测出边

直接检测，不可行

Where is the edge?

Source: S. Seitz

Solution: smooth first



① 先滤波

② 再求导

- To find edges, look for peaks in $\frac{d}{dx}(f*g)$

Source: S. Seitz

↑ 以上2次卷积费时

可以合并2次卷积过程，卷积具有结合律↓

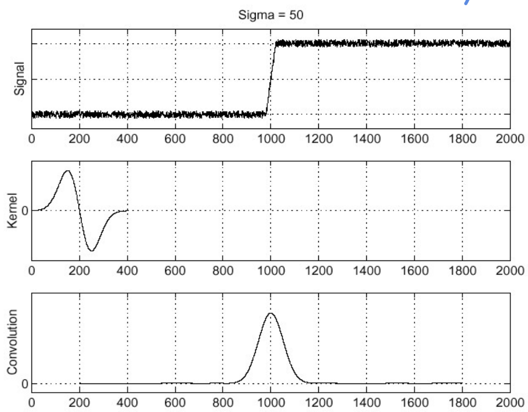
Derivative theorem of convolution

- Differentiation is convolution, and convolution is associative:
- This saves us one operation: $\frac{d}{dx}(f*g)=f*\left(\frac{d}{dx}g\right)$ 结合律

f

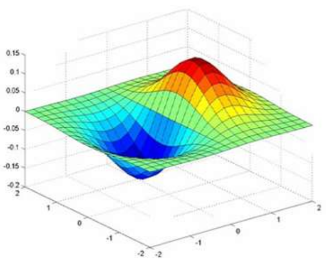
$\frac{d}{dx}g$

$f*\frac{d}{dx}g$

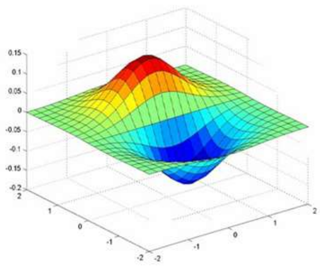


Source: S. Seitz

Derivative of Gaussian filter



x-direction

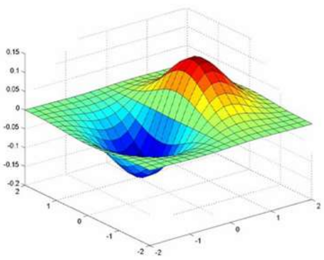


y-direction

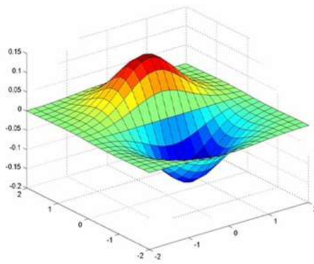
Are these filters separable?

Source：S. Lazebnik

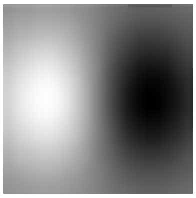
Derivative of Gaussian filter

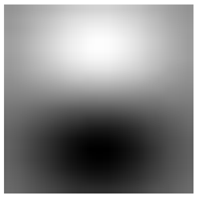


x-direction



y-direction

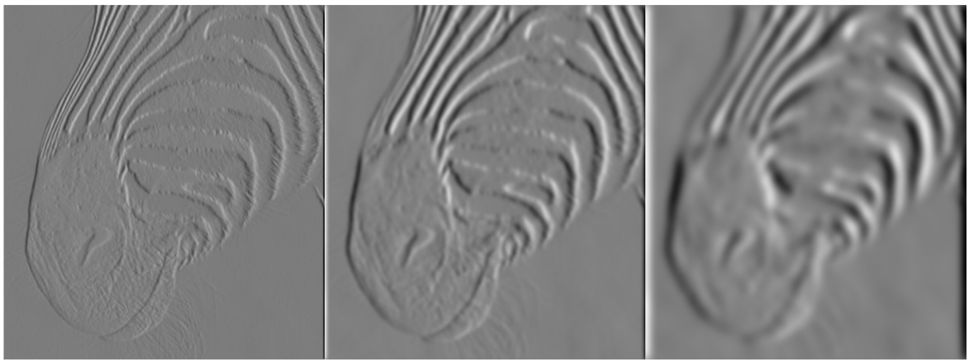




Which one finds horizontal/vertical edges?

Source： S. Lazebnik

Scale of Gaussian derivative filter



1 pixel 3 pixels 7 pixels

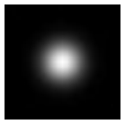
σ 越大 ⟶

Smoothed derivative removes noise, but blurs edge.
Also finds edges at different "scales"

Source: D. Forsyth

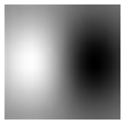
Review: Smoothing vs. derivative filters

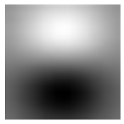
Smoothing filters



- Gaussian: remove "high-frequency" components; "low-pass" filter
- Can the values of a smoothing filter be negative?
- What should the values sum to?
  - **One:** constant regions are not affected by the filter

Derivative filters





- Derivatives of Gaussian
- Can the values of a derivative filter be negative?
- What should the values sum to?
  - **Zero:** no response in constant regions
- High absolute value at points of high contrast

Source: S. Lazebnik

The Canny edge detector 经典.有效

original image

The Canny edge detector

$\|\nabla f\|$

norm of the gradient

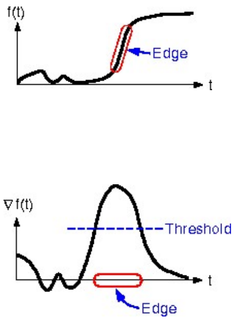
Source: S. Lazebnik

The Canny edge detector

thresholding

Source: S. Lazebnik

The Canny edge detector



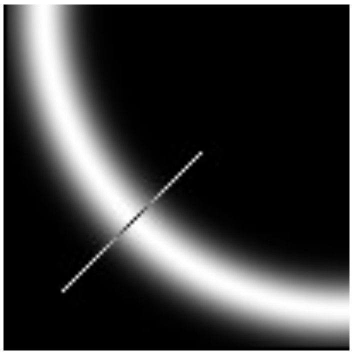
How to turn these thick regions of the gradient into curves?

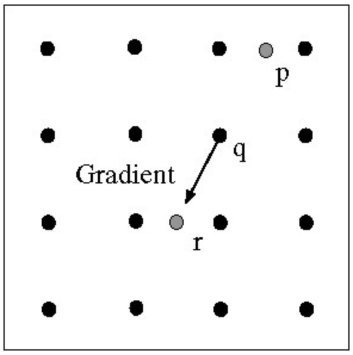
thresholding

Source: S. Lazebnik

Non-maximum suppression

非极大值抑制





Check if pixel is local maximum along gradient direction, select single max across width of the edge

Source: S. Lazebnik

The Canny edge detector

Problem: pixels along this edge didn't survive the thresholding

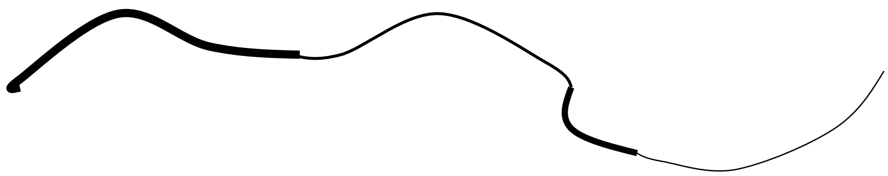
thinning

(non-maximum suppression)

Source： S. Lazebnik

Hysteresis thresholding

Use a high threshold to start edge curves, and a low threshold to continue them.



Source: S. Seitz

Hysteresis thresholding

original image

high threshold (strong edges)

low threshold (weak edges)

hysteresis threshold

Source: L. Fei-Fei

Recap: Canny edge detector

1. Filter image with derivative of Gaussian 高斯平滑
2. Find magnitude and orientation of gradient $\|\nabla f\|$ 梯度取模，找到轮廓 (threshold_1)
3. **Non-maximum suppression**: 非最大化抑制，宽边变细边
   - Thin wide "ridges" down to single pixel width
4. **Linking and thresholding** (**hysteresis**):
   - Define two thresholds: low and high 因为threshold_1在第2步时，滤掉了太细的边(不合理)
   - Use the high threshold to start edge curves and the low threshold to continue them ① 我们假设，与宽边相连的细边是要保留的

MATLAB: `edge(image, 'canny');`

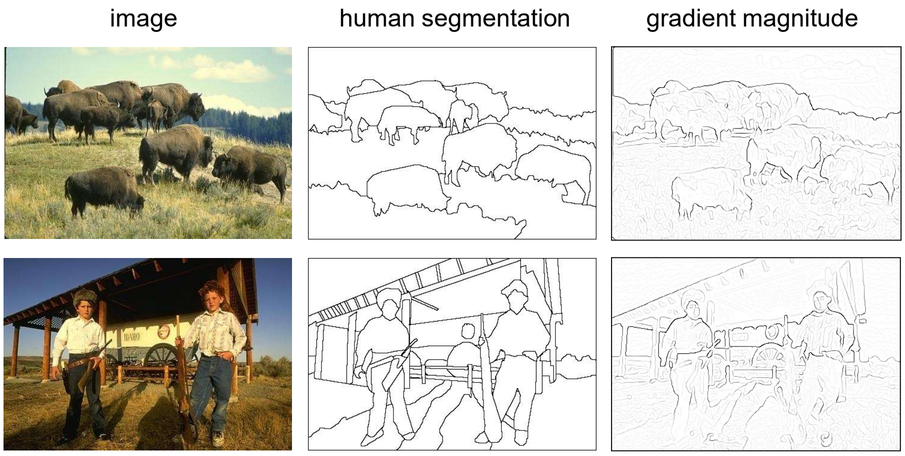
② 先用高threshold找宽边，再用低threshold找细边，

③ 留下与宽边相连的细边

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

Source： S. Lazebnik

Edge detection is just the beginning…



Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Source： S. Lazebnik