Edge Detection and Beyond

Introduction to Computer Vision

CSE 152

Lecture 9

Announcements

- Assignment 2: due today
- Assignment 3: Posted on Web page, Thursday May 6
- Midterm: Tuesday, May 4.
- Don't wait until after midterm to start assignment…

Rotation (Mistake in lec. 3 slide) was:

- About x axis:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- About y axis:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 1 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotation: Should be:

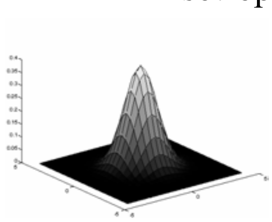
- About x axis:

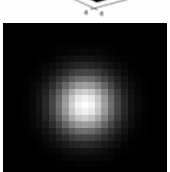
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- About y axis:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

An Isotropic Gaussian





- The picture shows a smoothing kernel proportional to

$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

(which is a reasonable model of a circularly symmetric fuzzy blob)

Median filters : Principle

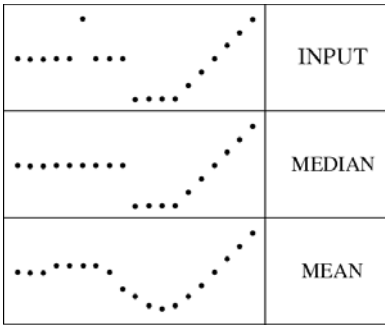
Method :

1. rank-order neighborhood intensities in a window
2. take middle value

- non-linear filter
- no new grey levels emerge...

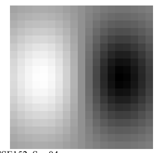
Median filters : example

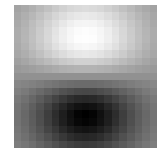
filters have width 5 :



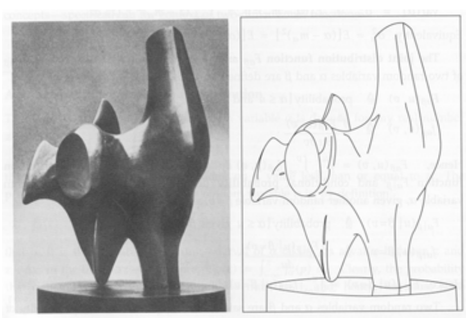
Filters are templates

- Applying a filter at some point can be seen as taking a dot-product between the image and some vector
- Filtering the image is a set of dot products
- Insight
  - filters look like the effects they are intended to find
  - filters find effects they look like





Edges



Physical causes of edges

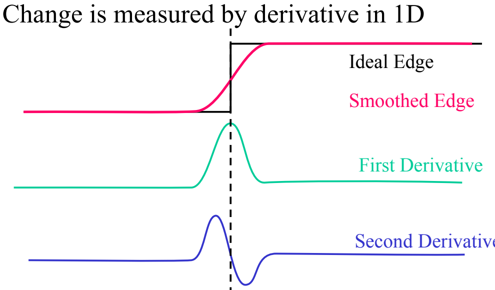
1. Object boundaries
2. Surface normal discontinuities
3. Reflectance (albedo) discontinuities
4. Lighting discontinuities

Noisy Step Edge

- Derivative is high everywhere.
- Must smooth before taking gradient.

Edge is Where Change Occurs: 1-D

- Change is measured by derivative in 1D



- Biggest change, derivative has maximum magnitude
- Or 2nd derivative is zero.

On numerical derivatives

Blackboard

Convolve with
First Derivative: [-1 0 1]
Second Derivative: [-1 2 -1]

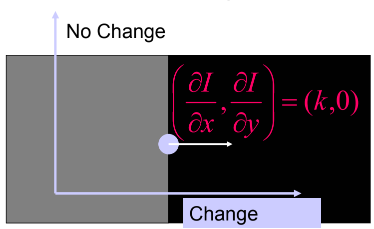
2D Edge Detection: Canny

1. Filter out noise
   – Use a 2D Gaussian Filter. $J = I * G$
2. Take a derivative
   – Compute the magnitude of the gradient:

$$\nabla J = (J_x, J_y) = \left(\frac{\partial J}{\partial x}, \frac{\partial J}{\partial y}\right) \text{ is the Gradient}$$

$$\|\nabla J\| = \sqrt{J_x^2 + J_y^2}$$

What is the gradient?

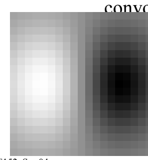


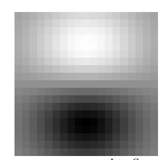
$\|\nabla J\|$ Gradient Magnitude: Edge Strength

$\frac{1}{\|\nabla J\|}\nabla J$ Gradient Direction: Edge Normal

Smoothing and Differentiation

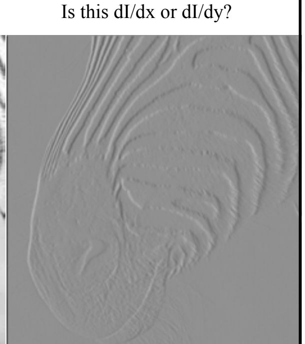
- Need two derivatives, in x and y direction.
- Filter with Gaussian and then compute Gradient, OR
- Use a derivative of Gaussian filter
  - because differentiation is convolution, and convolution is associative

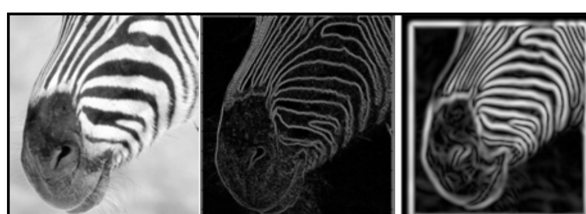




Finding derivatives

Is this dI/dx or dI/dy?



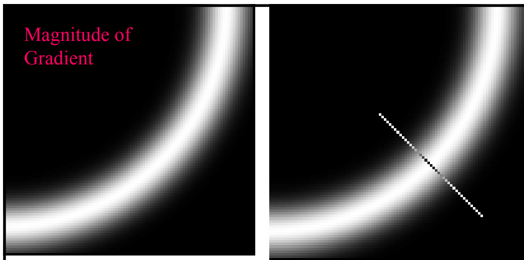


σ = 1 σ = 2

There are three major issues:

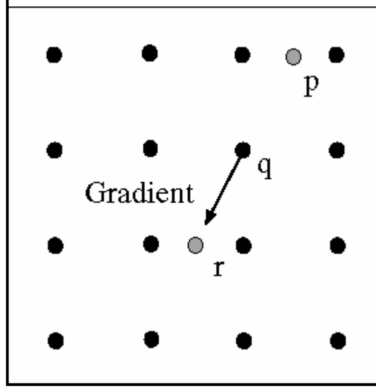
1. The gradient magnitude at different scales is different; which scale should we choose?
2. The gradient magnitude is large along thick trail; how do we identify the significant points?
3. How do we link the relevant points up into curves?

Magnitude of Gradient

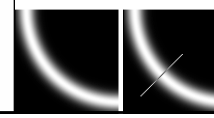


We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

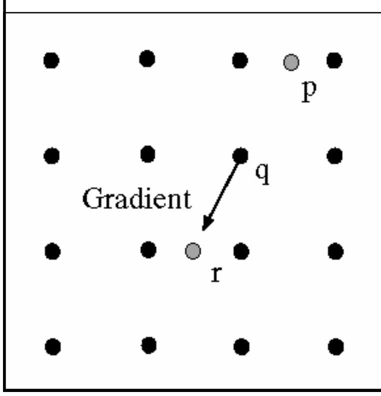
Non-maximum suppression



For every pixel in the image (e.g., q) we have an estimate of edge direction and edge normal (shown at q)



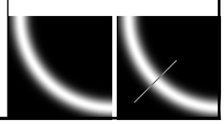
Non-maximum suppression



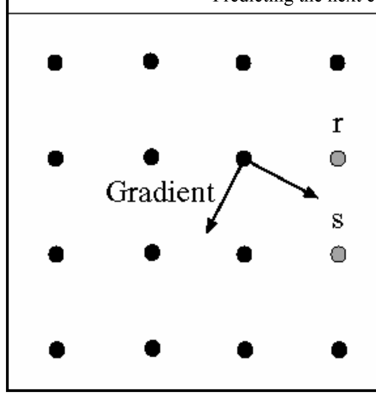
Using normal at q, find two points p and r on adjacent rows (or columns).

We have a maximum if the value is larger than those at both p and at r.

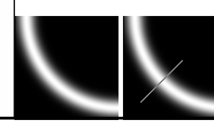
Interpolate to get values.



Non-maximum suppression
Predicting the next edge point

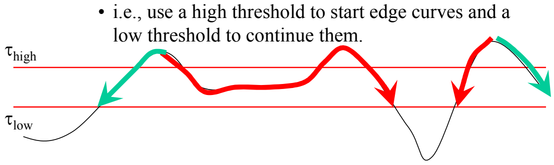


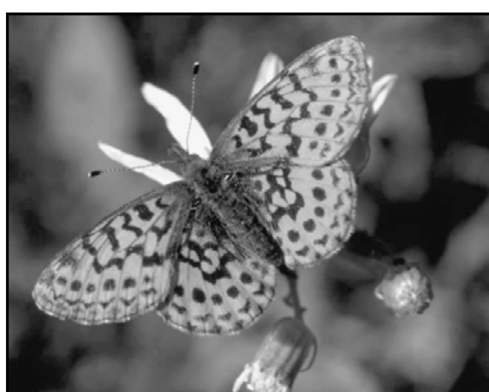
Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).

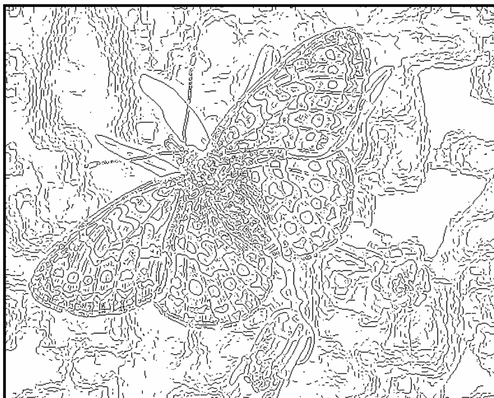


Hysteresis Tresholding

- Track edge points by starting at point where gradient magnitude > $\tau_{high.}$
- Follow edge in direction orthogonal to gradient.
- Stop when gradient magnitude < τ_{low}.
  - i.e., use a high threshold to start edge curves and a low threshold to continue them.

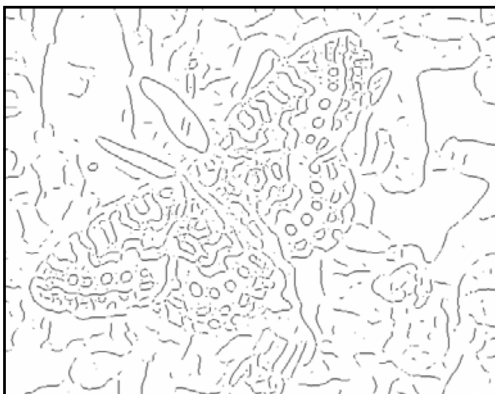






fine scale
high
threshold

coarse scale,
High threshold



coarse
scale
Low high
threshold

Why is Canny so Dominant

- Still widely used after 20 years.
  1. Theory is nice (but end result same,).
  2. Details good (magnitude of gradient, non-max suppression).
  3. Hysteresis an important heuristic.
  4. Code was distributed.

Hough Transform