Canny Edge Detector
and use of edges

Computer Vision
CSE 190-B
Lecture 7

Announcements

- Assignment 2 is posted to the web page.
- The Midterm will be Tuesday, May 6

Convolution

Image (I)  Kernel (K)

Note: Typically Kernel is relatively small in vision applications.

Convolution: \( R = K \ast I \)

Kernel size is \( m+1 \) by \( m+1 \)

\[
R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h,k)I(i-h,j-k)
\]

Linear filtering (warm-up slide)

Blurring

original  Pixel offset  Blurred (filter applied in both dimensions)

Filtered (no change)
Smoothing by Averaging

Kernel:

Smoothing with a Gaussian

• Notice “ringing”
  – apparently, a grid is superimposed
• Smoothing with an average actually doesn’t compare at all well with a defocussed lens
  – what does a point of light produce?
• A Gaussian gives a good model of a fuzzy blob

An Isotropic Gaussian

• The picture shows a smoothing kernel proportional to
  \[
  \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)
  \]

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

Finding derivatives
The scale of the smoothing filter affects derivative estimates.

Gradient Magnitude

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). There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Non-maximum suppression

At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.

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).
fine scale, high threshold

course scale, high threshold

course scale, low threshold