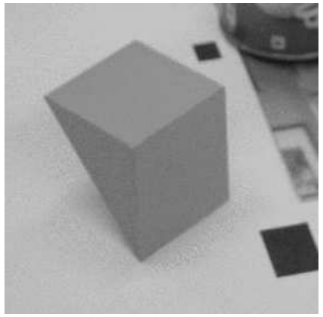
Canny Edge Detection

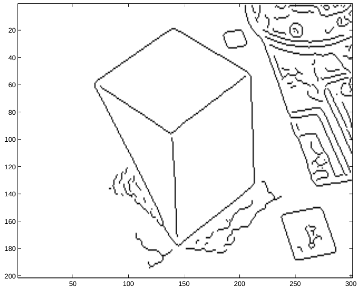
Robert B. Fisher
School of Informatics
University of Edinburgh

Edge Detector Introduction

- Edge detection: find pixels at large changes in intensity
- Much historical work on this topic in computer vision (Roberts, Sobel)
- Canny edge detector (1986) first modern edge detector and still commonly used today
- Edge detection never very accurate process: image noise, areas of low contrast, a question of scale. Humans see edges where none exist.

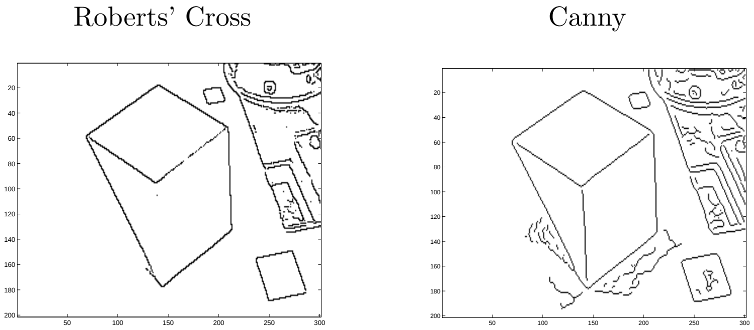
Canny Edge Detections





Finds most edges, even in low contrast
Finds lots of noisy detail too

Comparison Edge Detections



Differences in sensitivity and details, esp. in busy areas

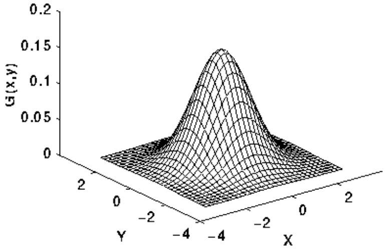
Canny Edge Detector

Four stages:

1. Gaussian smoothing: to reduce noise and smooth away small edges
2. Gradient calculation: to locate potential edge areas
3. Non-maximal suppression: to locate "best" edge positions
4. Hysteresis edge tracking: to locate reliable, but weak edges

Canny: Gaussian Smoothing

Convolve with a 2D Gaussian



$\frac{1}{273}$

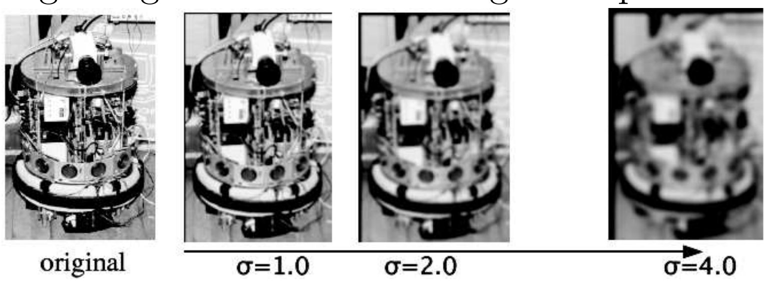
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Averages pixels with preference near center: smooths noise without too much blurring of edges

σ of Gaussian controls smoothing explicitly

$$\text{convolution } mask(r,c) = \frac{1}{2\pi\sigma} e^{-\frac{r^2+c^2}{2\sigma^2}}$$

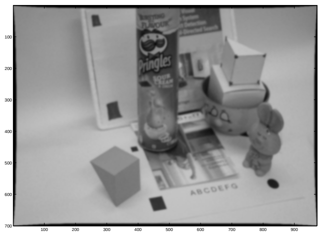
Larger σ gives more smoothing - low pass filter



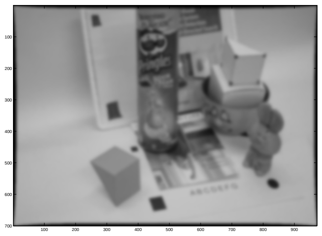
original σ=1.0 σ=2.0 σ=4.0

Gaussian Smoothing Examples

$\sigma = 5$

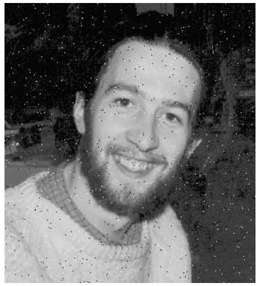


$\sigma = 20$

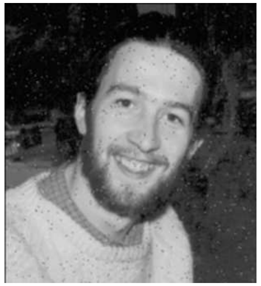


Conservative Smoothing

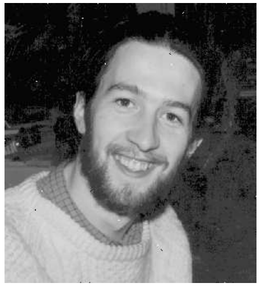
Gaussian smoothing inappropriate for salt&pepper/spot noise



Noisy image



Gauss smooth



Conservative

So do this before Canny or Gaussian Smoothing

Canny: Gradient Magnitude Calculation

$G(r, c)$ is smoothed image

Compute local derivatives in the r and c directions as $G_r(r, c)$, $G_c(r, c)$:

Edge gradient: $\nabla G(r, c) = (G_r(r, c), G_c(r, c))$

Gradient magnitude:

$$H(r,c) = \sqrt{G_r(r,c)^2 + G_c(r,c)^2}$$

$$\doteq \mid G_r(r,c) \mid + \mid G_c(r,c) \mid$$

Gradient direction

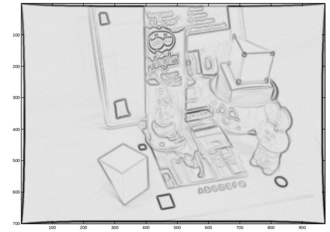
$\theta(r, c) = arctan(G_r(r, c), G_c(r, c))$

$$G_r(r,c) = \frac{\partial G}{\partial r} = lim_{h \to 0} \frac{G(r+h,c) - G(r,c)}{h}$$

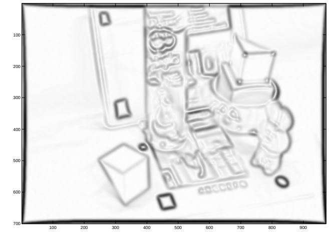
$$\doteq G(r+1,c) - G(r,c)$$

Gradient Magnitude Examples

$\sigma = 5$



$\sigma = 20$



σ controls amount of smoothing
Smaller σ gives more detail & noise
Larger σ gives less detail & noise

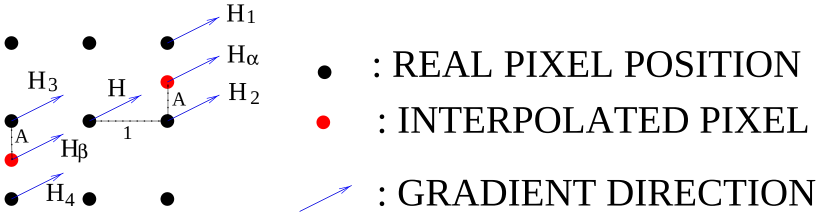
Canny: Non-maximal Suppression

Where exactly is the edge? peak of gradient

Suppress lower gradient magnitude values: need to check **ACROSS** gradient

0	0	3	12	4	0
0	0	6	10	2	0
0	2	8	7	1	0
0	3	11	4	0	0

Estimate gradient magnitudes using gradient direction:



● : REAL PIXEL POSITION

● : INTERPOLATED PIXEL

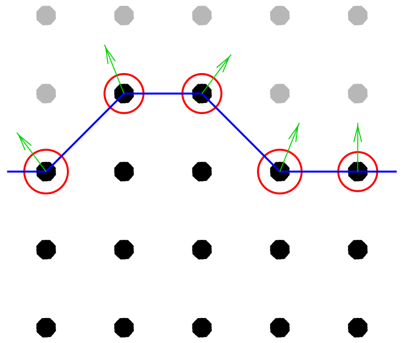
↗ : GRADIENT DIRECTION

$$A = \frac{|G_r|}{|G_c|}$$
$$H_\alpha = AH_1 + (1-A)H_2$$
$$H_\beta = AH_4 + (1-A)H_3$$

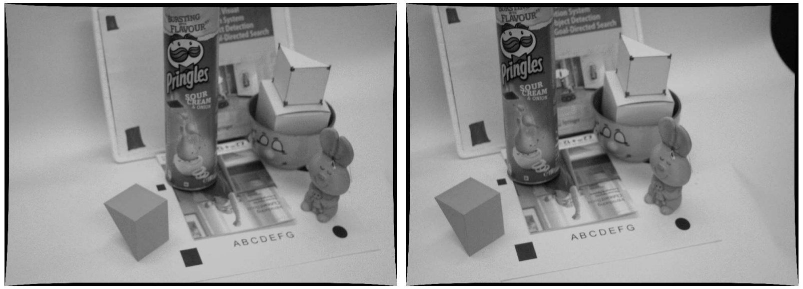
Suppress (set to 0) if $H < H_\alpha$ OR $H < H_\beta$

Canny: Hysteresis Tracking

Start edges at certainty: $H > \tau_{start}$
Reduce requirements at connected edges to get weaker edges: $H > \tau_{continue}$

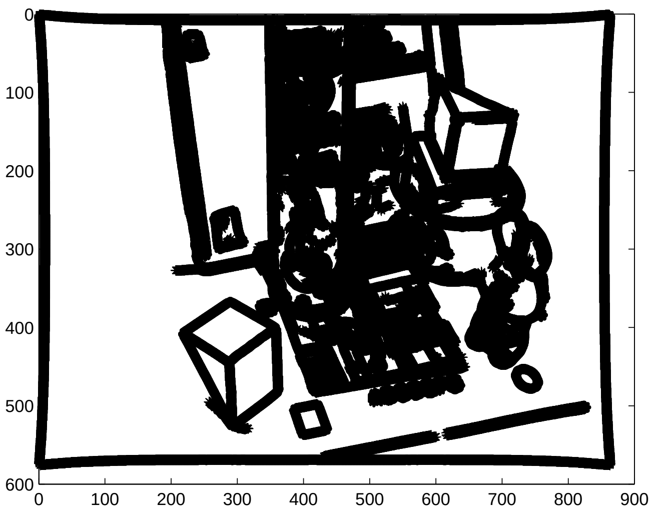


Stereo Canny Edges

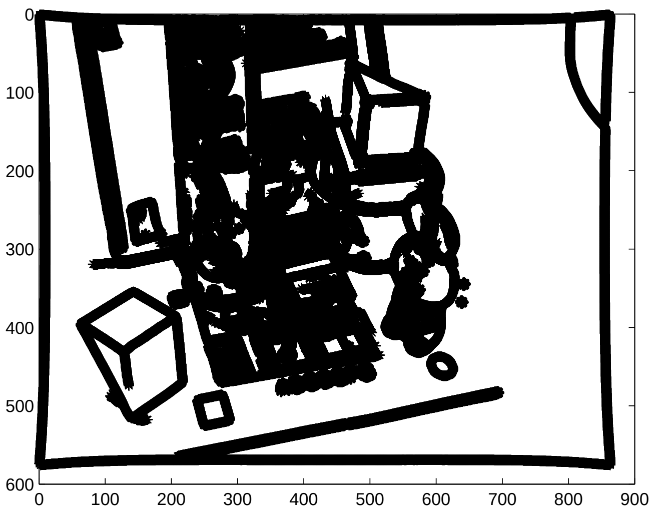


Matlab has Canny: `edge(left,'canny',[],4);`

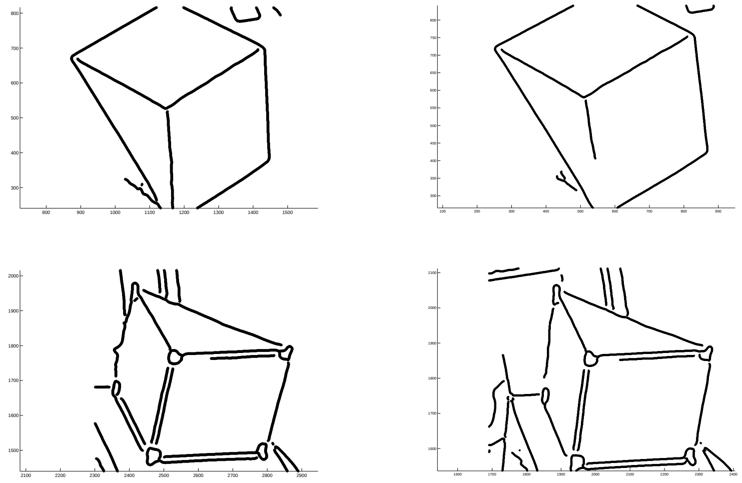
Stereo Left Canny Edges



Stereo Right Canny Edges



Zoom on Wedge Edges



What We Have Learned

- Capabilities of edge detectors
- Noise and detail reduction by smoothing
- Canny algorithm details
- Can find too many edges