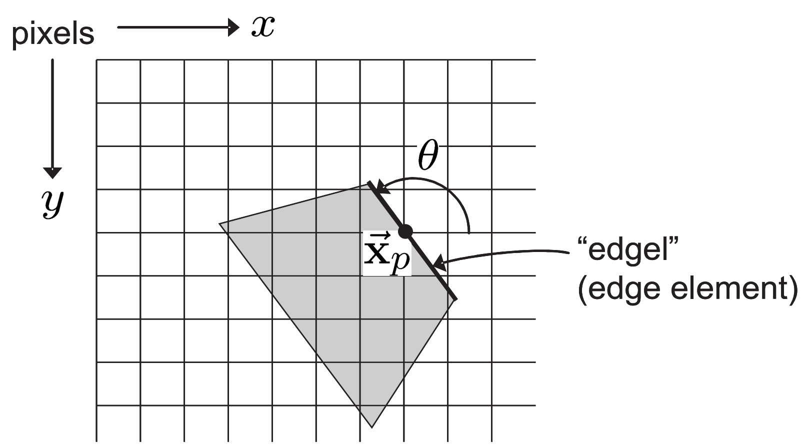
Edge Detection

Goal: Detection and Localization of Image Edges.

Motivation:

- Significant, often sharp, contrast variations in images caused by illumination, surface markings (albedo), and surface boundaries. These are useful for scene interpretation.
- **Edgels (edge elements):** significant local variations in image brightness, characterized by the position $\vec{\mathbf{x}}_p$ and the orientation θ of the brightness variation. (Usually $\theta \bmod \pi$ is sufficient.)



- **Edges:** sequence of edgels forming smooth curves

Two Problems:

1. estimating edgels
2. grouping edgels into edges

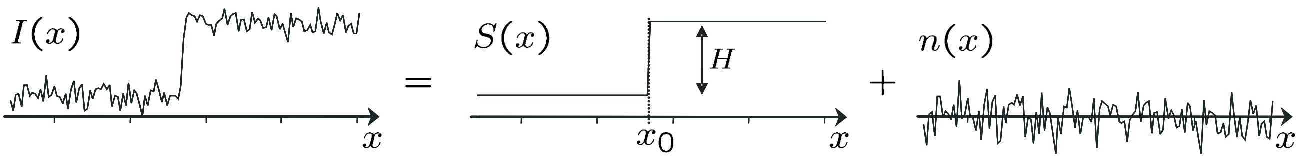
Readings: Chapter 8 of the text.
Matlab Tutorials: cannyTutorial.m

1D Ideal Step Edges

Assume an ideal step edge corrupted by additive Gaussian noise:

$$I(x) = S(x) + n(x)\ .$$

Let the signal S have a step edge of height H at location x_0, and let the noise at each pixel be Gaussian, independent and identically distributed (IID).



Gaussian IID Noise:

$$n(x) \sim N(0, \sigma_n^2)\ , \qquad p_n(n; 0, \sigma_n^2) \ = \ \frac{1}{\sqrt{2\pi}\sigma_n}\, e^{-n^2/\sigma_n^2}$$

Expectation:

$$\text{mean: E}[n] \ \equiv \ \int n\, p_n(n)\, dn = 0$$

$$\text{variance: E}[n^2] \ \equiv \ \int n^2\, p_n(n)\, dn = \sigma_n^2$$

Independence:

$$\text{E}[n(x_1)\, n(x_2)] = \begin{cases} 0 & \text{when } x_1 \neq x_2 \\ \sigma_n^2 & otherwise \end{cases}$$

Remark: Violations of the main assumptions, i.e., the idealized step edge and additive Gaussian noise, are commonplace.

Optimal Linear Filter

What is the optimal linear filter for the detection and localization of a step edge in an image?

Assume a linear filter, with impulse response $f(x)$:

$$\begin{aligned} r(x) \;=\; f(x) * I(x) \;&=\; f(x) * S(x) \;+\; f(x) * n(x) \\ &=\; \quad r_S(x) \quad\;\; + \quad\;\; r_n(x) \end{aligned}$$

So the response is the sum of responses to the signal and the noise.

The mean and variance of the response to noise $r_n(x)$,

$$r_n(x) \;=\; \sum_{k=-K}^{K} f(-k)\, n(x+k) \;,$$

where K is the radius of filter support, are easily shown to be

$$\begin{aligned} \mathrm{E}[r_n(x)] \;&=\; \sum_k f(-k)\, \mathrm{E}[n(x+k)] \;=\; 0 \\ \mathrm{E}[r_n^2(x)] \;&=\; \sum_k \sum_l f(-l)\, f(-k)\, \mathrm{E}[n(x+k)n(x+l)] \;=\; \sigma_n^2 \sum_k f^2(k) \end{aligned}$$

The response *Signal-to-Noise Ratio* (SNR) at the step location x_0 is:

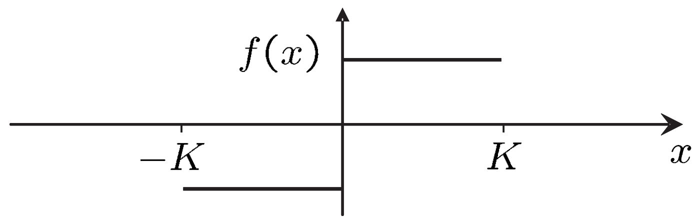
$$SNR \;=\; \frac{|(f * S)(x_0)|}{\sigma_n \sqrt{\sum_k f^2(k)}}$$

Next, consider criteria for optimal detection and localization ...

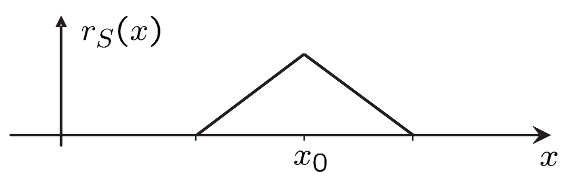
Criteria for Optimal Filters

Criterion 1: *Good Detection.* Choose the filter to maximize the the SNR of the response at the edge location, subject to constraint that the responses to constant sigals are zero.

For a filter with a support radius of K pixels, the optimal filter is a matched filter, i.e., a difference of square box functions:



Response to ideal step:



Explanation:
Assume, with out loss of generality that $\sum f^2(x) = 1$, and to ensure zero DC response, $\sum f(x) = 0$.
Then, to maximize the SNR, we simply maximize the inner product of $S(x)$ and the impulse response, reflected and centered at the step edge location, i.e., $f(x_0 - x)$.

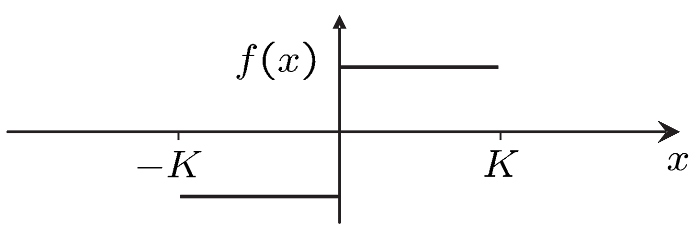
Criteria for Optimal Filters (cont)

Criterion 2: *Good Localization.* Let $\{x_l^*\}_{l=1}^L$ be the local maxima in response magnitude $|r(x)|$. Choose the filter to minimze the root mean squared error between the *true edge location* and the *closest peak* in $|r|$; i.e., minmize

$$LOC \;=\; \frac{1}{\sqrt{\mathrm{E}[\,\min_k |x_l^* - x_0|^2\,]}}$$

Caveat: for an optimal filter this does not mean that the closest peak should be the most significant peak, or even readily identifiable.

Result: Maximizing the product, $SNR \cdot LOC$, over all filters with support radius K produces the same matched filter already found by maximizing SNR alone.



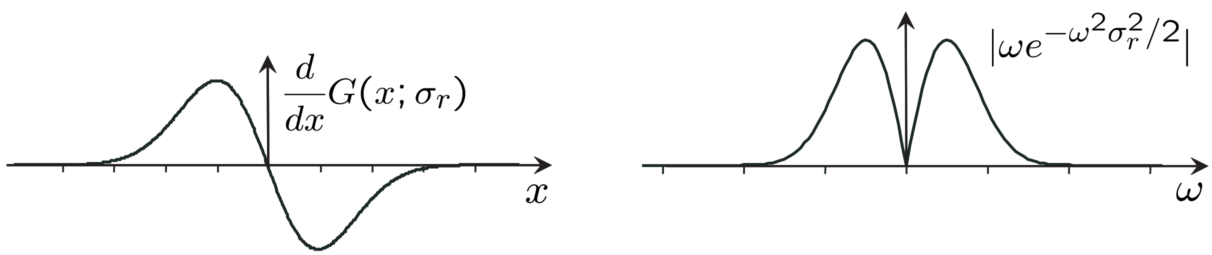
Criteria for Optimal Filters (cont)

Criterion 3: *Sparse Peaks.* Maximize $SNR \cdot LOC$, subject to the constraint that peaks in $|r(x)|$ be as far apart, on average, as a manually selected constant, $xPeak$:

$$\mathrm{E}[\,|x^*_{k+1} - x^*_k|\,] \;=\; xPeak$$

When $xPeak$ is small, $f(x)$ is similar to the matched filter above. But for $xPeak$ larger (e.g., $xPeak \approx K/2$) then the optimal filter is well approximated by a derivative of a Gaussian:

$$f(x) \;\approx\; \frac{dG(x;\sigma_r)}{dx} \;=\; \frac{-x}{\sqrt{2\pi}\sigma_r^3}\, e^{-\frac{x^2}{2\sigma_r^2}}\;, \text{ with } \mathcal{F}\left[\frac{dG(x;\sigma_r)}{dx}\right] \;=\; i\,\omega\, e^{-\frac{\omega^2\sigma_r^2}{2}}$$



Conclusion:

Sparsity of edge detector responses is a critical design criteria, encouraging a smooth envelope, and thereby less power at high frequencies. The lower the frequency of the pass-band, the sparser the response peaks.

There is a one parameter family of optimal filters, varying in the width of filter support, σ_r. Detection (SNR) improves and localization (LOC) degrades as σ_r increases.

Multiscale Edge Features

Multiple scales are also important to consider because salient edges occur at multiple scales:

1) Objects and their parts occur at multiple scales:

2) Cast shadows cause edges to occur at many scales:

3) Objects may project into the image at different scales:

2D Edge Detection

The corresponding 2D edge detector is based on the magnitude of the directional derivative of the image in the direction normal to the edge.

Let the unit normal to the edge orientation be $\vec{\mathbf{n}} = (\cos\theta, \sin\theta)$. The directional derivative of a 2D isotropic Gaussian, $G(\vec{\mathbf{x}};\, \sigma^2) \equiv \frac{1}{2\pi\sigma^2}\, e^{\frac{-(x^2+y^2)}{2\sigma^2}}$ is given by

$$\begin{aligned} \frac{\partial}{\partial \vec{\mathbf{n}}} G(\vec{\mathbf{x}};\, \sigma^2) \;&=\; \nabla G(\vec{\mathbf{x}};\, \sigma^2) \cdot \vec{\mathbf{n}} \\ &=\; \cos\theta\, G_x(\vec{\mathbf{x}};\, \sigma^2) \;+\; \sin\theta\, G_y(\vec{\mathbf{x}};\, \sigma^2) \end{aligned}$$

where $G_x \equiv \frac{\partial G}{\partial x}$ and $G_y \equiv \frac{\partial G}{\partial y}$.

The direction of steepest ascent/descent at each pixel is given by the direction of the image gradient:

$$\vec{\mathbf{R}}(\vec{\mathbf{x}}) \;=\; \nabla G(\vec{\mathbf{x}};\, \sigma^2) \,*\, I(\vec{\mathbf{x}})$$

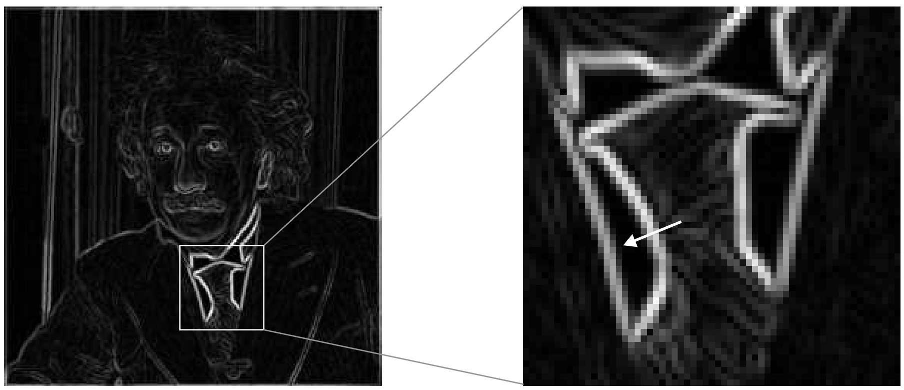
The unit edge normal is therefore given by

$$\vec{\mathbf{n}}(\vec{\mathbf{x}}) \;=\; \frac{\vec{\mathbf{R}}(\vec{\mathbf{x}})}{|\vec{\mathbf{R}}(\vec{\mathbf{x}})|}$$

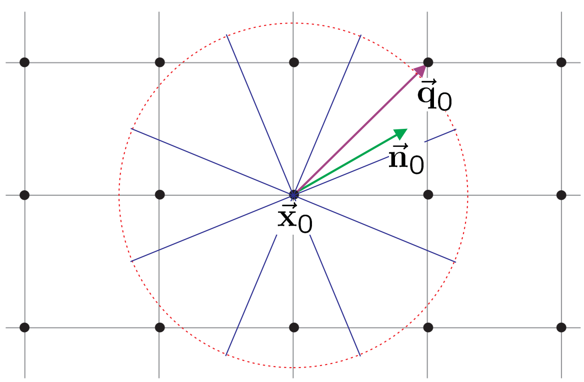
Edge Detection: Search for maxima in the directional image derivative in the direction $\vec{\mathbf{n}}(\vec{\mathbf{x}})$.

2D Edge Detection (cont)

Search for local maxima of gradient magnitude $S(\vec{\mathbf{x}}) = |\vec{\mathbf{R}}(\vec{\mathbf{x}})|$, in the direction normal to local edge, $\vec{\mathbf{n}}(\vec{\mathbf{x}})$, suppressing all responses except for local maxima (called non-maximum suppression).



In practice, the search for local maxima of $S(\vec{\mathbf{x}})$ takes place on the discrete sampling grid. Given $\vec{\mathbf{x}}_0$, with normal $\vec{\mathbf{n}}_0$, compare $S(\vec{\mathbf{x}}_0)$ to nearby pixels closest to the direction of $\pm\vec{\mathbf{n}}_0$, e.g., pixels at $\vec{\mathbf{x}} \pm \vec{\mathbf{q}}_0$, where $\vec{\mathbf{q}}_0$ is $\frac{1}{2\sin(\pi/8)}\vec{\mathbf{n}}_0$ rounded to the nearest integer.



The dotted (red) circle depicts points $\vec{\mathbf{x}} \pm \frac{1}{2\sin(\pi/8)}\vec{\mathbf{n}}_0$. Normal directions between (blue) radial lines all map to the same neighbour of $\vec{\mathbf{x}}_0$.

Canny Edge Detection

Algorithm:

1. Convolve with gradient filters (at multiple scales)

$$\vec{\mathbf{R}}(\vec{\mathbf{x}}) \;\equiv\; (R_x(\vec{\mathbf{x}}),\, R_y(\vec{\mathbf{x}})\,) \;=\; \nabla G(\vec{\mathbf{x}};\, \sigma^2) \,*\, I(\vec{\mathbf{x}})\;.$$

2. Compute response magnitude, $S(\vec{\mathbf{x}}) = \sqrt{R_x^2(\vec{\mathbf{x}}) + R_y^2(\vec{\mathbf{x}})}$.

3. Compute local edge orientation (represented by unit normal):

$$\vec{\mathbf{n}}(\vec{\mathbf{x}}) \;=\; \begin{cases} (R_x(\vec{\mathbf{x}}),\, R_y(\vec{\mathbf{x}}))/S(\vec{\mathbf{x}}) & \text{if } \; S(\vec{\mathbf{x}}) > threshold \\ 0 & \text{otherwise} \end{cases}$$

4. Peak detection (non-maximum suppression along edge normal)

5. Non-maximum suppression through scale, and hysteresis thresholding along edges (see Canny (1986) for details).

Implementation Remarks:

Separability: Partial derivatives of an isotropic Gaussian:

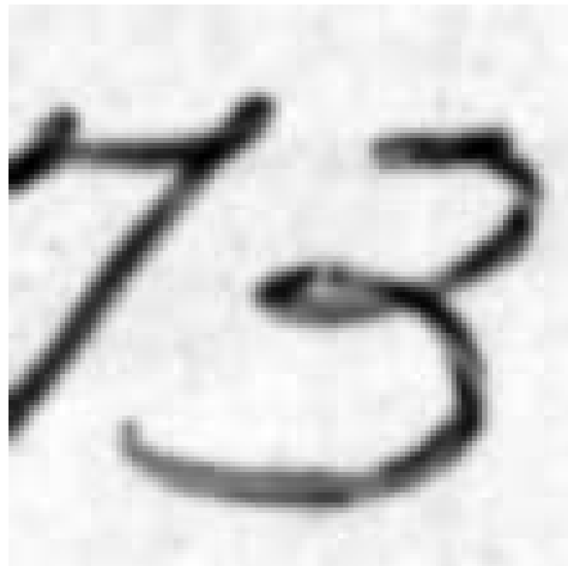
$$\frac{\partial}{\partial x} G(\vec{\mathbf{x}};\, \sigma^2) \;=\; -\,\frac{x}{\sigma^2}\, G(x;\, \sigma^2)\, G(y;\, \sigma^2)\;.$$

Filter Support: In practice, it's good to sample the impulse response so that the support radius $K \geq 3\sigma_r$. Common values for K are 7, 9, and 11 (i.e., for $\sigma \approx 1, 4/3$, and $5/3$).

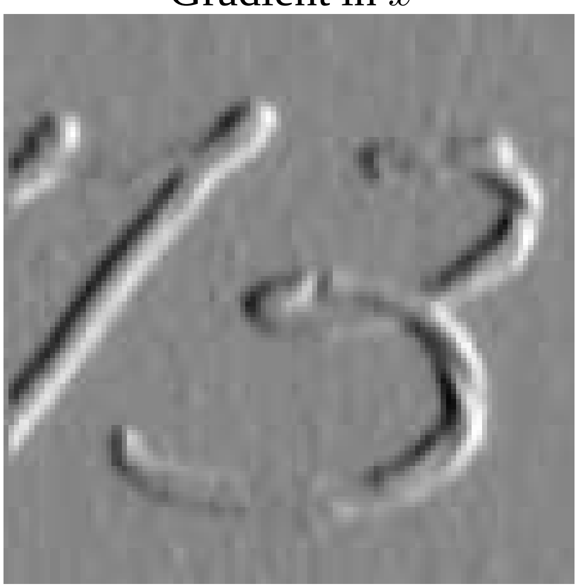
Filtering with Derivatives of Gaussians

Image `three.pgm`

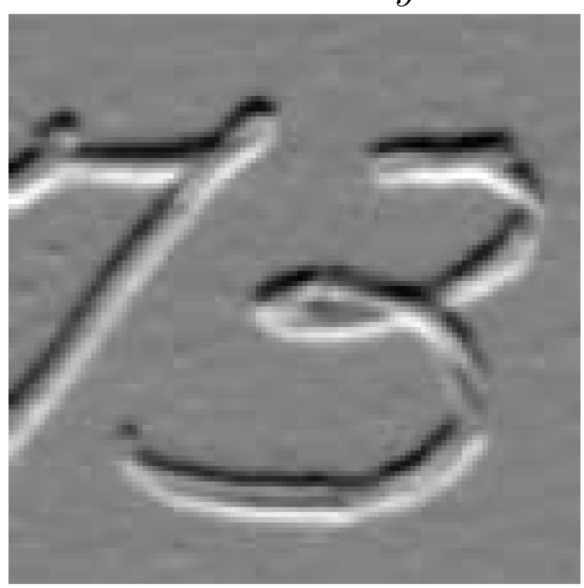
Gaussian Blur $\sigma = 1.0$



Gradient in x



Gradient in y



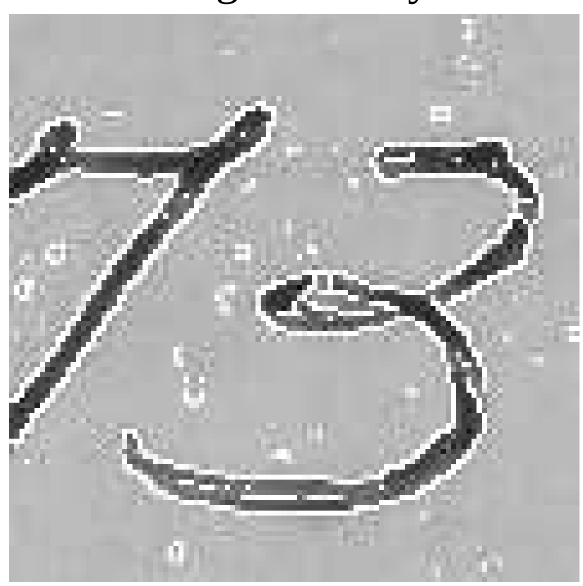
Canny Edgel Measurement

Gradient Strength

Gradient Orientations

Canny Edgels

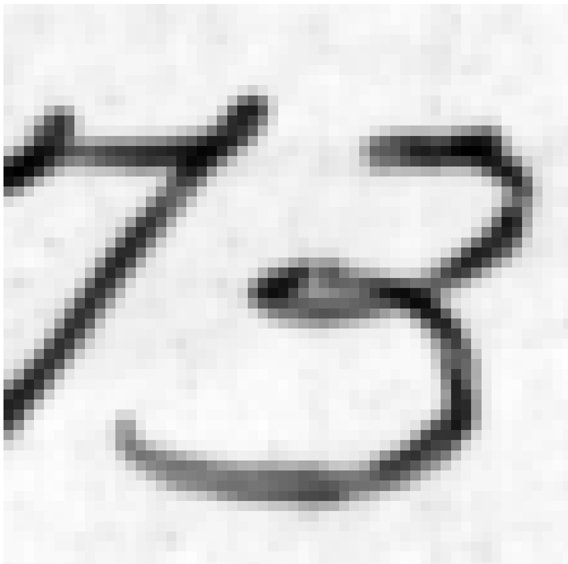
Edgel Overlay



Colour gives gradient direction (red – $0°$; blue – $90°$; green – $270°$)

Gaussian Pyramid Filtering (Subsample $\times$ 2)

Blurred and Down-Sampled ($\times 2$)



Gaussian Blur σ = 1.0

Gradient Magnitude (dec $\times 2$)

Gradient Orientations

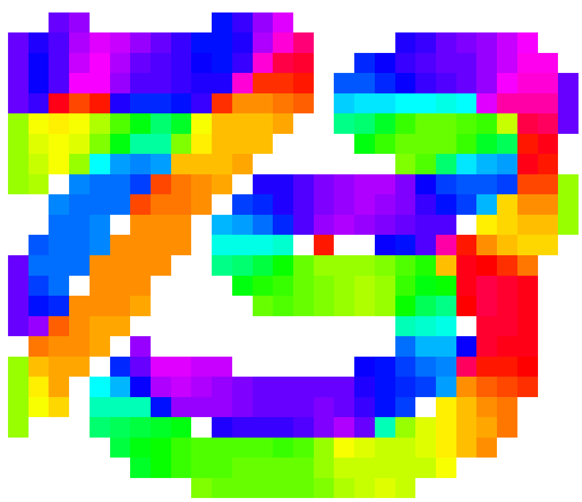
Gaussian Pyramid Filtering (Subsample $\times$ 4)

Blurred and Down-Sampled ($\times 4$)

Gaussian Blur $\sigma = 1.0$

Gradient Magnitude (dec $\times 4$)

Gradient Orientations

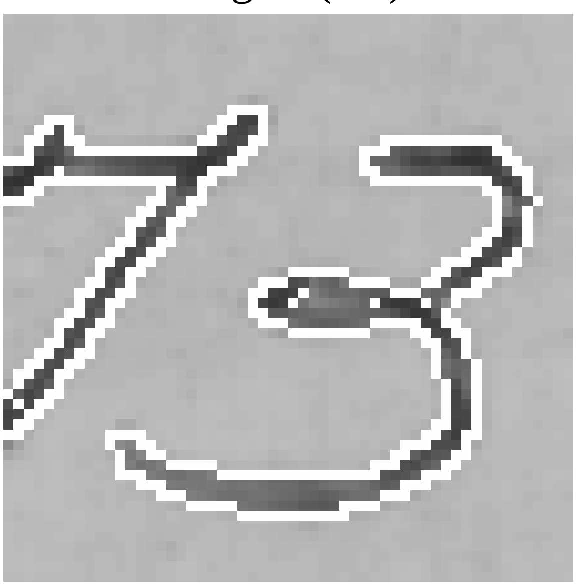


Multiscale Canny Edgels

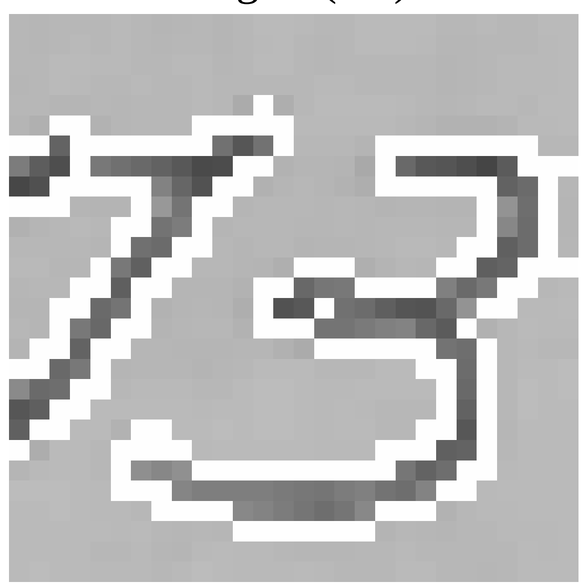
Image `three.pgm`

Edgels (x 1)

Edgels (x 2)

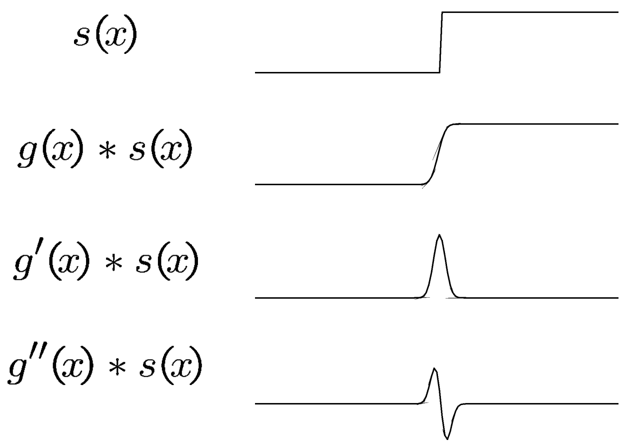


Edgels (x 4)



Subpixel Localization

Maximal responses in the first derivative will coincide with zero-crossings of the second derivative for a smoothed step edge:



Often zero-crossings are more easily localized to subpixel accuracy because linear models can be used to approximate (fit) responses near the zero-crossing. The zero-crossing is easy to find from the linear fit.

So, given a local maxima and its normal, $\vec{\mathbf{n}} = (\cos\theta, \sin\theta)$, we can compute the 2^{nd}-order directional derivative in the local region:

$$\frac{\partial^2}{\partial \vec{\mathbf{n}}^2} G(\vec{\mathbf{x}}) * I(\vec{\mathbf{x}}) = \cos^2\theta\, G_{xx}(\vec{\mathbf{x}}) * I(\vec{\mathbf{x}}) + 2\cos\theta\sin\theta\, G_{xy}(\vec{\mathbf{x}}) * I(\vec{\mathbf{x}}) + \sin^2\theta\, G_{yy}(\vec{\mathbf{x}}) * I(\vec{\mathbf{x}})\,. \tag{1}$$

Note that the three filters, $G_{xx} \equiv \frac{\partial^2 G}{\partial x^2}$, $G_{xy} \equiv \frac{\partial^2 G}{\partial x \partial y}$ and $G_{yy} \equiv \frac{\partial^2 G}{\partial y^2}$ can be applied to the image independent of $\vec{\mathbf{n}}$.

Steerable Filters

The first- and second-order Gaussian derivatives used in the Canny edge detector are orientation-tuned band-pass filters. Such filters will, in general, produce a better signal to noise ratio than isotropic band-pass filters when applied to 1D structure (like edges).

Original Laplacian Directional derivative

When using orientation-tuned filters, especially in the context of an image transform/representation, one can ask how many orientation-tuned filters are needed? The answer is that, if we have a complete basis for the filters, then we should be able to represent the response of any orientation-tuned filter in terms of the basis filters. Such basis filters are often called steerable filters [Freeman and Adelson, 1991], and the gradient of an isotropic Gaussian is an example.

Let $f(x, y)$ be the impulse response for a filter tuned to orientation 0.
Let $f^\alpha(x, y)$ be a rotated version of $f(x, y)$ to orientation α.
Let $\{f^{\theta_j}(x,\, y)\}_{j=1\ldots B}$ be a basis for $f(x, y)$ under rotation, i.e.

$$f^\alpha(x,y) \;=\; \sum_{j=1}^{B} a_j(\alpha)\, f^{\theta_j}(x,y)$$

Here, $a_j(\alpha)$ are called steering (interpolating) functions.

Distributivity of convolution over addition then yields:

$$\begin{aligned} f^\alpha(x,\, y) * I(x,\, y) \;&=\; \left(\sum_{j=1}^{B} a_j(\alpha)\, f^{\theta_j}(x,\, y)\right) * I(x,\, y) \\ &=\; \sum_{j=1}^{B} a_j(\alpha)\, \left(f^{\theta_j}(x,\, y) * I(x,\, y)\right) \end{aligned}$$

i.e. convolve basis functions, then synthesize oriented filter output

Steerable Filters – Directional Derivatives

Consider the directional first derivative of a Gaussian, $g(x, y) \;=\; e^{-(x^2+y^2)/2}$.
The first derivatives in the horizontal and vertical directions:

$$g_x(x,\, y) = \,-x\, e^{-(x^2+y^2)/2} \quad , \quad g_y(x,\, y) = \,-y\, e^{-(x^2+y^2)/2}$$

General derivative in direction $\vec{\mathbf{s}} = (s_1,\, s_2) = (\cos(\theta), \sin(\theta))$

$$g_s(x, y) \;=\; (s_1,\, s_2) \cdot (g_x(x, y),\, g_y(x, y))$$

So output of filter tuned to orientation θ is given by

$$g_s(x, y) \,*\, I(x, y) \;=\; (s_1\, g_x + s_2\, g_y) \,*\, I(x,\, y) \;=\; (s_1,\, s_2) \cdot (g_x * I,\; g_y * I)$$

Explanation: In polar coordinates, the horizontal Gaussian derivative is:

$$g_x(r, \theta) = -r\, e^{-r^2/2} \cos(\theta)$$

which is a polar separable product of $\cos(\theta)$ and $-r\, e^{-r^2/2}$. In polar coordinates it is easy to see that a rotation by $\pi/2$ is given by

$$\begin{aligned} g_x(r, \theta - \pi/2) &= -re^{-r^2/2} \cos(\theta - \pi/2) \\ &= -re^{-r^2/2} \sin(\theta) \\ &= g_y(r, \theta) \end{aligned}$$

And in general;, a rotation by θ_0, i.e., to direction $\vec{\mathbf{s}} = (\cos(\theta_0), \sin(\theta_0))$

$$\begin{aligned} g_x(r, \theta - \theta_0) &= -re^{-r^2/2} \cos(\theta - \theta_0) \\ &= -re^{-r^2/2} \left[\cos(\theta) \cos(\theta_0) + \sin(\theta) \sin(\theta_0)\right] \\ &= \cos(\theta_0)\, g_x \,+\, \sin(\theta_0)\, g_y \end{aligned}$$

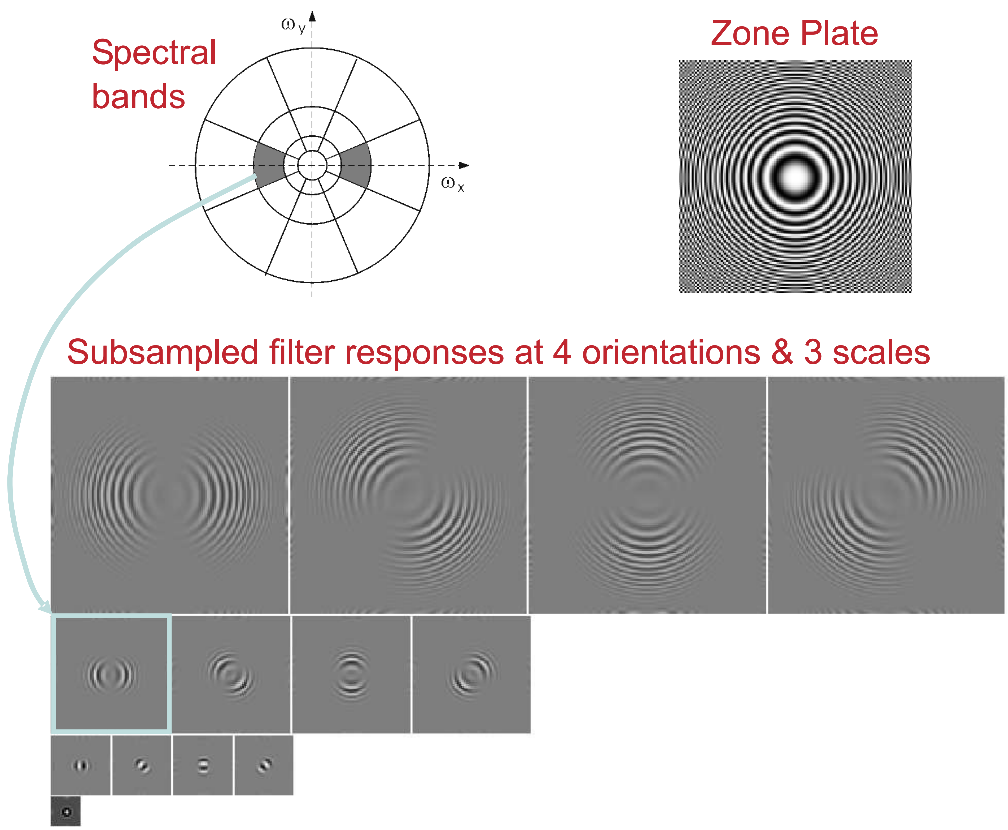
The same ideas also work with higher-order derivatives, separable polynomial functions, and polar separable functions having a limited numbers of angular frequency components. For example,

$$\frac{\partial^2}{\partial s^2} g(x, y) \;=\; \sum_{k=0}^{2} \frac{2!}{k!\,(2-k)!}\, s_x^k\, s_y^{2-k}\, \frac{\partial^2 g(x, y)}{\partial x^k\, \partial y^{2-k}}\;.$$

This is precisely the form of the second directional derivative given in Eqn (1), for direction $\vec{\mathbf{s}}$.

Steerable Pyramid

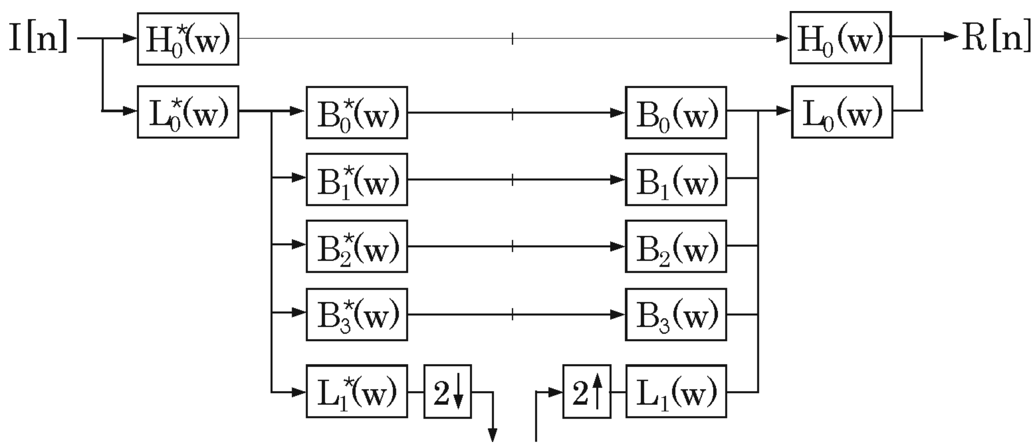
A Steerable pyramid is much like the Laplacian pyramid, except that each band-pass level is further decomposed into a set of oriented filters (a steerable basis).



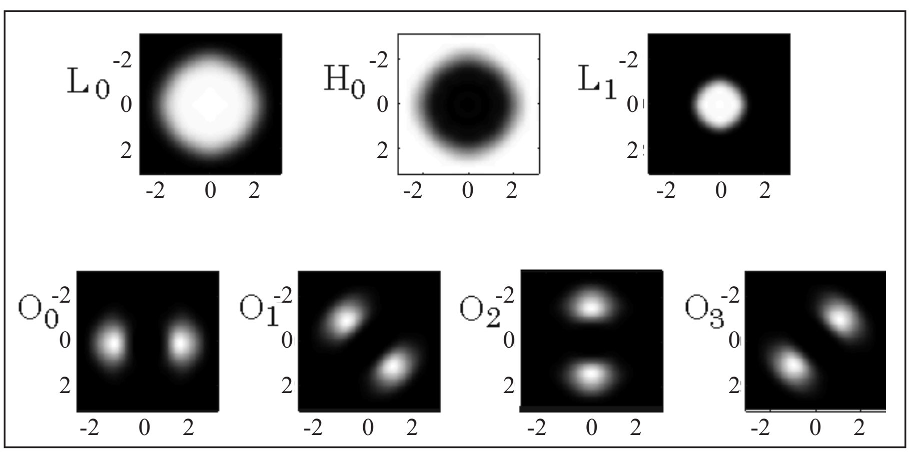
The pass-bands of the different channels in a steerable pyramid are shown in the top-left. One particular pass-band for a real-valued steerable filter is depicted by the two shaded regions. Below the rows of images show the responses of the different orientation tuned filters at each scale when applied to the zone plate (top-right). The filter corresponding to the shaded pass-band in the top-left has its response shown in the left-most image in the second row.

Steerable Pyramid (cont)

Analysis / Synthesis diagram for steerable pyramid:



The corresponding amplitude spectra for the different channels are shown below. The channels denotes O_j are spectra resulting from the cascade of L_0 and B_j.



Edge-Based Image Editing

Existing edge detectors are sufficient for a wide variety of applications, such as image editing, tracking, and simple recognition.

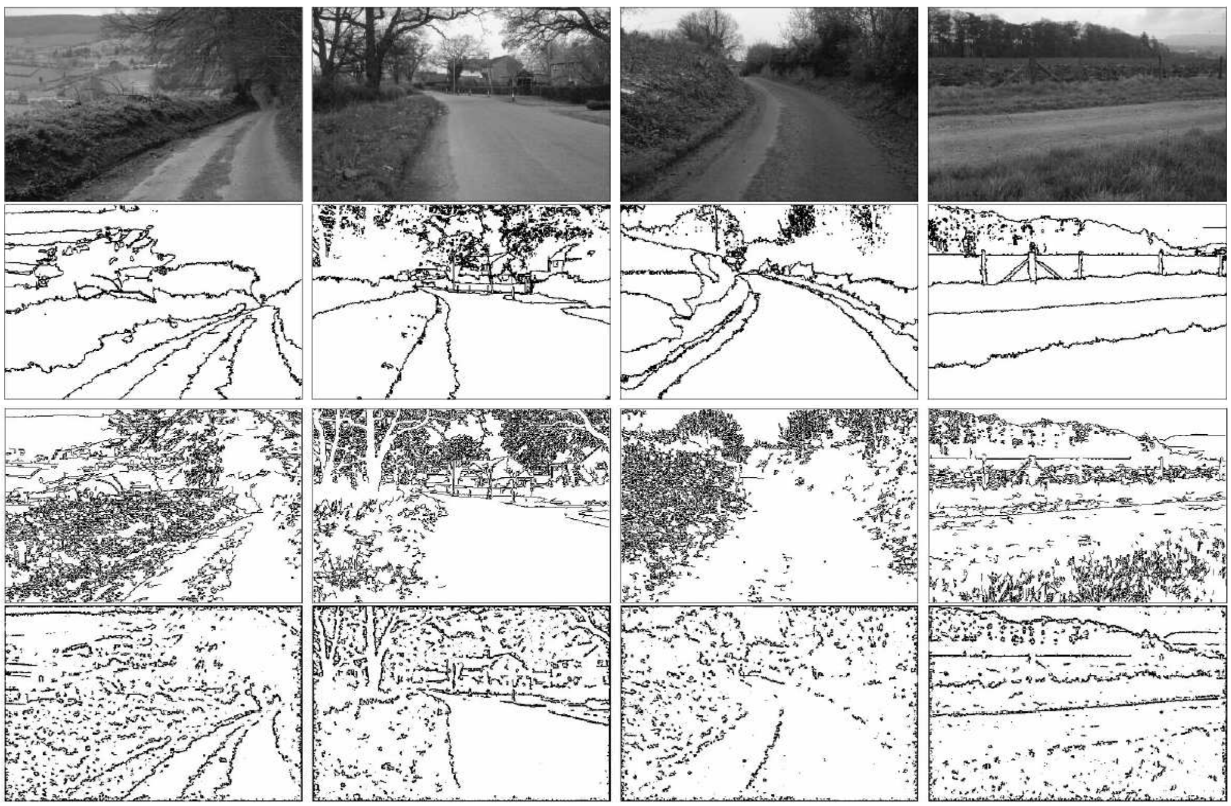
[from Elder and Goldberg (2001)]

Approach:

1. Edgels represented by location, orientation, blur scale (min reliable scale for detection), and asymptotic brightness on each side.
2. Edgels are grouped into curves (i.e., maximum likelihood curves joining two edge segments specified by a user.)
3. Curves are then manipulated (i.e., deleted, moved, clipped etc).
4. The image is reconstructed (i.e., solve Laplace's equation given asymptotic brightness as boundary conditions).

Empirical Edge Detection

The four rows below show images, edges marked manually, Canny edges, and edges found from an empirical statistical approach by Konishi et al (2003). (We still have a way to go.)



Row 2 – human;   Row 3 – Canny;   Row 4 – Konishi et al
[from Konishi, Yuille, Coughlin and Zhu (2003)]

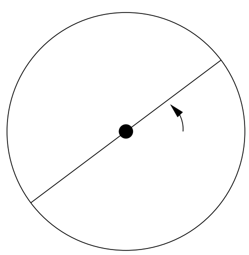
Context and Salience: Structure in the neighbourhood of an edgel is critical in determining the salience of the edgel, and the grouping of edgels to form edges.

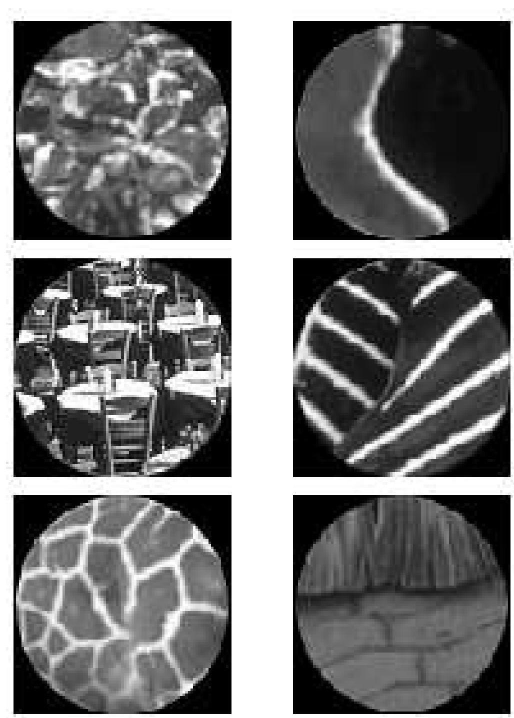
Other features: Techniques exist for detecting other features such as bars and corners. Some of these will be discussed later in the course.

Boundaries versus Edges

An alternative goal is to detect (salient) region boundaries instead of brightness edges.

For example, at a pixel $\vec{x}$, decide if the neighbourhood is bisected by a region boundary (at some orientation θ and scale σ)





From http://www.cs.berkeley.edu/~fowlkes/project/boundary

The Canny edge operator determines edgels $(\vec{x}, \theta, \sigma)$ based essentially on the difference of mean brightness in these two half disks.

We could also try using other sources of information, such as texture or contours (see Martin et al, 2004).

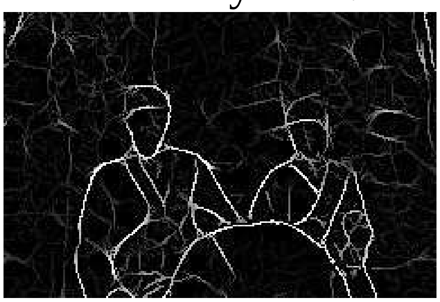
Boundary Probability

Martin et al (2004) trained boundary detectors using gradients of brightness, colour, and texture.

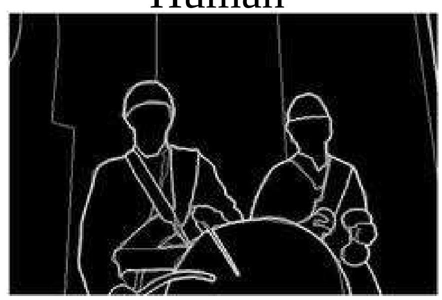
Image

Canny

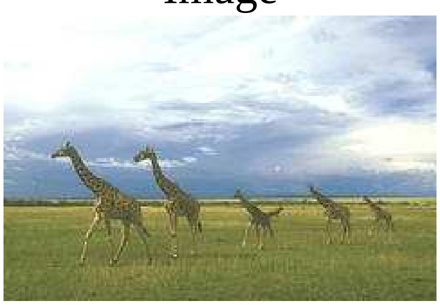
Boundary Prob.



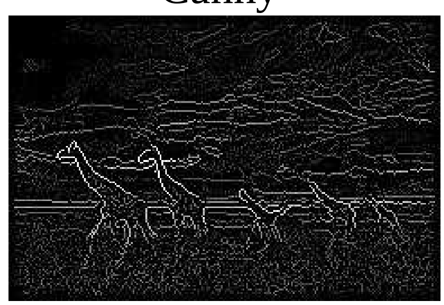
Human



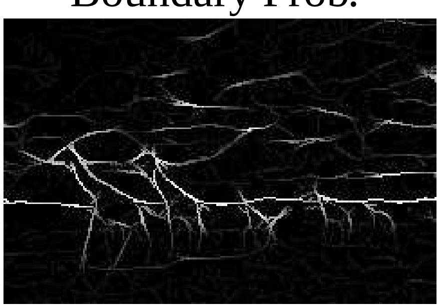
Image



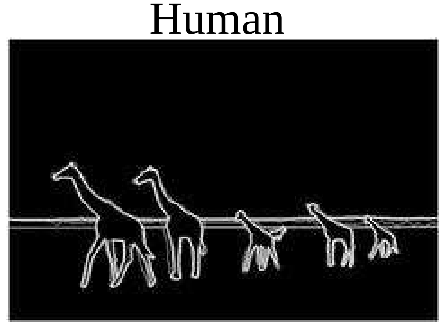
Canny



Boundary Prob.



Human



Further Readings

Castleman, K.R., **Digital Image Processing**, Prentice Hall, 1995

John Canny, ”A computational approach to edge detection.” *IEEE Transactions on PAMI*, 8(6):679–698, 1986.

James Elder and Richard Goldberg, ”Image editing in the contour domain.” *IEEE Transactions on PAMI*, 23(3):291–296, 2001.

Scott Konishi, Alan Yuille, James Coughlin, and Song Chun Zhu, ”Statistical edge detection: Learning and evaluating edge cues.” *IEEE Transactions on PAMI*, 25(1):57–74, 2003.

William Freeman and Edward Adelson, ”The design and use of steerable filters.” *IEEE Transactions on PAMI*, 13:891–906, 1991.

David Martin, Charless Fowlkes, and Jitendra Malik, ”Learning to detect natural image boundaries using local brightness, color, and texture cues.” *IEEE Transactions on PAMI*, 26(5):530–549, 2004.