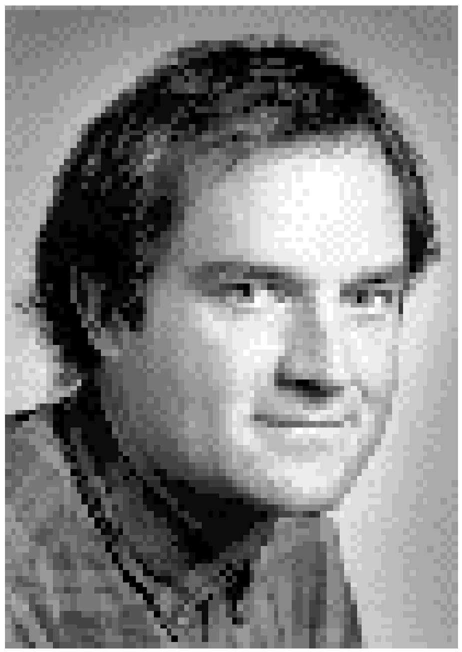
Canny Edge and Line Detection

CS/BIOEN 6640, Fall 2010

Guido Gerig

with some slides from Tsai Sing Lee, CMU and from J. Canny's Papers

Canny Edge Detector



- Canny (1984) introduces several good ideas to help.

- **References:** Canny, J.F. A computational approach to edge detection. IEEE Trans Pattern Analysis and Machine Intelligence, 8(6): 679-698, Nov 1986.

Optimal Edge Detector Design

- Canny derives his filter by optimizing a certain performance index that favors true positive, true negative and accurate localization of detected edges
- Analysis is restricted to linear shift invariant filter that detect unblurred 1D continuous step
- Other justifiable performance criteria are possible and will lead to different filters.

What are Canny's Criteria?

- **Good detection**: low probability of not marking real edge points, and falsely marking non-edge points.

$$SNR = \frac{\left| \int_{-w}^{w} G(-x) f(x) dx \right|}{n_o \sqrt{\int_{-w}^{w} f^2(x) dx}}$$

- *f* is the filter, *G* is the edge signal, denominator is the root-mean-squared response to noise *n(x)* only.

Localization Criterion

- Good localization: close to center of the true edge

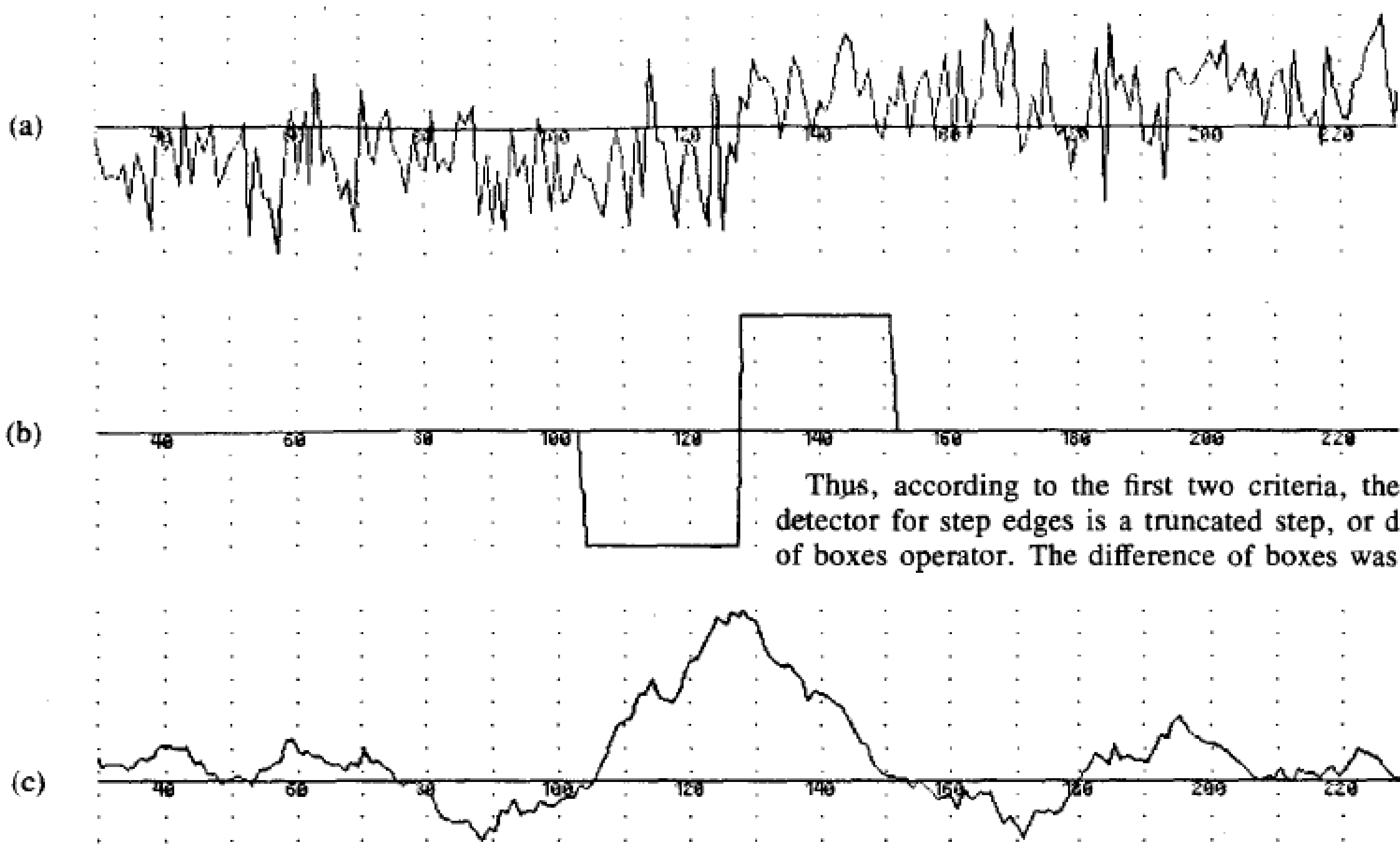
$$Localization = \frac{1}{\sqrt{E[x_0^2]}} = \frac{\left| \int_{-w}^{w} G'(-x) f'(x) dx \right|}{n_o \sqrt{\int_{-w}^{w} f'^2(x) dx}}$$

- a measure that increases as localization improves.
- Use reciprocal of the rms distance of the marked edge from the center of the true edge.

Eliminating Multiple Response

- Only one response to a single edge: implicit in first criterion, but make explicit to eliminate multiple response.
- The first two criteria can be trivially maximized by setting $f(x)=G(-x)$!
- What is this? This is a truncated step (difference of box operator).
- What is its problem?

“Optimal Operator” for Noisy Step Edge: SNR*LOC



Thus, according to the first two criteria, the optimal detector for step edges is a truncated step, or difference of boxes operator. The difference of boxes was used by

Inter-maximum Spacing

- Ideally, want to make the distance between peaks in the noise response approximate the width of the response of the operator to a single step.
- The mean distance between two adjacent maxima in the filtered response (or zero-crossing of their derivatives) can be derived as:

$$x_{zc}(f) = \pi \left(\frac{\int_{-\infty}^{\infty} f'^2(x)dx}{\int_{-\infty}^{\infty} f''^2(x)dx} \right)^{1/2}$$

- Set this distance a fraction k of the operator width W, Seek f satisfies this constraint with a fixed k.

$$x_{zc}(f) = kW$$

Optimization

Filter Parameters						
n	x_{max}	$\Sigma\Lambda$	r	α	ω	β
1	0.15	4.21	0.215	24.59550	0.12250	63.97566
2	0.3	2.87	0.313	12.47120	0.38284	31.26860
3	0.5	2.13	0.417	7.85869	2.62856	18.28800
4	0.8	1.57	0.515	5.06500	2.56770	11.06100
5	1.0	1.33	0.561	3.45580	0.07161	4.80684
6	1.2	1.12	0.576	2.05220	1.56939	2.91540
7	1.4	0.75	0.484	0.00297	3.50350	7.47700

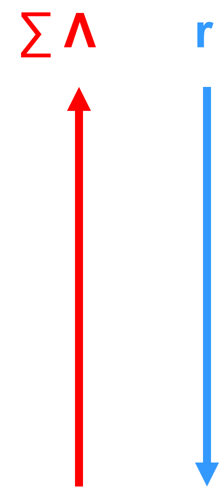


Fig. 4. Filter parameters and performance measures for the filters illustrated in Fig. 5.

∑: SNR

Λ: Localization (how close to true position)

X_max: distance between adjacent maxima (fraction of operator width)

r: multiple response performance

Optimal Operators

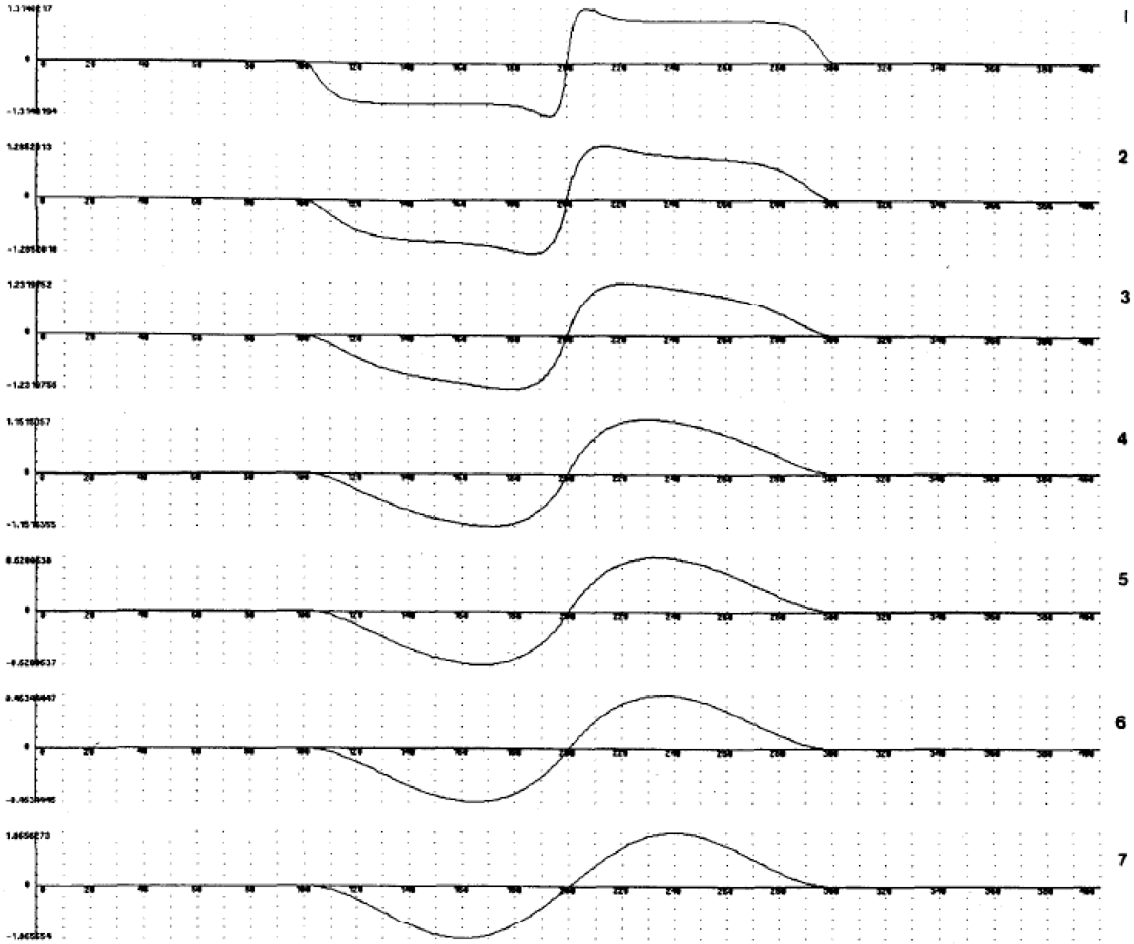
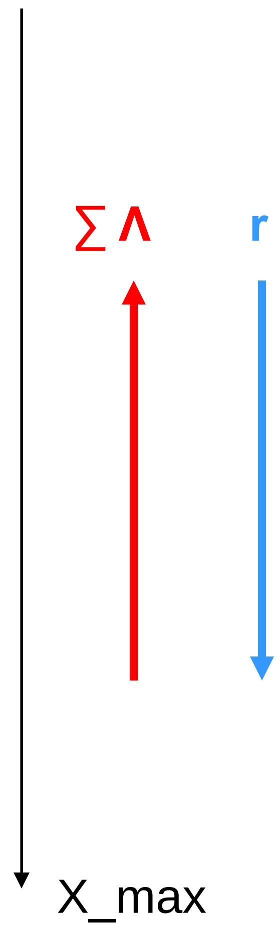


Fig. 5. Optimal step edge operators for various values of x_{max}. From top to bottom, they are x_{max} = 0.15, 0.3, 0.5, 0.8, 1.0, 1.2, 1.4.



Optimal Operator versus First Derivative of Gaussian

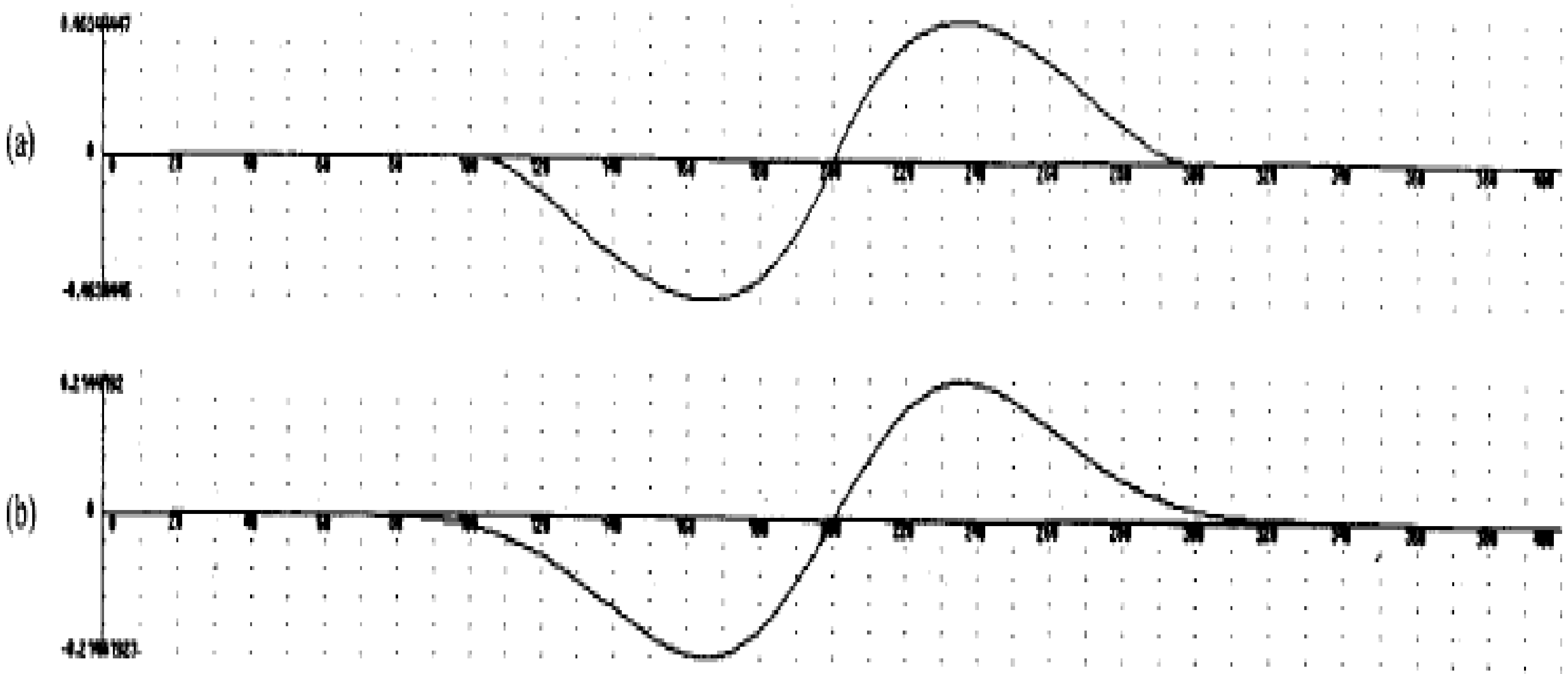


Fig. 6. (a) The optimal step edge operator. (b) The first derivative of a Gaussian.

“Optimal Operator” for Noisy Step Edge: SNR*LOC*MULT

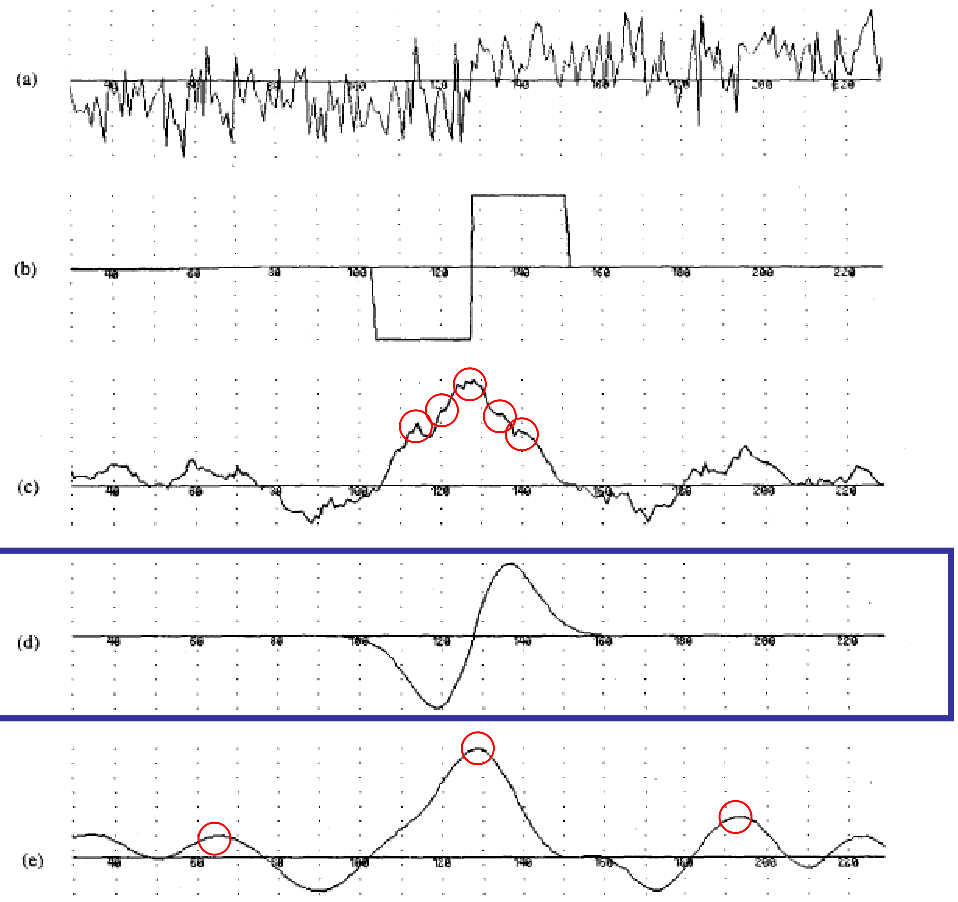
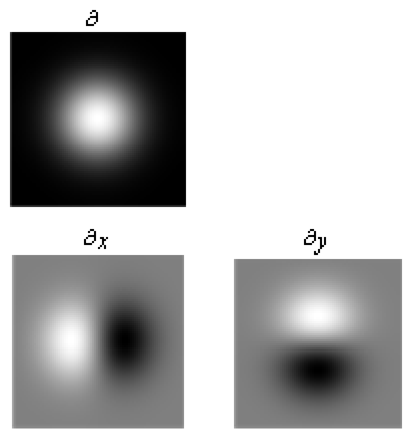


Fig. 1. (a) A noisy step edge. (b) Difference of boxes operator. (c) Difference of boxes operator applied to the edge. (d) First derivative of Gaussian operator. (e) First derivative of Gaussian applied to the edge.

2D Edge Filter: Output at different scales



1st order Gaussian Derivatives

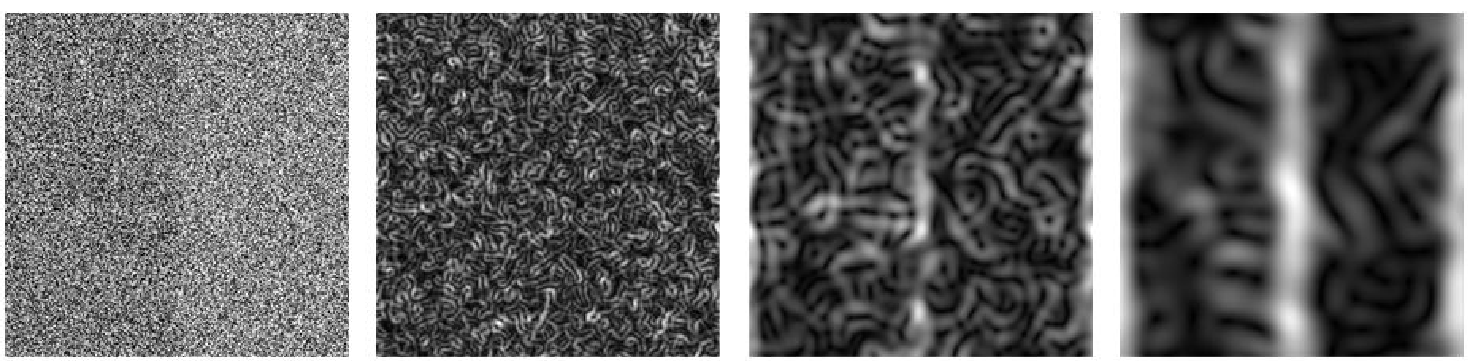
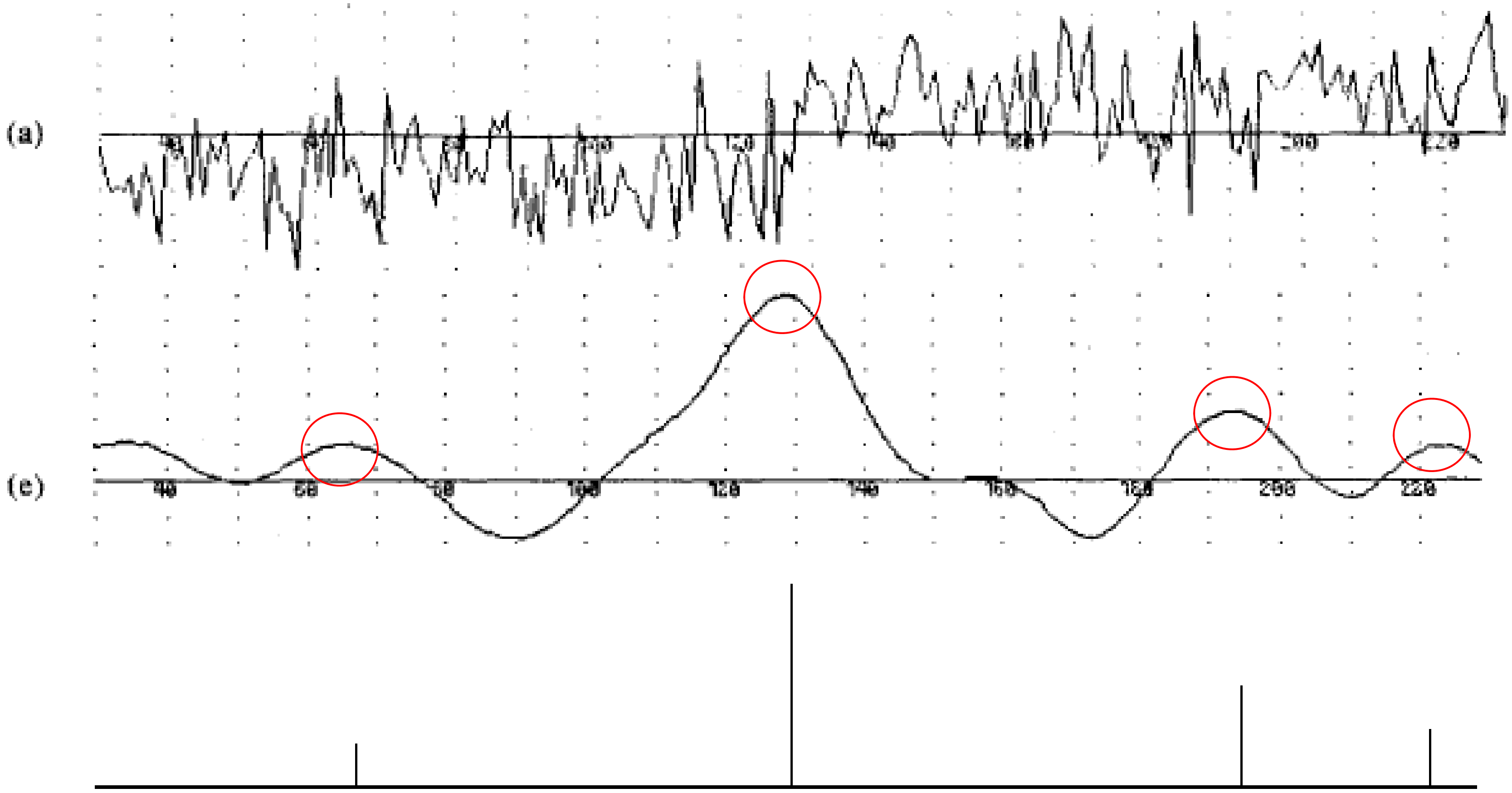


Figure 5.11 Detection of a very low contrast step-edge in noise. Left: original image, the step-edge is barely visible. At small scales (second image, $\sigma = 2$ pixels) the edge is not detected. We see the edges of the noise itself, cluttering the edge of the step-edge. Only at large scale (right, $\sigma = 12$ pixels) the edge is clearly found. At this scale the large scale structure of the edge emerges from the small scale structure of the noise.

Response at different scales

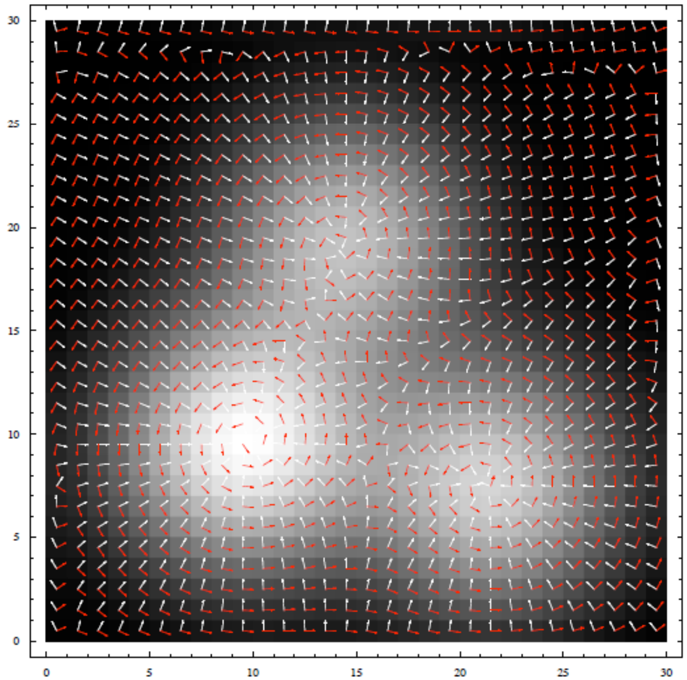
Figure 5.11 Gradient edges detected at different scales (σ = 0.5, 2, 5 pixels resp.).
coarser edges (right) indicate hierarchically more 'important' edges.

Non-Maximum Suppression

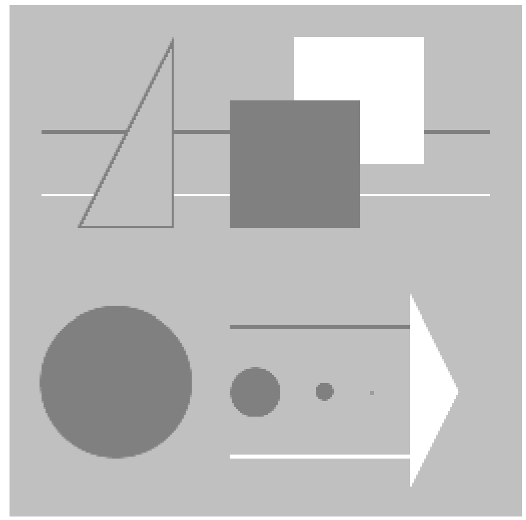


Detect local maxima and suppress all other signals.

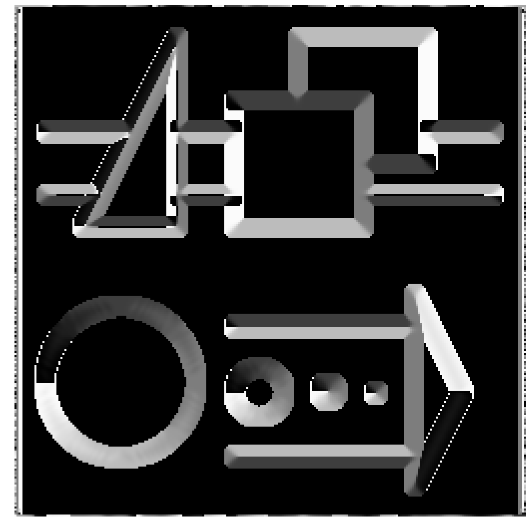
What about 2D?



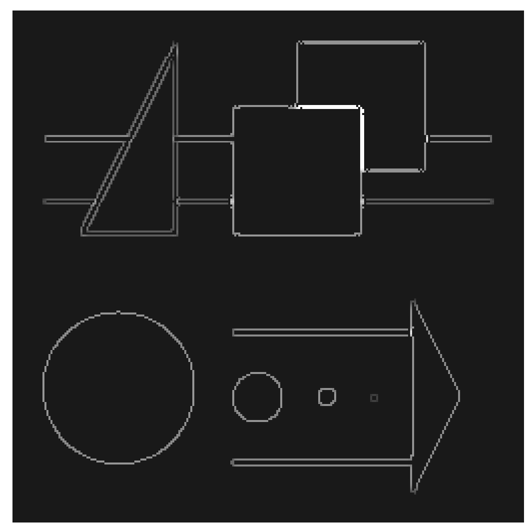
At every position in the edge-magnitude output, there is a coordinate system with normal and tangent.



original



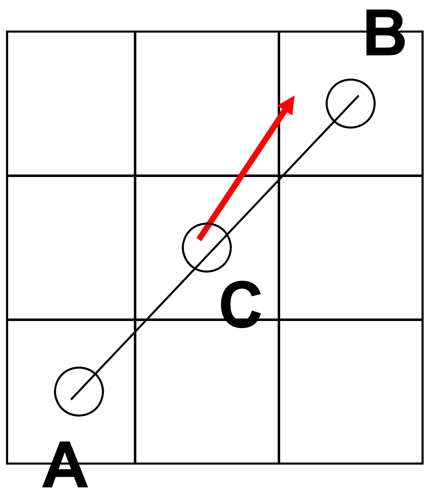
edge orientation



edge positions coded by edge magnitude

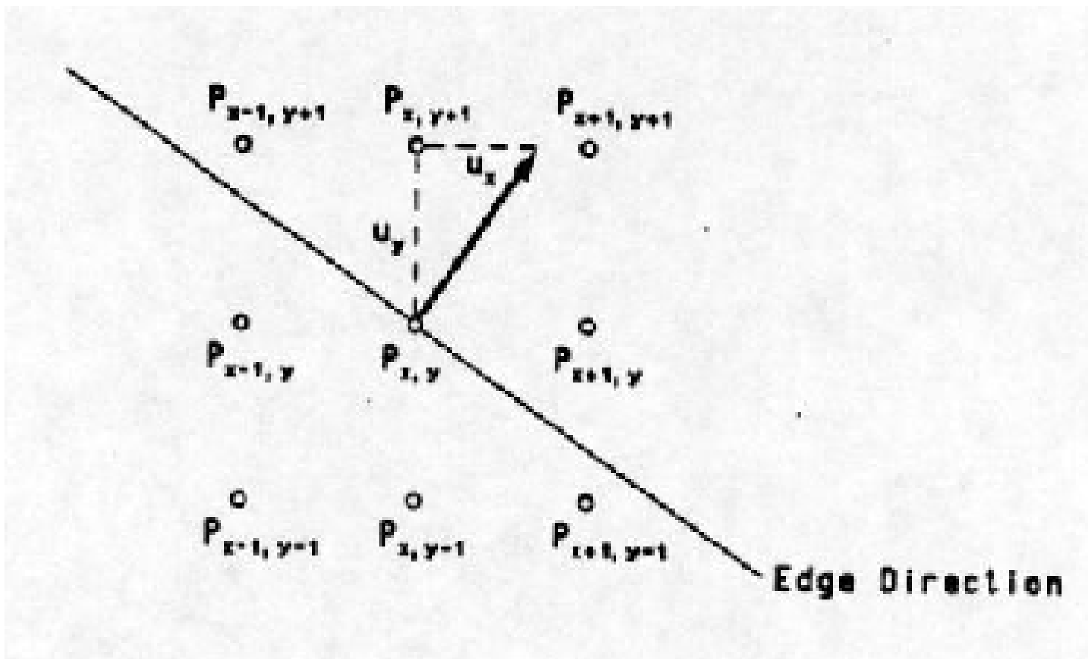
Non-Maximum Suppression

- Canny: Interpolate Gradient along gradients (plus and minus a certain distance) and check if center is larger than neighbors.
- Simplified: Test for each Gradient Magnitude pixel if neighbors along gradient direction (closest neighbors) are smaller than center: Mark C as maximum if A<C and B<C



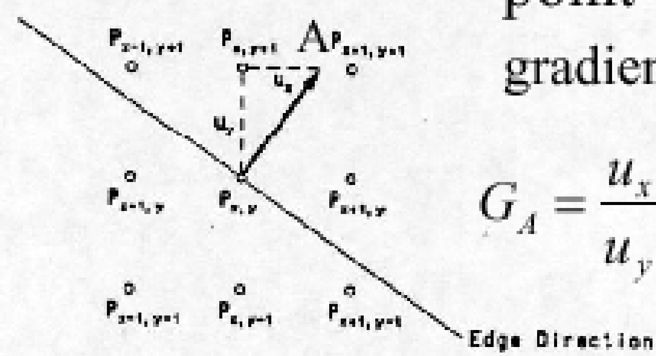
Non-maximum suppression

At each point, compute its edge gradient, compare with the gradients of its neighbors along the gradient direction. If smaller, turn 0; if largest, keep it.



Estimation of Gradient

- Sampling is discrete, how to estimate gradient?
- Pick 2 pts in support closest to u.
- The gradient magnitudes at 3 pts define a plane, use this plane to locally approximate the gradient magnitude surface and to estimate the value at a point on the line. The interpolated gradient magnitude at A, for example, is

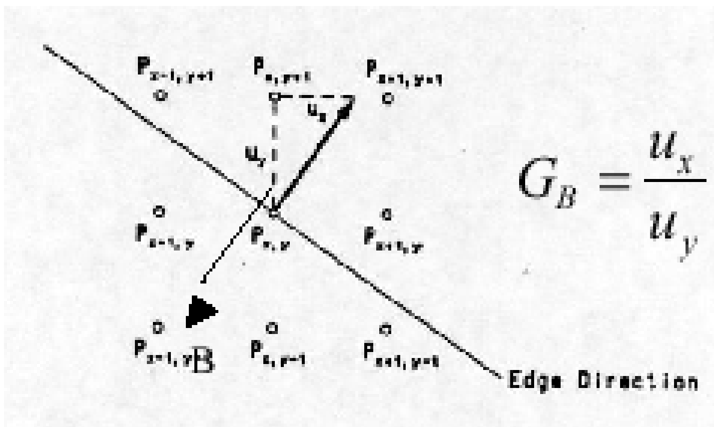


$$G_A = \frac{u_x}{u_y} G(x+1, y+1) + \frac{u_y - u_x}{u_y} G(x, y+1)$$

Note: $G(i, j), u_x, u_y$ are known.

Estimation of Gradient

- The interpolated gradient on the other side is given by:

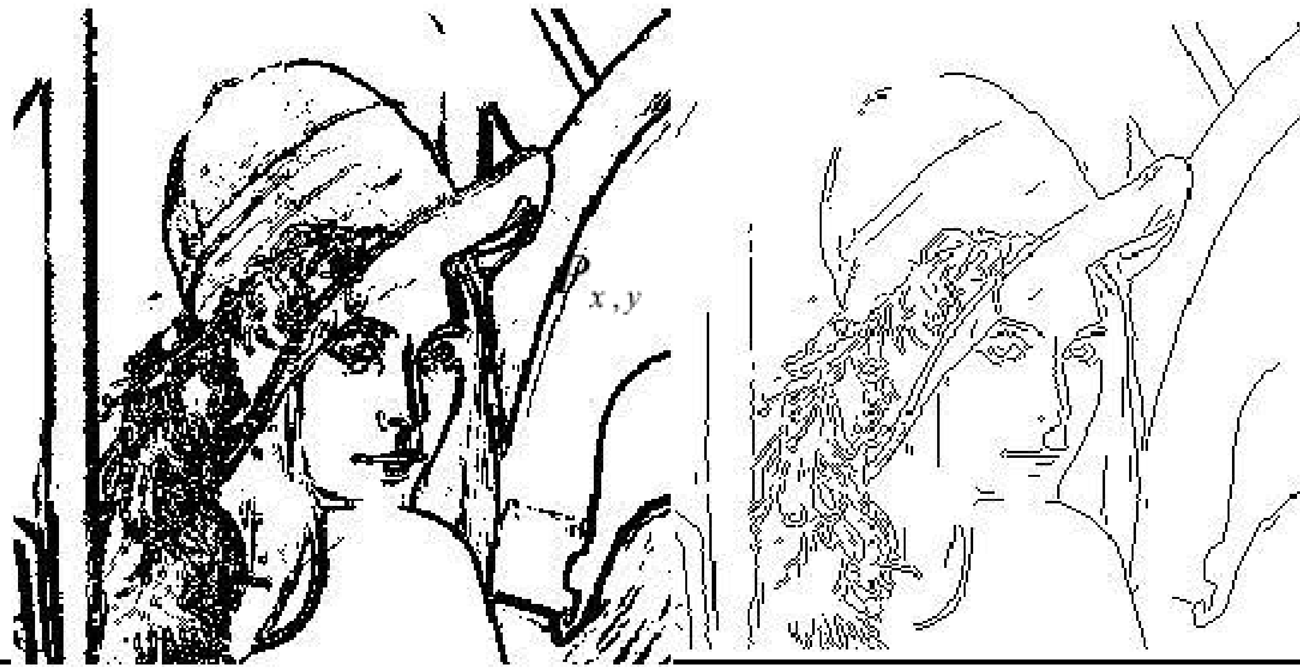


$$G_B = \frac{u_x}{u_y} G(x-1, y-1) + \frac{u_y - u_x}{u_y} G(x, y-1)$$

- Mark $P_{x,y}$ as a maximum if $G(x,y) > G_A$ and $G(x,y) > G_B$
- Interpolation always involve 1 diagonal and 1 non-diagonal point. Avoid division by multiplying through by u_y .

Non-maximum suppression

- This scheme involves 4 multiplication per point, but it is not excessive.
- Works better than simpler scheme which compares the points $P_{x,y}$ with two of its neighbors.



Results

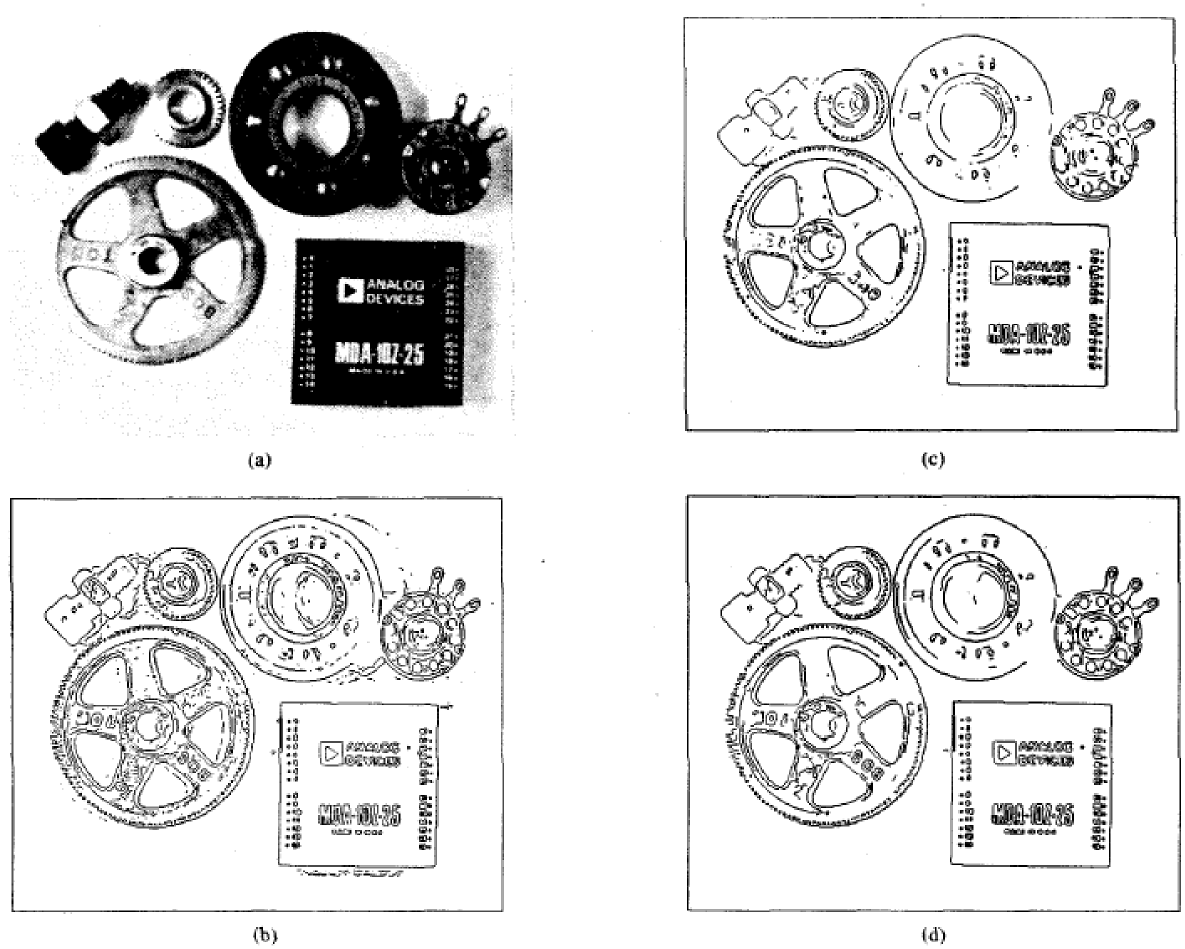


Fig. 7. (a) Parts image, 576 by 454 pixels. (b) Image thesholded at T_1. (c) Image thresholded at 2 T_1. (d) Image thresholded with hysteresis using both the thresholds in (a) and (b).

Canny Edge Detection

- Basic idea is to detect at the zero-crossings of the second directional derivative of the smoothed image
- in the direction of the gradient where the gradient magnitude of the smoothed image being greater than some threshold depending on image statistics.
- It seeks out zero-crossings of

$$\partial^2 (G * I) / \partial n^2 = \partial([\partial G / \partial n] * I) / \partial n$$

n is the direction of the gradient of the smoothed image.

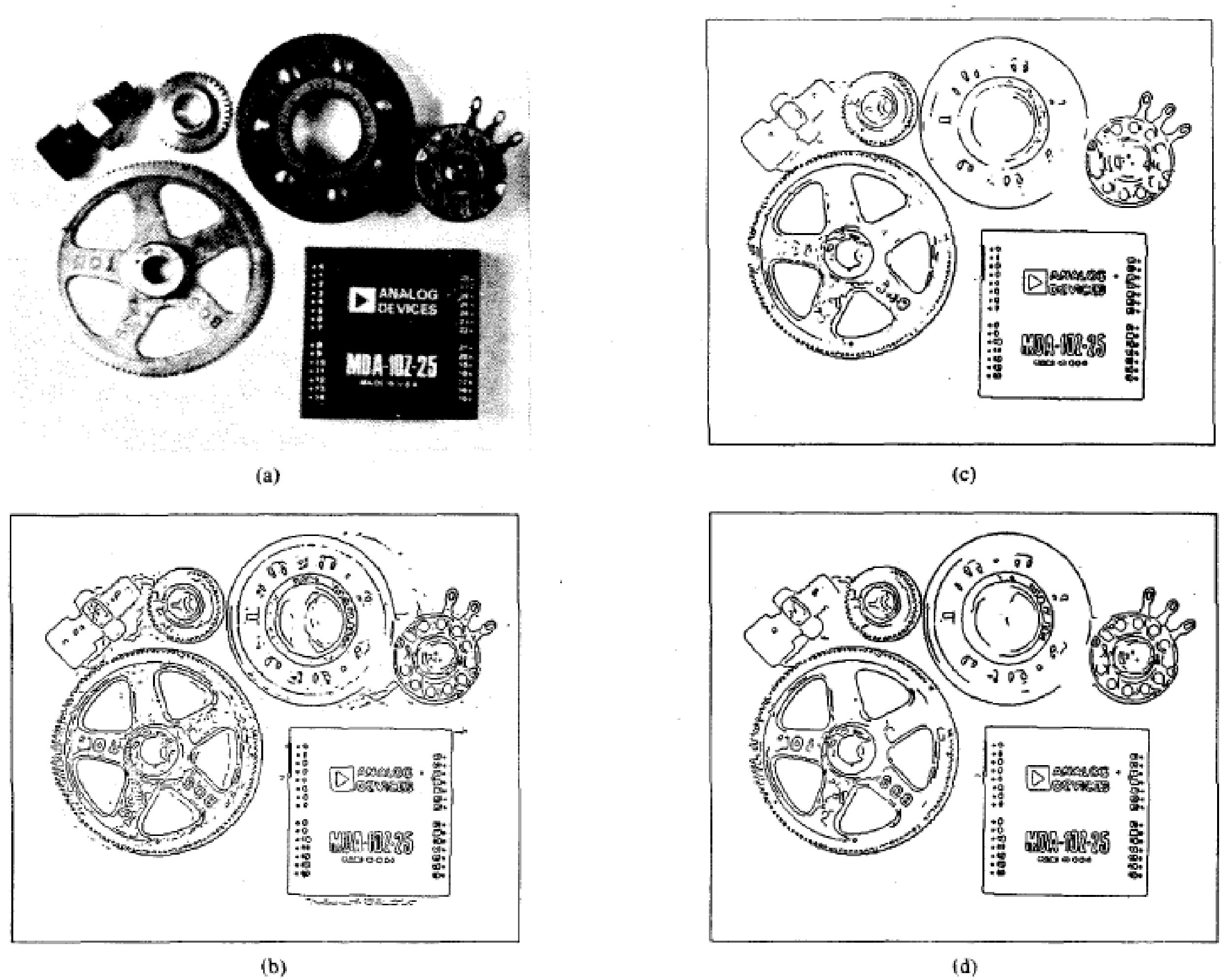
Canny's zero-crossings

- Canny zero-crossings correspond to the first-directional-derivative's maxima and minima in the direction of the gradient.
- Maxima in magnitude reasonable choice for locating edges.

Canny: Hysteresis Thresholding

- Thresholding/binarization of edge map:
  - Noise and image structures have different structure
  - Simple thresholding: If too low, too many structures appear, if too high, contours are broken into pieces
  - **Idea**: Hysteresis: Upper and lower threshold, keep all connected edges (d_m metric) that are connected to upper but above lower threshold

Hysteresis Thresholding



too little

too many

combi nation

Fig. 7. (a) Parts image, 576 by 454 pixels. (b) Image thesholded at T_1. (c) Image thresholded at 2 T_1. (d) Image thresholded with hysteresis using both the thresholds in (a) and (b).

Optimal Operators for Other Structures

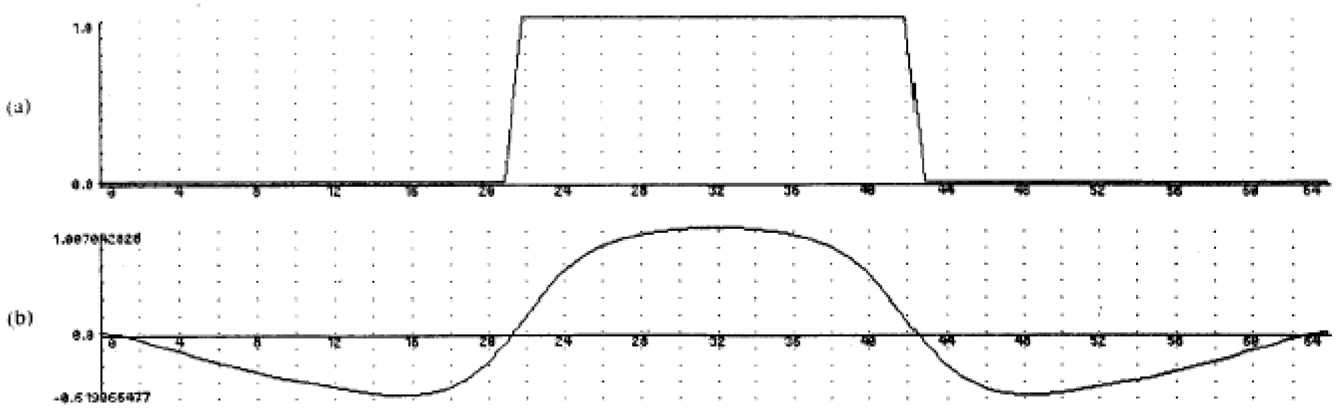


Fig. 2. A ridge profile and the optimal operator for it.

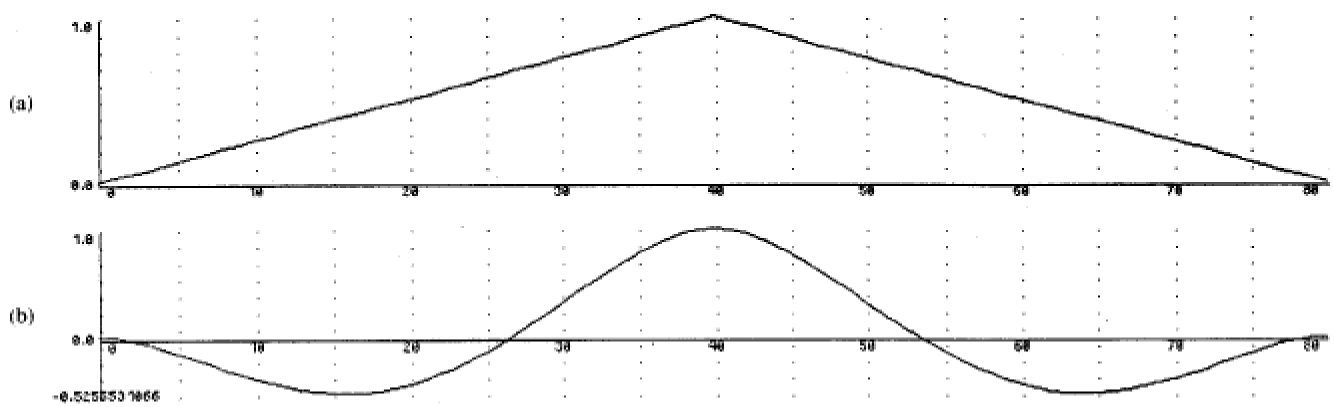
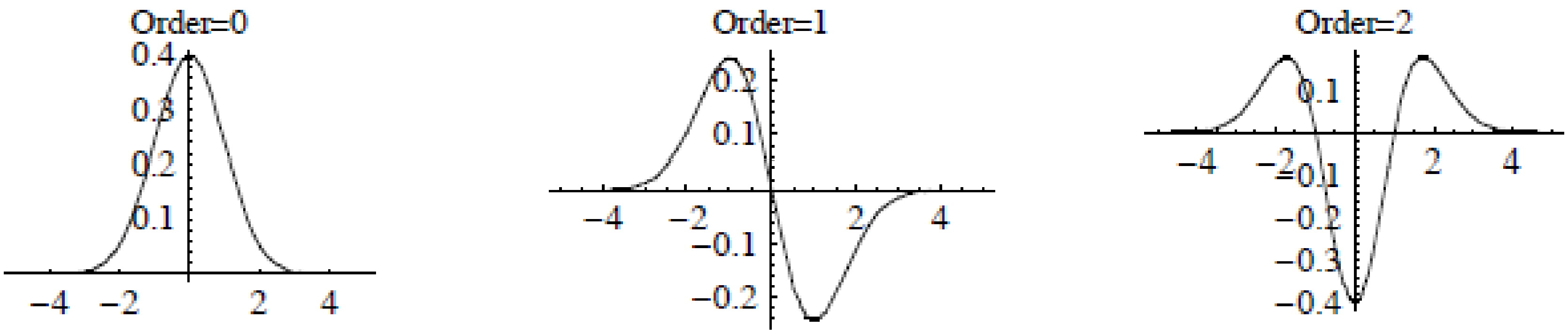


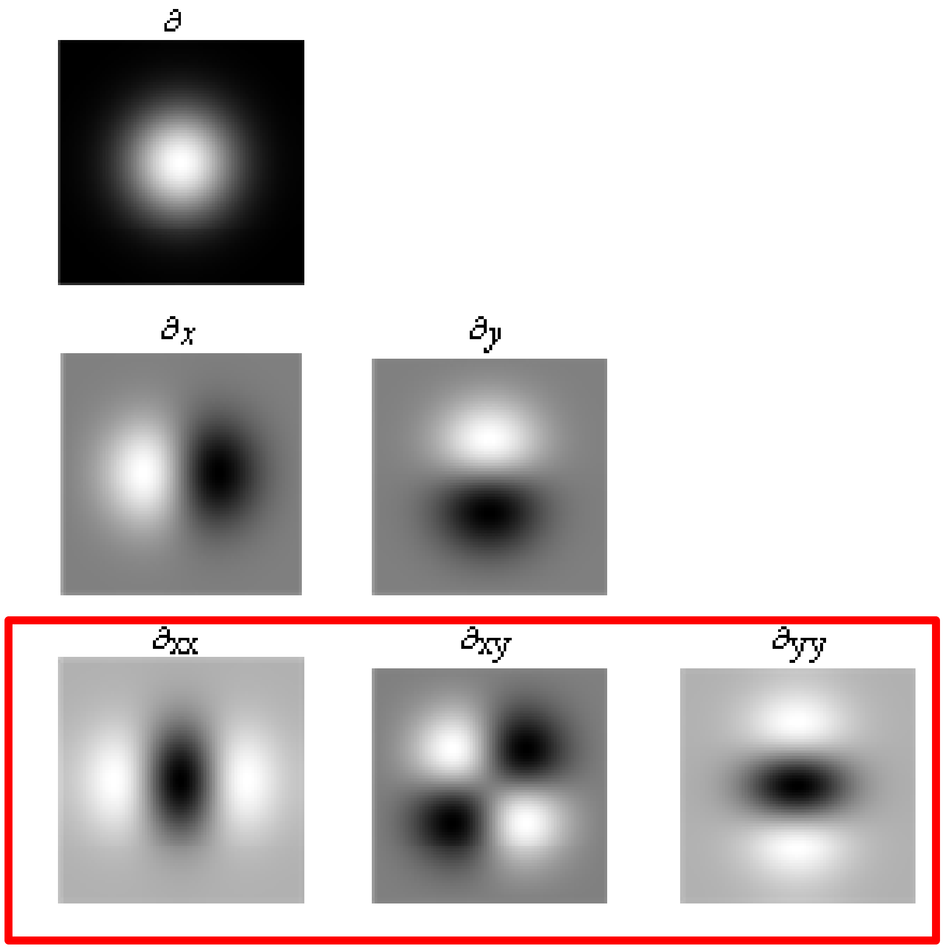
Fig. 3. A roof profile and an optimal operator for roofs.

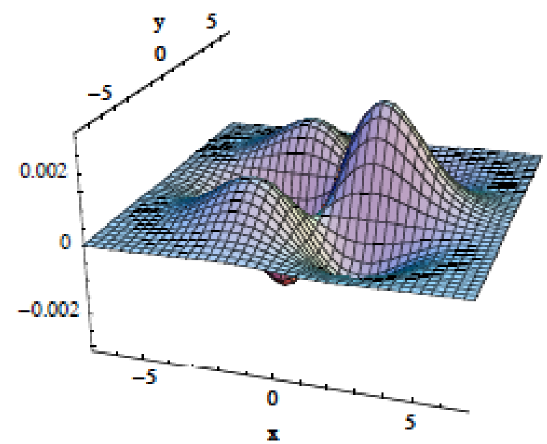
Resembles 2nd derivative of Gaussian

Gaussian Derivatives



2nd Derivative Operator to detect lines and curves





Multidimensional Derivatives

- Nabla operator: $\vec{\nabla} \equiv \left\{ \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right\}$
- Gradient: $\vec{\nabla} L$ is the gradient of L
- Laplacian: $\vec{\nabla} . \left(\vec{\nabla} L \right) = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$
- Hessian:

$$\vec{\nabla} \left(\vec{\nabla} L \right) = \begin{pmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{pmatrix}$$

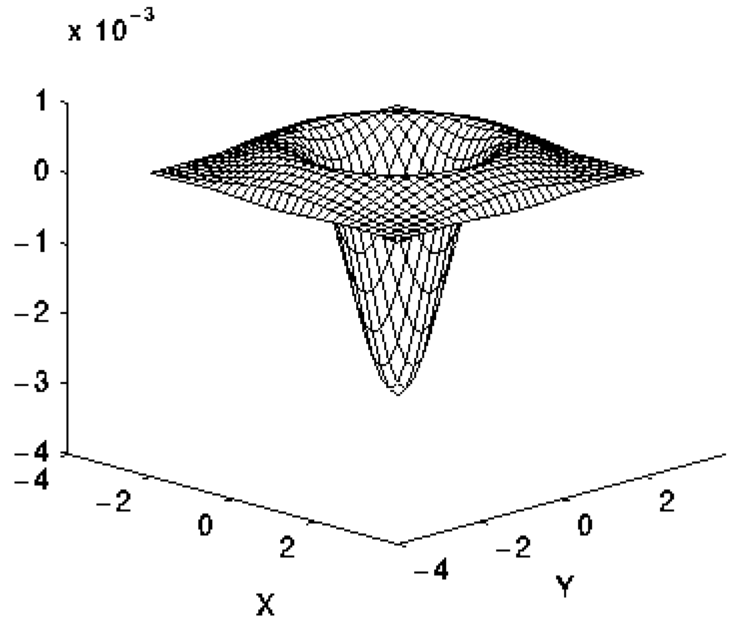
(matrix of 2nd derivatives, gradient of gradient of L)

Laplacian

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Local kernels



$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

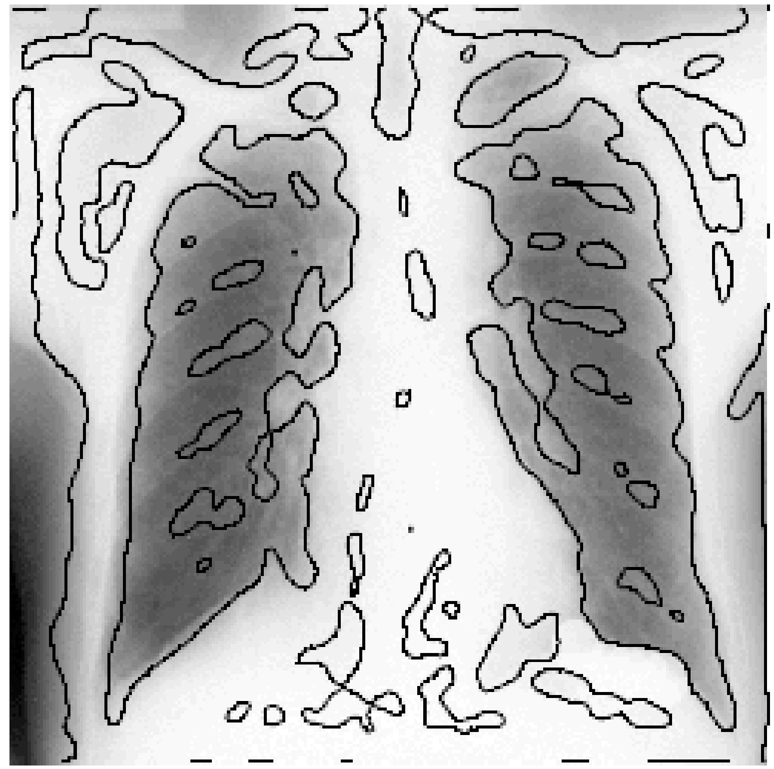
Laplacian of 2D Gaussian kernel

Laplacian of Gaussian (LoG)

Enhances line-like structures (glasses), creates zero-crossing at edges (positive and negative response at both sides of edges)

Source: http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

Often used: Zero-Crossings of LoG for Edge Detection



$\Delta L = 0 \quad \sigma = 4$ pixels

Hint: Remember that edge positions are extrema of first derivative → zero-crossings of 2nd derivatives. Be careful: Extrema or maxima & minima!

Line/Ridge Detection