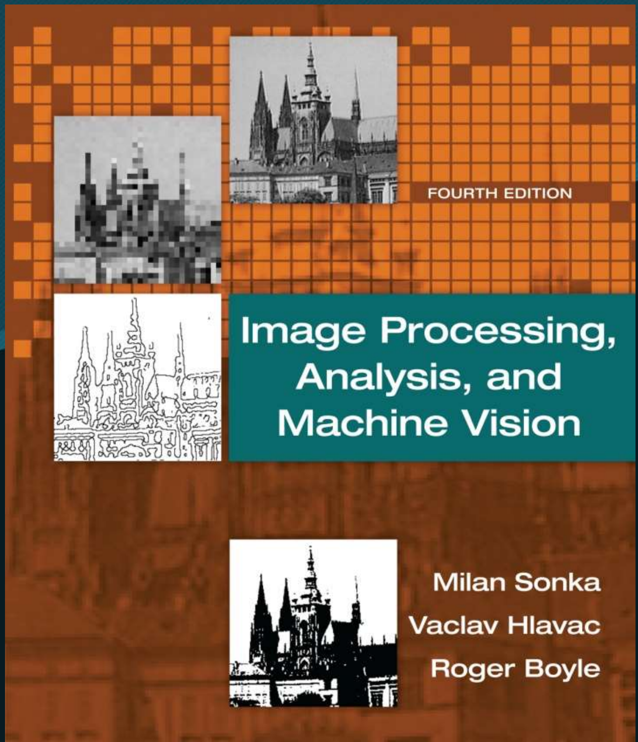
Chapter 5

Image pre-processing



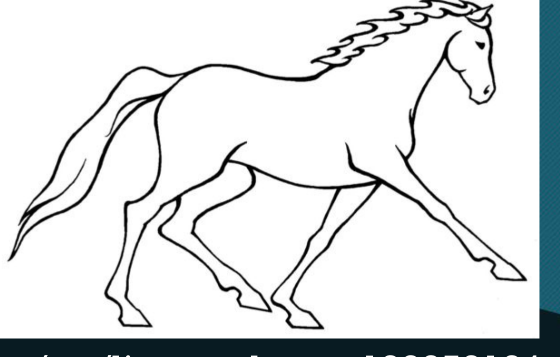
CENGAGE Learning

Image pre-processing

- Pixel brightness transformations
- Geometric transformations
- Local pre-processing
  - Image smoothing, edge detection, line detection, corner detection, and region detection.
- Image restoration

Edge detectors

- Edge detectors are a collection of very important local image pre-processing methods used to locate changes in the intensity function.
- Edges are pixels where the intensity function (brightness) changes abruptly.
- If only edge elements with strong magnitude are considered, such information often suffices (足夠) for image understanding.
- For example, line drawing images.



http://kelpie77.deviantart.com/art/line-art-horse-123352124

Edge detectors

- Physical phenomena (現象) in the image formation process lead to abrupt changes in image values

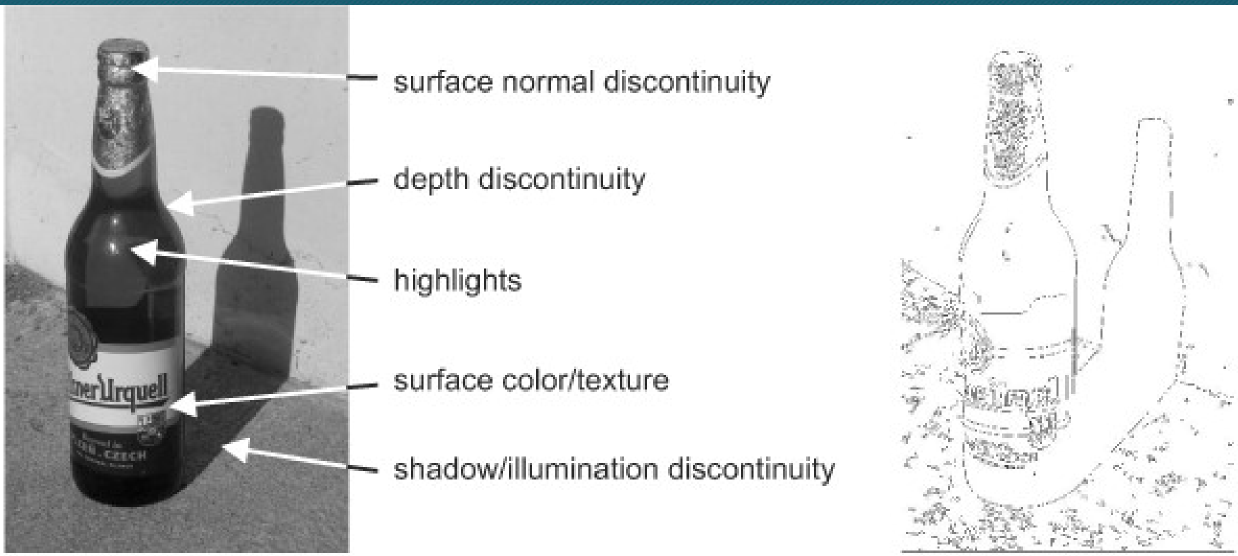
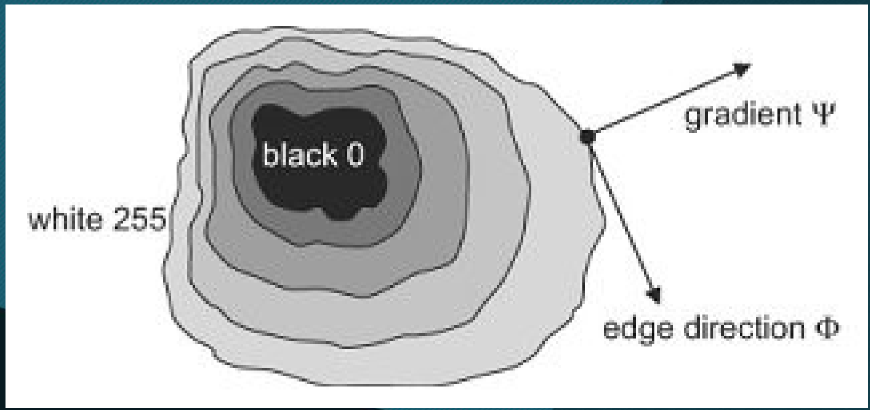


Figure 5.15: Origin of edges, i.e., physical phenomena in image formation process which lead to edges in images. At right, a Canny edge detection (see Section 5.3.5). © *Cengage Learning 2015.*

Edge detectors

- An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of that pixel.
- It is a vector variable with two components.
  - Magnitude
    - The edge magnitude is the magnitude of the gradient.
  - Direction
    - The edge direction ϕ is rotated with respect to the gradient direction ψ by -90°.
    - The gradient direction gives the direction of maximum growth of the function.



Edge detectors

- Edges are often used in image analysis for finding region boundaries.
- This figure shows examples of several standard edge profile (輪廓、外形).
- Edge detectors are usually tuned for some type of edge profile.

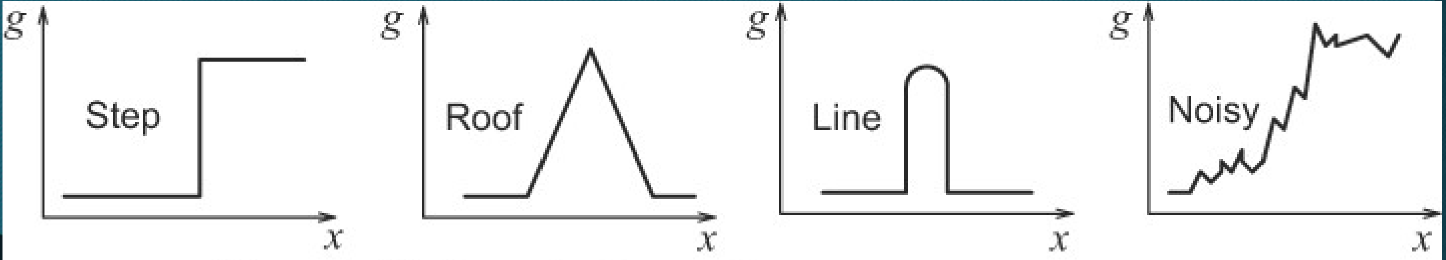


Figure 5.17: Typical edge profiles. © *Cengage Learning 2015.*

Edge detectors

- The gradient magnitude $|\mathrm{grad}\ g(x,y)|$ and gradient direction ψ are continuous image functions calculated as

$$|\mathrm{grad}\ g(x,y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

$$\psi = \arg\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)$$

where $\arg(x,y)$ is the angle (in radians;弳度) from the x axis to (x,y).

- Laplacian: (obtained only edge magnitudes without orientations)

$$\nabla^2 g(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2}$$

  - The Laplacian has the same properties in all directions and is therefore invariant to rotation.

Edge detectors

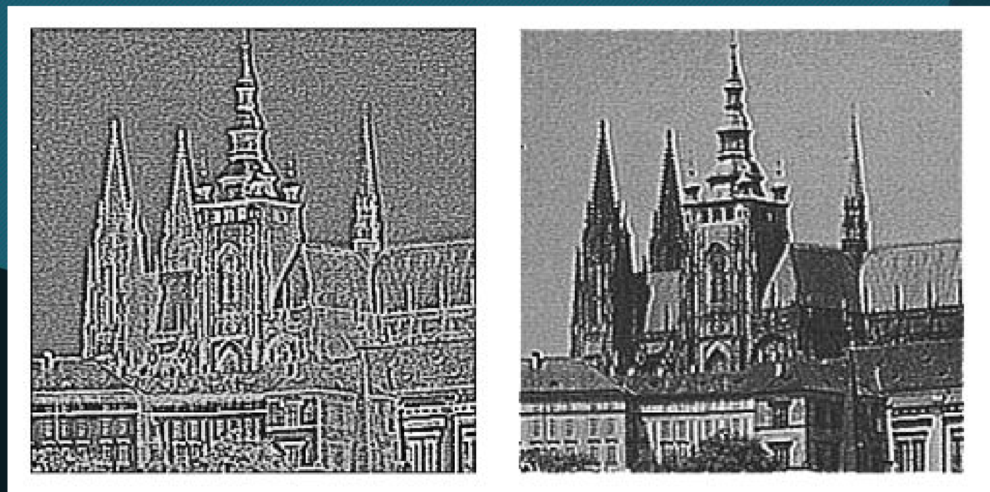
- Image sharpening has the objective of making edges steeper.
  - The sharpened output image f is obtained from the input image g as

$$f(i,j) = g(i,j) - C\,S(i,j)$$

    where C is positive coefficient which gives the strength of sharpening

    $S(i,j)$ is a measure of the image function sheerness (陡峭)
  - For example, image sharpening using a Laplacian

Original image



Edge detectors

- The derivatives (導數) can be approximated by differences in digital images.
  - The first differences of the image g in the vertical direction (for fixed i) and in the horizontal direction (for fixed j) are

$$\Delta_i g(i,j) = g(i,j) - g(i-n, \quad j)$$
$$\Delta_j g(i,j) = g(i,j) - g(\quad i, j-n)$$

    where n is a small integer, usually 1.
  - Symmetric expressions for the differences

$$\Delta_i g(i,j) = g(i+n, \quad j) - g(i-n, \quad j)$$
$$\Delta_j g(i,j) = g(\quad i, j+n) - g(\quad i, j-n)$$

Edge detectors

- Robert operator [1965]
  - It uses only a 2×2 neighborhood of the current pixel.

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

  - The magnitude of the edge is computed as

$$|g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)|$$

  - Disadvantage:
    - Its high sensitivity to noise: very few pixels are used to approximate the gradient.

Edge detectors

- Laplace operator
  - The Laplace operator ∇^2 is a very popular operator approximating the second derivative which gives the edge magnitude only.
  - A 3×3 mask is often used. For 4-neighborhoods and 8-neighborhoods it is defined as

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - A Laplacian with stressed significance of the central pixel or its neighborhood is sometimes used.

$$h = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \qquad h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

  - Disadvantage: It responds doubly to some edges in the image.

Edge detectors

- Prewitt operator
  - The Prewitt operator approximates the first derivative.
  - The gradient is estimated in eight (for a 3×3 convolution mask) possible directions, and the convolution result of greatest magnitude indicates the gradient direction.
  - Some examples of 3×3 masks

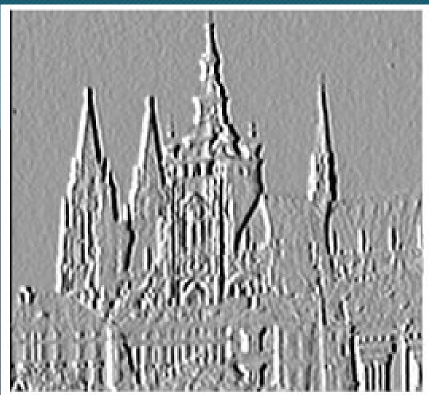
$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \ldots$$

  - Larger marks are possible, for example, 5×5 or 7×7.

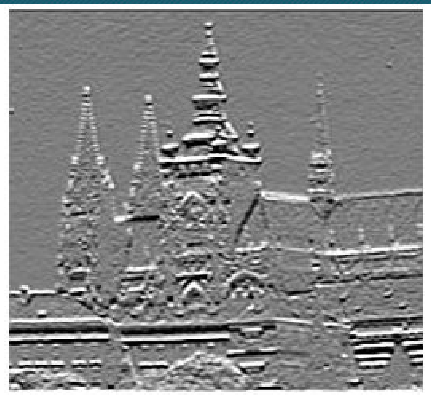
PS. The direction of the gradient is given by the mask giving maximal response. This is also the case for all the following operator approximating the first derivative.

Edge detectors

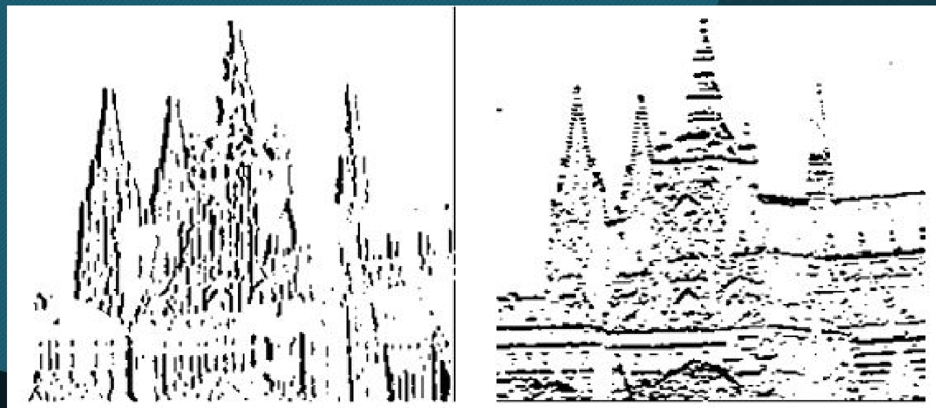
- First-derivative edge detection using Prewitt operators.
  - North direction
    - The brighter the pixel value, the stronger the edge.
  - East direction



North



East



Strong edges

Edge detectors

- Sobel operator
  - The Sobel operator approximates the first derivative.
  - Some examples of 3×3 masks

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \ldots$$

  - The Sobel operator is often used as a simple detector of horizontality (h_1) and verticality (h_3) of edges.
    - If the h_1 response is y and the h_3 response x, we might then derive edge magnitude as $\sqrt{x^2 + y^2}$ or $|x| + |y|$ and direction as $\arctan(\frac{y}{x})$ of edges.

Edge detectors

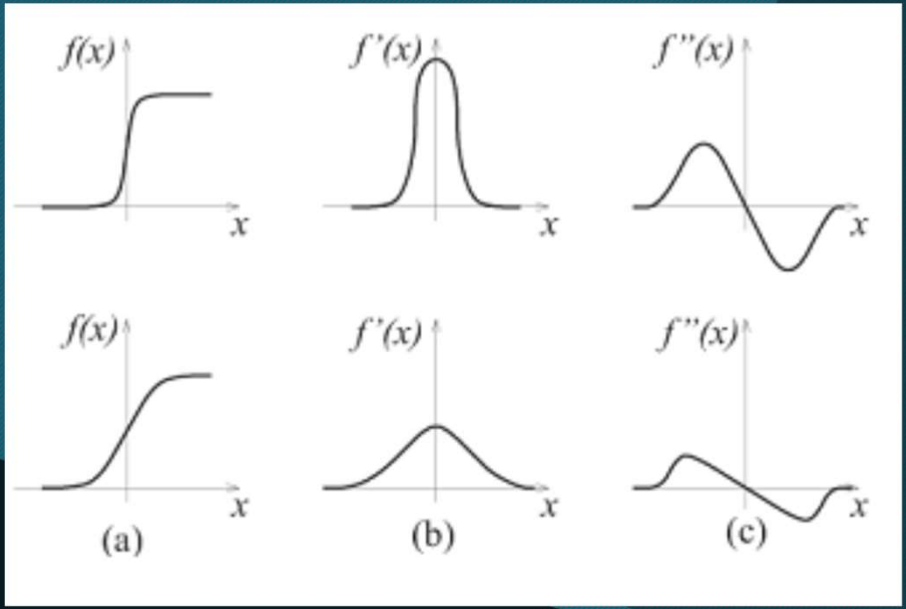
- Kirsch operator
  - The Kirsch operator approximates the first derivative.
  - Some examples of 3×3 masks

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \; h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \; h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \ldots$$

  - The direction of the gradient is given by the mask giving maximal response.

Zero-crossings of the second derivative

- Zero-crossings
  - An edge detection technique using the second derivative.
  - The first derivative of the image function should have an extremum at the position corresponding to the image.
  - The second derivative should be zero at the same position.
  - However, it is much easier and more precise to find a zero-crossing position than an extremum.



Zero-crossings of the second derivative

- Zero-crossings
  - How to compute the second derivative robustly?
    - To smooth an image first (to reduce noise)
    - The 2D Gaussian smoothing operator $G(x, y)$ (a Gaussian filter) is given by

$$G(x, y) = e^{-(x^2+y^2)/2\sigma^2}$$

where x, y: the image co-ordinates

σ: a standard deviation

    - Sometimes this is presented with a normalizing factor

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad \text{and} \quad G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x^2+y^2)/2\sigma^2}$$

Zero-crossings of the second derivative

- Zero-crossings
  - Our goal is to obtain a second derivative of a smoothed 2D function $f(x, y)$.
  - Laplacian of Gaussian (LoG)

$$\nabla^2[G(x, y) * f(x, y)]$$

where ∇^2: Laplace operator

$*$: convolution operator

  - Because of the linearity of the operators

$$\Rightarrow \nabla^2[G(x, y) * f(x, y)] = \nabla^2[G(x, y)] * f(x, y)$$

  - The derivative of the Gaussian filter $\nabla^2 G$ can be pre-computed.

Zero-crossings of the second derivative

- The derivative of the Gaussian filter $\nabla^2 G$

$$G(x,y) = e^{-(x^2+y^2)/2\sigma^2}$$

$$\Rightarrow \quad \frac{\partial G}{\partial x} = -\left(\frac{x}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \; ; \quad \frac{\partial G}{\partial y} = -\left(\frac{y}{\sigma^2}\right) e^{-(x^2+y^2)/2\sigma^2}$$

$$\Rightarrow \quad \frac{\partial^2 G}{\partial x^2} = \frac{1}{\sigma^2}\left(\frac{x^2}{\sigma^2} - 1\right) e^{-(x^2+y^2)/2\sigma^2} \; ; \quad \frac{\partial^2 G}{\partial y^2} = \frac{1}{\sigma^2}\left(\frac{y^2}{\sigma^2} - 1\right) e^{-(x^2+y^2)/2\sigma^2}$$

$$\Rightarrow \quad \nabla^2[G(x,y,\sigma)] = \frac{1}{\sigma^2}\left(\frac{x^2+y^2}{\sigma^2} - 2\right) e^{-(x^2+y^2)/2\sigma^2}$$

- The convolution mask of a LoG

$$h(x,y) = c\left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) e^{-(x^2+y^2)/2\sigma^2}$$

where c normalizes the sum of mask elements to zero. (!!)

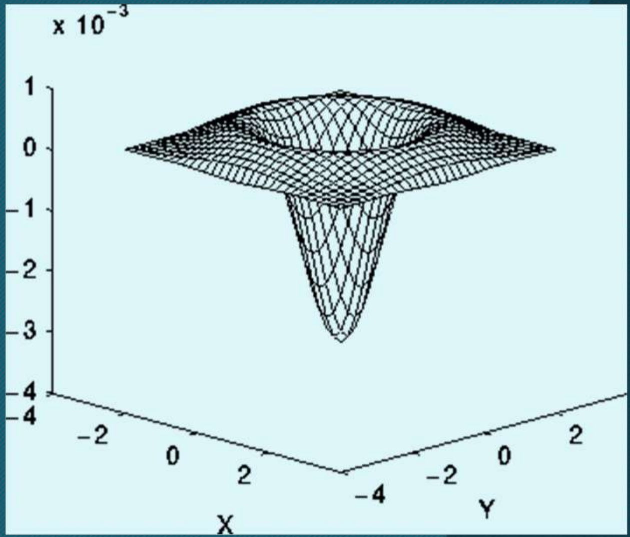
Zero-crossings of the second derivative

- The convolution mask of a LoG

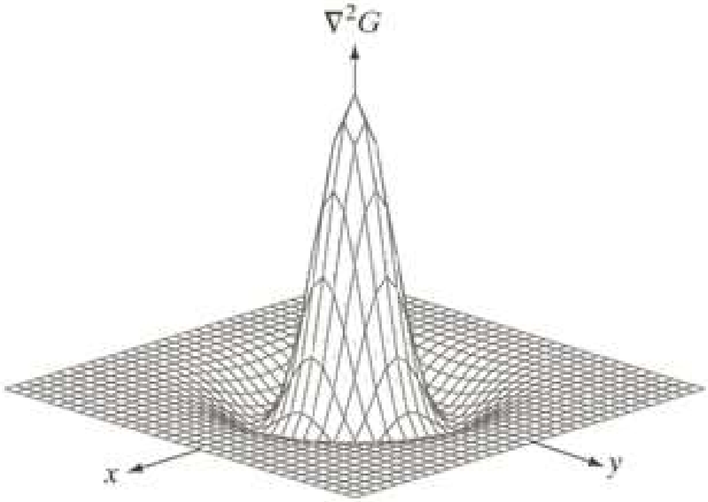
$$h(x,y) = c\left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right)e^{-(x^2+y^2)/2\sigma^2}$$

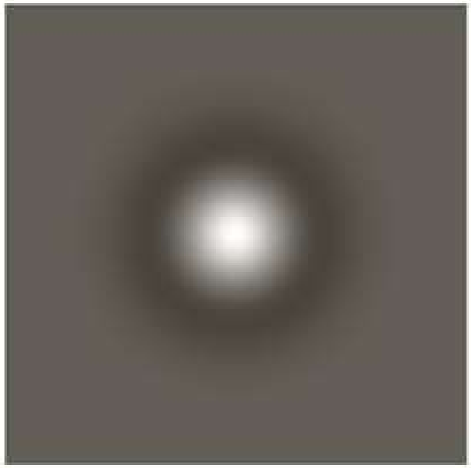
- The inverted LoG operator is commonly called a Mexican hat.
    - For example, a 5×5 discrete approximation is

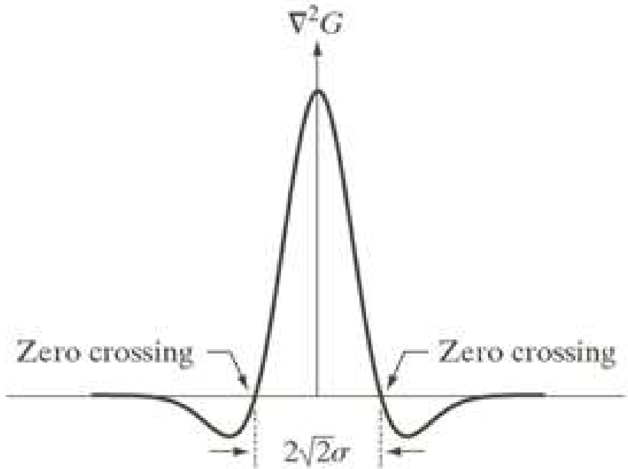
$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$



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







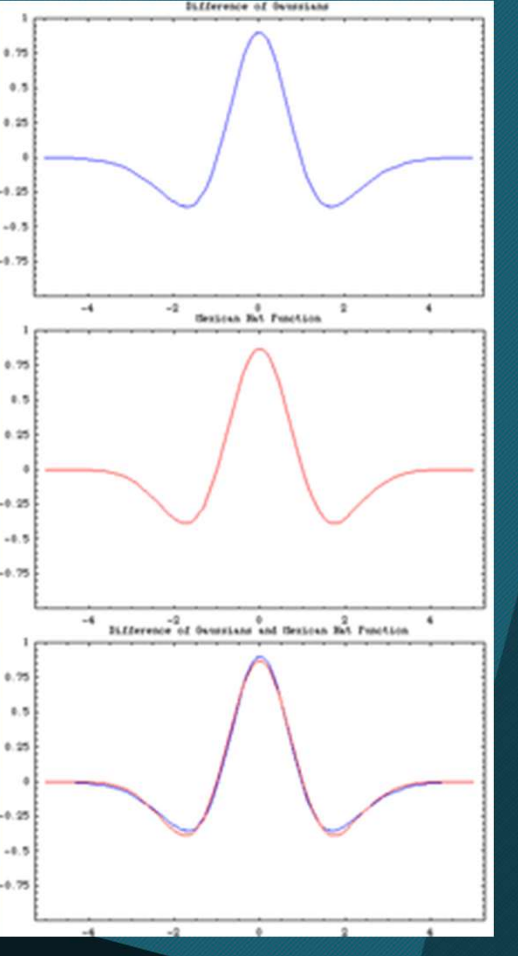
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

Zero-crossings of the second derivative

- LoG v.s. DoG
  - The $\nabla^2 G$ operator can be very effectively approximated by convolution with a mask that is the difference of two Gaussian averaging masks with substantially different σ.
  - The method is called the difference of Gaussians (DoG)



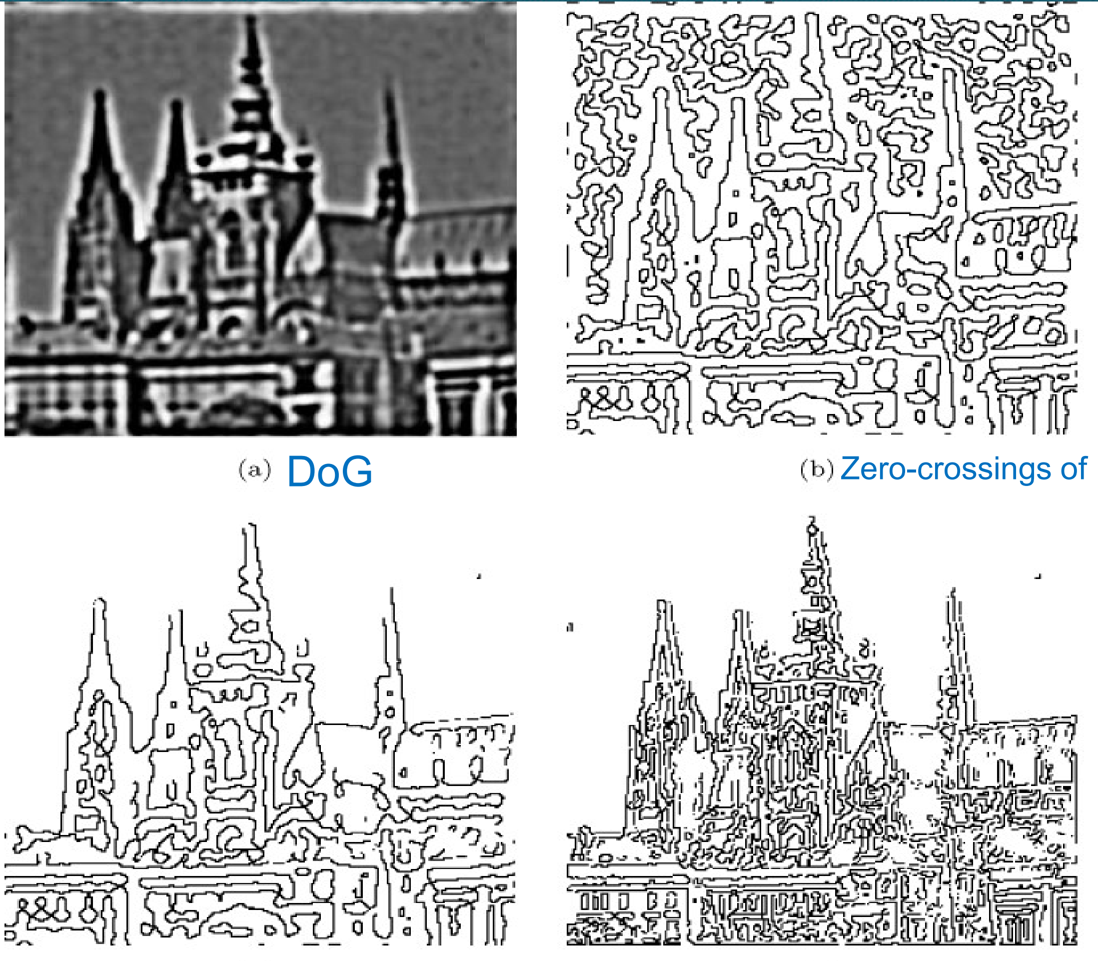
DoG

Mexican hat

Comparison of difference of Gaussian with Mexican hat wavelet
https://en.wikipedia.org/wiki/Difference_of_Gaussians

Original image

A post-processing step to avoid detection of zero-crossings corresponding to non-significant edges in regions of almost constant gray-level will admit only those zero-crossings for which there is sufficient edge evidence from a first-derivative edge detector.

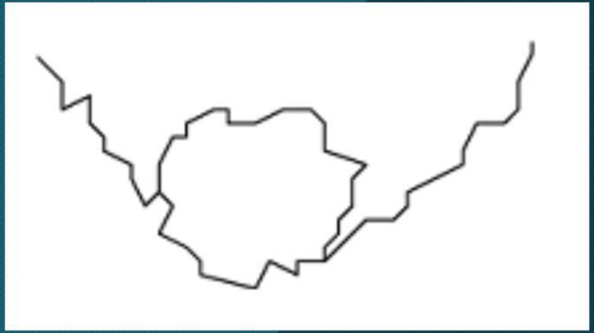


(a) DoG (b) Zero-crossings of DoG (c) DoG zc (d) LoG zc

Figure 5.21: Zero-crossings of the second derivative, see Figure 5.9a for the original image. (a) DoG image ($\sigma_1 = 0.10, \sigma_2 = 0.09$), dark pixels correspond to negative values, bright pixels to positive. (b) Zero-crossings of the DoG image. (c) DoG zero-crossing edges after removing edges lacking first-derivative support. (d) LoG zero-crossing edges ($\sigma = 0.20$) after removing edges lacking first-derivative support—note different scale of edges due to different Gaussian smoothing parameters. © *Cengage Learning 2015.*

Scale in image processing

- Different description levels are easily interpreted as different scales in the domain of digital images.
  - to eliminate fine scale noise
  - to separate events at different scales
- Scale-independent description of object can reduce the ambiguity.
- Three examples of the application of multiple scale description
  - To find the curve segments that represents the underlying structure of the scene needs to be found.
    - For example, the figure may be interpreted as a closed curve, or be described as two intersecting straight lines.



  - Scale-space filtering
  - Canny edge detector

Scale in image processing

- Scale-space filtering
  - Try to describe signals qualitatively (在質的方面) with respect to scale
  - The original 1D signal $f(x)$ is smoothed by convolution with a 1D Gaussian

$$G(x, \sigma) = e^{-x^2/2\sigma^2}$$

  - If the standard deviation σ is slowly changed, the function

$$F(x, \sigma) = f(x) * G(x, \sigma)$$

represents a surface on the (x, σ) plane that is called the scale-space image.

  - Inflexion points (反曲點) of the curve $F(x, \sigma_0)$ for a distinct value σ_0

$$\frac{\partial^2 F(x, \sigma_0)}{\partial x^2} = 0 \quad \text{and} \quad \frac{\partial^3 F(x, \sigma_0)}{\partial x^3} \neq 0$$

describe the curve $f(x)$ qualitatively (對$f(x)$質的描述).

Scale-space filtering

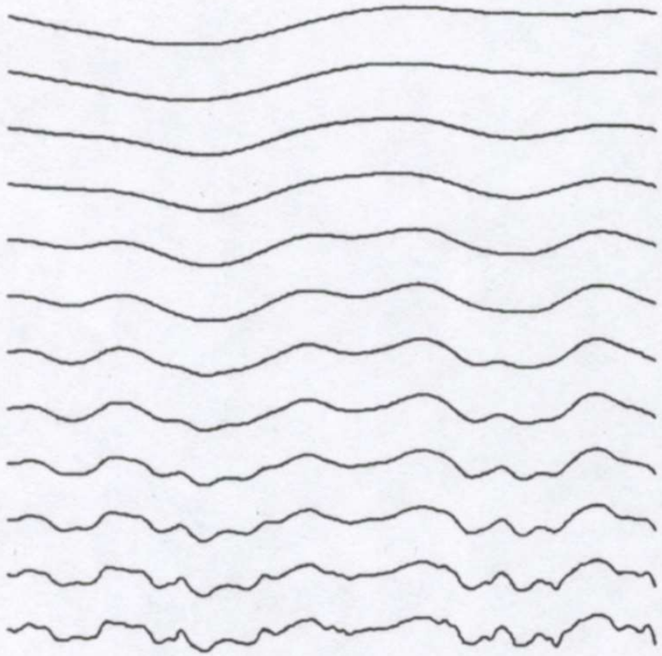


Figure 1. A sequence of gaussian smoothings of a waveform, with σ decreasing from top to bottom. Each graph is a constant-σ profile from the scale-space image.

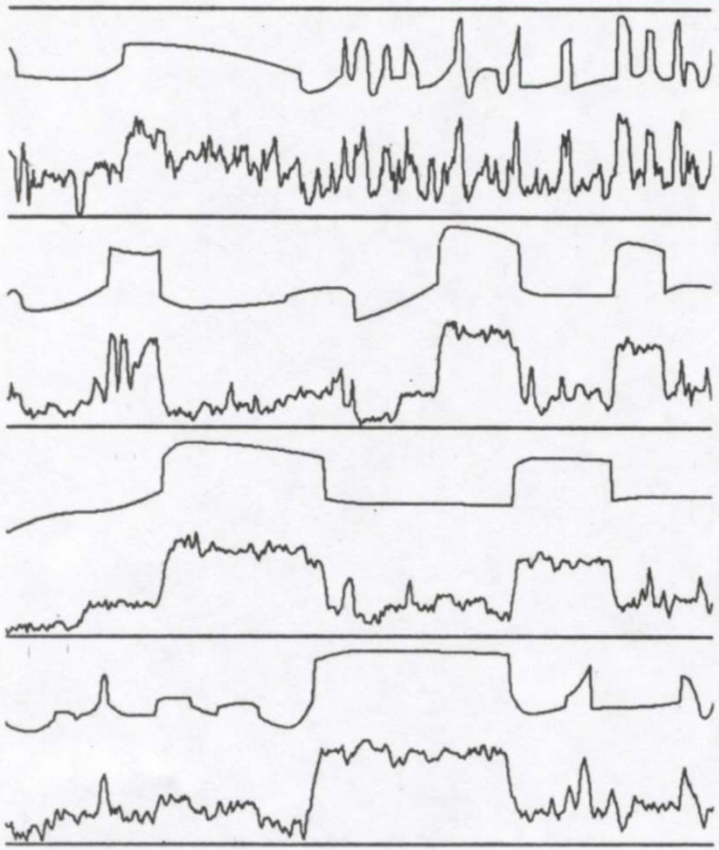


Figure 5. Several signals, with their maximum-stability descriptions. These are "top-level" descriptions, generated automatically and without thresholds. You should compare the descriptions to your own first-glance "top-level" percepts. (the noisy sine and square waves are synthetic signals.)

Andrew P. Witkin.
Fai'rchi'ld Laboratory for Artificial Intelligence Research

An example of 2D scale-space representation

Scale-space representation $L(x, y; t)$ at scale $t = 0$, corresponding to the original image f

Scale-space representation $L(x, y; t)$ at scale $t = 1$

Scale-space representation $L(x, y; t)$ at scale $t = 4$

Scale-space representation $L(x, y; t)$ at scale $t = 16$

- Larger scale t (or σ): fewer noises, less precise in location
- Smaller scale t (or σ): more noises, more precise in location

https://en.wikipedia.org/wiki/Scale_space

Canny edge detection

- Canny edge detection
  - It is optimal for step edges corrupted by white noise.
  - The optimality of the detector is related to three criteria.
    - The detection criterion
      - The important edges should not be missed.
      - There should be no spurious (假的) edges.
    - The localization criterion
      - The distance between the actual and located position of the edge should be minimal.
    - The one response criterion
      - Minimizing multiple responses to a single edge

Canny edge detection

- Suppose G is a 2D Gaussian and assume we wish to convolve the image with an operator G_n which is a first derivative of G in some direction $\mathbf{n}$

$$G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n}\nabla G \qquad (5.54)$$

We would like $\mathbf{n}$ is perpendicular (垂直) to the edge: this direction is not known in advance. (G_n: Sobel operator or Prewitt operators)

- To estimate $\mathbf{n}$: If f is the image, the normal ($\mathbf{n}$) to the edge is

$$\mathbf{n} = \frac{\nabla(G * f)}{|\nabla(G * f)|} \qquad (5.55)$$

- The edge location is then at the local maximum of the image f convolved with the operator G_n in the direction $\mathbf{n}$ (zero-crossing)

$$\frac{\partial}{\partial \mathbf{n}}(G_n * f) = 0 \qquad (5.56)$$

Canny edge detection

- Substituting in equation (5.56) for G_n from equation (5.54)

$$\frac{\partial}{\partial \mathbf{n}}(G_n * f) = \frac{\partial}{\partial \mathbf{n}}\left(\frac{\partial G}{\partial \mathbf{n}} * f\right) = \frac{\partial^2}{\partial \mathbf{n}^2}(G * f) = 0 \qquad (5.57)$$

  - This equation illustrates how to find local maxima in the direction perpendicular to the edge (non-maximal suppression—Algorithm 6.4)
    - Non-maximal suppression is an edge thinning technique.
- The magnitude of the edge is measured as

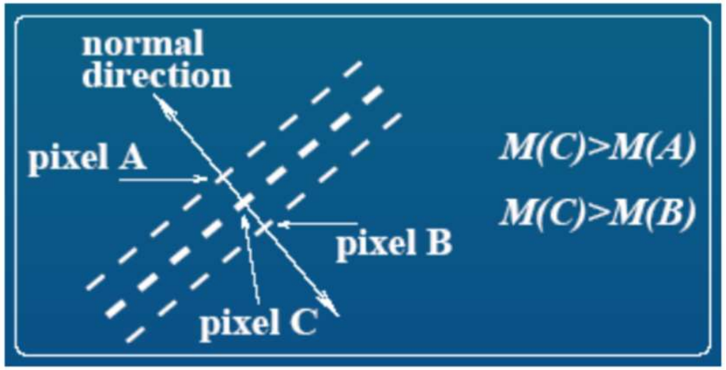
$$|G_n * f| = |\nabla(G * f)|$$

Non-maximal Suppression

- Non-maximal suppression is an edge thinning technique.
- After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. Non-maximal suppression can help to suppress all the gradient values (by setting them to 0) except the local maxima, which indicate locations with the sharpest change of intensity value.

— Non-maxima suppression

1. 對每一點C(x; y), 選定垂直於orientation方向兩個側邊的鄰近點，記作A和B；
2. 如果M(A) > M(C) or M(B) > M(C), 則C不為edge(設定M(C(x,y))=0);
3. 輸出(edge)強度影像$M_{NMS}(x; y)$



http://ccy.dd.ncu.edu.tw/~chen/course/vision/ch6/%E5%96%AE%E5%85%83%E5%85%AD%E3%80%81%E9%82%8A%E7%B7%A3%E5%81%B5%E6%B8%AC.pdf

Canny edge detection

Algorithm 5.4 Canny edge detection

1. Convolve an image f with a Gaussian of scale σ.
2. Estimate local edge normal directions $\mathbf{n}$ using equation

   $\mathbf{n} = \frac{\nabla(G*f)}{|\nabla(G*f)|}$ for each pixel in the image.
3. Find the location of the edges using $\frac{\partial^2}{\partial \mathbf{n}^2}(G * f) = 0$. (non-maximal suppression)
4. Compute the magnitude of the edge using $|G_n * f| = |\nabla(G * f)|$.
5. Threshold edges in the image with hysteresis (滞後作用) (Algorithm 6.5) to eliminate spurious (假的) responses.
6. Repeat steps (1) through (5) for ascending values of the standard deviation σ.
7. Aggregate the final information about edges at multiple scale using the 'feature synthesis(合成)' approach.

https://en.wikipedia.org/wiki/Canny_edge_detector

Hysteresis Thresholding

- **Double threshold:** Select high and low threshold values
  - If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel.
  - If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel.
  - If an edge pixel's value is smaller than the low threshold value, it will be suppressed.
- **Edge tracking by hysteresis (滯後作用)**
  - Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected.
  - To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels.
  - As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved.

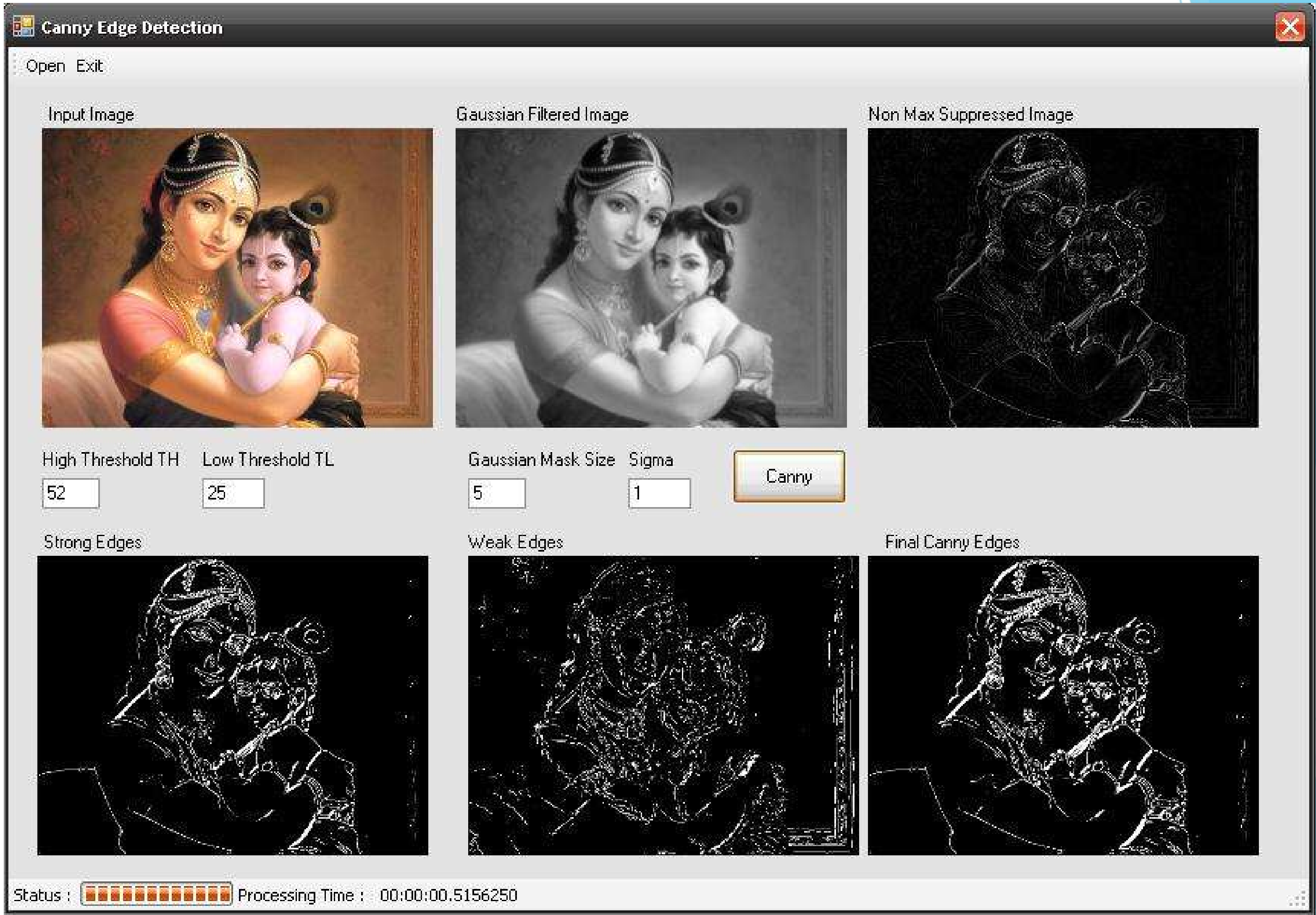
https://en.wikipedia.org/wiki/Canny_edge_detector

Canny edge detection

- Some examples

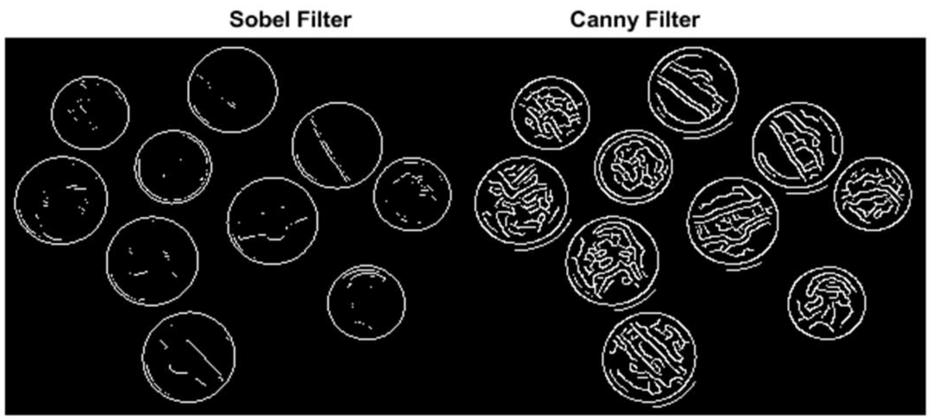
(a) $\sigma = 1.0$ (b) $\sigma = 2.8$

Figure 5.23: Canny edge detection at two different scales. © *Cengage Learning 2015.*



http://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C

Canny edge detection

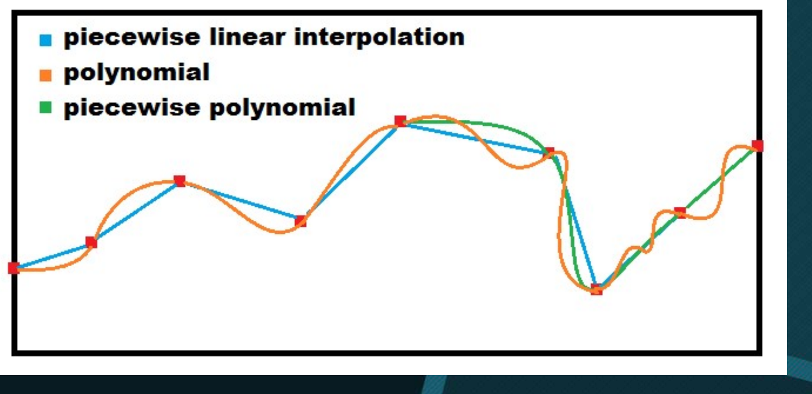


https://www.mathworks.com/examples/image/mw/images-ex48835658-detect-edges-in-images

Parametric edge models

- Parametric models are based on the idea that the discrete image intensity function can be considered a sampled and noisy approximation of an underlying continuous or piecewise continuous image intensity function.
- Piecewise (分段地) continuous function estimates called facets (小平面) are used to represent (a neighborhood of) each image pixel.
  - Such an image representation is called a facet model.
  - For example, linear, quadratic, and bi-cubic facet models.
  - An example of a bi-cubic facet model

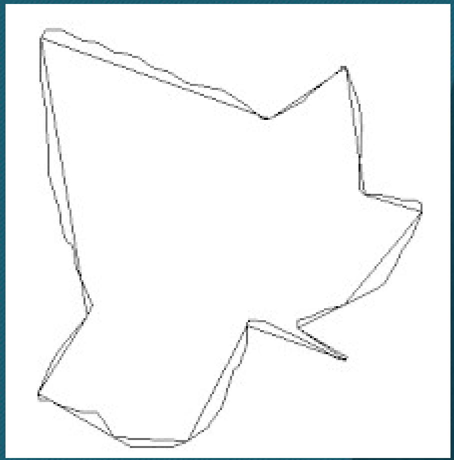
$$g(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$



https://stackoverflow.com/questions/30433391/how-can-i-produce-multi-point-linear-interpolation/30438865#30438865

Parametric edge models

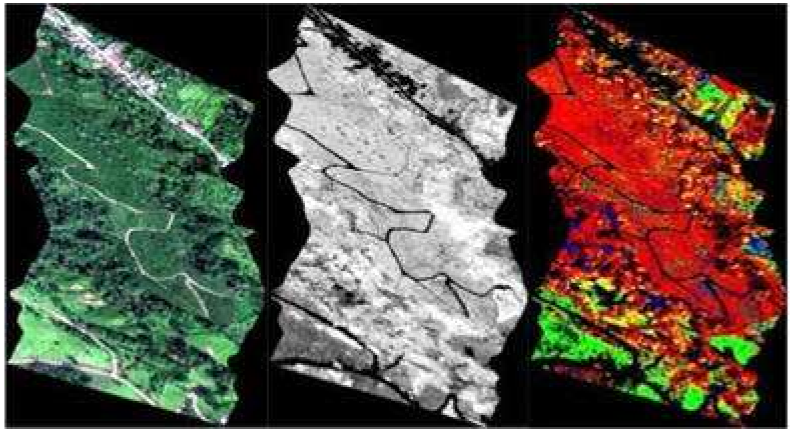
- Once the facet model parameters are available for each image pixel, edges can be detected as extrema of the first directional derivative and/or zero-crossings of the second directional derivative of the local continuous facet model functions.
- Edge detectors based on parametric models describe edges more precisely than convolution-based edge detectors.
    - They carry the potential for subpixel edge localization.
    - However, their computational requirements are much higher.



http://cs.joensuu.fi/~koles/approximation/Ch3_5.html

Edges in multi-spectral images

- One pixel in a multi-spectral image is described by an n-dimensional vector, and brightness values in n spectral bands are the vector components.
- There are several possibilities for the detection of edges in multi-spectral images.
  - Detect edges separately in individual image spectral components
  - Create a multi-spectral edge detector which uses brightness information from all n spectral bands



http://www.agricultureuavs.com/photos_multispectral_camera.htm

http://patingtoci24.soup.io/post/390611507/Download-multispectral-imaging