Image preprocessing

Filtering – edge detection

What is an edge?

- Locations in image in which the function value changes abruptly
- Important for image perception
- e.g. Line drawings, sketches ...

What is an edge?

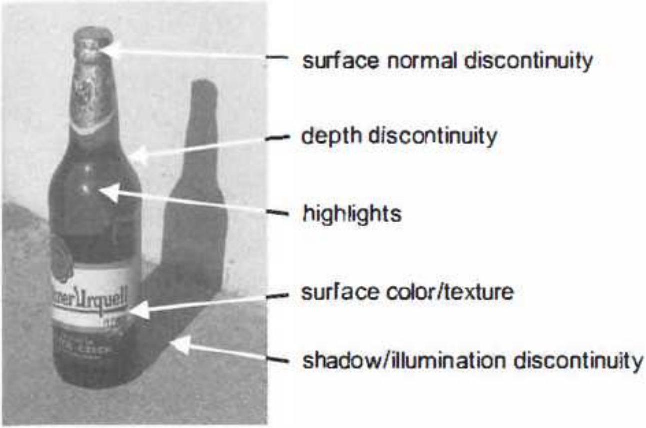


Figure 5.16: Origin of edges, i.e., physical phenomena in the image formation process which lead to edges in images.

Edge detection

- Changes in continuous functions are measured by derivatives
- Image is represented by 2-variable function, changes measured by partial derivatives.
- Change of image function can be described by gradient.

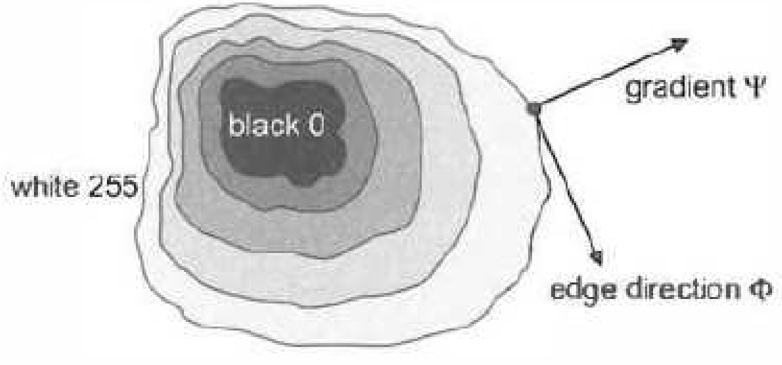
Edge detection

Edge is

- property attached to individual pixel,
- computed from image function behaviour in a neighbourhood of the pixel
- Vector variable, $(magnitude, direction)$

Edge detection

- Edge magnitude is the magnitude of the gradient,
- Edge direction is gradient direction minus 90 degrees



Edge detection

- Gradient magnitude

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

- Gradient direction

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

Edge detection

Laplacian

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

- Neglects edge orientations
- Invariant to rotation of the image

Edge detection

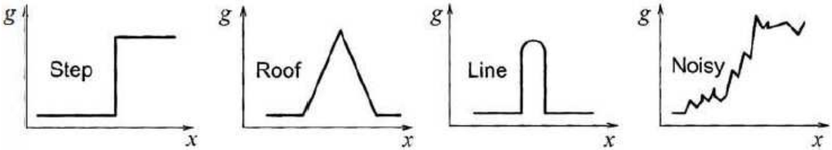


Figure 5.19: Typical edge profiles.

Edge detection

For discrete image

- Derivatives approximated by differences

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

- n is a small integer.

Categories of gradient operators

- Operators approximating derivatives of image function using differences
  - Rotationally invariant – Laplacian.
  - Approximation of first derivatives – several masks.
- Zero-crossing of the image function second derivatives
- Parametric models of edges

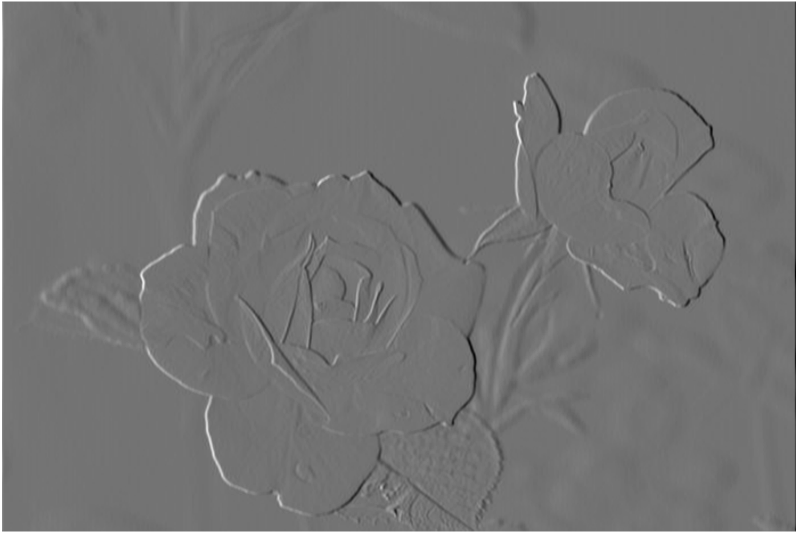
Prewitt operator

- Kernels

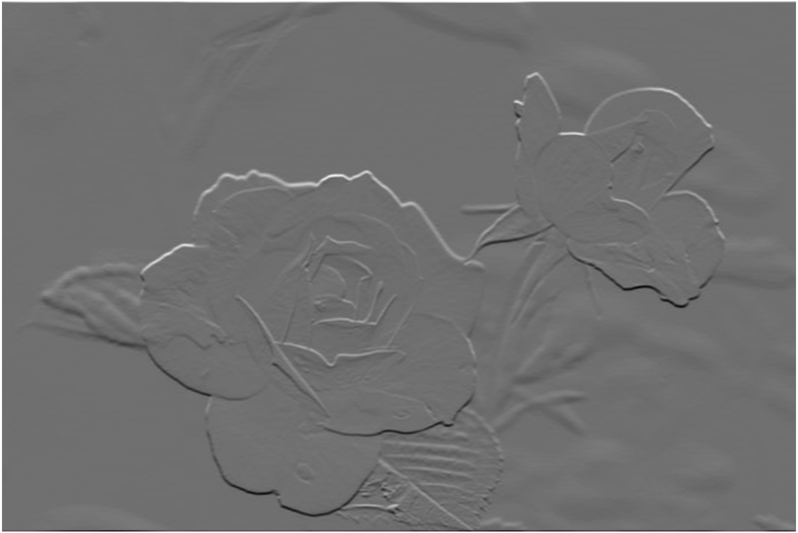
$$h_x = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \qquad h_y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

Prewitt

Prewitt



Prewitt



Prewitt

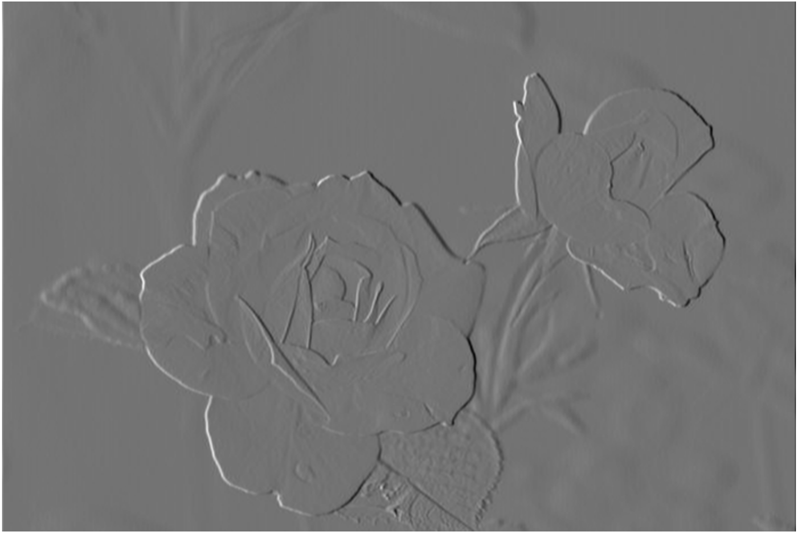
Sobel operator

- Kernels

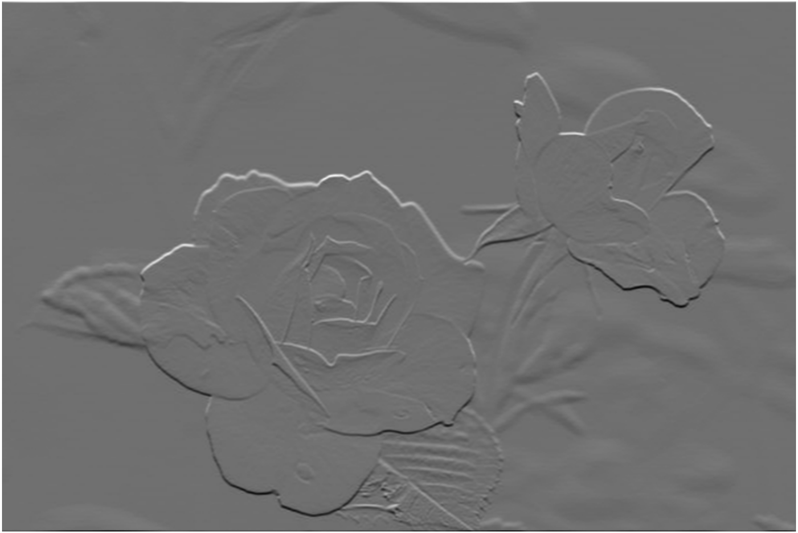
$$h_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad h_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Sobel

Sobel



Sobel



Sobel

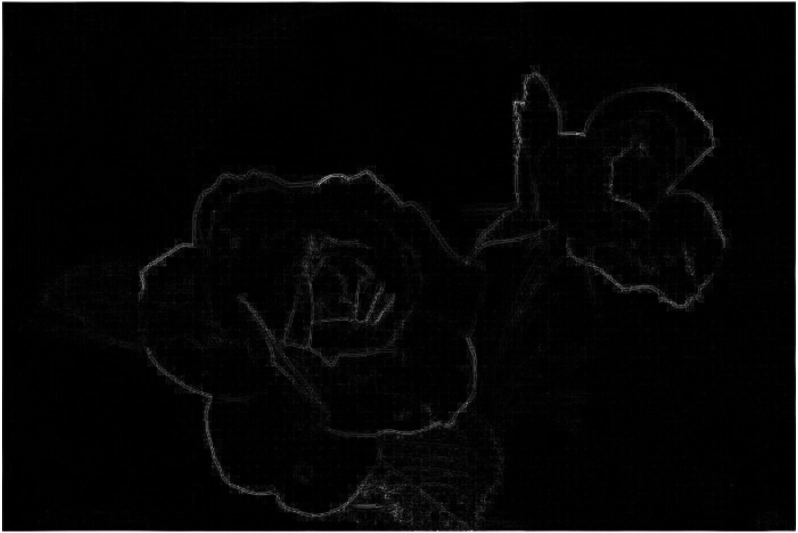
Laplace operator

- Kernels

$$h_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad h_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

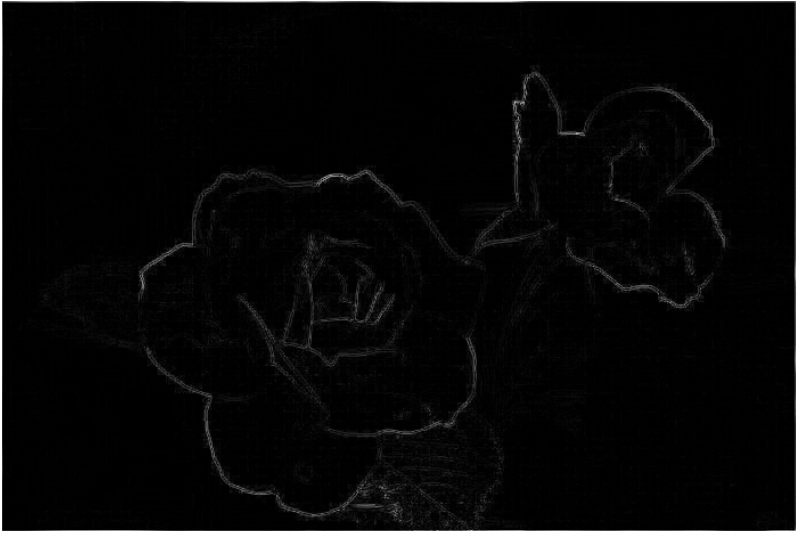
Laplace – 4-neighbourhood

Laplace – 4-neighbourhood



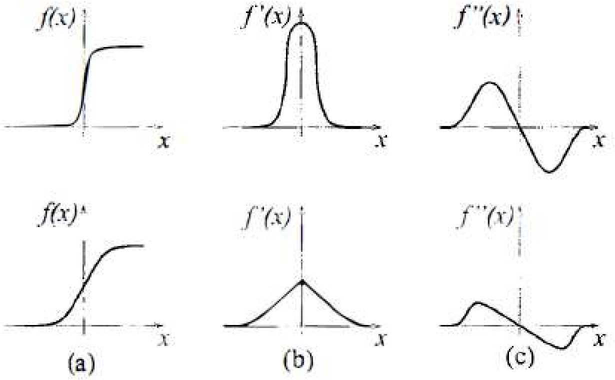
Laplace – 8-neighbourhood

Laplace – 8-neighbourhood



Zero-crossings of the second derivative

- Marr-Hildreth edge detector
- First derivative extreme corresponds to edge
- Zero of second derivative corresponds to edge



Zero-crossings of the second derivative

- Small changes (e.g. noise) create zero-crossing
- No change creates "zero-crossing".

Solution

- Smoothing
- High first derivative response at zero-crossing.

Zero-crossing

- Smoohting by Gaussian operator

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

- Smoothed image

$$G(x,y) * f(x,y)$$

- Second derivative by Laplacian

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

Zero-crossing

Convolution kernel of LoG – Laplacian of Gaussian

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

Can be approximated by difference of Gaussians (DoG)

$$h(x,y) \approx G(x,y,\sigma_1) - G(x,y,\sigma_2)$$

Zero-crossing

- Problem with zero-crossing in discrete image
- Detection of zero-crossing using small moving window
  - Same signs in window in LoG of image = no edge
  - Different signs = edge if first derivative is sufficiently high.