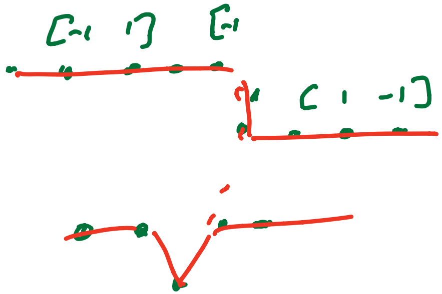
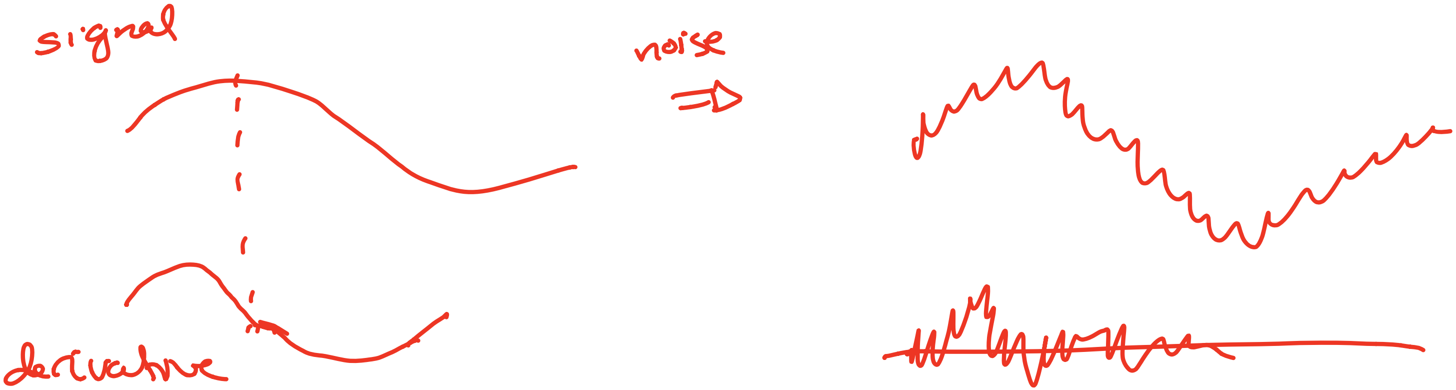
(5.3) Derivative Kernels

- We have talked about Gaussian filters and box filters, which are smoothing filters.
- Opposite of smoothing is enhancing edges/ local differences, and compute derivatives.
- The simplest approach to estimating the derivative is to compute **finite differences**.
  - Subtract one value in the signal from another.
  - Equivalent to convolving with the kernel [1 -1]



All signals and images have some noise. Therefore it is a good strategy to first smooth the signal somewhat, thereafter find the derivative.



first smooth → then derivative

Smoothing kernel $\frac{1}{2}[1 \quad 1]$ difference kernel $[1 \quad -1]$

$f(x)$

$$\left(f(x) \circledast \tfrac{1}{2}[1 \quad 1]\right) \circledast [1 \quad -1]$$

$$= f(x) \circledast \left(\tfrac{1}{2}[1 \quad 1] \circledast [1 \quad -1]\right)$$

in general

$$\frac{d}{dx}\left(f(x) \circledast g(x)\right)$$

$$= f(x) \circledast \left(\frac{d}{dx} g(x)\right)$$

$\ldots \; 0 \quad 0 \quad \frac{1}{2} \quad \frac{1}{2} \quad 0 \quad 0 \; \ldots$

$\ldots \; 0 \quad [-1 \quad 1] \quad 0 \; \ldots$

$\ldots \; 0 \quad \frac{1}{2} \quad 0 \quad -\frac{1}{2} \quad 0 \quad 0 \; \ldots$

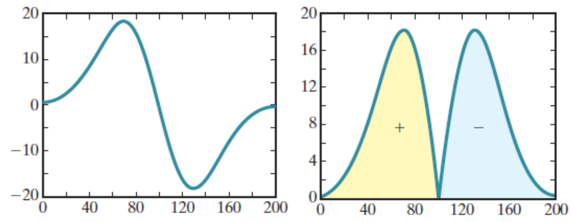
$$\Rightarrow \tfrac{1}{2}[1 \quad 0 \quad -1] \qquad f(x) \circledast \tfrac{1}{2}[1 \quad 0 \quad -1]$$

Gaussian is most used smoothing filter -> Derivative of Gaussian

$$\frac{d}{dx} gauss_{\sigma^2}(x) = \frac{d}{dx}\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}\right)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}}\left(\frac{-\cancel{2}x}{\cancel{2}\sigma^2}\right)\cdot e^{\frac{-x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} gauss_{\sigma^2}(x) = \dot{g}auss_{\sigma^2}(x)$$

Figure 5.7 The 1D Gaussian derivative (left), along with an equivalent view of the operation (right) in which a weighted sum of the values on one side are subtracted from a weighted sum of the values on the other side.



To construct a derivative kernel: sample the continous gaussian derivative and *normalize*.
Normalization: convolution with a ramp should give the slope of the ramp (the derivative!)

3x3 kernel $\sigma^2 = 0.5$

$x = -1, 0, 1$

$\dot{g}auss_{0.5} = \boxed{C} \cdot [1 \;\; 0 \;\; -1]$

$f(x) = [0 \;\; 1 \;\; 2]$

$\Rightarrow$: gradient $= 1$

normalize

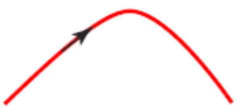
$-C \cdot 0 + 0 \cdot 1 + C \cdot 2 = 1$

$\Rightarrow$: $\boxed{C = \frac{1}{2}}$

Image gradient

- The generalization of the derivative to 2D is the **gradient**.
- The vector whose elements are the partial derivatives of the function along the two axes:

$$\nabla f(x, y) = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}\right]^{\mathsf{T}}$$



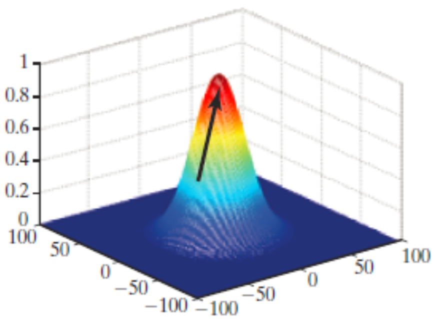
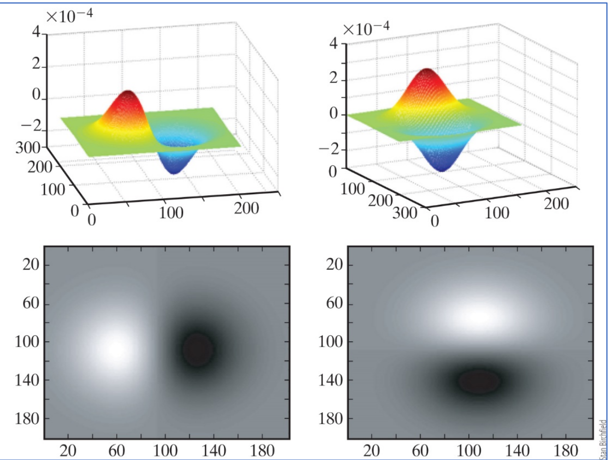


Figure 5.10 The 2D Gaussian partial derivatives in the *x* and *y* directions, shown as 3D plots (top) and images (bottom).



Gaussian in 2D:

$$Gauss(x, y) = C \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

1st. derivative in x and y direction, gradients:

$$g_x(x, y) = C_2 \cdot \boxed{x \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}}}$$

$$g_y(x, y) = C_3 \cdot y \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$\frac{d}{dx}\, gauss_{\sigma^2}(x)$$

$$= \boxed{\frac{-x}{\sigma^2}}\, gauss_{\sigma^2}(x) = \dot{g}auss_{\sigma^2}(x)$$

$$= h\; x \cdot gauss_{\sigma^2}(x)$$

What if we convolve with a gaussian to reduce noise, thereafter partial derivative (as in 1D example) :

$$\underline{\frac{\partial}{\partial x}\left[I(x,y) \circledast gaus(x,y)\right]} = I(x,y) \circledast \frac{\partial\, gaus(x,y)}{\partial x}$$

$$= C \cdot \iint -\frac{\alpha}{\sigma^2}\, \underline{I(x-\alpha, y-\beta)}\, \underbrace{e^{-\frac{(\alpha^2+\beta^2)}{2\sigma^2}}}_{e^{-\frac{\alpha^2}{2\sigma^2}} \cdot e^{-\frac{\beta^2}{2\sigma^2}}}\, d\alpha\, d\beta$$

$$= C \cdot \int -\frac{\alpha}{\sigma^2}\, e^{-\frac{\alpha^2}{2\sigma^2}} \boxed{\int I(x-\alpha, y-\beta)\, e^{-\frac{\beta^2}{2\sigma^2}}\, d\beta}\, d\alpha$$

$\hookrightarrow$ gauss smoothing in the y-direction

$$\Rightarrow \underline{\left[I(x,y) \circledast gauss_{\sigma^2}(y)\right] \circledast \dot{g}auss_{\sigma^2}(x)} = fx \qquad \underline{fy}$$

- Once we have computed the gradient of the image it is often desirable to compute the magnitude of the gradient and the phase of the gradient.

$$|\nabla f| = \sqrt{f_x^2 + f_y^2} \approx |f_x| + |f_y| \approx max(|f_x|, |f_y|)$$

Euclidean manhattan chessboard

$$\phi(x, y) = arctan\frac{f_y}{f_x}$$

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Prewitt operator

- The simplest 2D differentiating kernel is the **Prewitt operator**.
  - Obtained by convolving a 1D Gaussian derivative kernel with a 1D box filter in the orthogonal direction:

$$Prewitt_x = \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} \circledast \frac{1}{2}\begin{bmatrix}1 & 0 & -1\end{bmatrix} = \frac{1}{6}\begin{bmatrix}1 & 0 & -1\\1 & 0 & -1\\1 & 0 & -1\end{bmatrix}$$

$$Prewitt_y = \frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix} \circledast \frac{1}{2}\begin{bmatrix}1\\0\\-1\end{bmatrix} = \frac{1}{6}\begin{bmatrix}1 & 1 & 1\\0 & 0 & 0\\-1 & -1 & -1\end{bmatrix}$$

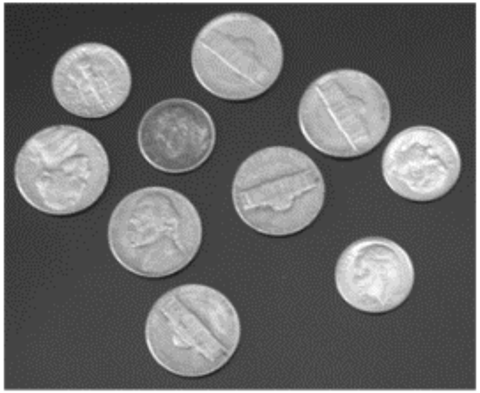
Sobel operator

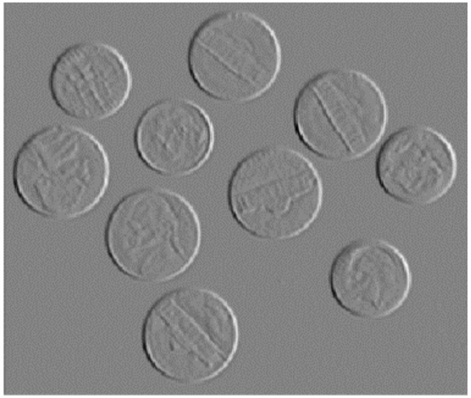
- The **Sobel operator** is more robust, as it uses the Gaussian $(\sigma^2 = 0.5)$ for the smoothing kernel:

$$Sobel_x = gauss_{0.5}(y) \circledast \dot{g}auss_{0.5}(x) = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \circledast \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

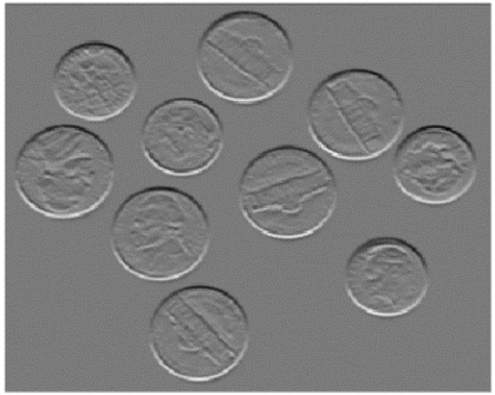
$$Sobel_y = gauss_{0.5}(x) \circledast \dot{g}auss_{0.5}(y) = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \circledast \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel operator





Δfx(i,j) Smooths over columns, derivative over rows.



Δfy(i,j) Smooths over rows, derivative over columns.

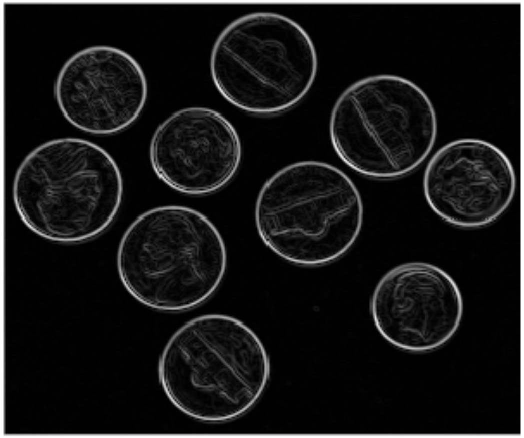
$$G(i,j) = \sqrt{\Delta f_x(i,j)^2 + \Delta f_y(i,j)^2}$$

Gradient image

$$\theta(i,j) = \mathrm{atan}\frac{\Delta f_x(i,j)}{\Delta f_y(i,j)}$$

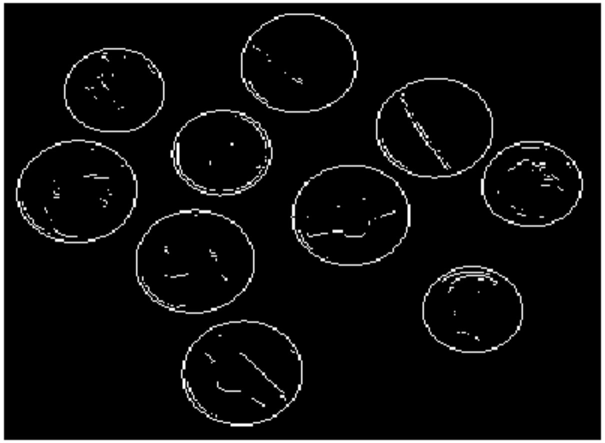
Angle image

Sobel – gradient image and edge map



$$G(i,j) = \sqrt{\Delta f_x(i,j)^2 + \Delta f_y(i,j)^2}$$

magnitude of gradient



Combining thresholded gradient image with angle information can give binary image/edge map.

Example: used in Canny – edge detection

1. Image is smoothed by Gaussian filter to reduce noise
2. Local gradient and edge direction are found (by sobel / prewitt)
3. The ridges of the gradient image is tracked, set to zero all pixels not on the ridge top -> thin line   Thereafter, hysteresis threshold

   Ridge pixel > T2 -> strong edge pixel

   T1<Ridge pixel<T2 -> weak edge pixel

4. Edge linking -> include weak edge pixels that are 8- connected to strong edge pixels.