Problem 1

First-order edge detection

Robert filter calculates diagonal edge gradients (Equation 4), susceptible to fluctuations and provides no information about edge orientation.

$$\begin{cases} G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * I \\ G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * I \end{cases} \tag{1}$$

Gradient map is composed of

$$\nabla I(x,y) = G(x,y) = \sqrt{G_x^2 + G_y^2} \tag{2}$$

and the edge map is determined by

$$E = \begin{cases} 1, 2 \text{ S.D.}(\sim 5\%) \\ 0, \text{otherwise} \end{cases} \tag{3}$$

Prewitt filter is a discrete differentiation operator, which behaves similar to the Sobel operator by computing the gradient for the image intensity function as well. It makes use of the maximum directional gradient. Though it is easy to implement, it is very sensitive to noise as well.

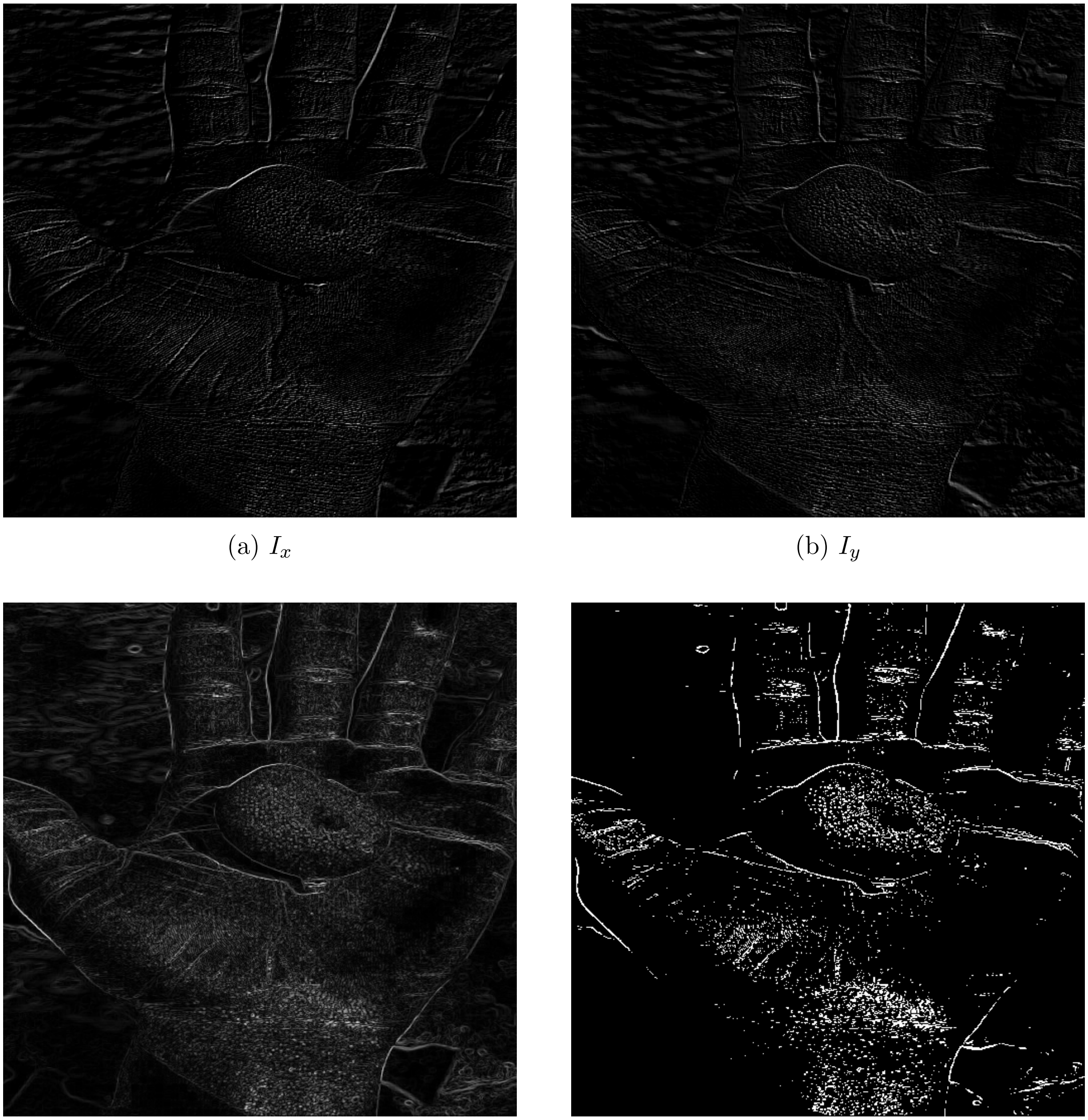
$$\begin{cases} G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I \\ G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * I \end{cases} \tag{4}$$

The final gradient result calculates through Equation 2 and Equation 3 as well.

Noted that since Prewitt kernel can be decomposed, G_x can be written as

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * I \tag{5}$$

which essentially means that it computes the gradient with smoothing, since it can be decomposed as the products of an averaging and a differential kernel.



(a) I_x

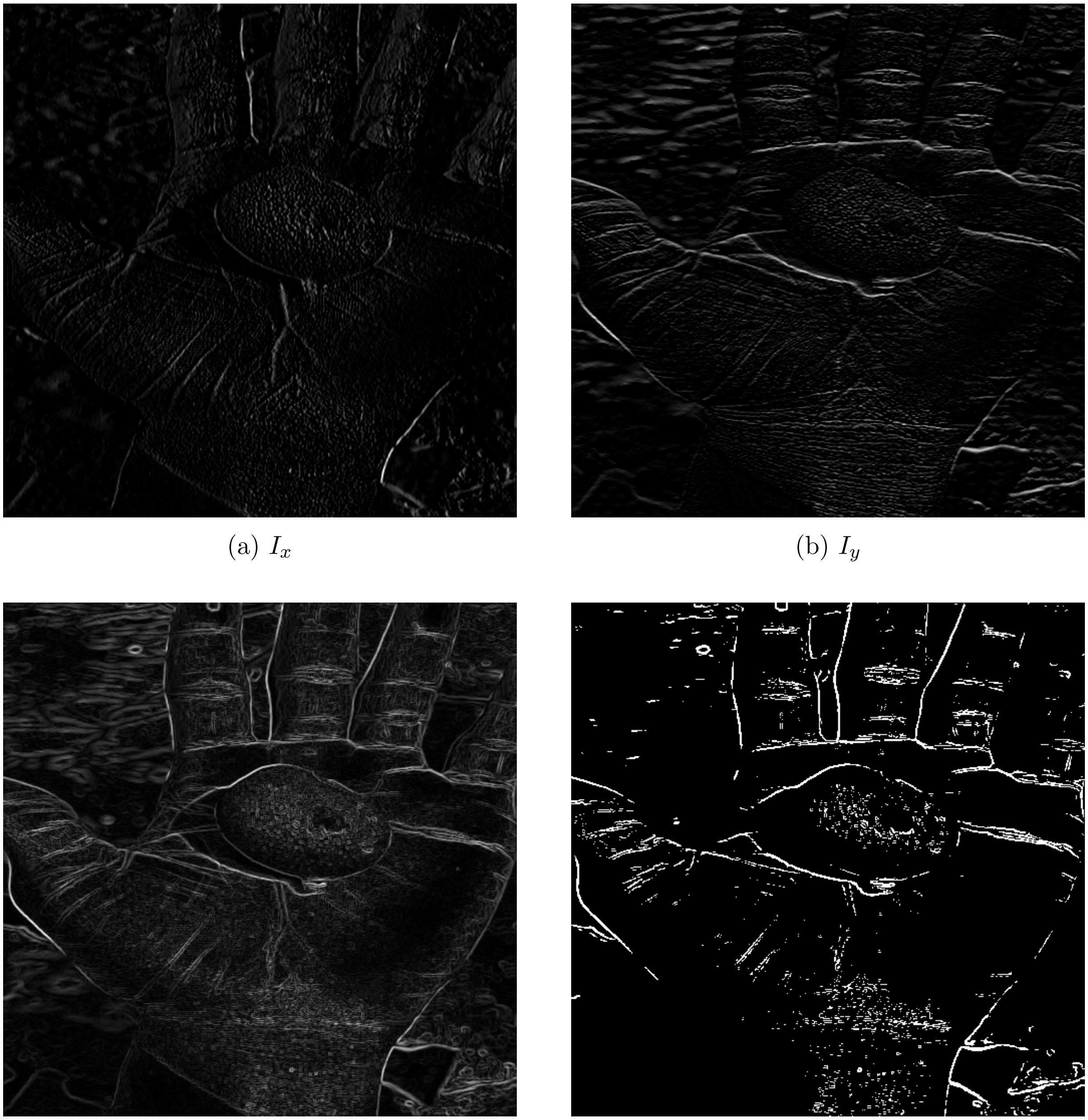
(b) I_y

(c) I_{xy}

(d) Edge map

Figure 1: Robert operator

Sobel filter detects edges where the gradient magnitude is high. This makes the Sobel edge detector more sensitive to diagonal edge than horizontal (Figure 3a) and vertical edges (Figure 3b). Both Sobel filter and Prewitt filter are very effective at providing good edge maps.



(a) I_x

(b) I_y

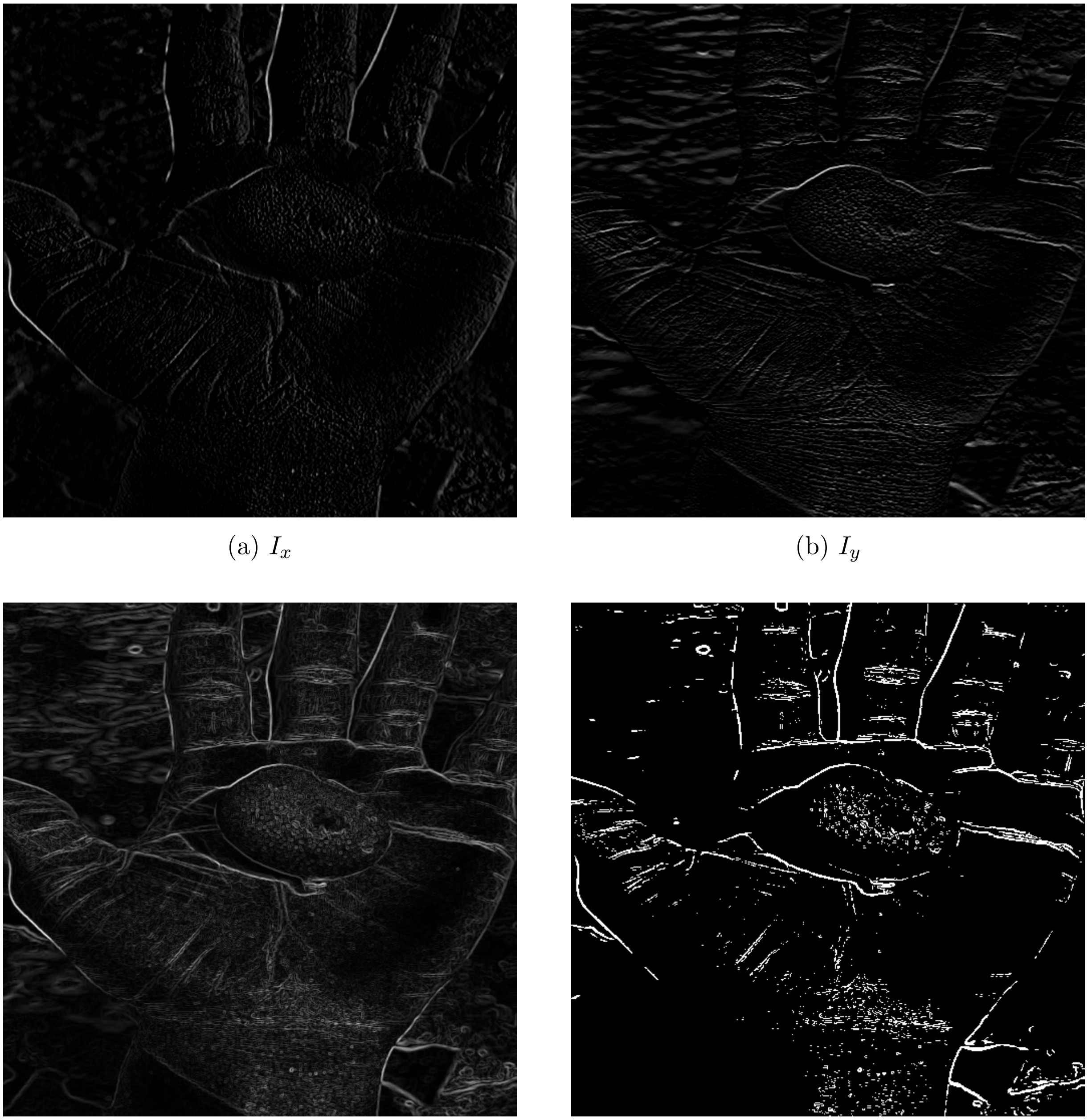
(c) I_{xy}

(d) Edge map

Figure 2: Prewitt operator

$$\begin{cases} G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I \\ G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \end{cases} \tag{6}$$

The final gradient result calculates through Equation 2 and Equation 3 as well.



(a) I_x

(b) I_y

(c) I_{xy}

(d) Edge map

Figure 3: Sobel operator

Second-order edge detection

Laplacian of Gaussian Laplacian filters are derivative filters used to find areas of rapid changing edges in images.

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{7}$$

Since derivative filters are very sensitive to noise, it is common to smooth the image using a Gaussian filter before applying the Laplacian.

A possible approximation for the effect of the Laplacian is

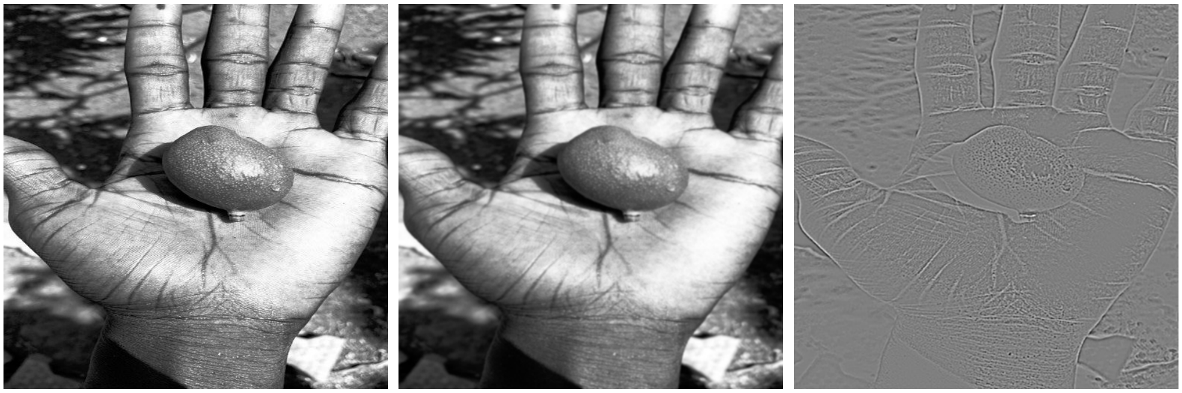
$$K_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{8}$$

One can reverse the sign of the elements of this negative Laplacian, but it does not affect the outcome.

To include a smoothing Gaussian lowpass filter, combine the Laplacian and Gaussian functions to obtain the result.

$$K_{LoG} = -\frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{9}$$

To utilize functions from previous homework, the calculation is still split instead of using a single kernel.



(a) I_1 (b) $LPF(I_1)$ (c) $LoG(I_1)$

Figure 4: LoG

Both Figure 4b and Figure 4c uses kernel size of 5 and $\sigma = 1$. Figure 4c is ranged from $[-1, 1]$.

Difference of Gaussians

Canny edge detection

Step 1: Noise reduction

Step 2: Compute gradient magnitude and orientation

Step 3: Non-maximal suppression

Step 4: Hysteretic thresholding

Step 5: Connected component labeling

Edges of noisy image

Problem 2

Edge crisping

Warping

Bonus