# DIP HW2 Report

R07922036 邱能賢

March 27, 2019

## Problem 1: Edge Detection

(a) Perform first order edge detection as image $E_1$.

To perform first order edge detection, I used Roberts cross differentiation and only kept its magnitude.

The threshold is calculated by first calculate the CDF and set the threshold at the intensity at 80% of pixels.

(b) Perform second order edge detection as image $E_2$.

To perform second order edge detection, I first convolute the image with a low-pass filter then with a separable 8-neighbor laplacian filter.

$$Gaussian = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}, H = \frac{1}{8}\begin{bmatrix}-2 & 1 & -2\\ 1 & 4 & 1\\ -2 & 1 & -2\end{bmatrix}$$

Then, with a threshold of $\pm 2.5$, set the value to $\pm 1$ if above the threshold, otherwise 0. At last, check zero-crossing considering the 8-neighbor (four directions).

(c) Perform Canny edge detection as image $E_2$.

Canny edge detection is done as the following method:

(a) Noise reduction with Gaussian filter:

$$Gaussian = \frac{1}{159}\begin{bmatrix}2 & 4 & 5 & 4 & 2\\ 4 & 9 & 12 & 9 & 4\\ 5 & 12 & 15 & 12 & 5\\ 4 & 9 & 12 & 9 & 4\\ 2 & 4 & 5 & 4 & 2\end{bmatrix}$$

(b) Compute gradient magnitude and orientation with Sobel filter:

$$G_x = \begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}, G_y = \begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$$

(c) Non-maximal suppression:

The orientation is modded to $[0, \pi/128]$ and the nearest neighbor is calculate by interpolation of the 8-neighbor.

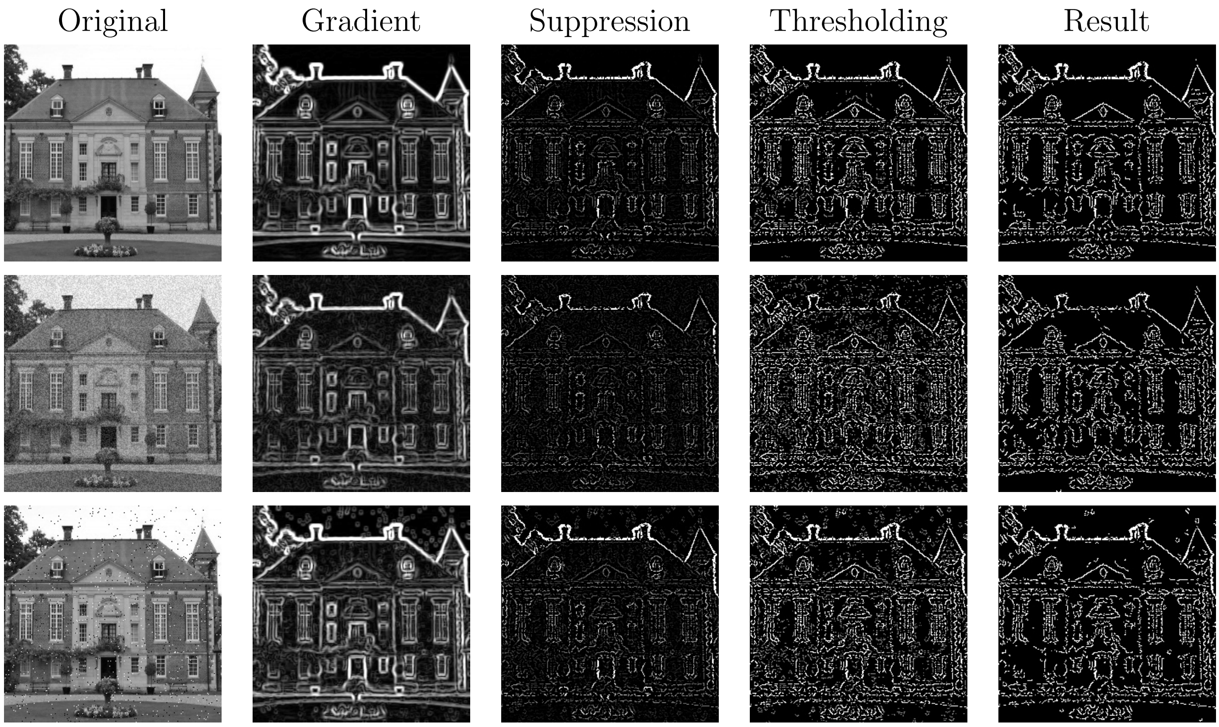


Table 1: Result after each step of Canny edge detection.

(d) Hysteretic thresholding:

The higher threshold is set to the intensity at 90% of CDF, and the lower threshold is the half of the higher value.

(e) Connected component labeling method:

Using breadth-first search considering all the 8-neighbor.

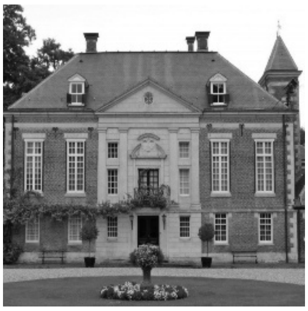
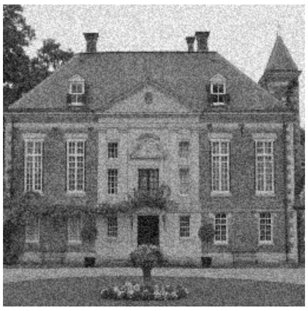
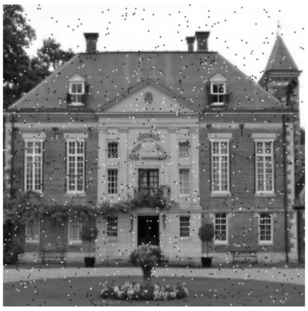
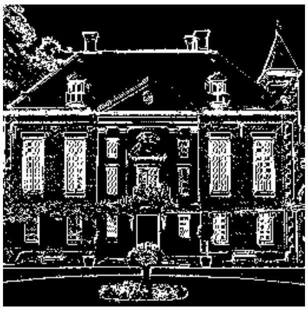
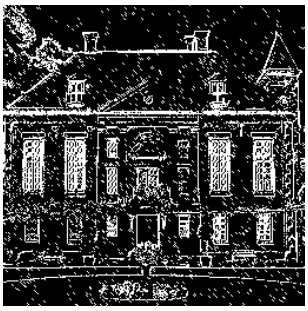
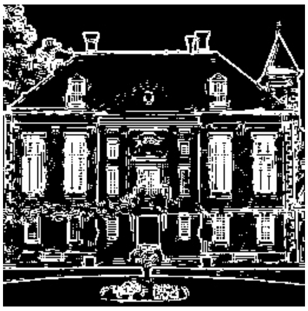
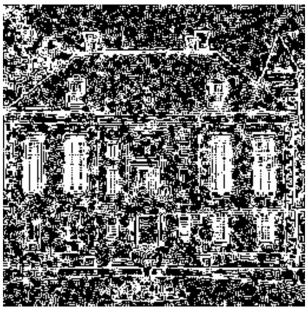
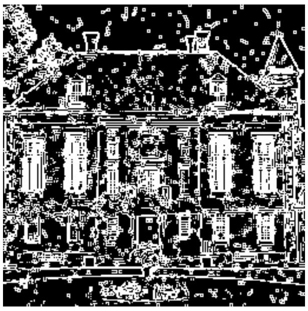
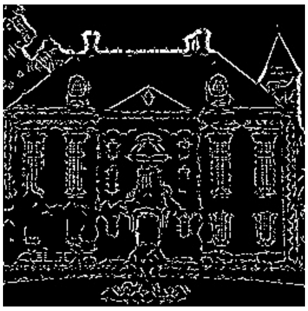
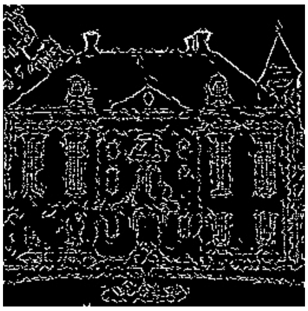
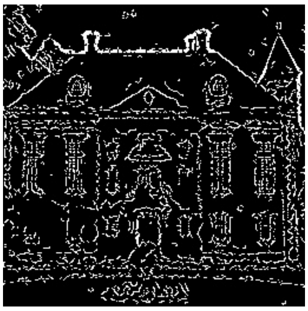
|  | sample1.raw | sample2.raw | sample3.raw |
|---|---|---|---|
| original |  |  |  |
| $E_1$ |  | |  |
| $E_2$ |  |  |  |
| $E_3$ |  |  |  |

Table 2: Result of different edge detecting method.

## Problem 2: Noise Removal

(a) Perform edge crispening on $I_2$ as image $C$.

To perform edge crispening, I simple convolute the image with the following high pass filter, which showed the best result in the textbook:

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

As a result, I think not only the edge is crispened, but also the contrast is increased.

$I_2$, sample4.raw

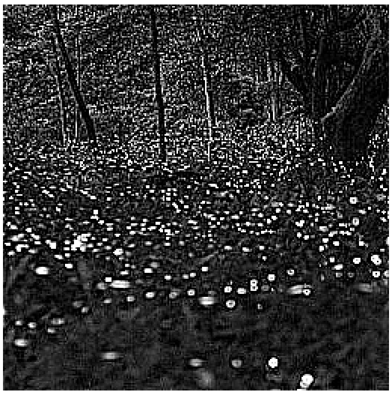


image $C$

(b) Design a warping method to produce $D$ from $C$. $D$ is a swirled disk with diameter of 256 pixels.

For coordinate $(i, j)$ relative to the center of the image:

(a) Wrap to a circle:

$$r = \sqrt{i^2 + j^2}, \theta = atan(\frac{j}{i})$$

$$(r, \theta) \rightarrow (r \times \max(|\cos(\theta)|, |\sin(\theta)|), \theta)$$

(b) Swirl transformation, $\rho$ is the radius of the image:

$$(r, \theta) \rightarrow (r, (\text{swirl angle}) \times (1 - \frac{r}{\rho})\theta), \rho = 128$$

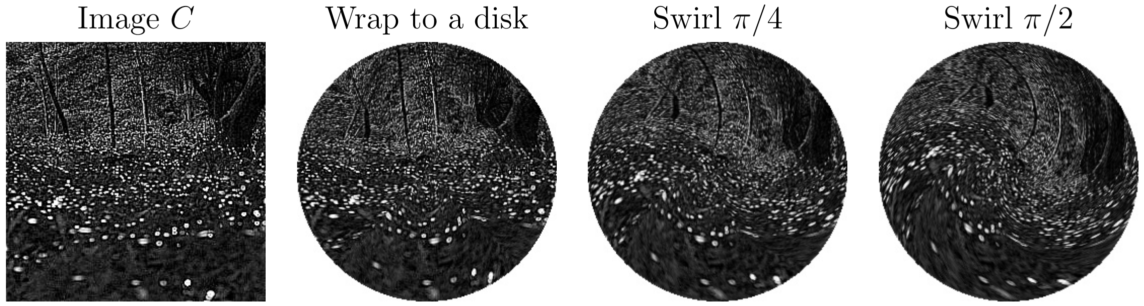


Image $C$ Wrap to a disk Swirl $\pi/4$ Swirl $\pi/2$