Computer Vision I: Homework 10

Department: CSIE Student ID: d11922016 Name: Jia-Wei, Liao

November 26, 2022

1 Problem description

Write the following programs to detect edge and use zero-crossing on the following four types of images to get edge images.

1. Laplacian (threshold: 15)
2. Minimum-variance Laplacian (threshold: 30)
3. Laplace of Gaussian (threshold: 3000)
4. Difference of Gaussian (threshold: 1)

2 Functions

2.1 Laplae Operator

```
function out_image = laplae_operator(image, kernel, threshold)
[m, n] = size(image);
[km, kn] = size(kernel);
out_m = m - (km-1);
out_n = n - (kn-1);
image = double(image);
out_image = zeros(out_m, out_n);
[km, ~] = size(kernel);
pad_pixel = (km-1) / 2;

for i = 1:out_m
    for j = 1:out_n
        local_image = get_local_image(image, kernel, i, j);
        value = convolution(local_image, kernel);

        if value >= threshold
            out_image(i, j) = 1;
        elseif value <= -threshold
            out_image(i, j) = -1;
        else
            out_image(i, j) = 0;
        end

    end
end

out_image = zero_crossing(out_image, pad_pixel);
out_image = uint8(out_image);

end
```

2.2 Zero-Crossing

Given the threshold t. A pixel $I(i, j)$ have a zero-crossing if it satisfies one of the following condition:

1. $I(i,j) \leq -t$ and $|\{I(p,q) \geq t \mid (p,q) \in \mathcal{N}_8(i,j)\}| \neq \varnothing$,
2. $I(i,j) \geq t$ and $|\{I(p,q) \leq -t \mid (p,q) \in \mathcal{N}_8(i,j)\}| \neq \varnothing$

```
function out_image = zero_crossing(image, pad_pixel)
[m, n] = size(image);
out_image = constant(m, n, 255);
image = padding(image, pad_pixel);

for i = 1:m
    for j = 1:n
        neighbor = get_neighbor(image, i+pad_pixel, j+pad_pixel, 1);

        if image(i+pad_pixel, j+pad_pixel) == 1 && is_value_in_array(neighbor, -1)
            out_image(i, j) = 0;
        end

    end
end

end
```

3 Experiment result

The kernel-based operator is an approximation of the gradient of image intensity. Given the following kernels, we can detect the edge by using the convolution algorithm.

3.1 Laplacian Operator

The Laplacian operator is

$$\Delta = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

The kernel 1 of Laplacian is

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 1: edge of image with Laplacian mask 1 of threshold 15

The kernel 2 of Laplacian is

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2: edge of image with Laplacian mask 2 of threshold 15

3.2 Minimum-Variance Laplacian Operator

The kernel of minimum-variance Laplacian is

$$\frac{1}{3}\begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

Figure 3: edge of image with minimum-variance Laplacian mask 2 of threshold 30

3.3 Laplace of Gaussian (LoG) Operator

The Laplace of Gaussian (LoG) operator is defined as

$$\mathrm{LoG}(x, y, \sigma) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

The kernel of Laplace of Gaussian is

$$\begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\ 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\ -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\ -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\ -2 & -9 & -23 & -1 & 103 & 178 & 103 & -1 & -23 & -9 & -2 \\ -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\ -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\ 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\ 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4: edge of image with Laplace of Gaussian mask of threshold 3000

3.4 Difference of Gaussian Operator

The Difference of Gaussian (DoG) operator is defined as

$$\mathrm{DoG}(x, y, \sigma_1, \sigma_2) = \mathrm{LoG}(x, y, \sigma_1) - \mathrm{LoG}(x, y, \sigma_2)$$

The kernel of Laplace of Gaussian with inhibitory $\sigma = 3$, excitatory $\sigma = 1$ is

$$\begin{bmatrix} -1 & -3 & -4 & -6 & -7 & -8 & -7 & -6 & -4 & -3 & -1 \\ -3 & -5 & -8 & -11 & -13 & -13 & -13 & -11 & -8 & -5 & -3 \\ -4 & -8 & -12 & -16 & -17 & -17 & -17 & -16 & -12 & -8 & -4 \\ -6 & -11 & -16 & -16 & 0 & 15 & 0 & -16 & -16 & -11 & -6 \\ -7 & -13 & -17 & 0 & 85 & 160 & 85 & 0 & -17 & -13 & -7 \\ -8 & -13 & -17 & 15 & 160 & 283 & 160 & 15 & -17 & -13 & -8 \\ -7 & -13 & -17 & 0 & 85 & 160 & 85 & 0 & -17 & -13 & -7 \\ -6 & -11 & -16 & -16 & 0 & 15 & 0 & -16 & -16 & -11 & -6 \\ -4 & -8 & -12 & -16 & -17 & -17 & -17 & -16 & -12 & -8 & -4 \\ -3 & -5 & -8 & -11 & -13 & -13 & -13 & -11 & -8 & -5 & -3 \\ -1 & -3 & -4 & -6 & -7 & -8 & -7 & -6 & -4 & -3 & -1 \end{bmatrix}$$

Figure 5: edge of image with Difference of Gaussian mask of threshold 1

4 Summary

In this homework, we use kernel-based methods to detect the edge of the image. The kernel with the convolution operator can detect the edge pattern, and we use zero-crossing algorithm to get the binary image which represent the edge of the image.