Department of Computer Science
Faculty of Science
Prince of Songkla University

…11st.. April 2018
…10:00-11:50…

LAB 10th **Edge detection**

Objectives

The goal of this tutorial is to learn how to implement edge detection and associated techniques in MATLAB.

1. Learn how to use the IPT edge function.
2. Explore the most popular first-derivative edge detectors: Roberts, Sobel, and Prewitt.

Procedure

1. From first-derivative, we can operate with digital image, *f(x,y)* as the following

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\right) \quad (1)$$

and we can find the magnitude of the gradient operator by

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (2)$$

For the gradient direction

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right) \quad (3)$$

A simple window gradient, Roberts edge detector is implemented as the following

$$|\nabla f(x,y)| = |f(x,y) - f(x-1,y-1)| + |f(x,y-1) - f(x-1,y-1)| \quad (4)$$

In matlab, it can implement the Eq. (4) by the following function.

```
function [Z]=Gradient2DRobert(f)
        % f : array of digital data, size M×N

        [M,N] = size(f);
        for x = 2:M
                for y=2:N
                        Z(x,y) = abs(f(x,y)-f(x-1,y-1))+abs(f(x,y-1)-f(x-1,y-1));
                end
```

```
        end
end
```

Use function Gradient2DRobert with the following commands.

```
>>f=imread('cameraman.tif');
>>g = Gradient2DRobert(f);
>>figure,imshow(g/max(g(:)))
```

And comparison the image result by the following instructions.

```
>>g = Gradient2DRobert(double(f));
>>figure,imshow(g/max(g(:)))
```

Question 1) Analyze the image results by considering how a difference between Gradient2DRobert(f) and Gradient2DRobert(double(f)).

2. From Eq. (4), we can operate by calculating with horizontal and vertical directions

$$\nabla_R f(x,y) = f(x,y) - f(x-1,y-1) \tag{5}$$

$$\nabla_C f(x,y) = f(x,y-1) - f(x-1,y-1) \tag{6}$$

We can formulate the gradient magnitude of Eq. 2 by the following function

```
function [Z]=Gradient2DRobert1(f)
        % f : array of digital data, size M×N
         [M,N] = size(f);
        for x = 2:M
                for y=2:N
                        r = f(x,y)-f(x-1,y-1);
                        c = f(x,y-1)-f(x-1,y-1);
                        Z(x,y) = sqrt(r*r + c*c);
                end
        end
end
```

Consider the image result from the following instructions.

```
>>g = Gradient2DRobert1(double(f));
>>figure,imshow(g/max(g(:)))
```

From Eq. (5) and (6), the linear combination of 2×2 windows

$$W_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad W_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

The windows can implement with the following function.

```
function [Z]=Gradient2D(f, w)
        % f : array of digital data, size M×N
        % w: array of gradient operator, size P×Q
        [M,N] = size(f);
        [P,Q] = size(w);
        P = floor(P/2);
        Q = floor(Q/2);
        for x = 2:M
                for y=2:N
                        X = f(x-P:x, y-Q:y);
                        Y = X.*w;
                        Z(x,y) = sum(Y(:));
                end
        end
end
```

Run each instructions

2.1 g1 = Gradient2D(double(f), w1);

2.2 g2 = Gradient2D(double(f), w2);

2.3 figure, imshow(g1/max(g1(:)))

2.4 figure, imshow(g2/max(g2(:)))

2.5 g3 = sqrt(g1.*g1 + g2.*g2);

2.6 figure, imshow(g3/max(g3(:)))

Question 2) Describe commands 2.1 to 2.6, what are they using for?

3. From the step 2, it is known that by the name of convolution, which can use imfilter replacing Gradient2D

3.1 Use imfilter function operating the commands 2.1 to 2.6.

3.2 Run w1 = fspecial('sobel'); to defend Sobel windows w2 = w1'; and following by the operating the commands as formulating in 3.1.

3.3 Use edge function to find the edge image by the given commands:

>> e = edge(f, ‘sobel’);

>> figure, imshow(e);

Question 3) Use the following instructions and describing the image results:

```
I = imread('circuit.tif');
BW1 = edge(I,'prewitt');
BW2 = edge(I,'canny');
figure, imshow(BW1)
figure, imshow(BW2)
```

Question 4) Use the following instructions and describing the image results:

```
>> BW2 = edge(I,'canny', [0.1 0.25], 0.75);
>> figure, imshow(BW2)
>> BW2 = edge(I,'canny', [0.1 0.25], 1.5);
>> figure, imshow(BW2)
>> BW2 = edge(I,'canny', [0.1 0.25], 1);
>> figure, imshow(BW2)
>> BW2 = edge(I,'canny', [0.1 0.35], 1);
>> figure, imshow(BW2)
```