



## ปฏิบัติการการประมวลผลภาพดิจิทัลครั้งที่ 10 Edge detection

### วัตถุประสงค์

เพื่อเรียนรู้การวิเคราะห์ข้อมูลภาพระดับล่างว่ามีความจำเป็นอย่างไร

1. ศึกษาลักษณะขอบวัตถุที่ปรากฏในภาพ และทราบถึงความสำคัญของขอบวัตถุ
2. เรียนรู้วิธีการหาขอบวัตถุด้วยเทคนิคต่างๆ

### การทดลอง

1. จากนิยามของอนุพันธ์อันดับที่หนึ่งที่ใช้ดำเนินการกับข้อมูลดิจิทัลสองมิติ  $f(x,y)$  ได้กำหนดไว้ดังนี้

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

และสามารถหาขนาดของอนุพันธ์ได้จาก

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

สำหรับทิศทางของเวกเตอร์เกรเดียนต์คือ

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

ตัวดำเนินการเกรเดียนต์ของวินโดว์ขนาดเล็ก ที่ใช้หาขอบของวัตถุในภาพ เช่นวิธีของ Robert มีวิธีการหาขอบดังนี้

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

เมื่อนำตัวดำเนินการในสมการที่ (4) มาใช้กับข้อมูลดิจิทัลสองมิติ มีขั้นตอนวิธีดังนี้

**function** [Z]=Gradient2DRobert(f)

% f : array of digital data, size M×N

% w: array of gradient operator, size P×Q

[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

```

ให้ทดลองประมวลผลอัลกอริทึม Gradient2DRobert โดยมีอินพุตของฟังก์ชันคือ

```

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

```

คำถาม 1) ให้วิเคราะห์ผลภาพจากคำสั่งทั้งสองกรณี จงอธิบายความเหมือนและความแตกต่างของผลการดำเนินการ Gradient2DRobert(f); กับ Gradient2DRobert(double(f));

2. จากสมการที่ (4) สามารถแยกสมการออกเป็นสองส่วน คือส่วนที่ดำเนินการตามแถว และส่วนที่ดำเนินการตามคอลัมน์ที่กำหนดได้ดังนี้

$$\nabla_R f(x, y) = f(x, y) - f(x - 1, y - 1) \quad (5)$$

$$\nabla_C f(x, y) = f(x, y - 1) - f(x - 1, y - 1) \quad (6)$$

โดยสามารถคำนวณหาขนาดของเกรเดียนต์ตามสมการที่ (2) ได้ดังนี้

```

function [Z]=Gradient2DRobert1(f)
% f : array of digital data, size M×N
% w : array of gradient operator, size P×Q
[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

```

ทดลองประมวลผลอัลกอริทึมโดยคำสั่ง

```

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

```

จากสมการที่ (5) และ (6) ที่อยู่ในรูปของการรวมเชิงเส้น สามารถกำหนดเป็นวินโดว์ขนาด 2×2 ได้ดังนี้

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

เมื่อเขียนโปรแกรมรองรับการประมวลผลด้วยวินโดว์ได้ดังนี้

```

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

```

ทดลองประมวลผลอัลกอริทึมโดยคำสั่ง

```

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(:)))

```

คำถาม 2) จงอธิบาย คำสั่งที่ 2.1 ถึงคำสั่งที่ 2.6 เป็นการดำเนินการเพื่ออะไร

3. จากข้อ 2. ที่มีการประมวลผลโดยใช้การรวมเชิงเส้นระหว่างข้อมูลภาพกับตัวดำเนินการเกรเดียนต์ ที่กำหนดเป็นวินโดว์ การประมวลผลภาพการดำเนินการดังกล่าวจะเรียกว่าคอนโวลูชัน ที่สามารถเรียกใช้ฟังก์ชัน imfilter แทนฟังก์ชัน Gradient2D

3.1 จงดำเนินการทดลองตามขั้นตอนที่เป็นคำสั่ง 2.1 ถึง 2.6 โดยใช้ฟังก์ชัน imfilter

3.2 ใช้คำสั่ง w1 = fspecial('sobel'); เพื่อสร้างตัวดำเนินการหาขอบแบบ Sobel โดย w2 = w1'; แล้วดำเนินการเช่นเดียวกับข้อ 3.1

3.3 ใช้คำสั่ง edge ในการหาขอบด้วยวิธีเดียวกันกับข้อ 3.2 โดยใช้คำสั่ง

```

>> e = edge(f, 'sobel');
>> figure, imshow(e);

```

3.4 ให้ดำเนินการตามคำสั่งต่อไปนี้ แล้วอธิบายความแตกต่างของผลลัพธ์ที่ได้

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

3.5 ให้ดำเนินการตามคำสั่งต่อไปนี้ แล้วอธิบายความแตกต่างของผลลัพธ์ที่ได้

```
>> 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)
```