

実践演習 1-1

```
(ア) size(P,'c')  
(イ) :,i
```

実践演習 1-2

```
[mindist, ans] = min(sqrt(sum((P-repmat(x,[1,size(P,'c')]))).^2,'r'))
```

実践演習 1-3

```
clear;  
P = [[0,1,1,1,0,...  
      1,0,0,0,1,...  
      1,0,0,0,1,...  
      1,0,0,0,1,...  
      0,1,1,1,0]','...  
      [0,0,1,0,0,...  
      0,0,1,0,0,...  
      0,0,1,0,0,...  
      0,0,1,0,0,...  
      0,0,1,0,0]','...  
      [0,1,1,1,1,...  
      1,0,0,1,0,...  
      0,0,1,0,0,...  
      0,1,0,0,0,...  
      1,1,1,1,1]','...  
      [0,1,1,1,0,...  
      1,0,0,0,1,...  
      0,0,1,1,0,...  
      1,0,0,0,1,...  
      0,1,1,1,0]','...  
      [0,0,1,0,0,...  
      0,1,0,0,0,...  
      1,0,0,1,0,...  
      1,1,1,1,1,...  
      0,0,0,1,0]'];  
  
x = [0,0,0,1,0,...  
      0,0,0,1,0,...  
      0,0,0,1,0,...  
      0,0,0,1,0,...  
      0,0,0,1,0]';  
  
function feature = feature_extraction(data)  
    feature = [];  
    for i = 1:size(data, 'c')  
        img = matrix(data(:,i), 5, 5)';  
        feature = [feature, [detect_line(img), detect_line(img')]]';  
    end  
endfunction  
  
function val = detect_line(m)  
    val = 0;  
    for i = 1:size(m,'c')  
        if regexp(strcat(string (m(:,i))), '111/') > 0  
            val = val + 1;  
        end  
    end  
endfunction  
  
F = feature_extraction(P);
```

```
x2 = feature_extraction(x);

[mindist, ans] = min(sqrt(sum((F-repmat(x2,[1,size(F,'c')]))).^2,'r')));
disp("Ans = "+string(ans-1))
```

実践演習 2-1

- (ア) y, x
- (イ) y-1:y+1, x-1:x+1

実践演習 2-2

```
clear;

// 画像データの読み込み
im = im2double(imread('test1.pgm'));
// 2次元配列 im のサイズ取得
[h w] = size(im);
// 結果格納用の配列 resultim を用意
resultim1 = ones(im);
resultim2 = ones(im);

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim1(y, x) = median(im(y-1:y+1, x-1:x+1));
        resultim2(y, x) = mean(im(y-1:y+1, x-1:x+1));
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim1, resultim2])
imwrite([im, resultim1, resultim2], 'out.png');
```

実践演習 2-3

```
clear;

// 画像データの読み込み
im = im2double(imread('test1.pgm'));
// 2次元配列 im のサイズ取得
[h w] = size(im);
// 結果格納用の配列 resultim を用意
resultim = ones(im);
// Sobelフィルタの定義
dx=[-1,0,1; -2,0,2; -1,0,1];
dy=[1,2,1; 0,0,0; -1,-2,-1];

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim(y, x) = sqrt(sum(im(y-1:y+1, x-1:x+1) .* dx).^2+...
            sum(im(y-1:y+1, x-1:x+1) .* dy).^2);
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim])
imwrite([im, resultim], 'out.png');
```

実践演習 2-4

```
clear;

// 画像データの読み込み
im = im2double(imread('test1.pgm'));
// 2次元配列 im のサイズ取得
[h w] = size(im);
// 結果格納用の配列 resultim を用意
resultim1 = ones(im);
resultim2 = ones(im);
// Sobelフィルタの定義
dx=[-1,0,1; -2,0,2; -1,0,1];
dy=[1,2,1; 0,0,0; -1,-2,-1];

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim1(y, x) = sqrt(sum(im(y-1:y+1, x-1:x+1) .* dx)^2+...
            sum(im(y-1:y+1, x-1:x+1) .* dy)^2);
    end
end

// maxプーリング
for y = 2:h-1
    for x = 2:w-1
        resultim2(y, x) = max(resultim1(y-1:y+1, x-1:x+1));
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim1, resultim2]);
imwrite([im, resultim1, resultim2], 'out.png');
```
