ELL 205 PROJECT
Palm-Print Recognition

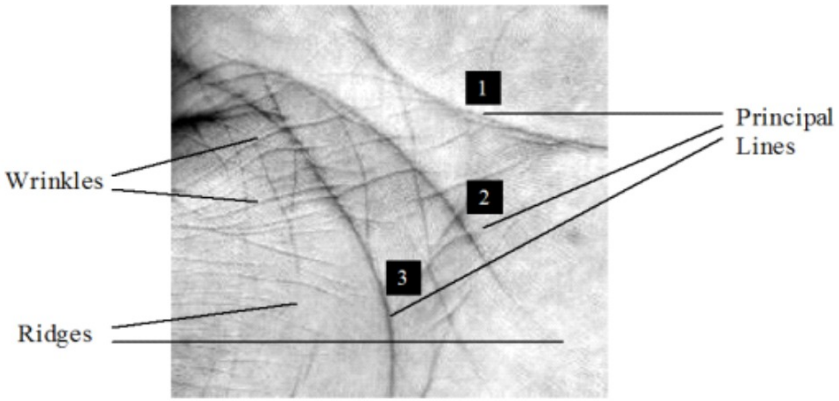
BY -

Shubham Mittal(2018EE10957)
Simran Rathore(2018EE30566)

What is Palm-Print Recognition

• Palm print recognition is a biometric authentication method based on the unique patterns of various characteristics in the palms of people's hands.

• The inner surface of palm contains principal lines, wrinkles, ridges, singular points, dat



Applications –

• For individual identification as the palm print is unique to individuals.

•To correlate palm patterns with medical disorders, e.g. genetic disorders and
Downs syndromes.

How is this done ?

Palmprint Acquisition → Feature Extraction/ Image Segmentation → Matching Features

Applications of Image Segmentation

- Biometrics
- Medical imaging tasks
- Automated traffic control system
- Identifying objects in a scene for object-based measurements such as size and shape
- Identifying objects which are at different distances from a sensor using depth measurements

Image Segmentation

1. **Thresholding**

   In thresholding we set a cut-off for the pixels of the image with respect to the pixels of the palm.

2. **Edge Detection**

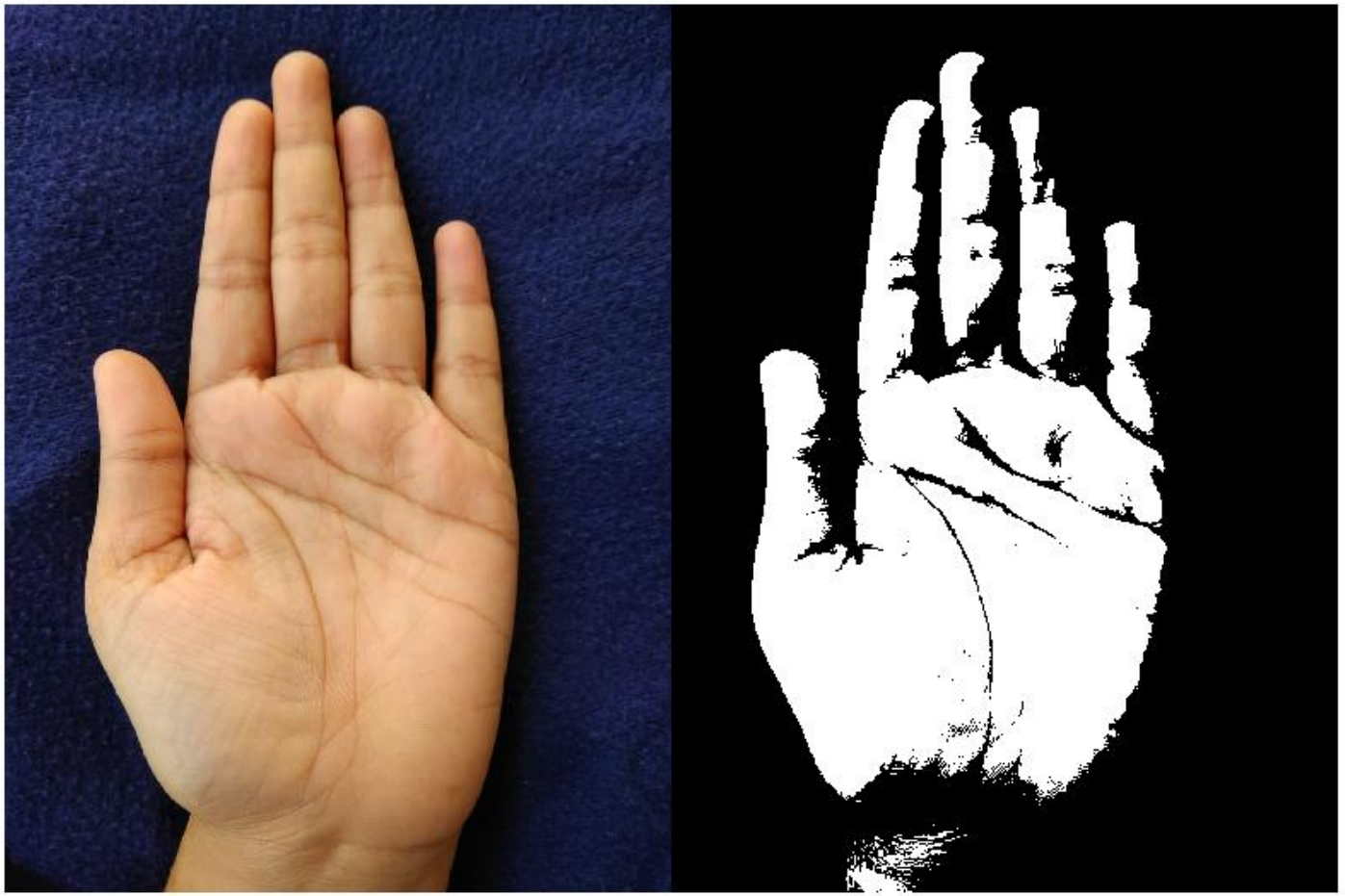
   At edges there are discontinuities in pixels which are impulses in time domain and are of high frequencies in Fourier domain.

Thresholding

Method 1 : Skin colour thresholding

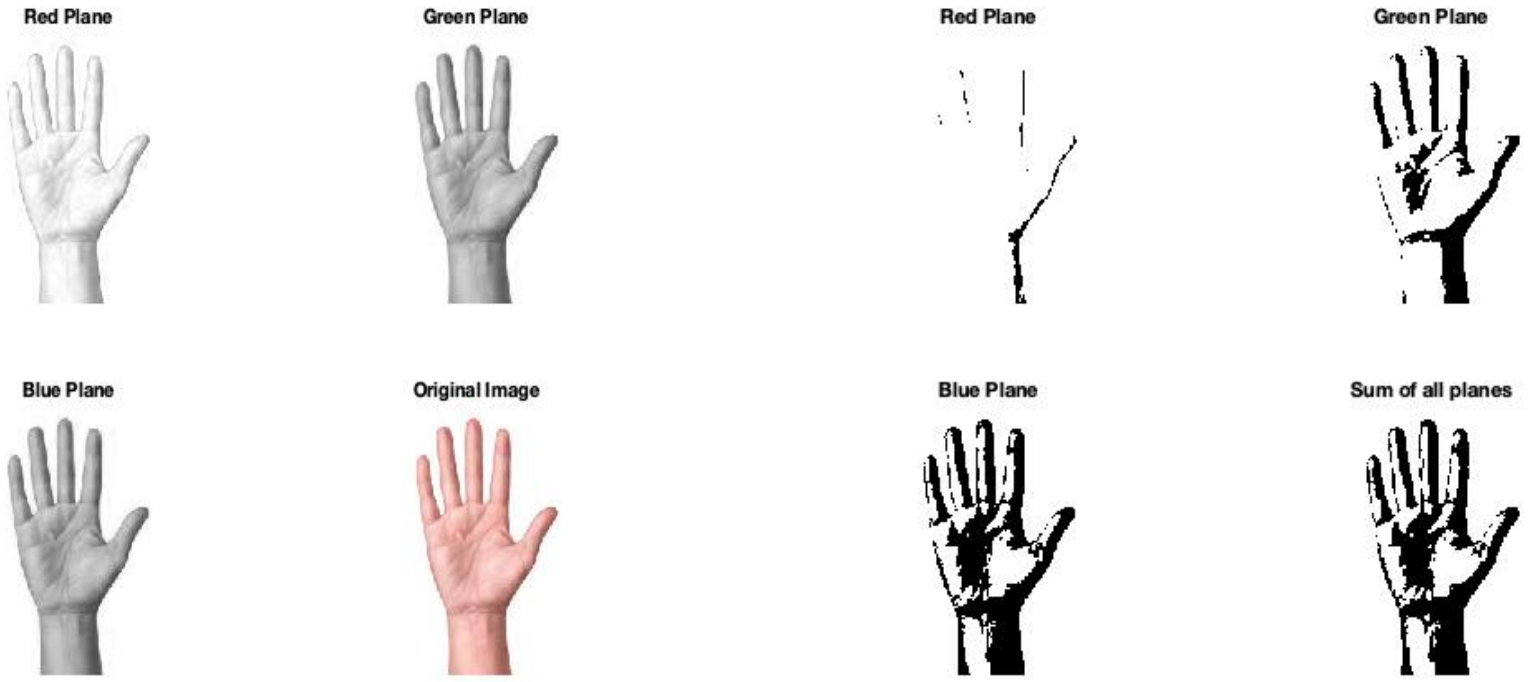
Pixel value at wrinkles and ridges of palm are lower than the overall palm. So if we set some right level then features can be extracted from the palm.

```
%% Read in image
img = imread('im1.jpeg');
imshow(img);

%% Convert to grayscale image
imggray = rgb2gray(img);
imshow(imggray);
% note here pixelXpixel not X3 since only one plane

%% Problem: illumination does not allow proper segmentation
level = 0.7; % belongs in [0,1] ; use increment and run option by right click
img_threshED = im2bw(imggray,level);
imshowpair(imggray,img_threshED,'montage');
```



Method 2 : Thresholding in RGB planes separately

- Separate RGB image into its three planes
- Apply threshold at each plane
- Sum all the three planes



Method 3 : Double thresholding

- Set a minimum and maximum threshold with respect to palm pixels
- Reset the pixels of the image which doesn't fall in this range

```
%% Read in image
img1 = imread('im1.jpeg');
imshow(img1);
```

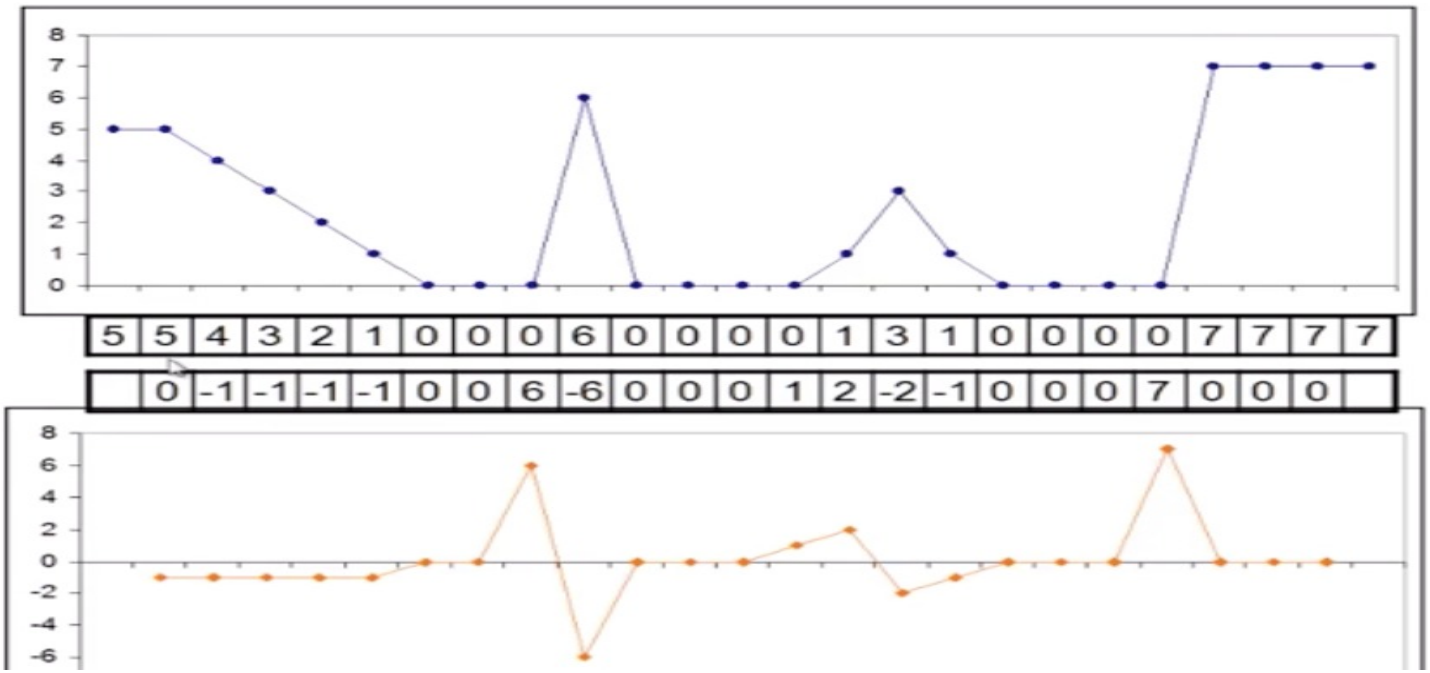
```
%% double threshing
T1 = 196;
T2 = 270;

NewImage = uint8(255 * (img1 < T1 | img1 > T2));
Compl = imcomplement(NewImage);


subplot(1,2,1), imshow(img1);
title('Original Image');
subplot(1,2,2), imshow(Compl);
title('Threshed Image');
```

Original Image

Threshed Image

Edge Detection

- Edges in any image are associated with high frequency in Fourier domain.
- So we pass the image through high pass filter.



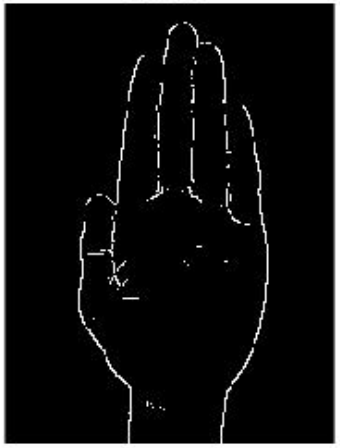
Edge Detection using inbuilt filters

Prewitt

Canny

Sobel

Roberts



Edge Detection using High Pass Filter

1st Derivative Filtering

For a function $f(x, y)$ the gradient of f at coordinates (x, y) is given as the column vector:

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of this vector is given by:

$$\nabla f = mag(\nabla \mathrm{f})$$

$$= \left[G_x^2 + G_y^2\right]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

-1	0	1
-1	0	1
-1	0	1

Gx (X derivative)

-1	-1	-1
0	0	0
1	1	1

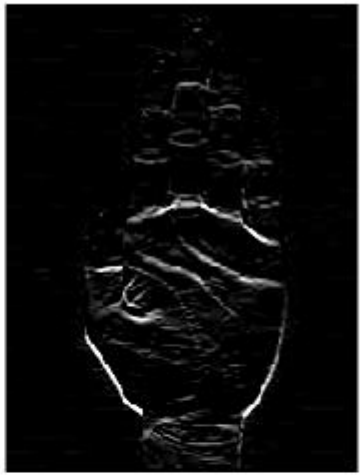
Gy (Y derivative)

V= (Gx ² +Gy ²) ^0.5

To filter an image it is filtered using both operators the results of which are added together

Original

Along X



Along Y

Sum

Original

Along X

Along Y

Sum

```
%% read image
img = imread('im4.jpeg');
b = rgb2gray(img);
imshow(b);
```

```
%% mask 1
f1 = [-1 0 1;-1 0 1;-1 0 1];
imgX = filter2(f1,b); % filter2 is a 2D digital filter

f2 = f1';
imgY = filter2(f2,b);

imgEdge = sqrt(imgX.^2 + imgY.^2);
%imgEdge = im2bw(imgEdge,0.3);

% print images
figure;
subplot(2,2,1) , imshow(b);
title('Original');
subplot(2,2,2) , imshow(imgX/255);
title('Along X');
subplot(2,2,3) , imshow(imgY/255);
title('Along Y');
subplot(2,2,4) , imshow(imgEdge/255);
title('Sum');
```

```
%% read image
img = imread('im4.jpeg');
b = rgb2gray(img);
imshow(b);
```

```
%% mask 2
f1 = [-1 -2 -1;0 0 0;1 2 1];
imgX = filter2(f1,b); % filter2 is a 2D digital filter

f2 = f1';
imgY = filter2(f2,b);

imgEdge = sqrt(imgX.^2 + imgY.^2);
%imgEdge = im2bw(imgEdge,0.3);

% print images
figure;
subplot(2,2,1) , imshow(b);
title('Original');
subplot(2,2,2) , imshow(imgX/255);
title('Along X');
subplot(2,2,3) , imshow(imgY/255);
title('Along Y');
subplot(2,2,4) , imshow(imgEdge/255);
title('Sum');
```

Matching Feature

Using Correlation :

- **Apply correlation on two segmented image**

If output is high => similar images

- **But alignment is a problem**

Alignment can be done by going through the diagonal and stopping at the first different pixel and then create a rectangle enclosing palm of known size

Shortcomings

- There must be no noise in background

- Sufficient contrast between palm and background

THANK YOU