

## Exercise 1

### 1.1 Copy here the two fingerprint images provided as examples example1\_1 and example1\_2

We are provided with the following to examples of fingerprints.



(a) Example1\_1



(b) Example1\_2

Figure 1: Examples of fingerprints.

As we can see, the pictures of the fingerprints have relatively high quality and its macro-singularities can be observed at first sight.

### 1.2 How many macro-singularities do you observe in each fingerprint?

As we already know, macro-singularities can be loops, deltas or whorls.

In the previous figure we can observe a single loop in each of the fingerprints. No deltas or whorls have been found.

### 1.3 Mark the macro-singularities in the images (deltas and loops).

We mark the previously mentioned loops using a red circle in Figure 2.



(a) Example1\_1 with a marked loop.



(b) Example1\_2 with a marked loop.

Figure 2: Marked loops.

## Exercise 2

### 2.1 Execute the provided code for Fingerprint Enhancement and paste the resulting image here

We execute the Matlab script `main.m` and stop it when the enhanced images are printed. The obtained enhanced fingerprints are shown in Figure 3.

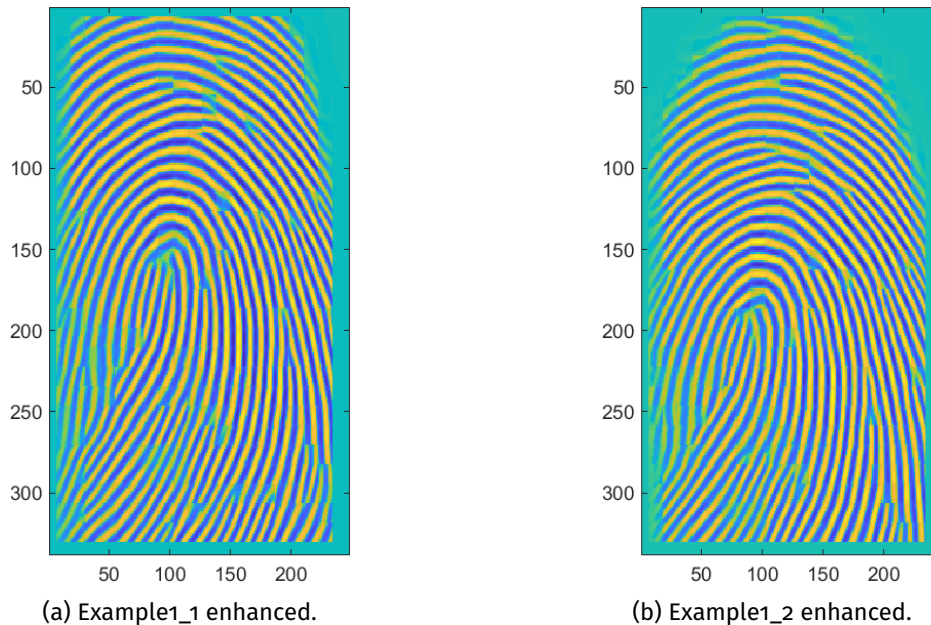


Figure 3: Enhanced fingerprint examples.

### 2.2 What differences do you observe with respect to the original fingerprints?

The aim of the enhancement techniques is to improve the fingerprint image quality in order to make the feature extraction task easier. Typically, we want to complete the ridge lines, deal with cuts or bruises on the finger or obtain a better separation between parallel ridges.

As we can see, the fingerprints in Figure 1 had incomplete ridge lines at the bottom part of the image which have been completed in the enhancement in Figure 3.

Also, the image has been *sharpened*, which means that now the contrast between the ridge lines and the *white spaces* has been increased, making ridge lines easier to identify.

## Exercise 3

### 3.1 Execute now the code for Quality Maps, and paste the resulting quality maps

If we keep executing the `main.m` script we obtain the following Quality Maps:

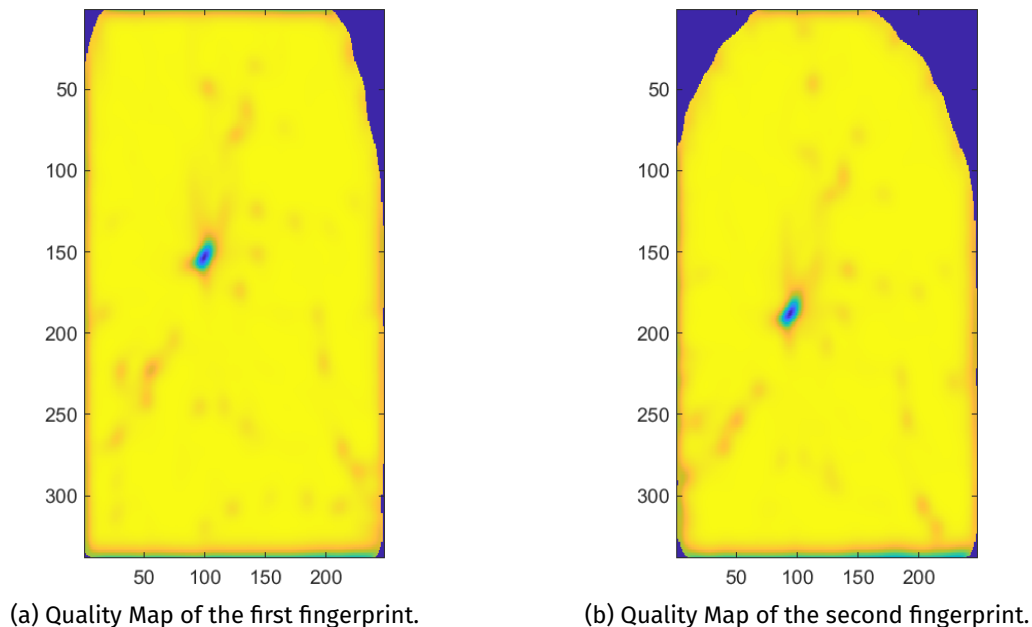


Figure 4: Quality maps.

### 3.2 What is the range of values for these quality maps?

To obtain this quantities, we have added the following code to the main script:

```
min(reliI1,[],'all')
max(reliI1,[],'all')

min(reliI2,[],'all')
max(reliI2,[],'all')
```

This code shows that:

- The minimum value for both Quality Maps is 0.0
- The maximum value for the first fingerprint Quality Map is 0.9991
- The maximum value for the second fingerprint Quality Map is 0.9988

Thus, we can say that both Quality Maps have all its values in the range  $[0, 1]$ .

### 3.3 What kind information (apart from the quality) can be inferred from such code?

We examine the file `testfin.m` to understand the result that we are obtaining.

If we read the first commented lines, we find that the function we applied returns

```
% Returns:    newim - Ridge enhanced image.
%             binim - Binary version of enhanced image.
%             mask - Ridge-like regions of the image
%             reliability - 'Reliability' of orientation data
```

Considering this and the code of the `main.m` file, what we just showed in Figure 4 is the **reliability** of the orientation data, that is, how secure we are of the orientation of the ridge-lines in each point of the image. As we can see in the figure, we can say that we can not rely on the results outside the fingerprint, and that the algorithm has some problems in the zone where the loop is detected. This makes sense to us, since in the top of the loop the orientation of the ridge-line changes completely. We also find that the process followed to obtain the quality maps is:

1. Identify the ridge-like regions and normalise the image. We can see that different blocksize and thresholds have been used but finally using `blocksize = 16` and `threshold = 0.1` are the values used.
2. Determine the orientations of the found ridge-like regions.

3. Determine ridge frequency values across the image. Then, the median frequency is multiplied in the final mask that will be applied to the image (a comment says that the median frequency performed better than the actual frequency.)
4. Lastly, the filter is applied to enhance the patterns

## Exercise 4

### 4.1 Execute the code in order to show the Binarized Fingerprint and the Segmented Fingerprint. Apply different values of quality threshold (0.1, 0.3, 0.6, 0.9) and paste here the resulting images

Let us first examine the code to see what we are doing in this section.

```
%Binarized and Segmented Fingerprints
threshold=0.1; %quality threshold
figure;
subplot(121)
imagesc(binI1+mask1+(relI1>threshold))
binI1(relI1<threshold)=0;
inv_binI1 = (binI1 == 0);
```

We can see that we establish a threshold (which we are told to modify) and then plots an image that is an addition of the images:

- Enhanced binarized image
- The mask that contains the ridge-like regions of the image (returned by testfin)
- The reliability of the orientation, selecting the pixels where the reliability is above the selected threshold

The result of the code is shown in Figures 5 and 6.

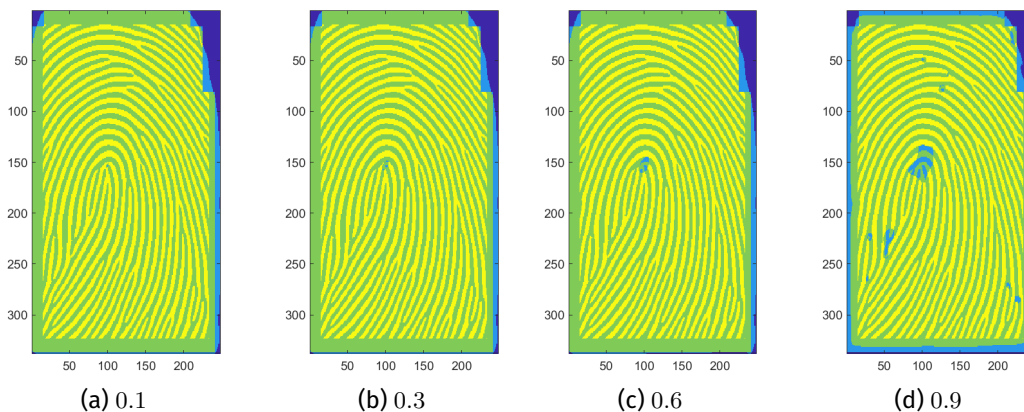


Figure 5: Different thresholds for the first fingerprint example.

We can observe in both cases that, as the threshold is augmented, the reliability around the loops decreases and the yellow mask disappears in both cases. This phenomenon also happens in other points that, until this point of the lab, we have not detected as important points.

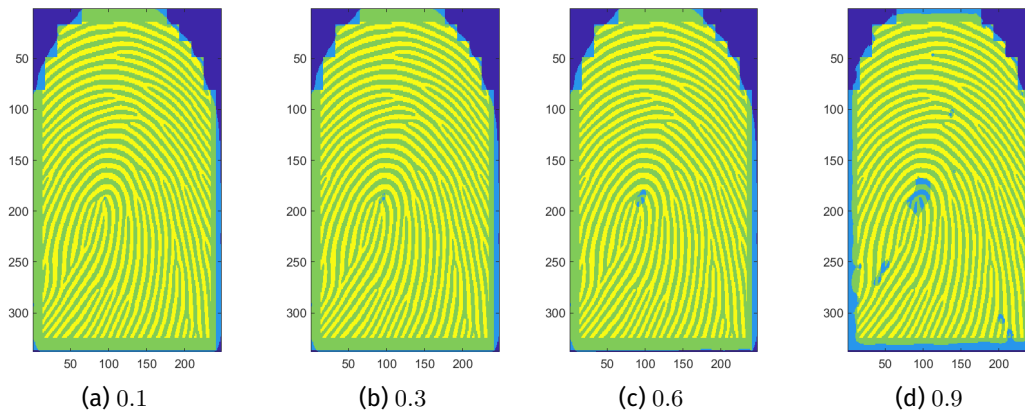


Figure 6: Different thresholds for the second fingerprint example.

## Exercise 5

- 5.1 Execute the code for generating the Fingerprint Skeleton and the Minutiae Extractor. Paste the resulting images for the original values  $window = 5$  and  $margin = 5$ .
- 5.2 Search heuristically by looking at the images for the optimal values of parameters window and margin. Paste the resulting images with your optimal parameters and justify your decision.

## Exercise 6

- 6.1 Execute the code corresponding to the Minutiae Validation for  $window=5$  and  $margin=5$ . Paste the resulting image including the minutiae extracted (red crosses) and validated (blue circles) of both fingerprints.
- 6.2 Execute the same code but with the optimal values of parameters window and margin. Paste the resulting image below.
- 6.3 Do you think it is a good idea to include the Minutiae Validation module? Justify your opinion.

## Extra Exercise

With all the previous exercises done correctly you can obtain a mark up to 7 points out of 10. Extra work: If you want to obtain a mark up to 10 points out of 10 you should complete the following: In folder “/ddbb” you have 20 fingerprint images. 19 of them are labeled with the subject identity (e.g., H0001), and 1 is Unknown. Search for the identity of the Unknown fingerprint in the set of 19 labelled reference fingerprints. You can use the provided code “identification\_1\_19.m” as basis. Paste here the resulting ranked list of scores of the Unknown fingerprint with respect each one of the 19 reference fingerprints.