A Filterbank-based Representation for Classification and Matching of Fingerprints

http://matlab-recognition-code.com

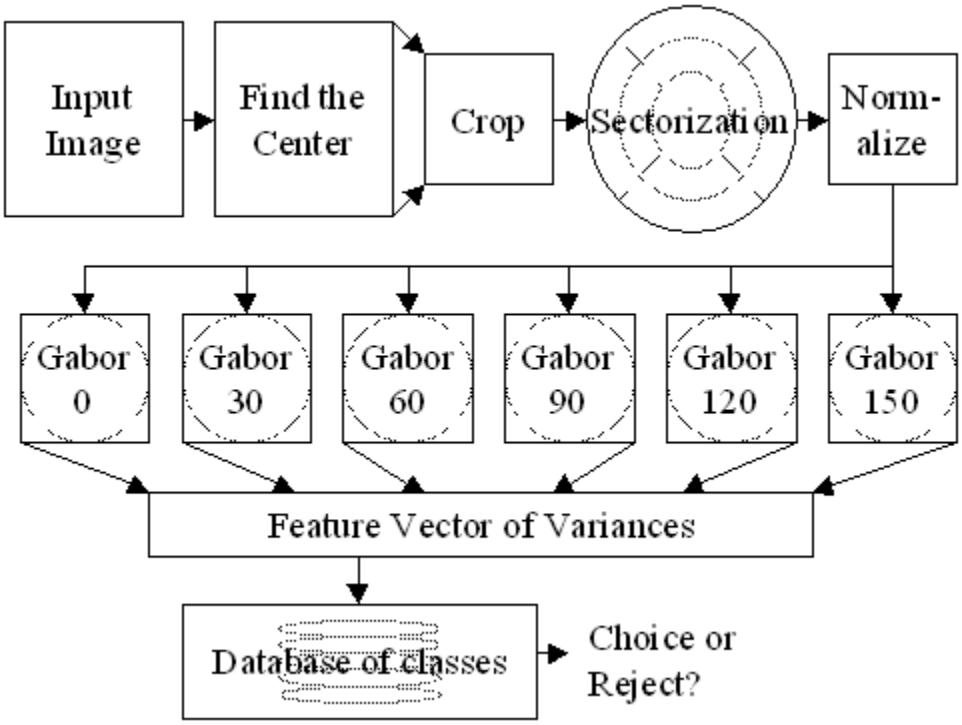

Figure 1 Description of the algorithm

Step	**Corresponding M-function**
Find the center (core point localization)	Supercore.m
Crop	Cropping.m
Sectorization	Whichsector.m
Normalize	Sector_norm.m
Filter-bank (8 filters)	Gabor2d_sub.m
Feature Vector of Variances	Sector_norm.m
Choice/Reject or add to DataBase	Fprec.m (main file)

Up to now, when a new fingerprint image is added to database, the FingerCode was calculated two times: one time for input image and a second time for the image rotated of a proper angle (22.5/2 degrees) in order to make the process rotation-invariant (see the cited reference for more details). The image was rotated using the Matlab function imrotate. This procedure can introduce noise. To avoid this behavior we calculated the FingerCode associated to the rotated image in this way: we rotate sectorization and the orientation of Gabor filters of filter-bank of the same angle (22.5/2 degree). This is equivalent to consider as filter-bank input a rotated image.

When a new fingerprint image is added to database, only one core point is found. On the other side, when an input image is selected for fingerprint matching, a list of candidates for core point is found and the matching is performed for each of them. At last only the candidate with the smallest distance is considered. For example , in database I have 3 images Img1, Img2 and Img3. Each of them is characterized only by one core point, so I will have 3 core points, each of them associated to an image present in database. If I select an image for fingerprint matching (let it be ImgNew) I found for it a certain number of core point (let it N). For each of these N core points (candidates) I will find the nearest fingerprint image present in database. At last I will obtain N distances (as the number of core points candidates): I say that the recognized image is the image with the nearest distance I have obtained (this distance is associated to one of the initial N core points candidates of ImgNew). This approach is very similar to the algorithm discussed in

Erian Bezhani, Dequn Sun, Jean-Luc Nagel, and Sergio Carrato, "Optimized filterbank fingerprint recognition", Proc. SPIE Intern. Symp. Electronic Imaging 2003, 20-24 Jan. 2003, Santa Clara, California.

The pixel-wise orientation field estimation (the M-function is **orientation_image_luigiopt.m**) is greatly accelerated reusing previous sum computations. The sum of elements of a block centered at pixel (I,J) can be used for the computation of the sum of block elements centered at pixel (I,J+1). This can be performed in the following way:

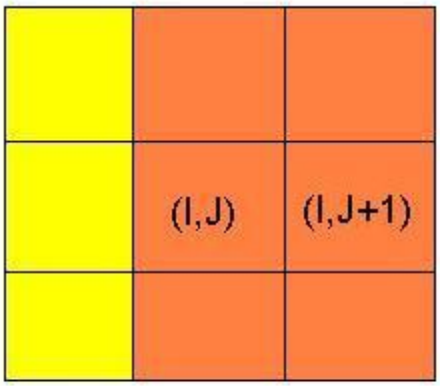

Figure 1

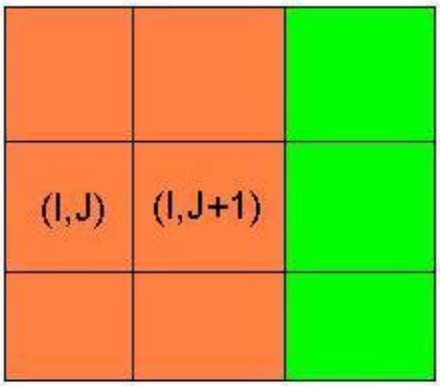

Figure 2

Once the sum of values centered at pixel (I, J) has been calculated (sum of yellow pixels and orange pixels of Figure 1), in order to calculate the sum centered at pixel (I, J+1) we simply subtract from the previous sum the yellow area and add the green area (see Figure 2): in this way it is possible to save a lot of computation. In other words
SUM (I, J) = yellow + orange
SUM (I, J+1) = SUM (I, J) – yellow + green.

For more details concerning the implementation of this algorithm please visit
http://www.ece.cmu.edu/~ee551/Old_projects/projects/s99_19/finalreport.html

The function Supercore.m is discussed in detail in the document corepoint.doc

References

A. K. Jain, S. Prabhakar, and S. Pankanti, "A Filterbank-based Representation for Classification and Matching of Fingerprints", International Joint Conference on Neural Networks (IJCNN), pp. 3284-3285, Washington DC, July 10-16, 1999.
http://www.cse.msu.edu/~prabhaka/publications.html

An improved scheme for core point positioning

Step 1

Given an input image, I enhance [1] the fingerprint in order to obtain a better image quality.

Figure 1. Input fingerprint image

Figure 2. Enhanced fingerprint image

Step 2

After this operation, the image is segmented and background is separated from fingerprint image. This can be performed using a simple block-wise variance approach, since background is usually characterized by a small variance. Image is first binary closed (Matlab command imclose), then eroded (Matlab command imerode), in order to avoid holes in fingerprint image and also undesired effect at the boundary (between fingerprint and background). The image segmentation is repeated several times, up to a desired condition is satisfied. This is done in order to avoid undesired boundary effects between fingerprint and background. The condition which has to be satisfied is chosen in the following way: the enhanced image is divided into non-overlapping blocks of given sizes (usually 32 x 32 or 64 x 64). The whole enhanced image is filtered with a complex filter. Let Cf_max be the maximum value of the filtered image **in the current region of interest** (previously calculated according to some initial parameters). For each non-overlapping block we calculate the relative maximum Cf_rel. We finally consider a logical matrix F whose element (I,J) is equal to 1 if (I,J) is a block relative maximum and this value is equal or greater than a threshold value (usually 0.65*Cf_max in our simulations); F(I,J) is equal to 0 in all the other cases (i.e. if F(I,J) is not a block relative maximum or a block relative maximum smaller than the threshold value). If the number of non-zero elements of the logical

matrix F is above a threshold value, the parameters for image segmentation are re-calculated and the whole process is repeated once again (complex filtering output does not vary, only the region of interest has been changed).

Note: these relative maxima larger than the threshold value are not considered core points but they will be considered candidates for it when a new image is selected for fingerprint matching.

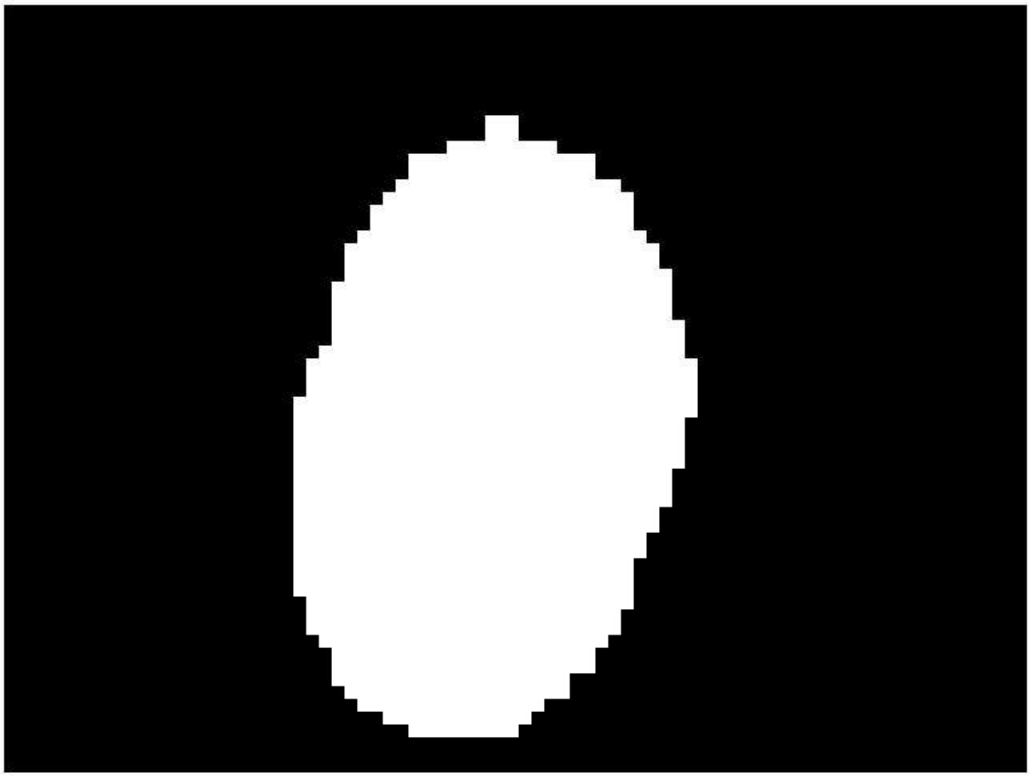

Figure 3. Image is segmented, closed and eroded (binary erosion and closing).

Step 3

Then we perform a fast pixel-wise orientation field computation [2].

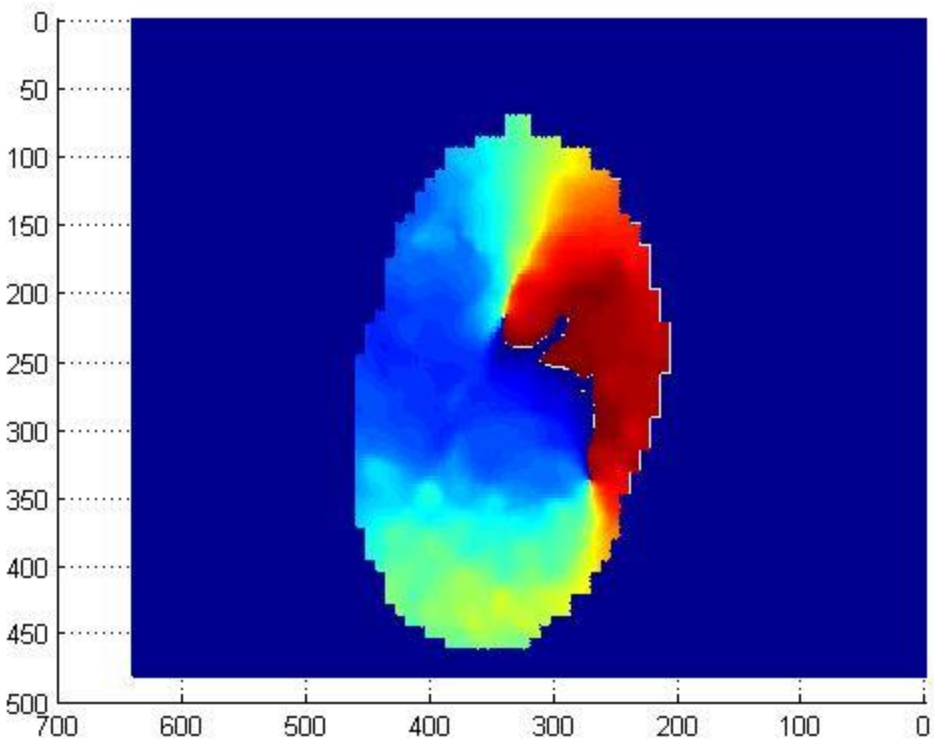

Figure 4. Pixel-wise orientation field estimation in the region of interest.

Step 4

The orientation field is used to obtain a logical matrix where pixel (I,J) is set to 1 if the angle of the orientation is <= PI/2 (PI 3.1415926535897...).

Figure 5. Logical matrix with local orientation field <= PI/2

Step 5

After this computation I find the border of this logical matrix, in the region of interest of fingerprint image.

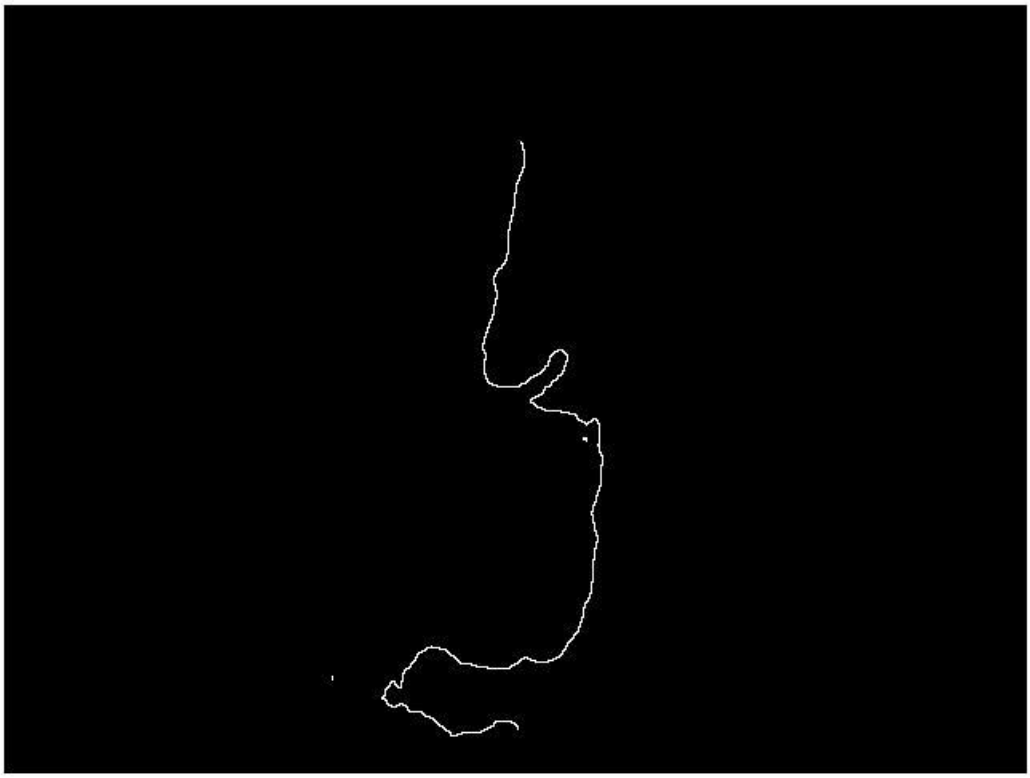

Figure 6. Border of the logical matrix in Fig.5 in the region of interest.

Step 6

After this computation I calculate the complex filtering output [3] of the enhanced fingerprint image. It is not necessary to re-calculate it, I use the complex filtered image calculated in step 2.

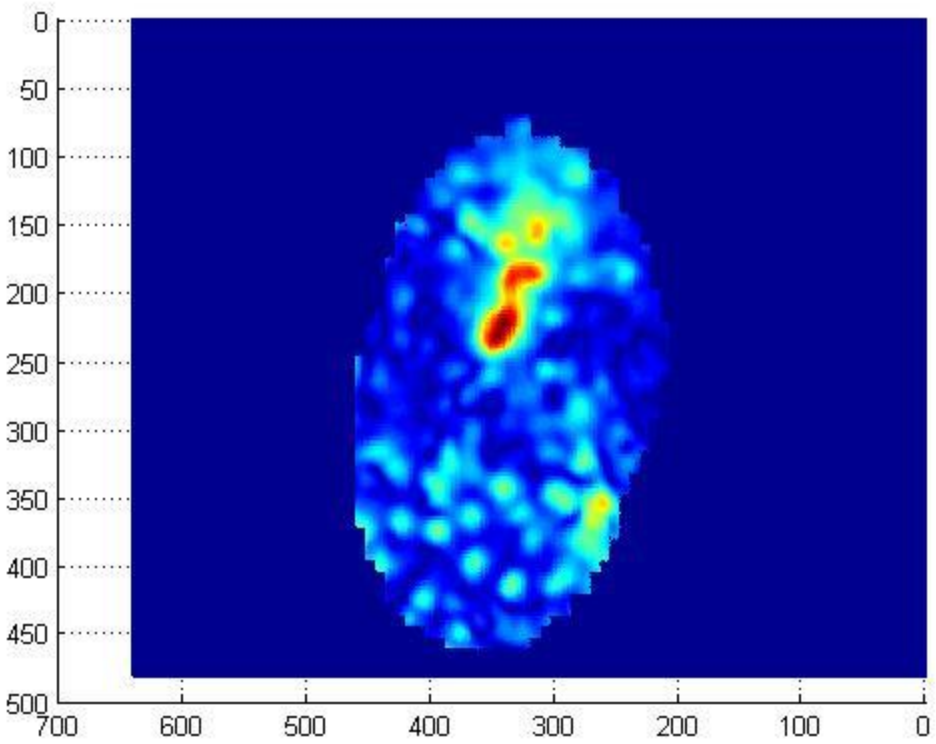

Figure 7. Complex filtering output of the enhanced fingerprint image.

Now I can find the maximum value of complex filtering output where the pixels of the logical image (shown in Fig. 6) are set to 1.

Step 7

I repeat steps 4-5-6 for a wide set of angles (…,PI/2-3*alfa, PI/2-2*alfa, PI/2-1*alfa, PI/2, PI/2+1*alfa, PI/2+2*alfa, PI/2 +3*alfa, …where alfa is an arbitrary angle). Each time I determine a point (**Note:** each of them will be a candidate for fingerprint matching).

Step 8

```
I subdivide all the points found in step 7 into subsets of
points which are quite near each other. I will have say N
subsets. For each subsets I will have a certain number of
candidates and I consider only subset with a number of
candidates >= 3. For each of this subset I consider the
subset with the greatest x-averaged coordinate. In this
subset I consider the core point the candidate with the
greatest x-coordinate. This is a good approximation in
standard fingerprint image.
```

See supercore7.m for more details concerning the algorithm used. The functions supercore7_list returns as candidates for core point:

- The points found at the end of step 8
- The point found as relative maxima of non-overlapping block at the end of step 2

Figure 8. Input fingerprint image and corresponding core point.

References

[1] S. Chikkerur,C. Wu and Govindaraju, "A Systematic approach for feature extraction in fingerprint images", ICBA 2004
[2] Ravishankar Rao, "A taxonomy of texture description", Springer Verlag
[3] Kenneth Nilsson and Josef Bigun, “Localization of corresponding points in fingerprints by complex filtering”, Pattern Recognition Letters 24 (2003), 2135-2144