CS663 Course Project: A Fingerprint Recognition Algorithm Using Phase-Based Image Matching for Low-Quality Fingerprints

Raavi Gupta, Anoushka Dey, Kartik Singhal
200070064, 210010010, 21D070035

Introduction

In this project we aim to recognize fingerprints using a phase-based method as proposed in [1]. The algorithm contains four major steps which we explain in detail and the corresponding resutls that we obtain. Finally, we test our implementation on the FVC2000 database.

Database

The database consists of 4 parts in which each registers the fingerprints of ten subjects acquired using different sensors.

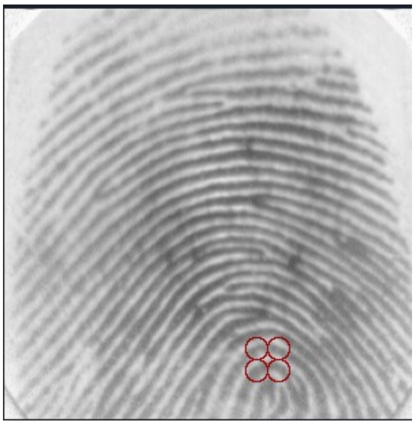
1. **DB1**: low-cost optical sensor ”Secure Desktop Scanner” by KeyTronic
2. **DB2**: low-cost capacitive sensor ”TouchChip” by ST Microelectronics
3. **DB3**: optical sensor ”DF-90” by Identicator Technology
4. **DB4**: synthetic fingerprint generation.

Algorithm

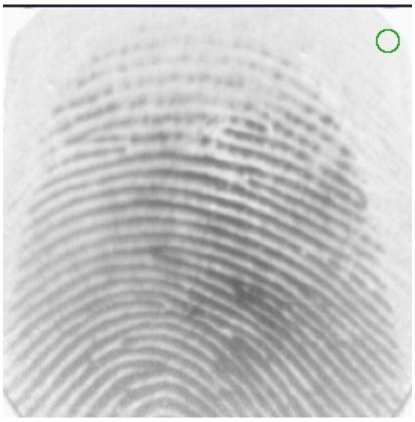
The algorithm comprises of four crucial steps which have been detailed below.

Core Detection

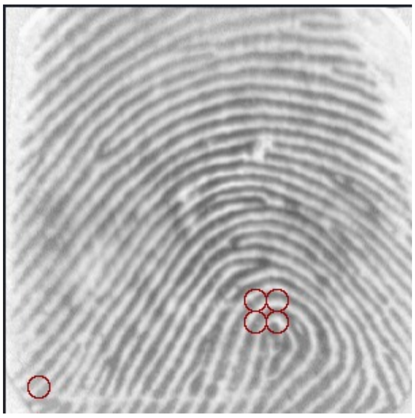
Poincaré index: Let G be a vector field and C be a curve immersed in G; then the Poincaré index $P_{G,C}$ is defined as the total rotation of the vectors of G along C. The path defining C is the ordered sequence of 8 elements d_k (k=0...7) surrounding the pixel in consideration at [i,j]. The direction of the elements is chosen such that the angle between d_i and d_{i+1} is less than ninety degrees. The Poincaré index is then just the sum of these differences.
For the case of core detection, the Poincare index must be equal to 180 degrees. Cores are represented by 4 red circles in the images below.



(a) Accurate Core detection



(b) Core absent in the image



(c) Extra singularity detected

Displacement and Rotation Alignment

The registered and input images need to be aligned in order to perform a high accuracy finger print matching. When both the images have their cores, we have first aligned their translational displacements. We then proceeded to normalize their rotation by first generating a set of rotated images of the registered fingerprint. We then compared each of these rotated images with the input fingerprint using the Band Limited Phase Only Correlation (BLPOC) function. When either the input fingerprint or the registered fingerprint did not have a core, we first normalized the rotation using the above mentioned technique and then aligned them using their translational displacement, which was obtained as the peak location of the Phase Only Correlation (POC) function between the two fingerprints. We thus obtained the normalized versions of the registered and the input image.

Common Region Extraction

The next step involved extracting the common region of the two images. The non-overlapping areas pertain to the uncorrelated noise components of the BLPOC function and hence ignoring them would increase the accuracy of the fingerprint matching technique. We examined the n1 and n2 axis projection of the pixel values and extracted only the common effective areas pertaining to the two images.

Fingerprint Matching

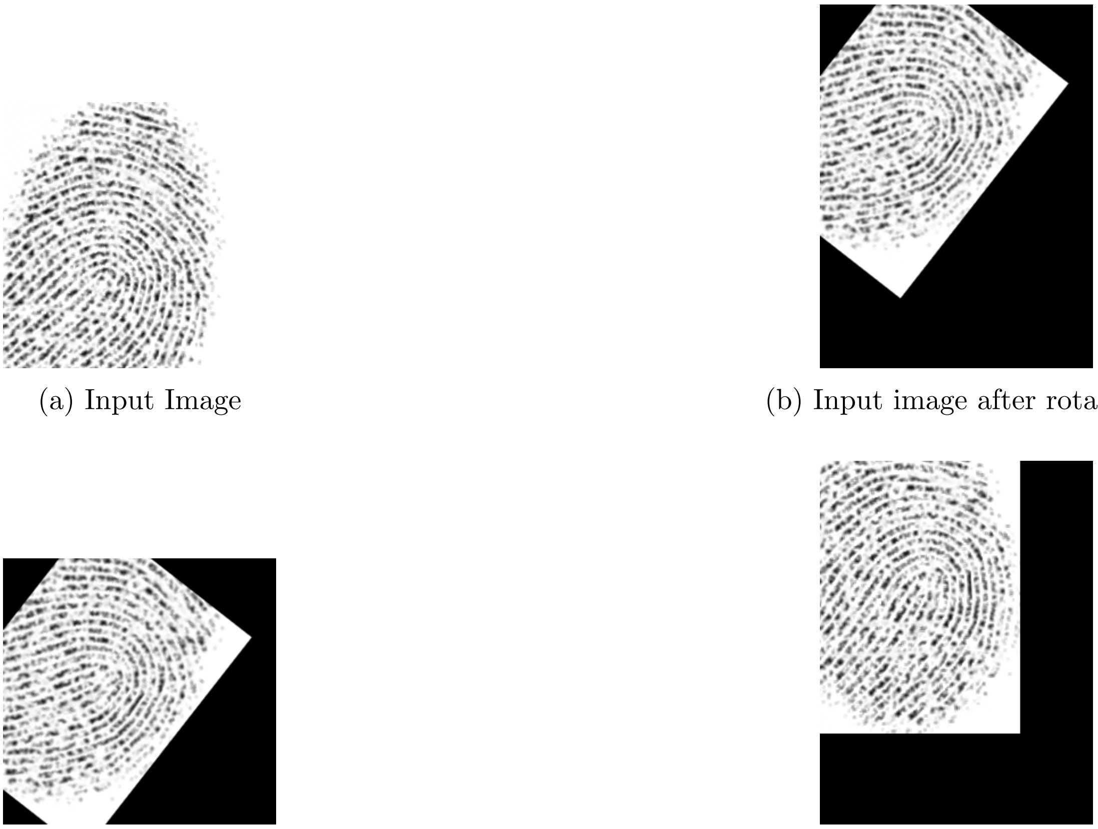
Following the above steps, we proceeded to calculate the value of the BLPOC between the two extracted images and evaluated the matching score. The matching score was defined as the sum of the highest two peaks of the BLPOC function in order to enhance the redundancy of the detection and make it slightly flexible to incorporate the effects of elastic fingerprint deformation.

Results

[1]

Database	Accuracy
DB1	1.0
DB2	0.95
DB3	0.93
DB4	0.9

[1] Please run the program using Python 2.7



(a) Input Image

(b) Input image after rotation

(c) Registered Image

(d) Displaced registered image

Figure 2: Example of fingerprint processing

Future work

We hope to extend the algorithm on colored images as well.

References

[1] K. Ito et al. “A fingerprint recognition algorithm using phase-based image matching for low-quality fingerprints”. In: *IEEE International Conference on Image Processing 2005*. Vol. 2. 2005, pp. II–33. DOI: 10.1109/ICIP.2005.1529984.