CS 663 Course Project

Rishabh Dahale, 17d070008
Shaurya Chopra, 170100108
Shyam Thombre, 170010023
Nitish Tongia, 170010042

Fingerprint Recognition using Phase-Based Image Matching

Autumn, 2020

Abstract

A major approach for fingerprint recognition is to extract minute details from high-quality fingerprint images using algorithms like Harris Corner Detection, and then matching these stored minute details to those of a probe image. However, such an approach depends heavily on fingerprint surface condition (for instance, dirt on the fingerprint sensor glass, rough fingertips and so on). This makes realistic recognition systems unreliable.

Using phase components of 2D discrete Fourier Transforms[1] makes it possible to achieve robust recognition with good accuracy even for low-quality fingerprint images, captured in unfavourable conditions.

Specifications

Dataset and Programming Language

We use the dataset provided by FVC2004 [2]. FVC2004 is the third international Fingerprint Verification Competition, a technology evaluation of fingerprint recognition algorithms open to companies, academic research groups and independent developers. Four databases constitute the FVC2004 benchmark. Three different scanners and the SFinGE synthetic generator, were used to collect fingerprints.

	Technology	Image	Resolution
DB1	Optical Sensor (CrossMatch V300)	640×480	500 dpi
DB2	Optical Sensor (Digital Persona U.are.U 4000)	328×364	500 dpi
DB3	Thermal Sweeping Sensor (Atmel FingerChip)	300×480	512 dpi
DB4	Synthetic Generator (SFinGe v3.0)	288×384	About 500 dpi

The dataset consists of a total of 320 fingerprints with 80 fingerprints from each sensor. For each finger we have 8 sample images.

We use Python + OpenCV for the implementation.

Algorithm

The core idea behind the algorithm [1] is that the Phase Only Correlation (POC) of two images (which in turn is defined using the cross-phase spectrum) gives a distinct sharp peak when they are similar.

The POC is defined as the inverse Fourier Transform of the cross-phase spectrum of two images *f,g* where

$$R_{FG}(k_1, k_2) = \frac{F(k_1,k_2)\overline{G(k_1,k_2)}}{|F(k_1,k_2)\overline{G(k_1,k_2)}|}$$

$$= e^{j\theta(k_1,k_2)}$$

Where $\overline{G(k_1,k_2)}$ is the complex conjugate of $G(k_1, k_2)$ and $\theta(k_1, k_2)$ denotes the phase difference $\theta_F(k_1, k_2) - \theta_G(k_1, k_2)$

We modify this and use the Band-Limited Phase Only Correlation (BLPOC), which eliminates the high frequency components in the calculation of cross-phase spectrum.

$$r_{fg}^{K_1K_2}(n_1, n_2) = \frac{1}{L_1L_2} \sum_{k_1,k_2}' R_{FG}(k_1, k_2) \times W_{L_1}^{-k_1n_1} W_{L_2}^{-k_2n_2}$$

Where – $K_1 : K_1, K_2 : K_2$ is the bandwidth of interest, and $W_{N_1} = e^{-j\frac{2\pi}{N_1}}$ and $W_{N_2} = e^{-j\frac{2\pi}{N_2}}$

The algorithm is divided into the following parts:

Core Detection

Detect the core of the registered fingerprint image and input fingerprint image in order to align the displacement between them. “Core” is defined as a singular point in a fingerprint image that exhibits the maximum ridge line curvature. The Poincare index method is used to detect the core [3]. In this method, we compute the direction pattern of the image, that is compute the probabilities of each direction of the fingerprint curve within a block (image divided into non-overlapping blocks). This is done by creating histograms of various possibilities of 2x2 slices, each of which corresponds to directions 0,45,90,135.

This direction pattern is also “relaxed” [3] in order to reduce noise.

Then, for a given block, given the direction pattern of its neighbors, assign a code according to the amount of curvature of pattern around a block, and use that to classify blocks as whorl, core, delta or normal.

Displacement Alignment

Normalize the displacement and rotation between the registered and input fingerprints in order to perform high accuracy matching. Bicubic interpolation is done for rotation.

Use core points to align two fingerprint images. If core points not available, rotate the probe image from -40 to 40 degrees and use the BPLOC function to check for maximum similarity.

Common Region Extraction

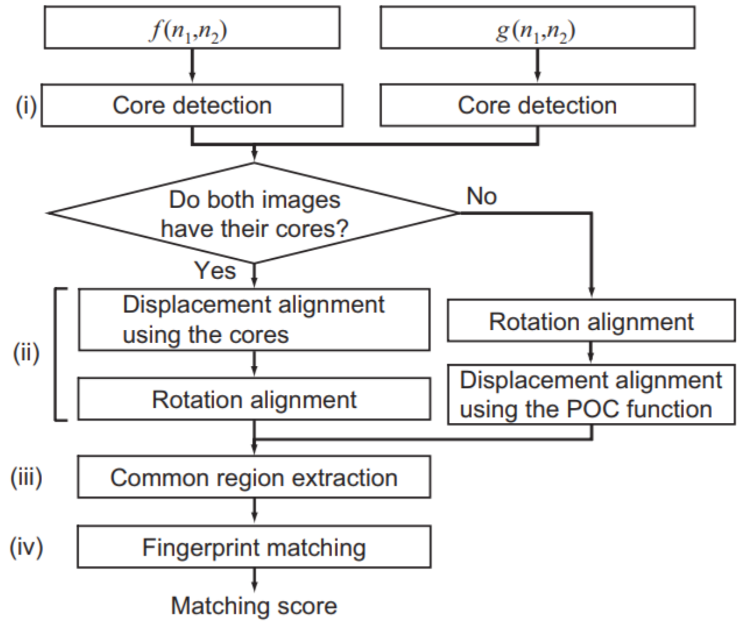
Intersection of two images is then extracted, since non-overlapping regions are uncorrelated and of no use.

Remove regions with no information (whitespace) by taking projections of image on both axes and removing points that correspond to whitespace only.

Fingerprint Matching

Finally, BLPOC is used to get a matching score. Since multiple correlation peaks are possible, the matching score is defined as the sum of the two highest peaks. Bandwidths are tuned to a point after which an increase does not give a significant improvement in the matching score.

A flow diagram of the algorithm is as follows:



Evaluation and Validation

We use the evaluation metrics that are used in the paper for validating the performance of our model. We first evaluate the False Non-Matching Rate (FNMR) for all possible combinations of genuine attempts. Next, we evaluate False Matching Rate (FMR) for impostor attempts, where we select a single image for each fingerprint and make all the possible combinations of impostor attempts. Based on the above definitions, following two metrics were used for evaluation:

1. ZeroFMR - This is defined as the lowest FNMR where FMR=0%.
2. Equal Error Rate (EER) - The EER is defined as the error rate where the FNMR and the FMR are equal.

Results

References

[1] Ito, Kiyoshi & Morita, A. & Aoki, Takiko & Higuchi, Takuma & Nakajima, H. & Kobayashi, K.. (2005). A fingerprint recognition algorithm using phase-based image matching for low-quality fingerprints. Proceedings - International Conference on Image Processing, ICIP. 2. II - 33. 10.1109/ICIP.2005.1529984.

[2] FVC2004 web site: http://bias.csr.unibo.it/fvc2004

[3] M. Kawagoe and A. Tojo, “Fingerprint pattern classification,” Pattern Recognition, vol. 17, no. 3, pp. 295–303,1984