Synthetic Fingerprint-Database Generation

R. Cappelli, D. Maio and D. Maltoni
DEIS - CSITE CNR - University of Bologna
{rcappelli,dmaio,dmaltoni}@deis.unibo.it

Abstract

This work complements our previous efforts in generating realistic fingerprint images for test purposes. The main variability which characterizes the acquisition of a fingerprint through an on-line sensor is modeled and a sequence of steps is defined to derive a series of impressions from the same master-fingerprint. This allows large fingerprint databases to be randomly generated according to some given parameters. The experimental results validate our technique and prove that it can be very useful for performance evaluation, learning and testing in fingerprint-based systems.

1. Introduction

Fingerprint recognition is nowadays one of the most widely diffused biometric authentication techniques [1]. Great efforts are continuously spent in designing new algorithms both in academic and industrial environments, but not enough care is usually dedicated to performance evaluation and benchmarking. In fact, testing a fingerprint recognition algorithm requires a large database of samples (thousands or tens of thousands), due to the small errors which have to be estimated [1], and collecting large databases of fingerprint images is:

- expensive both in terms of money and time;
- boring for both the people involved and for the volunteers, which are usually submitted to several acquisition sessions at different dates;
- delicate due to the privacy legislation which protects such personal data.

FVC2000 [2] and FVC2002 [3] are examples of performance evaluation events, where real fingerprint databases have been collected and different algorithms have been tested with the aim to track the state of the art in fingerprint verification.

To overcome the problem of gathering large databases of fingerprint images for testing purposes, some years ago we started working on synthetic fingerprint-image generation. In [4] we presented an effective technique which allows a single impression of a finger to be randomly generated according to some input parameters. Section 2 briefly summarizes the above method. Generating random impressions of different fingers is not enough for testing a fingerprint recognition algorithm: in fact, while these can be used for measuring impostor errors or FAR (the probability of falsely matching two fingerprints), in order to measure genuine errors or FRR (the probability of false non-match), a set of instances have to be generated for each finger. This paper is aimed at describing a new original technique for generating multiple impressions of the same finger, starting from a master-fingerprint which is a sort of template randomly transformed according to some given generation rules.

This problem is quite difficult because, for a realistic generation, all the variability which in the practice characterizes the fingerprint acquisition process has to be captured. We focus on fingerprint acquisition with on-line sensors, since this kind of acquisition allows practical on-line recognition systems to be developed. On the other hand, with a few changes, the generation of impressions produced by the classic “ink-technique” is also possible.

Section 3 of this paper describes the basic steps involved in the generation of fingerprint impressions from the same master-fingerprint. In section 4, we provide some experimental results and significant validations of our generation technique. Finally, section 5 draws some conclusions and outlines the future work on this topic.

2. Generating master-fingerprints

As discussed in [4], generating a master-fingerprint involves the following steps (see fig. 1):

1) fingerprint shape generation;
2) directional map generation;
3) density map generation;
4) ridge pattern generation.

Step 1 defines the global shape of the fingerprint (fig. 1.1); step 2, starting from the positions of cores and deltas, exploits a mathematical ridge-flow model to generate a consistent directional map (fig. 1.2). Step 3 creates a density map on the basis of some heuristic criteria inferred by the visual inspection of several real fingerprints (fig. 1.3). In step 4, the ridge-line pattern and the minutiae are created through a space-variant linear filtering; the filters orientation and frequency are locally adjusted according to the directional map and the density map, respectively (fig. 1.4). The iterative application of the filters to an initially white image, enriched with few random seeds, determines the generation of the ridge-line pattern where minutiae appear at random positions.
3. Generating fingerprint impressions from a master-fingerprint

Several factors contribute to make substantially different the impressions of a given finger as captured by an on-line acquisition sensor:
- displacement in X and Y direction and rotation;
- different touching area;
- non-linear distortion produced by non-orthogonal pressure of the finger against the sensor;
- variations in the ridge-line thickness given by pressure intensity or by skin dampness (wet or dry);
- small cuts or abrasions on the fingertip;
- background noise and other random noise.

For each fingerprint impression to be generated, our technique sequentially performs the following steps, starting from the master-fingerprint:
1) variation of the ridge average-thickness;
2) distortion;
3) noising and rendering;
4) global translation/rotation.

3.1. Variation of the ridge average thickness

Skin dampness and finger pressure against the sensor platen have similar effects on the acquired images: when the skin is dry or the pressure is low, ridges appear thinner, whereas, when the skin is wet or the pressure is high, ridges appear thicker (see fig. 2.1).

![Fig. 1](image.png) Fig. 1. A graphical representation of the steps sequentially performed to generate a master-fingerprint. Steps 2, 3 and 4 are shown for two different master-fingerprints.

![Fig. 2](image.png) Fig. 2. Fig. 2.1 shows three fingerprints of the same real finger as captured when the finger is dry, normal and wet. In 2.2 the application of different levels of erosion/dilation to the same master-fingerprint is illustrated.

Morphological operators [5] are here applied to the master-fingerprint, to simulate different degrees of dampness/pressure. In particular, the erosion operator is applied to simulate low pressure or dry skin, while the dilation operator is adopted to simulate high pressure or wet skin (fig. 2.2). The structuring element used is a square box; its size varies from 2×2 to 4×4 to modulate the magnitude of the ridge thickness variation.

3.2. Distortion

In [6] we introduced a skin-distortion model to cope with the non-linear distortion caused by non-orthogonal
pressure of the finger against the sensor surface (see fig. 3.1).

Fig. 3. Fig. 3.1 shows two impressions of the same real finger where corresponding points are marked to highlight distortion; in fig. 3.2 the effect of traction and torsion are illustrated, referring to the three regions defined by our model; fig. 3.3 shows the application of our distortion model (for different parameters) to a square-meshed grid.

By noting that the finger pressure against the sensor is not uniform but decreases moving from the centre toward the borders, our distortion model defines three distinct regions (see fig 3.2):

a) a close-contact region where the high pressure and the surface friction do not allow any skin slippage;

b) an external region, whose boundary delimits the fingerprint visible area, where the light pressure allows the finger skin to be dragged by the finger movement;

c) a transitional region where an elastic distortion is produced to smoothly combine regions a and c. The skin compression and stretching is restricted to region b, since points in a remain almost fixed and points in c rigidly move together with the rest of the finger.

The distortion model is defined by a mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ which can be viewed as an affine transformation (with no scale change) which is progressively “braked” as it moves from c towards a. Each point $v$ is mapped into $\text{distortion}(v)$ such that:

$$\text{distortion}(v) = v + \Delta(v) \cdot \text{brake}(\text{shapedist}(v), k)$$

where $\Delta$ specifies the affine transformation of the external region c; $\text{shapedist}(.)$ is a shape function describing the boundary of region a; $\text{brake}(.)$ is a monotonically increasing function that rules the gradual transition from region a toward region c. Fig. 3.3 shows some examples of distortion varying the parameters.

Unlike in [6], where the distortion model was applied to re-map minutiae points in order to improve fingerprint matching, in this work the mapping has to be applied to the whole master-fingerprint image and therefore Lagrangian interpolation is employed to obtain a smooth gray-scale target image. Performing Lagrangian interpolation requires the inverse mapping function $\text{distortion}^{-1}(.)$ to be computed, but unfortunately this function cannot be analytically derived. Therefore, for each pixel involved in the mapping, the Newton-Raphson method [7] is used for numerically calculating the inverse.

3.3. Final noising and rendering

In the practice, several factors contribute to deteriorate the quality of an acquired fingerprint, thus producing a gray-scale noisy image: irregularity of the ridges and their different contact with the sensor surface, presence of small pores within the ridges, presence of very-small-prominence ridges, gaps and cluttering noise due to non-uniform pressure of the finger against the sensor. Our noising and rendering stage sequentially performs the following sub-steps:

1) isolate the valley white pixels into a separate layer;
2) add noise in the form of small white blobs of variable size and shape;
3) smooth the image over a $3 \times 3$ window;
4) superimpose the valley layer to the image obtained.

Steps 1 and 4 are necessary to avoid an excessive overall image smoothing. Fig. 4 shows the intermediate images produced after sub-steps 2 and 4, respectively.

Fig. 4. The fingerprint image after sub-step 2 (left) and sub-step 4 (right).
4. Experimental results and validation

We developed an automated tool for generating fingerprint images according to the method described in this paper. A demo version of this tool, called SFinGe, can be downloaded from http://bias.csr.unibo.it/research/biolab/sfinge.html. SFinGe allows a database to be batch-generated given a relatively small set of input parameters (number of fingers, impressions per finger, image size, seed for the random numbers generator, maximum amount of translation/rotation, maximum amount of noise, maximum amount of deformation, global database difficulty).

In our experimentation, on a Pentium IV PC, a fingerprint database of 10,000 images (240×336 pixels) was generated in 11 hours. Fig. 5 shows some samples.

![Fig. 5. Two sets of fingerprint impressions (one for each row) derived from two master-fingerprints (not shown).](image)

Several experimentations have been carried out to validate our synthetic generator. In particular:

- about 90 people (most of them having a discrete or good background in fingerprint analysis) have been asked to find a synthetic fingerprint image among four images (3 of which were real fingerprints). Just 23% of them identified the artificial image;

- an important test has been performed in conjunction with the First International Fingerprint Verification Competition (FVC2000) [2], where one of the four databases used (DB4) was synthetically generated by SFinGe. Not only the participant algorithms performed on DB4 similarly to the other DBs, but the genuine/imposter distributions and the ROC curves are also surprisingly close. This clearly proves that the main inter-class and intra-class variation of fingerprints in nature are very well captured by SFinGe.

5. Conclusions

Synthetic fingerprint generation is an effective technique to overcome the problem of collecting large fingerprint databases for test purposes. Obviously real-fingerprint databases cannot be completely substituted, especially when performance has to be measured referring to a given real environment/application. The use of synthetic fingerprints is not only limited to the problem of performance evaluation:

- many classifiers and pattern recognition techniques (i.e. neural network, PCA, SVM, ...) require a large training-set for an accurate learning stage. Synthetic fingerprint images are very well suited to this purpose: in fact the generator input parameters allow to explicitly control the type and features of the generated datasets (e.g. class, type of noise, distortion, ...) and this can be exploited in conjunction with boosting techniques to drive the learning process. In [8] we successfully used a large synthetic training set to derive optimal MKL subspaces for fingerprint indexing.

- we are currently investigating the use of synthetic fingerprint images to study the robustness of matching algorithms with respect to fake fingerprints. In this case, SFinGe allows to generate large sets of fingerprints whose features (e.g. minutiae distribution) can be varied independently of other fingerprint characteristics (e.g. directional image).

As to future improvement, the main aspect we will consider is an ad-hoc stage aimed at creating realistic (sensor-dependent) backgrounds.

10. References