Demonstrator of a Fingerprint Recognition Algorithm into a Low-Power Microcontroller

Javier Arcenegui, Rosario Arjona, and Iluminada Baturone
Instituto de Microelectrónica de Sevilla (IMSE-CNM)
Universidad de Sevilla, Consejo Superior de Investigaciones Científicas (CSIC)
Seville, Spain
{arcenegui, arjona, lumi}@imse-cnm.csic.es

***Abstract*—A demonstrator has been developed to illustrate the performance of a lightweight fingerprint recognition algorithm based on the fingerprint feature QFingerMap16, which is extracted from a window of the directional image (containing 16 direction values) centered at the convex core point of the fingerprint. The algorithm has been implemented into a low-power ARM Cortex-M3 microcontroller included in a Texas Instruments LaunchPad CC2650 evaluation kit. It has been also implemented in a Raspberry Pi 2 so as to show the results obtained at the successive steps of the recognition process with the aid of a Graphical User Interface (GUI).**

Keywords—Biometrics; Fingerprint recognition; Lightweight algorithms; Microcontrollers

I. INTRODUCTION

Among biometric traits, fingerprints are the most accepted by the end users because they are non-invasive and easy to use. Fingerprint recognition has been employed traditionally in forensic applications with large-scale databases that require complex algorithms, high computational resources, and high power consumption. More recently, fingerprint recognition is widely used in smart phones, which include microprocessors at moderate frequencies and, hence, moderate power consumption. The context of constrained hardware devices, such as smart cards, key fobs, smartbands or Internet-of-Thing devices, work at low frequency with low power consumption, and, hence, they should implement lightweight algorithms.

Fingerprint features can be classified into three levels [1]. Level-1 features are related to global information of fingers, like textures (frequencies or orientations of the ridges) and geometric information (singular points such as convex and concave cores and deltas). Level-2 features contain local information, like minutiae (bifurcations and endings of the ridges). Finally, level-3 features are related to very fine details like scars or pores. Higher distinctiveness is obtained with algorithms that process features of higher levels, but they are more complex. The algorithm implemented in this demonstrator is based on level-1 features because the hardware considered as target is a low-power microcontroller with constrained resources. Higher distinctiveness is achieved with the fusion of several samples from several fingers (at enrollment and matching stages).

The paper is structured as follows. The recognition algorithm is summarized in Section II. Section III presents the main features of the demonstrator. Finally, Section IV shows the conclusions.

II. FINGERPRINT RECOGNITION ALGORITHM

The algorithm for the extraction of the feature named QFingerMap16 (proposed in [2]) has the following steps:

1. For each pixel, the dominant direction value of the ridges (out of 16 possible dominant directions) is selected by using a 9x9 window. The standard deviations of the luminance of the pixels associated to the 16 possible directions are evaluated. The dominant direction is selected as the one with the maximum difference of the standard deviations between it and its orthogonal one.
2. In order to remove noise and isolated direction values, smoothing is applied to obtain a directional image with homogeneous direction regions. For each pixel, a 27x27 window is considered to select the direction value with the highest number of occurrences.
3. The computation of the directional image lets extract quality indexes that determine how the fingerprint image has been captured [3]: an index that evaluates the number of reliable pixels, and three indexes that estimate the reliable pixels associated to three groups of direction values (around 0º, 45º, and 135º). The pixels with directions around 90º are not considered since there are fingerprints that have few pixels with that direction.
4. Convex core location is performed, in a first step, by computing the Poincaré index in a 3x3 window. If the number of convex core candidates is higher than 5, another technique is applied to reduce the number of convex core candidates. It is based on finding 3x3 direction patterns (as proposed in [3]) within a second 27x27 smoothing of the directional image, which now clusters the 16 directional values into 4 values.

This work was supported by the TEC2014-57971-R project funded by the Ministerio de Economía y Competitividad of the Spanish Government (with support from the PO FEDER-FSE).

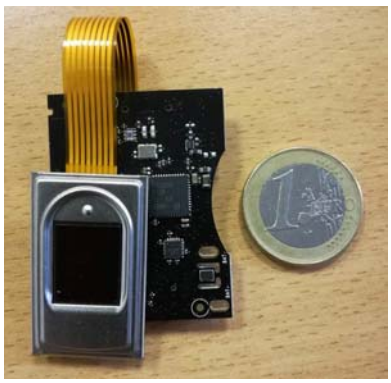

Fig. 1. A CC2650 Sensor Tag with the fingerprint sensor.

5. For each candidate convex core, a 128x128 distinctive window of the first smoothed directional image centered at that point is extracted.
6. Down-sampling is applied to each candidate distinctive window to generate a feature vector of 16x16 direction values, named QFingerMap16.

At the enrollment phase, the candidate feature vectors are stored as template. At the matching phase, the feature vectors extracted from the input fingerprint image are compared to the template. The similarity (score) between two feature vectors is computed as the number of direction values that coincide in both of them. Only the direction values within the ROI are considered and the similarity is given in percentage of the total number of direction values compared. The similarity between one finger sample enrolled and another captured at the matching phase is selected as the maximum similarity of all the possible combinations between input and template feature vectors. The fusion of several samples of several fingers (instances) is performed at score level (as described in [2]).

III. Realization of a Demonstrator

The fingerprint recognition algorithm has been implemented into the low-power microcontroller of a Texas Instruments LaunchPad CC2650 evaluation kit. The microcontroller is based on ARM Cortex-M3 architecture working at 48 MHz, with 128 KB of flash memory and 20 KB of SRAM. The sensor used is a Fingerprints FPC1011F3 capacitive sensor which provides images of 152x200 pixels, with 8 bits representing the grayscale values. Figure 1 shows the small size that can be achieved when using the capacitive sensor with the CC2650 Sensor Tag.

The fingerprint recognition algorithm was programmed in C, taking into account fixed-point arithmetic. Each feature vector extracted is composed of a maximum of 1,024 bits (16x16x4=128 bytes). The variables stored in SRAM occupy 17,159 bytes and the variables stored in flash memory (including the template) occupy 20,480 bytes. 34,757 bytes from the flash memory are used for the program. The extraction of the feature vector is done in average in 5.52 s and 654 µs are needed in average for the matching operation between two feature vectors. In active mode, the power consumption of the CC2650 is 11.53 mW, according to the datasheet for the Texas Instruments CC2650EM-5XD reference design with T_c=25º and V_{DDS}=3.0 V.

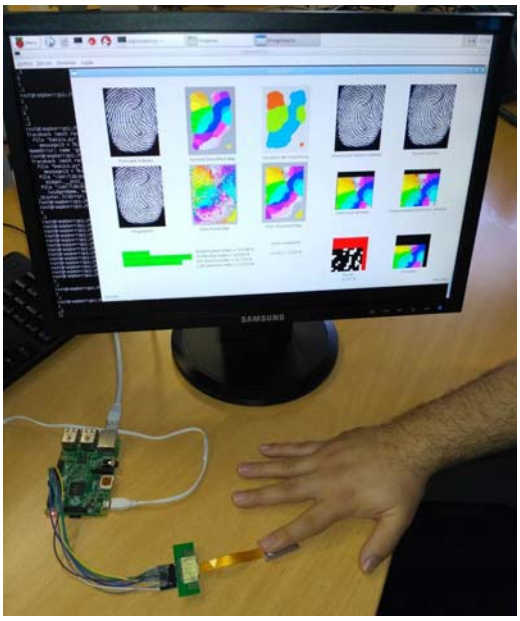

Fig. 2. GUI with the Raspberry Pi 2.

In order to illustrate the steps of the fingerprint recognition, the algorithm was also implemented in a Raspberry Pi 2 and a Graphical User Interface (GUI) was developed. Figure 2 shows the sensor connected to the Raspberry and the GUI with the results of a matching phase. All the steps summarized in Section II are shown in the example in Figure 2 because the fingerprint was acquired with enough quality. If the algorithm detects that the capture is not good, the GUI informs the user about the reasons of the low quality and requests another sample.

IV. Conclusions

This work demonstrates the performance of a lightweight fingerprint recognition algorithm implemented into a low-power ARM Cortex-M3 microcontroller working at 48 MHz. The memory required for a fingerprint feature vector stored as template is 128 bytes. The feature extraction is performed in average in 5.52 s and the matching between two vectors in 654 µs. The power consumption of the microcontroller in active mode is 11.53 mW.

References

[1] D. Maltoni, D. Maio, A. Jain, S. Prabhakar, “Handbook of Fingerprint Recognition”, Springer, 2009.

[2] R. Arjona, I. Baturone, “A Dual-Factor Access Control System based on Device and Use Intrinsic Identifiers”, Proccedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), pp. 1-6, 2016.

[3] R. Arjona, I. Baturone, “A Hardware Solution for Real-Time Intelligent Fingerprint Acquisition”, Journal of Real-Time Image Processing, Springer, Vol. 9, No. 1, pp. 95-109, 2014.