FPGA Implementation of Orientation Field Estimation of Fingerprint Recognition Process

S. Gayathri 1 and V Sridhar 2
1Dept of E & C, SJCE, Mysore, Karnataka, India
Email: sgmurthy_65@yahoo.com
2Dept of E&C, PESCE, Mandya, Karnataka, India
Email: venusridhar@yahoo.com

Abstract— Most of the automatic fingerprint verification systems are based on minutiae pattern matching. The accuracy of the system depends on the quality and location of minutiae. Most of the time, the quality of the ridges in the fingerprint image is not well defined. This leads to failure of precise location of minutiae. Hence, it is essential to get distinct ridges from the fingerprint images.

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. Distinct ridges provide accurate extraction of the minutiae from the fingerprint images. Implementation of orientation field estimation of fingerprint recognition process on Vertex-II Pro FPGA development board with verilog HDL is proposed.

Index Terms— matching, ridges, orientation, enhancement, filtering, implementation

I. INTRODUCTION

Fingerprint based identity verification, is one of the most widely used biometric systems [1] due to its easiness of acquisition, high distinctiveness, persistence, and acceptance by the public. FPGA is a good choice for implementing fingerprint recognition application because it has a large logic capacity and memory resources [2].

Usually, the fingerprint recognition process [2] involves a series of image enhancement and minutiae extraction steps classified as follows: Fingerprint normalization, Orientation and field frequency estimation, Filtering, Image binarization, Image thinning, Minutiae extraction, and False minutiae elimination.

Among these steps, orientation and filed frequency estimation is the most important of all because the accuracy of the system depends on the quality and location of minutiae. In turn, this depends on the ridges in the fingerprint image. In an ideal fingerprint image, ridges, and valleys alternate and flow in a locally constant direction.

Usually in a poor quality fingerprint image, ridges are not well defined. This leads to inaccurate extraction of minutiae. Thus, image enhancement techniques are employed to improve the quality of fingerprint images. Success of automatic fingerprint identification systems (AFISs) strongly depends on the accurate detection of a reference point on the fingerprints [3].

Reference points are obtained using orientation field and edge detection algorithms with secondary filter. This ensures high accuracy, high performance, and tolerance against orientation and translation of the input.
Two different forms of ridge extraction algorithm [4] are implemented on FPGA. The first one by micro-blaze soft-processor, and the second one based on a hardware coprocessor. Hardware coprocessor shows a higher performance.

The quality of fingerprint images and extraction of minutiae have an important role in the performance of automatic identification and verification. The minutiae extraction algorithm starts with a preprocessing of fingerprint image for improving the quality of images without changing the local and global properties of the image [5].

Commonly used features for improving fingerprint image quality are Fourier spectrum energy, Gabor filter energy, and local orientation. Accurate segmentation of fingerprint ridges from noisy background is necessary. For efficient enhancement and feature extraction algorithms, the segmented features must be void of any noise. It is very important to detect singular points (core and delta) accurately and reliably for classification and matching of fingerprints. An improved method of singularity detection based on continuous orientation field [6], improves accuracy of the position and reliability of the singular points. The proposed work overcomes the shortcoming of the traditional methods. The main benefit of this algorithm is its fast operational speed.

One of the main challenges in building an efficient and scalable automatic fingerprint identification system is to identify features, which are highly discriminative and are reproducible across different prints of the same finger. Most existing fingerprint matching approaches rely on minutiae geometry. Relatively, little effort has done in analyzing ridge flow patterns present in the fingerprint, partly due to difficulty in extracting robust discriminative features from the fingerprint images. The curvature information as proposed in [7], can be utilized along with the existing minutiae geometry-based features, for further reducing the number of potential candidates for fingerprint identification.

Most of the automatic fingerprints verification systems are based on minutiae pattern matching [8]. System accuracy depends on the ridge quality for precise extraction of minutiae. Preprocessing of Fingerprint image improves the quality of ridges. Many kinds of enhancement methods [9] for fingerprint image are proposed. Most of them are based on image binarization, and others enhance images directly from gray scale images. The orientation field of a fingerprint image defines the local orientation of the ridges in the fingerprint. It is a matrix of direction vectors representing the ridge orientation at each location of the image. Gradient-based approach is widely used to calculate the gradient [10], which makes use of the fact that the orientation vector is orthogonal to the gradient.

The orientation estimation is a fundamental step in the image enhancement process. The subsequent Gabor filtering stage relies on the local orientation, to enhance the fingerprint image. Least mean square estimation method employed to get orientation field from the normalized fingerprint image [11].

Reconfigurable computing adds a new degree of freedom to the traditional hardware/software design flow in the development of electronic systems [12]. However, the physical implementation of automatic fingerprint recognition system is still challenging task. Until now, only the initial stages of biometric recognition algorithm are tested.

Fingerprint image enhancement through reconfigurable hardware accelerators using hardware time multiplexing saves silicon area as compared with the existing general-purpose microcontroller system [13]. Fingerprint image processing through reconfigurable hardware [14] implemented on Virtex-4 by means of hardware-software co-design techniques, is faster in terms of execution time as compared to the existing techniques.

II. METHODOLOGY

The fingerprint is composed of ridges and valleys. The interleaved pattern of ridges and valleys are the most evident structural characteristic of a fingerprint. There are three main fingerprint features: Global Ridge Detail, Local Ridge Detail, and Intra Ridge Detail [5].

A. Global ridge detail

There are two types of ridge flows: the pseudo-parallel ridge flows and high-curvature ridge flows, which are located around the core point and/or delta point(s). This representation relies on the ridge structure, global landmarks, and ridge pattern characteristics. The commonly used global fingerprint features are:

i) Singular points:
They are discontinuities in the orientation field. There are two types of singular points called core and delta. A core is the uppermost of a curving ridge, and a delta point is the point where three ridge flows meet. Singular points are used for fingerprint registration and classification.

**ii) Ridge orientation map:**
They are local direction of the ridge-valley structure. It is helpful in classification, image enhancement, feature verification and filtering.

**iii) Ridge frequency map:**
They are the reciprocal of the ridge distance in the direction perpendicular to local ridge orientation. Ridge frequency map is used for filtering of fingerprint images.

**B. Local Ridge Detail**
This is the most widely used detail in fingerprint representation. Local ridge details are the discontinuities of local ridge structure referred as minutiae. Forensic experts use ridge detail to match two fingerprints. There are about 150 different types of minutiae. Among these minutiae types, ridge ending, and ridge bifurcation are the most widely used, as depicted in Figure 1.

![Figure 1: Features of fingerprint image](image)

Most of the automatic fingerprint recognition systems use minutiae as their fingerprint representations. The location information and the direction of a minutiae point alone are not sufficient for achieving high performance. Minutiae-derived secondary features are being used.

**C. Intra Ridge Detail**
On every ridge of the finger epidermis, there are many tiny sweat pores and other permanent details. Pores are distinctive in terms of their number, position, and shape. However, extracting pores is feasible only in high-resolution fingerprint images and with very high image quality. Thus, the cost is very high. This representation is not adopted by current automatic fingerprint identification systems (AFIS).

The advantages of fingerprint recognition system are: they are universal, reliable, accurate, non-intrusive, and they remain structurally unchanged [5]. There are three approaches for fingerprint recognition. They are image-based approach, texture-based approach, and minutiae-based approach [5].

Minutiae-based approach is widely used because it is invariant to translation, rotation, and scale changes. It is however error prone in low quality images. Usually before minutiae extraction, image preprocessing is performed. Fingerprint enhancement is a preprocessing technique used to reduce the noise and improve the clarity of ridges against valleys. Thus the ridges becomes more distinctive in fingerprint images.

**III. DESIGN**
The fingerprint recognition technique involves different processes like Normalization, Orientation Field Estimation, Filtering, Binarization, Thinning, Minutiae extraction, and False minutiae elimination.

**A. Orientation Field Estimation**
The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint image. It is determined by convolving each pixel with Gabor filter [8]. Gabor filters are employed because they have frequency and orientation selective property.

For an image edge, its direction is defined in terms of the gradient, pointing in the direction of maximum image intensity increasing from dark to bright. This implies that two edges can have the same orientation but the corresponding image gradients point in opposite directions if the edges go in different directions.

A local image orientation representation is computed for fingerprint image data. This information is used for solving line or edge consistency estimation, curvature information estimation, corner point’s detection, and adaptive or anisotropic noise reduction.
B. Algorithm

Based on least mean square estimation method, proposed by Hong [10], let $\theta$ be defined as the orientation field of a fingerprint image. $\theta(i, j)$ represent the local ridge at pixel $(i, j)$. Local ridge, however, is usually specified for a block rather than that for every pixel. Figure 2 depicts the estimation of local ridge orientation in a fingerprint image.

![Figure 2: Estimation of local ridge Orientation](image)

The main steps of the algorithm are as follows:
1. Divide the input image into blocks of size $w \times w$.
2. Compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel.
3. Estimate the local orientation using the following equations.

$$V_x(i, j) = \sum_{u=-w/2}^{w/2} \sum_{v=-w/2}^{w/2} 2\partial_x(u,v)\partial_x(u,v)$$

$$V_y(i, j) = \sum_{u=-w/2}^{w/2} \sum_{v=-w/2}^{w/2} 2\partial_y(u,v)\partial_y(u,v)$$

4. Assume that the local ridge orientation varies slowly in a local neighborhood where no singular point appears. The discontinuity of ridge and valley due to noise can be smoothened by applying a low pass filter. To apply a low pass filter the orientation image is converted into a continuous vector field. This continuous vector field, for which its x and y components are defined as $\Phi_x$ and $\Phi_y$, respectively, are as follows:

$$\Phi_x(i, j) = \cos(2\theta(i, j))$$

$$\Phi_y(i, j) = \sin(2\theta(i, j))$$

5. With the resulting vector field, the two dimensional low pass filter $G$ with unit integral is applied. The specified size of the filter is $w_b \times w_b$. As a result,

$$\Phi_x(i, j) = \sum_{u=-w_b/2}^{w_b/2} \sum_{v=-w_b/2}^{w_b/2} G(u,v)\Phi_x(i-uw, j-vw)$$

$$\Phi_y(i, j) = \sum_{u=-w_b/2}^{w_b/2} \sum_{v=-w_b/2}^{w_b/2} G(u,v)\Phi_y(i-uw, j-vw)$$

6. The smoothened orientation field (local ridge orientation at $(i, j)$) can then be computed as follows:
IV. HARDWARE / SOFTWARE REQUIREMENT

A. Virtex-II Pro FPGA development board

Figure 3 depicts the block diagram of the hardware specifications provided by the XUP Virtex-II Pro FPGA Development board. Virtex-II Pro based designs have the PPC405, Rocket IO transceiver, and potentially a large part of the user's system within a single FPGA. These additional capabilities add complexities to the development and verification of the system. The user must build software applications to be transferred to the hardware domain so that the evolving design can be verified.

The PPC405 and the Rocket IO transceiver require accurate simulation models that integrate into the verification procedure. Designs using the processor are more likely to be collections of various cores rather than a custom design, proper management of design libraries becomes important.

\[
o(i, j) = \frac{1}{2}\tan^{-1}\left(\frac{\Phi'(i,j)}{\Phi''(i,j)}\right)
\]

(8)

B. MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
C. Xilinx ISE

Xilinx ISE is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize (“compile”) the designs, perform timing analysis, examine RTL schematics, simulate a design’s reaction to different stimuli and configure the target device with the programmer. The Xilinx ISE is an integrated development environment for Xilinx FPGAs and CPLDs.

Xilinx ISE is a complete FPGA/CPLD programmable logic design suite providing
1) Specifications of programmable logic, via schematic capture or Verilog/VHDL.
2) Synthesis and fit of specified logic into Xilinx Cool Runner and Spartan devices.
3) Behavioral and post-fit simulation.
4) Download of configuration into target device via communications cable.
5) To design and implement any kind of electronic circuit, connecting cables, LED’s and chips ends up with a big and confusing pile of small colored cables. Xilinx ISE simply takes this idea to next level (or several levels above). It helps in having access to hundreds of industrial standard chips, from the standard AND or NOT gates, all the way to the more complex ones. After the design phase, the design can be tested by changing input values and checking the results. This will take some time, as each circuit has to be completely compiled before a test can be done.

Advantages of FPGA
(1) Real time.
(2) Portable.
(3) Embedded.
(4) Small size.
(5) Low power consumption.
(6) High processing speed.
(7) Architecture can be changed whenever required.
(8) Parallel processing.

D. Displaying Output on VGA Monitor

VHDL source code for a simple VGA controller (VgaCon) is provided. The purpose of VgaCon is to isolate the details of VGA signal generation from all the other modules in a design. VgaCon allows the pixel
information to be written into its internal video memory using a very simple interface, while it is alone responsible for generating the required signals for displaying the pixel information on a VGA monitor. Thus, for modules interfacing with VgaCon, the process of drawing on the screen consists of a request to set the intensity a point located at any valid pixel location. Figure 5 shows the displaying module of VGA monitor.

The VGA monitor can be thought of as a grid of pixels (picture elements which can be individually set to a specific intensity). It contains 480 rows of 640 horizontal pixels. Most monitors, including VGA, use a serial scheme to set the intensity of each pixel. This means that the VGA controller sends the intensity information for each pixel one at a time, rather than being able to set all of them at once in a parallel scheme.

The VGA monitor can be thought of as a grid of pixels (picture elements which can be individually set to a specific intensity). It contains 480 rows of 640 horizontal pixels. Most monitors, including VGA, use a serial scheme to set the intensity of each pixel. This means that the VGA controller sends the intensity information for each pixel one at a time, rather than being able to set all of them at once in a parallel scheme.

Thus, the controller is only able to provide full intensity. The VGA monitor does not save any of the pixels written to it so the pixels must be continuously written to the monitor for the image to remain stable.

Database
Database is a collection of information either in the form of data or images. Image used for simulation and implementation of the proposed block is taken from the database generated using fingerprint images acquired from NFD HU 3.8 version optical scanner.

V. IMPLEMENTATION OF ORIENTATION BLOCK

A. Matlab implementation

The Figure 6 shows the flowchart of Orientation block. Normalized image is taken as input to the orientation block. The gradients of the Normalized image is calculated in both x and y directions. The sobel operator used to detect the edges of the image and to estimate the orientation angles of the image.

The normalized fingerprint image is used to find the gradients at each pixel level. Second order derivatives are calculated for each image pixels of the previous process. The local orientation and the orientation angle of each pixel is calculated.

B. Verilog implementation

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint image. The orientation algorithm for the fingerprint image is realized using verilog HDL. The orientation block consists of ROM, RAM memory blocks, counters, multipliers, adders, address control blocks, Math function unit and divider.

The normalized image from the ROM memory is copied to a RAM memory block during the positive edge of the clock. The output from the RAM is used to calculate the gradients of the image both in x and y directions. The calculated gradient values are used to find the orientation field of the image with the help of divider and Math function blocks.

The Core generator software is used to generate the IP cores for different blocks like Divider Generator, Block Memory Generator, CORDIC, etc. To implement the orientation process these IP core blocks are used. Figure 7 shows the corresponding RTL schematic of the orientation block. The orientation block is implemented with the help of individual modules like ROM, RAM, Sobel operator, Adder, Divider Multiplier, etc. These blocks are interconnected together to form the complete orientation block.
VI. EXPERIMENTAL RESULTS

A. Simulation results using Matlab

Figure 8 shows the MATLAB output of the orientation block. The Normalized image’s gradients in both x and y directions are calculated and used to compute the orientation angle of the image. The orientation process provides the horizontal and vertical derivatives of the normalized image. The normalized image of the input fingerprint is as shown in Figure 8(a). Using Sobel operator, the edge-detected image is obtained as shown in Figure 8(b). The gradients of the first derivative of the normalized fingerprint image provide horizontal and vertical derivatives as shown in Figure 8(c) and Figure 8(d) respectively.
The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. Combining the horizontal and vertical derivatives of the normalized image provides the orientation angles of the fingerprint image as shown in Figure 9.

B. Simulation results using verilog HDL

Figure 10(a) to Figure 10(f) shows the simulation waveforms of the various process of orientation block. The outputs are synchronized with 100 MHz clock input. The outputs from the individual modules of orientation block are at a high impedance state until the image copied to the RAM block. The gradients are calculated for each pixel values of the image. These gradient values are used to calculate the orientation angle of the image. The outputs are in the signed decimal value. The normalized fingerprint image is stored in the ROM initially. At the rising edge of the clock, the image is copied into RAM from ROM for processing. This is depicted in the simulation waveform as shown in Figure 10 (a) and Figure 10 (b) respectively.
The gradient values both in the horizontal and vertical direction for the first derivative are depicted in the simulation waveform as shown in Figure 10 (c) and in Figure 10 (d) respectively. Second order derivatives are obtained by combining the first derivative as depicted in simulation waveform shown in Figure 10(e) and Figure 10(f) respectively.

Results obtained from simulation and implementations are discussed in three different sections considering different aspects: firstly hardware resources used, followed by processing time and finally the performance.

A. Hardware resources used

The orientation block is implemented on Vertex-II Pro FPGA development board. The hardware occupied on the FPGA board is tabulated as in Table I.

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice Registers</td>
<td>3752</td>
<td>69120</td>
<td>5</td>
</tr>
<tr>
<td>Number of Slice LUTs</td>
<td>2545</td>
<td>69120</td>
<td>3</td>
</tr>
<tr>
<td>Number of fully used LUT-FF pairs</td>
<td>1538</td>
<td>4759</td>
<td>32</td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>34</td>
<td>640</td>
<td>5</td>
</tr>
<tr>
<td>Number of BUFG/BUFGCTRLs</td>
<td>1</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>Number of DSP48Es</td>
<td>2</td>
<td>64</td>
<td>3</td>
</tr>
</tbody>
</table>
B. Processing time

The processing time from the FPGA implementation is compared with that of Matlab simulation time. The processing time for the orientation block implemented on Virtex-II Pro FPGA development board is in nanoseconds (< 2.86ns), whereas the same is in terms of few microseconds using Matlab.

C. Performance

For the purpose of the performance evaluation, the database containing the fingerprint image of individuals (irrespective of age and gender) is generated. The database comprises the images acquired using NFD scanner. The input normalized fingerprint image of FPGA is as shown in Figure 11 and the output oriented fingerprint image of FPGA is as shown in Figure 12.

VII. CONCLUSION AND FUTURE SCOPE

The Fingerprint recognition process involves a series of image enhancement and minutiae extraction steps like Normalization, Orientation field estimation, Filtering, Image binarization, Image thinning, Minutiae extraction, and False minutiae elimination. Only the orientation block is implemented on virtex-II pro FPGA development board with verilog HDL.

The simulation results obtained are satisfactory to carry out the further blocks of the fingerprint recognition process. The processing time will be in the order of nanoseconds in FPGA whereas it is in terms of microseconds in Matlab.

The hardware solution significantly reduces the processing time by almost 1000 times and the reduction in silicon area prompt to carry further fingerprint recognition processes. Further, it is suggested to carry out the remaining blocks of the fingerprint recognition process. In future, it is proposed to integrate all the blocks of the Fingerprint recognition process on higher version of FPGA board.

REFERENCES


