

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de
Eletrónica e Telecomunicações e de Computadores

ISEL

Mestrado em Engenharia de Eletrónica e Telecomunicações

**Sistema de Reconhecimento de Impressões Digitais
Baseado em FPGA**

Ricardo Miguel Coelho Leitão

(Licenciado em Engenharia Eletrónica e Telecomunicações e de Computadores)

TRABALHO DE PROJETO PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA DE
ELETRÓNICA E TELECOMUNICAÇÕES

Orientadores:

Prof. Doutor Artur Jorge Ferreira
Eng. Tiago Miguel Braga da Silva Dias

Júri:

Presidente e Vogal: Prof. Doutor Mário Pereira Véstias
Vogal: Prof. Doutor Pedro Miguel Torres Mendes Jorge
Vogal: Prof. Doutor Artur Jorge Ferreira

Dezembro de 2014

Agradecimentos

Gostaria de começar por agradecer aos professores Artur Ferreira e Tiago Dias pelo apoio e disponibilidade durante a orientação deste trabalho. Os conselhos dados e o apoio prestado, permitiram levar o projeto a bom porto. Gostaria também de agradecer ao professor António Serrador pela cedência de espaço no Laboratório de Investigação e Desenvolvimento para desenvolver este trabalho.

Gostaria ainda de agradecer aos meus colegas e amigos Filipe Palhinha, João Rosário, João Zagalo, Milton Barbosa, Nuno Sousa, Romil Argi e Telmo Anacleto, não só pela contribuição na criação da base de dados, como também por me terem acompanhado ao longo deste trabalho.

Deixo também um agradecimento especial aos meus pais pelo apoio dado ao longo de todo o meu percurso académico. Sem eles este trabalho não seria possível.

Resumo

No presente trabalho é proposta a implementação de um sistema embebido de reconhecimento de impressões digitais em *Field Programmable Gate Array* (FPGA). Este sistema é suportado num algoritmo de extração de características baseado em textura orientada. Para otimizar o desempenho do sistema proposto, a execução dos componentes mais exigentes deste algoritmo é realizada recorrendo a hardware dedicado, especialmente desenvolvido no âmbito deste trabalho usando a linguagem *Very-High-Speed Integrated Circuits* (VHSIC) *Hardware Description Language* (VHDL).

Na análise em *software* do sistema proposto é efetuada uma otimização dos parâmetros do algoritmo de extração de características para o sensor ótico adotado, obtendo-se um sistema biométrico com *Equal-Error Rate* (EER) de 7,18%. Após a definição dos parâmetros ótimos, o algoritmo foi descrito em linguagem C e implementado no *softcore MicroBlaze*. Para reduzir o impacto no desempenho do sistema do módulo de filtragem de Gabor, que consome 67,11% do tempo total da execução em *software*, foi desenvolvido um acelerador em *hardware*.

A filtragem pretendida é realizada mediante uma redução da dimensão das janelas, que são separadas em filtros a uma dimensão. Durante o processo de filtragem, os dados são convertidos de vírgula flutuante para vírgula fixa. O periférico desenvolvido requer 18 blocos DSP48E1 e 22 memórias RAMB36E1 para realizar a filtragem. Para partilhar dados entre o *MicroBlaze* e o periférico também são utilizadas duas memórias, com 64 kB cada, ocupando no total 32 memórias RAMB36E1. Comparando o desempenho do sistema entre as soluções com e sem aceleração por *hardware*, verifica-se que o EER se mantém nos 7,18%. O ganho no tempo de processamento no algoritmo é de 2,72 vezes, enquanto que na filtragem de Gabor é de 731,99 vezes. Como consequência, a percentagem do tempo total de execução associado à filtragem de Gabor diminui de 67,11% para aproximadamente 2,3%.

Palavras-chave:

Impressão digital, FPGA, sistema biométrico, reconhecimento, filtragem de Gabor, VHDL.

Abstract

In this work, it is proposed an implementation of an embedded system for fingerprint recognition using Field Programmable Gate Array (FPGA) devices. The system is supported by a feature extraction algorithm based on oriented texture. In order to optimize the system performance, the Very-High-Speed Integrated Circuits (VHSIC) Hardware Description Language (VHDL) is used to develop specialized computational circuits to achieve hardware acceleration for the most demanding parts of the algorithm.

The software analysis of the proposed system allowed to fine tune some of the parameters of the considered feature extraction algorithm to the characteristics of the adopted optical sensor, and thus obtain a biometric system with an Equal-Error Rate (EER) of 7.18%. After settling the optimal parameters, the algorithm was described by using the C programming language and ported to the *MicroBlaze* softcore. To reduce the impact of the Gabor filtering module in the system overall performance, which holds 67.11% of the total execution time of the software implementation, a hardware accelerator was developed.

The required filtering task is accomplished by reducing the size of the kernels, which are separated into one dimension filters. During the filtering process, the data are converted from floating-point to fixed-point representation. The developed peripheral uses 18 DSP48E1 blocks and 22 RAMB36E1 memories to perform the filtering process. In addition, two 64 kB memories are used to share the data between the *MicroBlaze* and the peripheral, which involves a total of 32 RAMB36E1 memories. By comparing the overall system performance for the solutions with and without hardware acceleration, it was found that the EER is kept at 7.18%. The gain in terms of the algorithm's processing time is 2.72, while for the Gabor filtering it's 731.99. As a result, the percentage of the total execution time associated with Gabor filtering decreases from 67.11% to, approximately, 2.3%.

Keywords:

Fingerprint, FPGA, biometric system, recognition, Gabor filtering, VHDL.

Índice

Resumo	v
Abstract	vii
Lista de Figuras	xi
Lista de Tabelas	xv
Lista de Acrónimos	xvii
Lista de Símbolos	xix
1 Introdução	1
1.1 Enquadramento	1
1.2 Estado da arte	1
1.3 Principais contributos do projeto	6
1.4 Organização do documento	7
2 Biometria	9
2.1 Indicadores biométricos	9
2.2 Aplicações	12
2.2.1 Caracterização das aplicações	12
2.2.2 Tipo de aplicações	13
2.3 Tipos de reconhecimento biométrico	15
2.3.1 Sistema biométrico	15
2.4 Erros do sistema biométrico	17
2.4.1 Tipos de erro	18
2.4.2 Comparação	19
2.4.3 Autenticação	20
2.4.4 Identificação	21
2.5 Avaliação de sistemas biométricos	21
2.6 Impressão digital	24
2.6.1 Constituição	25
2.6.2 Aquisição de impressões digitais	27
3 Reconhecimento de impressões digitais: técnicas e ferramentas	31
3.1 Extração e comparação de características baseado em minúcias	31
3.1.1 Estimação da orientação das riscas	32
3.1.2 Extração de singularidades	33

3.1.3	Extração das riscas	34
3.1.4	Extração de minúcias	35
3.1.5	Comparação de características	36
3.1.6	Avaliação de desempenho	37
3.1.7	Desafios no reconhecimento de impressões digitais	38
3.2	Algoritmo de extração de características baseado em textura orientada	39
3.2.1	Determinação do ponto de referência	40
3.2.2	Seleção da zona de interesse e divisão em setores	45
3.2.3	Normalização da imagem por setores	46
3.2.4	Aplicação dos filtros de Gabor	47
3.2.5	Cálculo do vetor de características	49
3.3	Comparação de características de impressões digitais	50
4	Estudo e análise do sistema biométrico em <i>software</i>	53
4.1	Caracterização do sistema biométrico	53
4.1.1	Caracterização do sensor	54
4.1.2	Caracterização da base de dados	55
4.2	Estudo e análise em MATLAB	56
4.2.1	Resultados experimentais	58
4.3	Estudo e análise em linguagem C	58
4.3.1	Arquitetura de teste	59
4.3.2	Resultados experimentais	59
4.4	Otimização dos filtros de Gabor	60
4.4.1	Decomposição em valores singulares	62
4.4.2	Resultados experimentais	63
5	Implementação do sistema biométrico em <i>hardware</i>	67
5.1	Caracterização da FPGA	67
5.2	Arquitetura do sistema biométrico com aceleração por <i>hardware</i>	68
5.3	Arquitetura do periférico <i>filter_Gabor</i>	69
5.3.1	<i>AXI Slave IPIF e User Logic</i>	69
5.3.2	<i>Main</i>	70
5.4	Resultados experimentais	75
5.4.1	Ocupação de recursos	75
5.4.2	Avaliação de desempenho	76
6	Conclusões	81
6.1	Trabalho futuro	82
	Referências	85
A	Constituição da base de dados	93
B	Decomposição SVD dos filtros de Gabor	97
C	Análise de sensibilidade aos parâmetros do algoritmo de reconhecimento de impressões digitais	101

Lista de Figuras

1.1	Linha temporal de publicações sobre reconhecimento de impressões digitais, em sistemas embebidos no período 1995-2013, inclusivamente.	2
1.2	Tipos de implementação de sistemas de reconhecimento de impressões digitais com recurso a FPGA: (a) sistema controlado por um processador e com aceleração por <i>hardware</i> (co-processador), (b) implementação de um processo do algoritmo unicamente em <i>hardware</i>	4
2.1	Indicadores biométricos: íris, face, geometria da mão, voz, assinatura e impressão digital.	10
2.2	Diagrama de blocos de um sistema biométrico genérico e respetivos componentes individuais.	15
2.3	FMR e FNMR através da intersecção da distribuição genuína e da distribuição impostora (adaptado de [34]).	22
2.4	FMR e FNMR em função do nível de decisão t (adaptado de [34]).	23
2.5	Diferentes tipos de aplicações através da ROC (adaptado de [34]).	24
2.6	Constituição das impressões digitais: (a) riscas (<i>ridges</i>) e vales (<i>valleys</i>), (b) ponto central (<i>core</i>) e singularidades - laço (<i>loop</i>), delta e espiral (<i>whorl</i>) [34].	25
2.7	Classificação de impressões digitais: laço esquerdo (<i>left loop</i>), laço direito (<i>right loop</i>), espiral (<i>whorl</i>), arco (<i>arch</i>) e arco em tenda (<i>tented arch</i>) [34].	26
2.8	Constituição das impressões digitais em cinco classes: (a) terminações (<i>ridge ending</i>) e bifurcações (<i>bifurcation</i>), (b) largura, forma e poros.	26
2.9	Impressões digitais adquiridas por: (a) processo de <i>rolled-ink</i> , (b) sensor ótico [34].	27
2.10	Tipos de sensor <i>live-scan</i> : (a) ótico, (b) estado sólido, (c) ultrassons [34].	29
3.1	Diagrama de blocos do processo de extração de características (minúcias) (adaptado de [27]).	32
3.2	Exemplos de <i>Poincaré index</i> para as situações: (a) não existe uma singularidade, (b) existe um <i>loop</i> , (c) existe um <i>delta</i> , (d) existe um <i>whorl</i> (adaptado de [27]).	34
3.3	Melhoramento de impressões digitais através de filtros de Gabor: (a) impressão digital original, (b) impressão digital após melhoramento [27].	35
3.4	Deteção de minúcias: (a) não existe terminação ou bifurcação (ponto intermédio), (b) terminação, (c) bifurcação [34].	36
3.5	Diagrama de blocos do processo de comparação de características (minúcias) (adaptado de [27]).	38
3.6	Algoritmo de extração de características.	40
3.7	Exemplo de cálculo do gradiente: (a) amostra original, (b) gradiente da amostra ao longo do eixo dos x (∂_x), (c) gradiente da amostra ao longo do eixo dos y (∂_y).	42

3.8	Cálculo da orientação de campo: orientação de campo sobreposta com a amostra original antes (a) e após (b) a filtragem; orientação de campo representada como distribuição de intensidade antes (c) e após (d) a filtragem.	43
3.9	Determinação do ponto de referência: (a) resultado da aplicação do filtro de variância, (b) localização do ponto de referência na amostra original.	45
3.10	Zona de interesse: (a) impressão digital com a zona de interesse centrada no ponto de referência (\times) com os parâmetros $b = 20$, $B = 5$ e $k = 16$ [42], (b) constituição da zona de interesse.	46
3.11	Exemplo de zona de interesse: (a) antes da normalização, (b) depois da normalização.	47
3.12	Máscaras de Gabor para as orientações $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$	48
3.13	Exemplo de filtragem de Gabor para as orientações $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$	49
3.14	Constituição do <i>FingerCode</i>	50
3.15	Ângulos de rotação dos <i>FingerCodes</i> para $R \in \{-1, 0, 1\}$ com $\alpha < \frac{360^\circ/k}{2}$	51
4.1	Diagrama de blocos do sistema biométrico utilizado na avaliação de desempenho.	54
4.2	Exemplo de impressões digitais recolhidas através do sensor GT-511C3: (a) <i>left loop</i> , (b) <i>right loop</i> , (c) <i>whorl</i> , (d) <i>arch</i> e (e) <i>tented arch</i>	55
4.3	Exemplo de três impressões digitais do mesmo dedo, recolhidas através do sensor GT-511C3.	56
4.4	FMR e FNMR em função do nível de decisão t	59
4.5	Arquitetura utilizada na avaliação do sistema biométrico.	60
4.6	Distribuição de processamento do algoritmo, obtido através da opção <i>profile</i> na ferramenta XDSK.	61
4.7	Filtragem de Gabor com: (a) janela 9x9, (b) janela separada em uma linha e uma coluna.	62
4.8	Aplicação dos filtros de Gabor separáveis com o conjunto de ângulos proposto em [5].	63
4.9	Aplicação dos filtros de Gabor separáveis por SVD.	64
4.10	EER e FNMR em função da dimensão da janela dos filtros de Gabor.	64
4.11	EER e FNMR em função do número de bits da parte fracionária.	65
4.12	FMR e FNMR em função do nível de decisão t com precisão <i>double</i> e com 10 bits na parte fracionária.	65
5.1	Arquitetura do sistema biométrico.	69
5.2	Arquitetura do periférico <i>filter_Gabor</i>	69
5.3	Configuração dos registos partilhados.	70
5.4	Diagrama RTL do componente <i>main</i>	71
5.5	Diagrama RTL do componente <i>core</i>	72
5.6	Máquina de estados do componente <i>core</i>	73
5.7	Comparação entre as curvas de FMR e FNMR em função do nível de decisão t , obtidas com e sem aceleração por <i>hardware</i>	77
5.8	SNR e MSE das imagens após filtragem de Gabor.	78
6.1	Diagrama de blocos da arquitetura proposta com recurso a janela deslizante.	83
6.2	Diagrama de blocos da arquitetura proposta.	84

C.1	EER e FNMR em função do parâmetro b com: $B = 3, k = 16, gap = 1, 5,$ $ang = 7^\circ$ e $f = 0, 1.$	101
C.2	EER e FNMR em função do parâmetro b com: $B = 2, k = 16, gap = 1, 5,$ $ang = 7^\circ$ e $f = 0, 1.$	102
C.3	EER e FNMR em função do parâmetro k com: $B = 3, b = 15, gap = 1, 5,$ $ang = 7^\circ$ e $f = 0, 1.$	102
C.4	EER e FNMR em função do parâmetro k com: $B = 2, b = 15, gap = 1, 5,$ $ang = 7^\circ$ e $f = 0, 1.$	102
C.5	EER e FNMR em função do parâmetro gap com: $B = 3, b = 13, k = 12,$ $ang = 7^\circ$ e $f = 0, 1.$	103
C.6	EER e FNMR em função do parâmetro gap com: $B = 3, b = 14, k = 12,$ $ang = 7^\circ$ e $f = 0, 1.$	103
C.7	EER e FNMR em função do parâmetro gap com: $B = 3, b = 15, k = 12,$ $ang = 7^\circ$ e $f = 0, 1.$	103
C.8	EER e FNMR em função do parâmetro ang com: $B = 3, b = 14, k = 11,$ $gap = 1, 5$ e $f = 0, 1.$	104
C.9	EER e FNMR em função do parâmetro f com: $B = 3, b = 14, k = 11,$ $gap = 1, 5$ e $ang = 27^\circ.$	104

Lista de Tabelas

2.1	Comparação entre indicadores biométricos. E, M e B significam, respectivamente, Elevado, Médio e Baixo (adaptado de [34]).	12
2.2	Principais aplicações de sistemas biométricos nas áreas forense, governamental e comercial.	14
4.1	Especificações do sensor GT-511C3 [1].	55
4.2	Caracterização da base de dados criada.	56
4.3	Parâmetros do algoritmo de reconhecimento de impressões digitais.	57
5.1	Principais características da Virtex-7 XC7VX485T-2FFG1761C [60].	68
5.2	Resumo da configuração dos <i>IP LogiCOREs Float2Fixed</i> e <i>Fixed2Float</i>	74
5.3	Resumo da configuração do <i>IP LogiCORE FIR</i>	74
5.4	Ocupação de recursos da arquitetura.	75
5.5	Ocupação de recursos do periférico <i>filter_Gabor</i>	76
5.6	Tempos de execução do algoritmo de extração de características com o <i>MicroBlaze</i> e o componente <i>filter_Gabor</i> a 100 MHz.	78
A.1	Constituição da base de dados.	93

Lista de Acrónimos

Lista com os principais acrónimos.

AFIS - *Automatic Fingerprint Identification Systems*

AXI - *Advanced eXtensible Interface*

BRAM - *Block Random Access Memory*

CCD - *Charge Coupled Device*

CLB - *Configurable Logic Block*

CMOS - *Complementary Metal-Oxide-Semiconductor*

CORDIC - *COordinate Rotation DIgital Computer*

DDR - *Double Data Rate*

DPI - *Dots Per Inch*

DSP - *Digital Signal Processing / Digital Signal Processor*

EER - *Equal-Error Rate*

FAR - *False Acceptance Rate*

FBI - *Federal Bureau of Investigation*

FFT - *Fast Fourier Transform*

FIR - *Finite Impulse Response*

FMR - *False Match Rate*

FNIR - *False Negative Identification-Error Rate*

FNMR - *False Non-Match Rate*

FPGA - *Field Programmable Gate Array*

FPIR - *False Positive Identification-Error Rate*

FPU - *Floating-Point Unit*

FRR - *False Rejection Rate*

FTA - *Failure To Acquire*
FTC - *Failure To Capture*
FTD - *Failure To Detect*
FTE - *Failure To Enroll*
FTIR - *Frustrated Total Internal Reflection*
FTP - *Failure To Process*
FVC - *Fingerprint Verification Competition*
LED - *Light Emitting Diode*
LMS - *Least Mean Square*
LUT - *Lookup Table*
MATLAB - *MATrix LABoratory*
MSE - *Mean Squared Error*
NFIS - *NIST Fingerprint Image Software*
NIST - *National Institute of Standards and Technology*
PI - *Poincaré Index*
PPP - *Pontos Por Polegada*
ROC - *Receiver Operating Characteristic*
RTL - *Register-Transfer Level*
SB - *Sistema Biométrico*
SNR - *Signal-to-Noise Ratio*
SRL - *Shift Register Lookup Table*
SVD - *Singular Value Decomposition*
VHDL - *VHSIC Hardware Description Language*
VHSIC - *Very-High-Speed Integrated Circuits*
XSDK - *Xilinx Software Development Kit*
ZeroFMR - *Zero FMR*
ZeroFNMR - *Zero FNMR*

Lista de Símbolos

Lista com os principais símbolos.

b - largura de cada banda

B - número de bandas

C - conjunto de características

d_r - distância entre riscas

D_0 - *non-matching* entre T e C

D_1 - *matching* entre T e C

f_0 - frequência das riscas

$F_{i\theta}$ - imagem filtrada i segundo a direção θ

$G_{(x,y,f_0)}$ - filtro de Gabor a duas dimensões

h - resposta impulsional

H_0 - C e T não pertencem ao mesmo indicador biométrico

H_1 - C e T pertencem ao mesmo indicador biométrico

I - imagem

I_{bin} - imagem binarizada

I_{thin} - imagem após *thinning*

I_N - imagem normalizada

k - número de setores por banda

M_0 - média pretendida na normalização

ns - número de valores significativos da matriz S

$N_{amostras\ mesmo\ dedo}$ - número de amostras do mesmo dedo

$N_{amostras}$ - número de amostras

$N_{dedos\ diferentes}$ - número de amostras de dedos diferentes

$N_{dedos\ iguais}$ - número de amostras de dedos iguais
 O - orientação de campo
 O' - orientação de campo após filtragem
 $P_{i\theta}$ - valor médio dos píxeis
 s - nível de semelhança
 S - setor
 t - nível de decisão
 T - registo padrão
 V - variância da imagem
 V_0 - variância pretendida na normalização
 $V_{i\theta}$ - vetor de características
 x_{ref} - coordenada no eixo dos x do ponto de referência
 x_θ - coordenada no eixo dos x após rotação dos eixos
 y_{ref} - coordenada no eixo dos y do ponto de referência
 y_θ - coordenada no eixo dos y após rotação dos eixos
 α - ângulo de rotação
 δ_x - derivada segundo o eixo dos x
 δ_y - derivada segundo o eixo dos y
 ε - componente em seno de O'
 Φ_x - componente segundo o eixo dos x do vetor de campo
 Φ'_x - componente segundo o eixo dos x do vetor de campo após filtragem
 Φ_y - componente segundo o eixo dos y do vetor de campo
 Φ'_y - componente segundo o eixo dos y do vetor de campo após filtragem
 σ_x - constante espacial da envolvente gaussiana segundo o eixo dos x
 σ_y - constante espacial da envolvente gaussiana segundo o eixo dos y
 θ - ângulo segundo o eixo dos x

Capítulo 1

Introdução

1.1 Enquadramento

As impressões digitais são amplamente utilizadas no reconhecimento de indivíduos através de mecanismos de identificação e autenticação [34]. Para identificar ou autenticar um indivíduo através de uma impressão digital é necessário recolher e analisar diferentes características da mesma, sendo este um processo que se pode revelar ineficiente, em termos de tempo útil de resposta (latência) [23], ao ser realizado exclusivamente por *software*. Por outro lado, existem vantagens na realização deste tipo de reconhecimento através de sistemas embebidos, portáteis e compactos, os quais têm instalação e utilização simples. O avanço tecnológico verificado nas últimas décadas, e conseqüente redução nos custos de implementação em sistemas embebidos, impulsionou a comunidade científica e académica a procurar soluções com recurso a *hardware* reconfigurável [23]. Na figura 1.1 resume-se, na forma de linha temporal, alguns trabalhos científicos publicados que estão relacionados com sistemas de reconhecimento de impressões digitais; verifica-se que os primeiros estudos envolvendo *Field-Programmable Gate Arrays* (FPGAs) [11] ocorreram em meados da década de 1990. Analisando a quantidade de artigos científicos publicados a partir de 2005, verifica-se interesse crescente no desenvolvimento de sistemas baseados em FPGA, mantendo-se até aos dias de hoje.

1.2 Estado da arte

O conceito de biometria remonta às civilizações mais antigas da humanidade. Sabe-se que os sumérios consideravam a impressão de uma mão como um bom identificador e que os egí-

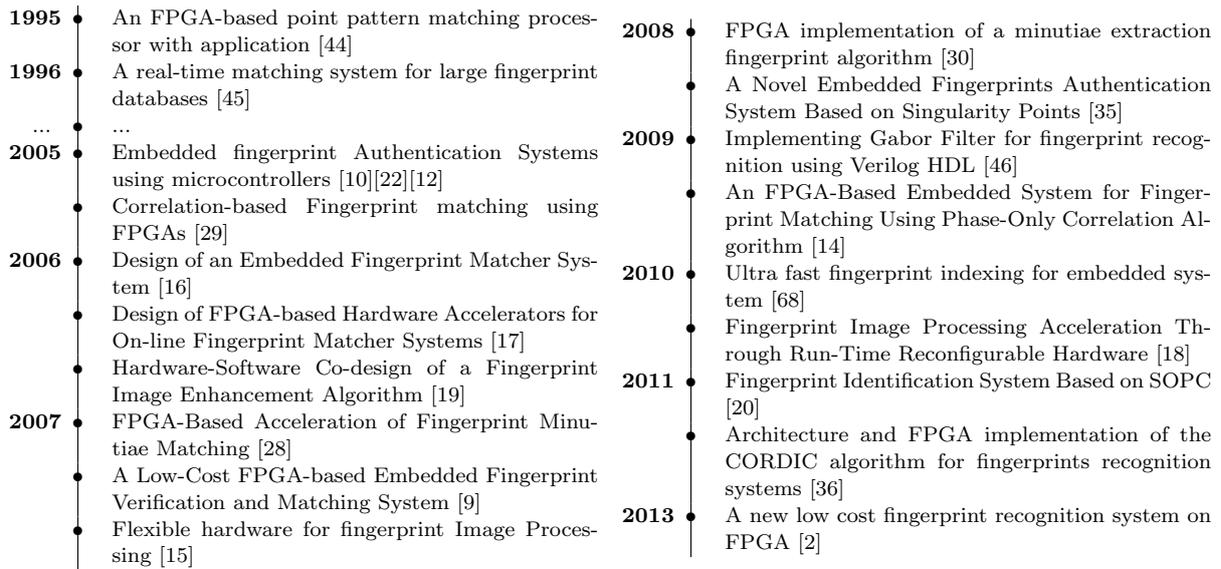


Figura 1.1: Linha temporal de publicações sobre reconhecimento de impressões digitais, em sistemas embebidos no período 1995-2013, inclusivamente.

cios utilizavam características como a altura, estrutura física, várias características da face, cicatrizes ou até a distância entre o polegar e o cotovelo, como forma de confirmar a identidade das pessoas [6]. A biometria contemporânea tem o seu grande avanço no século XIX [6] com várias vertentes de pensamento a surgirem, essencialmente com a motivação em relacionar características físicas com atividades criminais. Por exemplo, no início dos anos 1800, Franz Joseph Gall, um médico e anatomista alemão, foi um dos pioneiros ao desenvolver a frenologia. Segundo esta teoria, seria possível determinar as características da personalidade e o grau de criminalidade de uma pessoa através da forma do crânio. Esta teoria gerou bastante controvérsia e acabou por ser abandonada, ficando apenas a sua contribuição para o estudo do cérebro e a abertura do caminho para investigadores como Cesare Lombroso, Adolphe Quetelet e Alphonse Bertillon [6]. Em 1882, o criminologista francês Alphonse Bertillon propôs um sistema antropométrico, conhecido por Bertillonage, com o objetivo de identificar criminosos condenados. Este sistema tem como base a medição de várias partes do corpo e características individuais como cicatrizes e tatuagens. O sistema revelou-se falível ao verificar-se que duas pessoas poderiam apresentar as mesmas características.

A falha do Bertillonage motivou o estudo das impressões digitais por parte de investigadores como Edward Henry, Francis e Juan Vucetich. Embora Juan Vucetich tenha sido o pioneiro na utilização de impressões digitais a nível criminal, foi o sistema Henry [24] introduzido em 1900 que se tornou uma referência no mundo da criminologia. Este sistema consiste na divisão das impressões digitais por diferentes categorias consoante as suas características, permitindo assim

otimizar o processo de identificação que até então não possuía qualquer critério de indexação. Desde o estabelecimento da asserção de que não existem duas pessoas com impressões digitais iguais, conseguido no final do século XIX [34], o reconhecimento biométrico por impressões digitais tem sido alvo de muitos estudos. No século XIX e no início do século XX foram definidos os principais conceitos de reconhecimento de indivíduos, mantendo-se até aos dias de hoje. O final da Segunda Guerra Mundial, e conseqüente desenvolvimento eletrónico, proporcionou um novo avanço nos sistemas de reconhecimento biométrico com o estudo de sistemas automatizados. Estes sistemas visam acelerar o processo de pesquisa e comparação de impressões digitais, bem como realizar o reconhecimento de impressões digitais em tempo real.

Em 1963 surge uma das primeiras publicações sobre sistemas automatizados de identificação de impressões digitais (*Automatic Fingerprint Identification Systems - AFIS*) [51]. O *National Institute of Standards and Technology* (NIST) iniciou o estudo de sistemas automatizados em 1970 [56], enquanto que o *Federal Bureau of Investigation* (FBI) instalou o seu primeiro protótipo em 1972 [27].

A massificação dos AFIS que ocorreu nos anos seguintes deve-se em grande parte ao desenvolvimento de: sensores *live-scan* (secção 2.6.2), algoritmos de análise e extração de características de imagens de impressões digitais (secção 3.1) e de técnicas de comparação de impressões digitais (secção 3.1). Isto permitiu que o reconhecimento de impressões digitais passasse a ter também uma vertente de autenticação, aumentando assim o número de aplicações para estes sistemas. Estas aplicações podem ser divididas em diferentes categorias [34], como por exemplo, controlo de acesso físico, controlo de acesso lógico, controlo de acesso a dispositivos, autenticação de transações, controlo temporal ou de assiduidade, identificação civil e identificação forense. A evolução dos sensores e dos algoritmos de reconhecimento de impressões digitais levou a que em 2000 ocorresse a primeira competição internacional de reconhecimento de impressões digitais, a *Fingerprint Verification Competition* (FVC) [32], tendo-se repetido em 2002 [33], 2004 [31], 2006 [53] e mantendo-se atualmente com uma competição ininterrupta, a FVC-onGoing [52]. A competição de 2000 tinha como objetivo principal encontrar um meio de comparação entre algoritmos diferentes, definindo para tal um conjunto de bases de dados elaboradas a partir de sensores diferentes. Estas bases de dados, de acesso público, foram as primeiras criadas a partir de sensores do tipo *live-scan*. As bases de dados da NIST, utilizadas até então, eram essencialmente constituídas por impressões digitais do tipo *rolled inked*, não sendo as ideais para analisar o desempenho de AFIS.

A implementação de AFIS, e de sistemas biométricos em geral, pode ser realizada por *software* em sistemas genéricos ou por sistemas dedicados. No final do século XX, principalmente nas décadas de 80 e 90, o estudo e a implementação de sistemas biométricos incidia sobre computadores pessoais ou computadores com arquiteturas genéricas. No entanto, ao surgirem sistemas embebidos mais económicos e com maior capacidade de processamento, passou a existir a possibilidade de desenvolver sistemas dedicados capazes de rivalizar com os sistemas genéricos em termos de capacidade de processamento, custo e consumo energético. No contexto dos sistemas embebidos, as FPGAs enquanto sistemas de *hardware* reconfigurável permitem implementar circuitos dedicados por forma a realizar tarefas computacionais de forma mais eficiente, em comparação com soluções baseadas puramente em *software*. Desde o seu aparecimento na década de 80 do século XX que as FPGA têm sido utilizadas em diversas áreas, tais como: aeroespacial e defesa, áudio, eletrónica de consumo e telecomunicações [62]. Em sistemas biométricos, as FPGAs são geralmente utilizadas num de dois cenários:

- para acelerar a realização de uma determinada parte de um algoritmo com recurso a um co-processador, sendo que o resto do algoritmo é resolvido e controlado por um processador;
- na implementação de um determinado processo de forma dedicada, como por exemplo a comparação de características, em que este apenas necessita de uma interface para recolher os dados a processar.

Na figura 1.2 apresentam-se os dois tipos de utilização de FPGA descritos acima. Em 1995 [44]

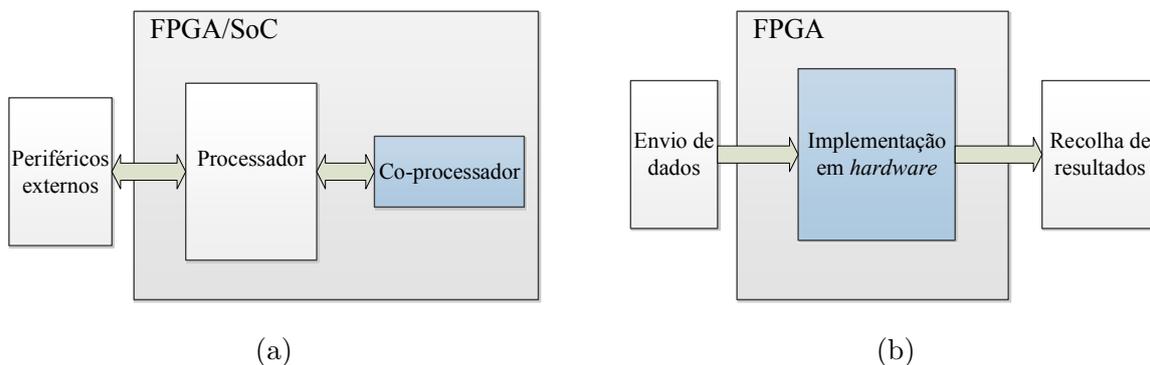


Figura 1.2: Tipos de implementação de sistemas de reconhecimento de impressões digitais com recurso a FPGA: (a) sistema controlado por um processador e com aceleração por *hardware* (co-processador), (b) implementação de um processo do algoritmo unicamente em *hardware*.

e em 1996 [45] Nalini K. Ratha propõe duas implementações de sistemas de reconhecimento

de impressões digitais em FPGA. A primeira descreve uma implementação em paralelo de um sistema de comparação de impressões digitais baseado em minúcias [34], enquanto que a segunda, recorrendo ao mesmo algoritmo, foca-se na otimização para bases de dados de grandes dimensões. Estas implementações são duas das primeiras de que há registo com recurso a FPGA, sendo necessário chegar a 2005 para encontrar novas propostas de implementação.

Dos três primeiros sistemas de 2005, presentes na figura 1.1, em [12][22] são propostos sistemas vocacionados para sistemas integrados e com algoritmos baseados em minúcias. Por sua vez, o sistema apresentado por C. Chen e J. Dai [10] recorre a um microcontrolador com arquitetura de 8 bits a operar a 25 MHz para realizar a implementação de um algoritmo que utiliza a orientação da imagem e cria um perfil circular da mesma para efetuar o reconhecimento. Com esta solução procuraram encontrar uma alternativa aos sistemas por minúcias, que se revelam mais exigentes ao nível do processamento dos dados. Nesse mesmo ano é proposta uma nova arquitetura em FPGA [29], desta feita com base na correlação recorrendo à *Fast Fourier Transform* (FFT) [13]. Os resultados experimentais reportados mostram que é possível implementar um sistema em tempo real segundo este método. No entanto, o paralelismo oferecido é limitado pela quantidade de recursos necessários. Em 2006, surgem três arquiteturas propostas por investigadores de universidades espanholas [16][17][19], todas elas baseadas na utilização de um processador para controlar o sistema e, utilizando a parte reconfigurável do mesmo, para realizar aceleração por *hardware*. As duas primeiras apresentam um sistema de reconhecimento baseado em minúcias, enquanto que na última é implementado um algoritmo de melhoramento de impressões digitais, realizando-se a implementação dos filtros de Gabor [21] em *hardware*. Em 2007 são apresentados novos artigos científicos com origem em universidades espanholas. No primeiro [28], são estudadas duas implementações em FPGA de algoritmos de comparação de minúcias. No segundo [9], é realizada a implementação de um sistema de reconhecimento de impressões digitais baseado no algoritmo BOZORTH3 [38], usado no *NIST Fingerprint Image Software* (NFIS) da NIST, com aceleração por *hardware* em FPGA. Em [15], é proposta uma arquitetura flexível em que o componente que realiza a aceleração por *hardware* pode resolver um de três algoritmos de processamento de imagem [34]: segmentação e normalização, supressão de ruído e cálculo da orientação de campo. Em 2008 é proposta uma arquitetura para realizar a extração de minúcias recorrendo a três co-processadores [30], e outra que apresenta um sistema de autenticação baseado nas singularidades *delta* e *core* [27][34], mostrando bons resultados através da utilização das bases de dados FVC [35]. Em 2009 foi proposta uma outra implemen-

tação dos filtros de Gabor em *hardware* [46] e foi apresentado um sistema de reconhecimento de impressões digitais baseado em correlação de fase [14].

Em 2010 é introduzida uma nova arquitetura que acelera o processo de indexação de impressões digitais [68], sendo também proposta uma arquitetura reconfigurável, em tempo real, de um sistema de reconhecimento de impressões digitais [18]. Em 2011 surge a proposta de um sistema de identificação baseado em minúcias com aceleração por *hardware* [20], sendo também proposta uma versão do algoritmo de *COordinate Rotation DIgital Computer* (CORDIC) [54] otimizada para sistemas de reconhecimento de impressões digitais [36]. Por fim, em 2013 é apresentada uma arquitetura de reconhecimento de baixo custo, para diminuir a complexidade de um sistema baseado em minúcias. A divisão da imagem (após binarização) em blocos, através de uma grelha de onde são retiradas as características que permitem identificar cada impressão digital é proposta em [2].

1.3 Principais contributos do projeto

As implementações em *hardware* da filtragem de Gabor descritas em [19] e [46], utilizam janelas de 13×13 e 3×3 , sendo baseadas em janela deslizante. Em [4] e em [5] é proposta a separação das janelas de Gabor num filtro passa-baixo e num filtro passa-banda. Nas duas soluções é realizado um ajuste nos parâmetros dos filtros, sendo que em [5] essa alteração resulta numa degradação no desempenho do sistema. Em [4], para minimizar essa mesma degradação, é realizado o processo de reamostragem. O processo de reamostragem implica processamento extra, não sendo este adequado para uma implementação em *hardware*.

Neste trabalho são realizadas as seguintes contribuições:

- análise (*profiling*) de uma implementação de referência, exclusivamente em software, de algoritmo de reconhecimento (autenticação e identificação) de impressões digitais;
- otimização e ajuste dos parâmetros da implementação de referência para as imagens adquiridas pelo sensor ótico de contacto, GT-511C3 [1] da ADH Technology;
- projeto e realização de componentes do algoritmo de reconhecimento por *hardware*;
- proposta (projeto e realização) de decomposição de filtros de Gabor a duas dimensões, em dois filtros a uma dimensão, através da decomposição em valores singulares (*Singular Value Decomposition* - SVD)[7][49] sobre a máscara do filtro original.

1.4 Organização do documento

O restante relatório encontra-se organizado da forma que se descreve em seguida.

No capítulo 2 são apresentados os conceitos relacionados com biometria, desde os indicadores biométricos mais comuns, até à descrição dos sistemas biométricos e métodos de avaliação associados aos mesmos. No final do capítulo, a impressão digital é analisada com detalhe com mais detalhe, descrevendo-se a sua constituição e os métodos de aquisição da mesma.

No capítulo 3 são apresentadas as principais técnicas e ferramentas utilizadas no reconhecimento de impressões digitais. A partir da secção 3.2, trata-se o algoritmo de extração de características utilizado neste trabalho.

No capítulo 4 é descrito o estudo do algoritmo de extração de características apresentado no capítulo anterior, considerando a sua implementação em *software*. Este estudo, efetuado recorrendo às implementações realizadas em MATrix LABoratory (MATLAB) e no processador *MicroBlaze* usando a linguagem C, tem como objetivo encontrar a parte do algoritmo candidata à aceleração em *hardware*.

No capítulo 5 é apresentada a arquitetura desenvolvida para realizar a extração de características no processador *MicroBlaze*, recorrendo-se a um co-processador especializado para acelerar parte do processamento em *hardware*.

No capítulo 6 são apresentadas as conclusões do trabalho realizado e são efetuadas as propostas de trabalho futuro.

No anexo A é apresentada a constituição da base de dados de impressões digitais, recolhida e utilizada neste trabalho.

No anexo B é descrito o processo de decomposição SVD dos filtros de Gabor, apresentando as respostas impulsionais resultantes.

No anexo C, analisa-se a sensibilidade do algoritmo de reconhecimento de impressões digitais, à variação dos seus parâmetros.

Capítulo 2

Biometria

A palavra biometria deriva do grego *bios metron* e significa, literalmente, medida de vida. No contexto atual, pode definir-se a biometria como sendo a medida de uma característica fisiológica ou comportamental que pode ser quantificada e armazenada com um elevado grau de confiança, tal que permita autenticar ou identificar univocamente um indivíduo [6] [43]. Tendo em conta esta definição, e escolhendo um indicador biométrico, é então possível projetar um sistema de reconhecimento biométrico para um determinado tipo de aplicação.

Este capítulo, inicia-se com uma breve introdução e comparação de alguns dos principais indicadores biométricos. Na secção 2.2 são apresentadas as diferentes características de um sistema biométrico, assim como as várias categorias em que estes podem ser aplicados. Segue-se na secção 2.3 uma descrição dos tipos de reconhecimento biométrico existentes e a exposição de um sistema biométrico genérico. Na secção 2.4 é realizada a caracterização dos tipos de erros existentes, destacando a influência destes na avaliação de um sistema biométrico na secção 2.5. Este capítulo termina com o estudo das impressões digitais na secção 2.6, realizando uma breve descrição da sua constituição. No final desta secção, são apresentados os principais métodos de aquisição de impressões digitais, descrevendo as características dos sensores mais utilizados.

2.1 Indicadores biométricos

Existem diversos indicadores biométricos do ser humano tais como: íris, face, geometria da mão, voz, assinatura ou impressões digitais. Na figura 2.1 apresentam-se exemplos de representações digitais destes indicadores, denominadas por amostras. Uma vez que este trabalho se foca essencialmente em impressões digitais, este tipo de indicador será abordado com maior

detalhe na secção 2.6, sendo que para os restantes é realizada uma breve introdução [27][34].

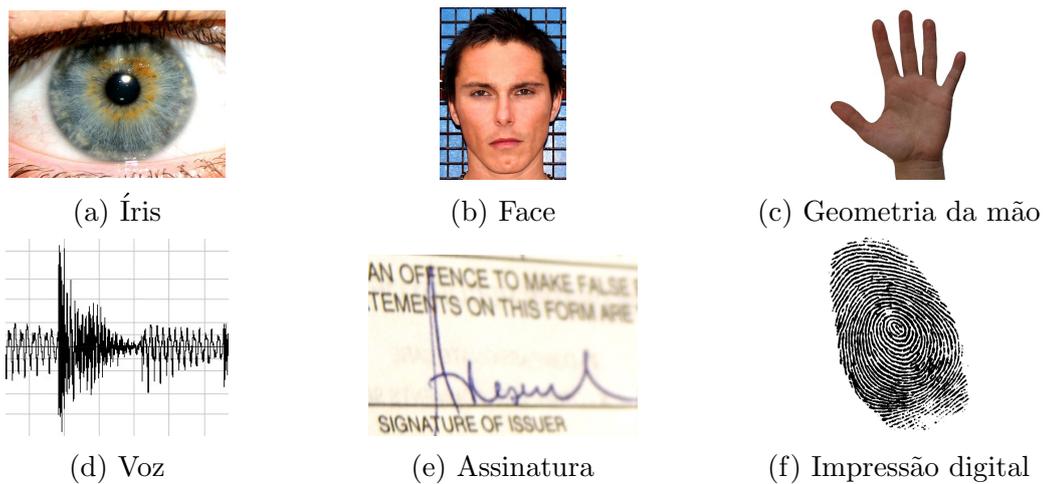


Figura 2.1: Indicadores biométricos: íris, face, geometria da mão, voz, assinatura e impressão digital.

- **Íris** - embora seja uma parte visível do corpo, a íris é um órgão interno do olho que se encontra localizado entre a córnea e o cristalino. Pode ser descrita como sendo a região anular do olho, delimitada pela pupila e pela esclera (ou branco do olho). Acredita-se que desde a sua formação, durante o desenvolvimento do feto, que a íris se mantém desde os oito meses de gestação [43]. Devido à elevada complexidade da textura da íris acredita-se também que não existem duas pessoas com uma textura de íris igual, justificando-se assim o grande interesse da comunidade científica neste indicador biométrico [43]. Apesar das potencialidades de utilizar um sistema biométrico baseado em íris, estes sistemas em comparação com os restantes, são menos utilizados por necessitarem em demasia da colaboração do utilizador e por possuírem um custo elevado [27].
- **Face** - o reconhecimento facial é um método não intrusivo, sendo o mais utilizado para efetuar o reconhecimento de humanos. Os sistemas biométricos que utilizam este tipo de reconhecimento recorrem à captura de imagens para efetuar a extração das características. Independentemente do tipo de características que o sistema utiliza, este terá dificuldade em realizar a comparação entre faces quando as imagens são capturadas com perspetivas diferentes, sobre outras condições de iluminação ou quando existiram alterações físicas [27]. Estas limitações, juntamente com as dúvidas quanto à unicidade deste indicador biométrico, tornam-no menos fiável em comparação com os restantes.
- **Geometria da mão** - os sistemas biométricos baseados na geometria das mãos recorrem

essencialmente ao comprimento e largura dos dedos, ao tamanho da palma e à forma da mão. Desta forma, o desempenho destes sistemas não é influenciado por tempo seco ou por pele seca, sendo também pouco exigente a nível computacional e dando origem a sistemas económicos. No entanto, existem dúvidas se a extração destas características é suficiente para considerar a geometria das mãos como identificador único [27]. Consequentemente, estes sistemas não são recomendados para serem utilizados em grandes populações ou numa aplicação em que a segurança seja uma prioridade. É também necessário considerar que a geometria das mãos varia ao longo do crescimento.

- **Voz** - a voz consiste no som produzido pelas cordas vocais aquando a passagem de um fluxo de ar e é definida em função das características físicas (forma e tamanho) dos tratos vocais, boca, cavidades nasais e lábios. Após terminarem o seu desenvolvimento, estas características físicas são invariantes [27]. No entanto, o padrão da voz pode ser influenciado ao longo do tempo pela idade, condição médica ou pelo estado emocional. Tal como o reconhecimento pela geometria da mão, os sistemas têm um custo baixo mas como a voz não é muito distintiva, não é a mais aconselhada para efetuar reconhecimento de uma grande população.
- **Assinatura** - o reconhecimento através da maneira como um indivíduo escreve o seu nome é um dos métodos de reconhecimento mais antigos em utilização, sendo amplamente aceite por instituições governamentais e possuindo aceitação legal. Este indicador destaca-se essencialmente por ser o mais aceite socialmente e por, ao contrário dos restantes, poder ser alterado ao longo do tempo, tal como uma *password*. No entanto, estas alterações podem ser involuntárias. Por exemplo, podem ser influenciadas pelo estado físico e emocional de um indivíduo, provocando assim mais erros comparativamente com outros indicadores. Outro aspeto a ter em conta prende-se com a variação de uma assinatura introduzida digitalmente de uma obtida por digitalização a partir de uma folha de papel.

Os indicadores biométricos podem ser caracterizados e comparados segundo um conjunto de parâmetros [26]:

- **Unicidade** - o grau de variação do indicador dentro da população de interesse.
- **Universalidade** - indica o nível de presença do indicador dentro da população de interesse.

- **Invariância** - refere a variação do indicador ao longo do tempo.
- **Usabilidade** - indica a facilidade de utilização do indicador.
- **Desempenho** - define o tempo que demora a realizar o processo de reconhecimento.
- **Aceitação** - indica o nível de aceitação social do indicador.
- **Fiabilidade** - define o nível de segurança e fiabilidade.

Tendo em conta estes parâmetros é possível visualizar na tabela 2.1 (adaptado de [34]) uma comparação entre os indicadores apresentados, em que E, M e B significam, respetivamente, Elevado, Médio e Baixo.

Indicadores biométricos	Unicidade	Universalidade	Invariância	Usabilidade	Desempenho	Aceitação	Fiabilidade
Íris	E	E	E	M	E	B	B
Face	B	E	M	E	B	E	E
Geometria das mãos	M	M	M	E	M	M	M
Voz	B	M	B	M	B	E	E
Assinatura	B	B	B	E	B	E	E
Impressão digital	E	M	E	M	E	M	M

Tabela 2.1: Comparação entre indicadores biométricos. E, M e B significam, respetivamente, Elevado, Médio e Baixo (adaptado de [34]).

2.2 Aplicações

Com o avanço dos sistemas biométricos, estes têm sido implementados nas mais diversas áreas. Como consequência, tem sido necessário adaptar as características destes sistemas ao tipo de aplicação alvo.

2.2.1 Caracterização das aplicações

Os sistemas biométricos podem ser classificados segundo sete características [55]:

- **Cooperativo ou não cooperativo** - refere-se à cooperação dos utilizadores ao utilizarem o sistema biométrico. Por exemplo, num acesso legítimo, o utilizador será cooperativo uma vez que é do interesse do mesmo que o acesso seja validado. Numa situação em que o utilizador não queira ser reconhecido este não irá colaborar, podendo a sua postura provocar uma falsa identificação.
- **Frequente ou não frequente** - caracteriza a frequência de utilização do sistema por parte dos utilizadores. Por exemplo, um utilizador habitua-se mais facilmente a um sistema que usa diariamente (acesso a uma sala) do que a um que utilize esporadicamente (renovação de um documento).
- **Acompanhado ou não acompanhado** - indica se a utilização do sistema é acompanhada, observada ou guiada por alguém. Por exemplo, no acesso a uma infraestrutura pode não existir outra pessoa a efetuar o controlo do acesso. O mesmo não acontece na recolha de impressões digitais para um documento de identificação.
- **Ambiente controlado ou não controlado** - refere se o sistema biométrico necessita de um ambiente controlado (humidade, iluminação ou ruído) para funcionar corretamente.
- **Público ou privado** - indica se o sistema é utilizado num meio privado como por exemplo numa empresa, em que a informação fica confinada à mesma, ou se é utilizado num meio público, como é o caso das instalações usadas para obtenção e renovação de documentos de identificação.
- **Aberto ou fechado** - se o sistema implementado é compatível com aplicações desenvolvidas por outras pessoas/empresas ou se é um sistema proprietário.
- **Consciente ou inconsciente** - indica se o utilizador tem consciência, ou não, de que está a ser alvo de reconhecimento biométrico. Por exemplo, pode ser realizado reconhecimento biométrico através de câmaras de videovigilância sem que o utilizador se aperceba.

2.2.2 Tipo de aplicações

As aplicações dos sistemas de reconhecimento biométrico podem ser divididas em diferentes categorias [34]:

- **Controlo de acesso físico** - utilizado quando é necessário restringir o acesso a instalações físicas (edifícios ou salas), como por exemplo em edifícios militares, entidades bancárias, empresas ou instituições governamentais.
- **Controlo de acesso lógico** - usado para restringir o acesso a informação num dispositivo eletrónico, tal como por exemplo, início de sessão num sistema operativo instalado num computador ou limitação de acesso a um servidor remoto.
- **Controlo de acesso a dispositivos** - permite limitar por completo o acesso a dispositivos eletrónicos, sendo necessário autenticar o acesso antes de utilizar o mesmo. Este tipo de acesso pode ser considerado lógico e/ou físico.
- **Autenticação de transações** - utilizado em transações comerciais, por forma a confirmar a identidade do indivíduo envolvido.
- **Controlo temporal ou de assiduidade** - geralmente adotado por empresas para controlar a assiduidade dos seus empregados.
- **Identificação civil** - utilizado em documentos como a carta de condução, passaporte ou um documento de identificação nacional para confirmar a identidade de um indivíduo e é utilizado, por exemplo, no controlo de fronteiras.
- **Identificação forense** - usado como auxílio na investigação criminal, sendo principalmente utilizado para identificação.

Cada uma destas categorias pode ser aplicada em áreas diferentes, sendo as principais: forense, governamental e comercial. Na tabela 2.2 resumem-se algumas das principais aplicações nestas áreas.

Forense	Governamental	Comercial
Identificação de corpos.	Controlo de fronteiras.	Acesso a infraestruturas.
Utilização como prova em processos judiciais.	Passaporte.	Acesso a informação.
Desaparecimento de pessoas.	Carta de condução.	Acesso a dispositivos móveis.
	Documento de identificação nacional.	Transações comerciais.
		Acessos bancários.

Tabela 2.2: Principais aplicações de sistemas biométricos nas áreas forense, governamental e comercial.

2.3 Tipos de reconhecimento biométrico

Pode-se considerar a existência de dois tipos de reconhecimento biométrico: um-para-um (autenticação) ou um-para-muitos (identificação). No primeiro caso, está-se a comparar uma amostra recolhida, em tempo real ou não, com outra previamente armazenada por forma a validar ou não a autenticação de um indivíduo. No segundo caso, efetua-se a comparação de uma amostra recolhida com várias amostras armazenadas, sem se saber de antemão quem se pretende identificar. Em ambos os casos efetua-se a correspondência entre as características das várias amostras. Uma vez que não é possível existir uma correspondência perfeita, mesmo quando as amostras têm a mesma origem (ver secção 2.4), o resultado da comparação é determinado através de um nível de decisão o qual valida ou não, a autenticação ou identificação. Isto significa que não se verifica igualdade entre amostras (do mesmo dedo com o mesmo sensor), mas sim um nível de semelhança entre amostras. Estes dois tipos de reconhecimento juntamente com o armazenamento de características biométricas constituem as principais funcionalidades de um sistema biométrico. Assim, pode-se definir um sistema biométrico como uma entidade que utiliza características fisiológicas recolhidas dos seus utilizadores (amostras) como forma de os reconhecer.

2.3.1 Sistema biométrico

Na figura 2.2 apresenta-se um sistema biométrico genérico, sendo este constituído pelos seguintes componentes:

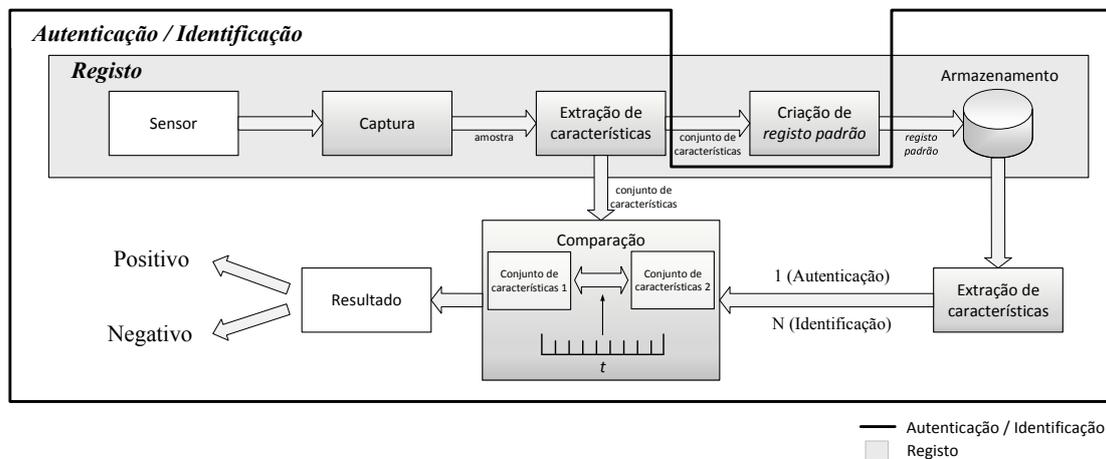


Figura 2.2: Diagrama de blocos de um sistema biométrico genérico e respetivos componentes individuais.

- **Captura** - recorrendo a um sensor biométrico, tem como objetivo criar uma representação digital do indicador biométrico. A representação digital, ou seja, a amostra, é criada a partir do tratamento da informação recolhida do sensor.
- **Extração de características** - através de uma amostra biométrica cria um conjunto de características que, segundo o tipo de sistema implementado, constitua uma representação compacta da amostra recolhida.
- **Criação de registo padrão** - organiza as características biométricas associadas a uma amostra num tipo de dados armazenável (registo padrão - *template*).
- **Armazenamento** - organiza os vários registos padrão numa estrutura de dados (base de dados). Este componente pode ser interno ou externo ao sistema.
- **Comparação** - compara conjuntos de características biométricas retornando, segundo um determinado nível de decisão, o resultado dessa comparação. No modo de autenticação é sempre retornado apenas um resultado, já no modo de identificação podem existir vários. Neste componente pode ainda existir uma funcionalidade de pré-seleção de registos padrão, de modo a reduzir o número de comparações efetuadas, sendo principalmente útil quando os sistemas se encontram a funcionar no modo de identificação e com bases de dados de grandes dimensões.

Com os diferentes componentes do sistema caracterizados, pode-se então descrever os três modos de funcionamento do sistema: registo, autenticação e identificação.

- **Registo** - adiciona à base de dados um novo registo padrão. Para tal, e como se pode verificar na figura 2.2, necessita de criar uma nova amostra, extrair as várias características e criar um registo para que este possa ser armazenado.
- **Autenticação** - no modo de autenticação é realizado reconhecimento biométrico de um-para-um. Isto significa que é necessário recolher uma nova amostra e extrair a partir desta um conjunto de características. É também preciso criar um segundo conjunto de características que tem origem no registo padrão associado ao indivíduo que se pretende autenticar. Segue-se a comparação dos dois conjuntos, dando origem a um resultado que pode ser positivo ou negativo.

- **Identificação** - o modo de identificação é semelhante ao de autenticação, mas como se trata de um tipo de reconhecimento de um-para-muitos, no componente de comparação são realizadas N comparações, de acordo com a dimensão da base de dados ou da existência ou não de pré-seleção de registos. Ao serem efetuadas várias comparações podem então ser gerados vários resultados que respeitem o nível de decisão escolhido, ou seja, pode existir mais do que um a cumprir o limiar de decisão. Nesta situação, dependendo da implementação realizada, pode ser escolhido o candidato que possui maior probabilidade de ser o correto (aquele que respeita o limiar de decisão com margem maior) ou em alternativa, apresenta-se a lista ordenada de candidatos que cumprem o limiar de decisão. Caso nenhum dos registos existentes na base de dados cumpra o limiar de decisão, então reporta-se que não foi possível identificar qualquer indivíduo a partir da amostra adquirida.

Independentemente do funcionamento em modo de autenticação ou identificação, o sistema pode operar em modo *online* ou *offline*. Em modo *online*, todos os componentes (com exceção para a criação prévia de registos) estão a funcionar em tempo real. Isto obriga a que os componentes realizem os seus processos o mais rápido possível, de modo a não comprometer o desempenho. Em modo *offline*, alguns ou até mesmo todos os componentes podem ter os seus processos a serem realizados manualmente, por exemplo realizando inspeção visual de uma amostra, não havendo nesta situação grande preocupação com o tempo de processamento. Os sistemas automatizados, como é o caso do sistema estudado neste documento, são tipicamente considerados *online*. Contudo, o modo de registo pode ser realizado em *offline*, atendendo a que pode não existir a necessidade de atualizar a base de dados em tempo real para que os modos de autenticação e identificação funcionem.

2.4 Erros do sistema biométrico

Em qualquer sistema biométrico automatizado, independentemente de este se encontrar a operar em modo *online* ou *offline*, podem surgir erros que irão influenciar os resultados obtidos pelo mesmo. A origem dos erros pode ser dividida em três classes:

- **Limitação de informação** - o tipo de indicador biométrico utilizado pode influenciar na quantidade e qualidade de informação distinta que se pode retirar de um indivíduo. Por exemplo, a quantidade de informação distinta que se consegue retirar da geometria de

uma mão será inferior à obtida de uma impressão digital, dando origem a um menor número de amostras distintas entre indivíduos. Também podem surgir erros quando não existe consistência na informação recolhida, como por exemplo na validação de uma assinatura quando existe uma variação em relação à assinatura tomada como referência. Por exemplo, tal acontece num processo de autenticação de impressões digitais quando é introduzido um dedo diferente do que deu origem ao registo padrão utilizado.

- **Limitação na representação** - embora alguns indicadores biométricos sejam reconhecidos por possuírem uma grande distinguibilidade, como a íris e a impressão digital, em termos práticos os sistemas biométricos impõem limitações à quantidade de informação que se consegue obter de um indivíduo. Consequentemente, esta limitação pode introduzir erros no sistema.
- **Limitação na invariância** - um sistema biométrico pode não conseguir lidar com todas as variações possíveis de uma representação biométrica, ou seja, este pode não conseguir efetuar um reconhecimento correto quando as condições de recolha do identificador biométrico mudam em relação a um ou vários registos padrão. Por exemplo, a rotação do dedo na aquisição da impressão digital pode dar origem a um erro no sistema quando é realizada a comparação com um registo padrão.

As duas primeiras limitações podem ser minimizadas na fase de projeto do sistema biométrico. A limitação na invariância também pode ser minimizada na fase de projeto, no entanto não é possível prever todas as variações.

2.4.1 Tipos de erro

Para avaliar o impacto dos erros no desempenho dos sistemas existe um conjunto de métricas que é tipicamente utilizado [34]:

- **Taxa de Falsa Correspondência (*False Match Rate* - FMR)** - probabilidade de ocorrer uma falsa validação durante o processo de reconhecimento.
- **Taxa de Falsa Não-Correspondência (*False Non-Match Rate* - FNMR)** - probabilidade de um acesso legítimo não ser validado durante o processo de reconhecimento.

- **Taxa de Erro de Identificações Negativas (*False Negative Identification-Error Rate - FNIR*)** - quantidade de identificações falhadas de um utilizador que se encontra registado no sistema.
- **Taxa de Erro de Identificações Positivas (*False Positive Identification-Error Rate - FPIR*)** - número de identificações positivas de um utilizador que não se encontra registado no sistema.
- **Insucesso na Detecção (*Failure To Detect - FTD*)** - quantifica os erros que ocorrem aquando o sensor não deteta a presença de um indicador biométrico.
- **Insucesso na Captura (*Failure To Capture - FTC*)** - ocorre quando o sistema deteta a presença de um indicador biométrico mas não consegue capturar a amostra.
- **Insucesso no Processamento (*Failure To Process - FTP*)** - indica quando o sistema falha no processo de extração de características de um indicador biométrico.
- **Insucesso na Aquisição (*Failure To Acquire - FTA*)** - métrica que resulta da conjugação dos tipos de erro FTD, FTC e FTP, uma vez que estes tipos de erro são comuns em todos os processos (registo, autenticação e identificação).
- **Insucesso no Registo (*Failure To Enroll - FTE*)** - quantifica o número de erros que ocorrem na criação de um registo padrão.

Este documento foca-se essencialmente nos indicadores FMR e FNMR. Note-se que em alguma literatura estas duas métricas, FMR e FNMR, são denominadas por *False Acceptance Rate* (FAR) e *False Rejection Rate* (FRR) [27][34], respetivamente.

2.4.2 Comparação

Admitindo um cenário em que é utilizada a impressão digital como indicador biométrico, quando é realizada a comparação de duas impressões digitais o módulo responsável por esta operação tem como retorno um número que quantifica a semelhança entre as mesmas. Assumindo que o resultado se encontra normalizado entre 0 e 1, quanto mais próximo este estiver do valor 1, mais confiança terá o sistema para indicar que as impressões digitais têm origem no mesmo dedo. A escolha é baseada num nível de decisão, em que as impressões digitais com

valor abaixo desse nível de decisão são descartadas, enquanto que as que possuem um valor igual ou superior são consideradas no processo de reconhecimento.

Assumindo um sistema a funcionar em modo um-para-um, ou seja, no modo de autenticação, podem surgir dois tipos de erros:

- Indicar erradamente que duas impressões digitais pertencem ao mesmo dedo (falsa correspondência ou *false match*).
- Indicar que as duas impressões digitais não pertencem ao mesmo dedo, quando na realidade têm origem no mesmo dedo (falsa não-correspondência ou *false non-match*).

2.4.3 Autenticação

Considerando que o registo padrão de uma pessoa é denominado por T e que um conjunto de características é designado por C , então tem-se que [34]:

$H_0 : C \neq T$, o conjunto de características e o registo padrão não pertencem ao mesmo indicador biométrico.

$H_1 : C = T$, o conjunto de características e o registo padrão pertencem ao mesmo indicador biométrico.

Para se chegar a H_0 ou a H_1 é necessário realizar a comparação entre T e C , por forma a obter o nível de semelhança, $s(T, C)$ e decidir, segundo um determinado nível de decisão se existe *matching* (D_1) ou *non-matching* (D_0). Realizando uma analogia com os sistemas de comunicação digital, pode-se considerar que H_0 é a hipótese de o sinal recebido não ser a mensagem e H_1 é a hipótese de o sinal recebido ser a mensagem. Admitindo estas duas hipóteses pode-se obter dois tipos de erros de autenticação mencionados:

Tipo I: *false match*, é decidido D_1 quando ocorre H_0 ;

Tipo II: *false non-match*, é decidido D_0 quando ocorre H_1 .

Pode-se então definir as métricas FMR e o FNMR descritas anteriormente como sendo, respetivamente, a probabilidade de ocorrer um erro do tipo I como

$$\text{FMR} = P(D_1|H_0), \quad (2.1)$$

e a probabilidade de ocorrer um erro do tipo II como

$$\text{FNMR} = P(D_0|H_1). \quad (2.2)$$

O FMR e o FNMR são duas das principais métricas utilizadas na avaliação de sistemas biométricos e que serão exploradas na secção 2.5.

2.4.4 Identificação

Tal como para os erros de autenticação, pode obter-se dois tipos distintos de erros no processo de identificação: o FNIR e o FPIR apresentados anteriormente. Colocando algumas condições, nomeadamente, que não existe qualquer pré-seleção dos registos presentes na base de dados utilizada e que cada registo é de um indivíduo diferente, estes erros podem ser calculados a partir do FNMR e do FMR:

- FNIR_N - a probabilidade de se obter uma identificação falhada de um utilizador que se encontra registado no sistema, quando se efetua a pesquisa em N registos padrão. É igual ao FNMR obtido no modo de autenticação ($\text{FNIR}_N = \text{FNMR}$).
- FPIR_N - a probabilidade de se obter uma identificação positiva de um utilizador que não se encontra registado no sistema, quando se efetua a pesquisa em N registos padrão. É igual ao complemento da probabilidade dada por $(1-\text{FMR})^N$ obtida no modo de verificação ($\text{FPIR}_N = 1-(1-\text{FMR})^N$).

2.5 Avaliação de sistemas biométricos

A avaliação de um sistema pode ser dividida em três categorias distintas [41]: tecnológica, operacional, ou num determinado cenário de testes, da forma que se resume de seguida.

- **Avaliação tecnológica** - o objetivo principal deste tipo de avaliação prende-se, fundamentalmente, com a comparação do desempenho de algoritmos. Para tal, e de modo a que a comparação seja fiável, é necessário isolar os efeitos do processo de captura de amostras do sensor, passando a utilizar uma base de dados comum constituída por amostras do tipo do indicador biométrico utilizado.

- **Avaliação operacional** - é utilizada para determinar o desempenho de um sistema biométrico numa aplicação específica e para uma determinada população.
- **Avaliação num cenário de testes** - neste tipo de avaliação é definido um cenário de testes, por norma num cenário real e vocacionado para a aplicação pretendida, para avaliar o desempenho geral do sistema.

Em qualquer uma destas categorias deve-se recorrer às métricas referidas na secção 2.4 de modo a quantificar essa mesma avaliação. Primeiro, é necessário recolher um elevado número de resultados obtidos, através da comparação entre conjuntos de características (C) e registos padrão (T) do mesmo indivíduo, dando origem à distribuição $p(s|H_1)$, também conhecida por distribuição genuína (*genuine distribution*). Em seguida, recolhem-se os resultados obtidos através da comparação entre C e T de indivíduos diferentes, dando origem à distribuição $p(s|H_0)$, também conhecida por distribuição impostora (*impostor distribution*). A intersecção das duas distribuições e a conjugação com nível de decisão dá origem ao FMR e FNMR, tal como se pode observar na figura 2.3 (adaptado de [34]). Desta figura, pode-se concluir que existe um compromisso entre o FMR e FNMR consoante o nível de decisão escolhido. Isto significa que se o nível de decisão diminuir, o FNMR também diminui, favorecendo os utilizadores que possuem um acesso legítimo ao mesmo tempo que aumenta o FMR. Por outro lado, se o nível de decisão aumentar, e como consequência FMR diminuir, o sistema torna-se mais seguro recusando mais acessos não autorizados.

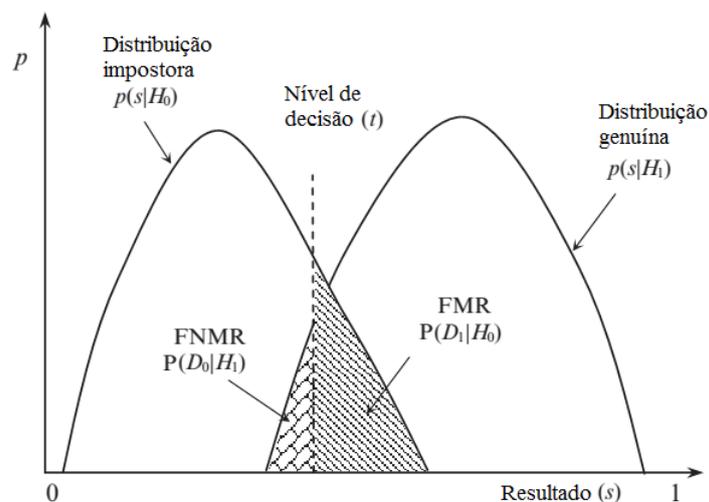


Figura 2.3: FMR e FNMR através da intersecção da distribuição genuína e da distribuição impostora (adaptado de [34]).

Pode-se então escrever estas métricas em função das distribuições apresentadas, através de:

$$\text{FMR} = \int_t^1 p(s|H_0)ds, \quad (2.3)$$

$$\text{FNMR} = \int_0^t p(s|H_1)ds. \quad (2.4)$$

Na figura 2.4 [34] representam-se o FMR e FNMR em função do nível de decisão, ou seja, o $\text{FMR}(t)$ e o $\text{FNMR}(t)$. A representação conjunta do $\text{FMR}(t)$ e do $\text{FNMR}(t)$ dá origem a três métricas auxiliares:

- **Taxa de Erro Igual (*Equal-Error Rate* - **EER**)** - o valor onde o $\text{FMR}(t)$ e o $\text{FNMR}(t)$ são iguais. Uma vez que na prática a distribuição impostora e a distribuição genuína são distribuições finitas, devido ao número finito de comparações que se podem realizar, nem sempre é possível encontrar o ponto exato do EER. Nestes casos é necessário estimar o valor do mesmo dentro de um intervalo.
- **ZeroFMR** - o ponto em que $\text{FNMR}(t)$ é mínimo, quando $\text{FMR}(t)$ é zero.
- **ZeroFNMR** - o ponto em que $\text{FMR}(t)$ é mínimo, quando $\text{FNMR}(t)$ é zero.

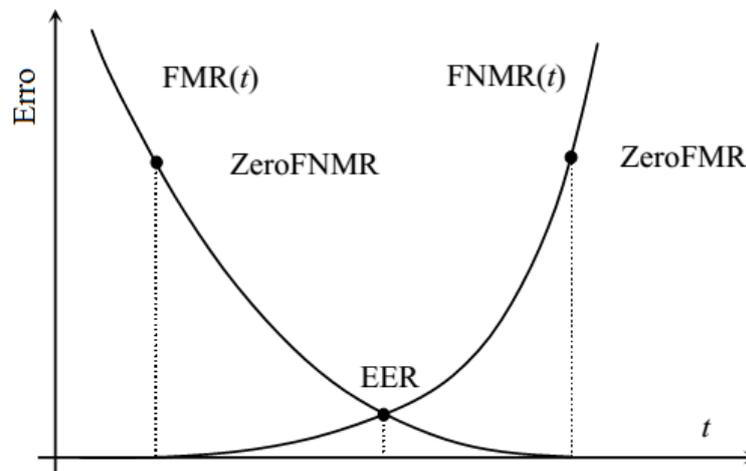


Figura 2.4: FMR e FNMR em função do nível de decisão t (adaptado de [34]).

Em sistemas biométricos, geralmente o EER serve apenas como indicador de desempenho, sendo que na prática o sistema irá funcionar com valores de t diferentes, consoante a aplicação do sistema biométrico. Na figura 2.5 (adaptado de [34]) encontra-se a *Receiver Operating Characteristic* (ROC) que dá uma análise visual do compromisso entre $\text{FMR}(t)$ e $\text{FNMR}(t)$. Na mesma figura é possível observar que o nível de decisão escolhido depende diretamente dos requisitos de cada tipo de aplicação. Por exemplo, em aplicações forenses existe uma maior

preocupação em garantir que não são excluídos candidatos, aumentando as falsas correspondências, do que correr o risco de eliminar candidatos por existir um maior número de falsas não-correspondências. Nestas situações é necessário realizar uma análise manual de forma a eliminar as falsas correspondências. No extremo oposto, em aplicações de alta segurança é menos preocupante o sistema recusar um acesso genuíno, dando origem a um maior número de falsas não-correspondências, do que aceitar um acesso ilegítimo (um maior número de falsas correspondências).

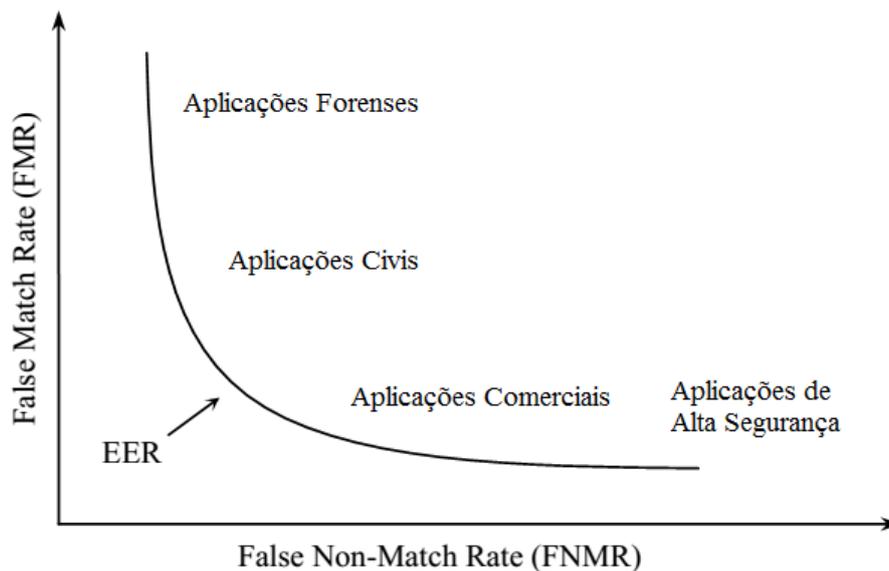


Figura 2.5: Diferentes tipos de aplicações através da ROC (adaptado de [34]).

2.6 Impressão digital

Há mais de 100 anos que as impressões digitais são utilizadas no reconhecimento de pessoas recorrendo ao conjunto de padrões formados por riscas e vales presentes na superfície das pontas dos dedos [34]. A formação das impressões digitais ocorre durante o desenvolvimento do feto, sendo que após os sete meses de gestação estas se encontram totalmente desenvolvidas e não sofrem qualquer tipo de alteração com exceção da existência de cortes ou feridas resultantes de acidentes [42]. É também aceite que as características de cada impressão digital são únicas. Assim, devido à unicidade, invariância e desempenho (ver tabela 2.1) deste indicador biométrico, o reconhecimento biométrico por impressões digitais é um dos métodos mais estudados e implementados até aos dias de hoje [34].

2.6.1 Constituição

Uma impressão digital representa a parte exterior da epiderme de um dedo, sendo constituída essencialmente por riscas (*ridges*) e vales (*valleys*). Na figura 2.6a identificam-se as riscas como sendo as linhas mais escuras e os vales como as linhas mais claras. A caracterização da constituição de uma impressão digital pode ser dividida em três níveis, designadamente:

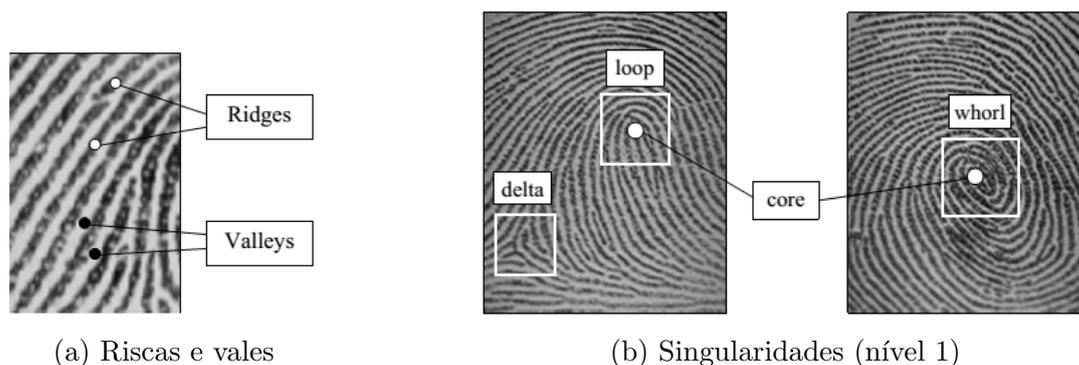


Figura 2.6: Constituição das impressões digitais: (a) riscas (*ridges*) e vales (*valleys*), (b) ponto central (*core*) e singularidades - laço (*loop*), delta e espiral (*whorl*) [34].

- **Nível 1** - análise global, o padrão gerado pelas riscas e vales, considerando as zonas em que estas deixam de estar em paralelo. Os padrões gerados são denominados por singularidades e podem ser classificados como laço (*loop*), delta ou espiral (*whorl*), tal como se pode observar na figura 2.6b. Pode-se também observar o ponto central (*core*), sendo este definido como o ponto onde se concentram mais riscas, tal que estas exibem curvatura máxima. Este ponto central é único na impressão digital e é utilizado como ponto de referência em vários algoritmos para realizar o alinhamento de impressões digitais. As singularidades são utilizadas para classificar o tipo de impressão digital, encontrando-se na figura 2.7 os cinco tipos de impressões digitais propostos por Edward Henry [24] - laço esquerdo (*left loop*), laço direito (*right loop*), espiral (*whorl*), arco (*arch*) ou arco em tenda (*tented arch*).
- **Nível 2** - análise local, na qual são consideradas as minúcias, i.e., as perturbações existentes nas riscas. Existem dois tipos principais de minúcias, terminações (*ridge ending*) e bifurcações (*bifurcation*), tal como exemplificado na figura 2.8a. As terminações são definidas nos pontos onde uma risca termina, enquanto que as bifurcações são os pontos em que uma risca se separa em duas ou em que duas riscas se combinam.

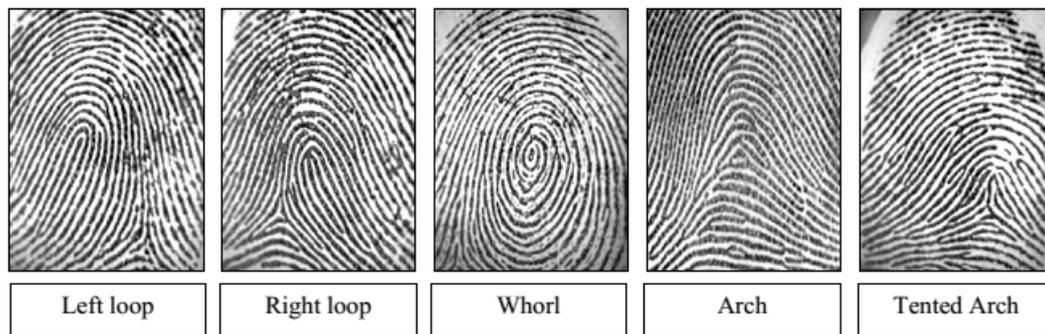


Figura 2.7: Classificação de impressões digitais: laço esquerdo (*left loop*), laço direito (*right loop*), espiral (*whorl*), arco (*arch*) e arco em tenda (*tented arch*) [34].

- **Nível 3** - caracterização a nível local, tal como no nível 2. No entanto, em vez de se recolher minúcias é realizado um levantamento mais detalhado das perturbações encontradas. Ao utilizar uma amostra de uma impressão digital em alta resolução, ou seja, superior a 500 Pontos Por Polegada (PPP), ou *Dots Per Inch* (DPI), é possível identificar características como a largura ou forma das riscas e a existência de poros. Na figura 2.8b apresenta-se uma imagem com 1000 DPI, onde é possível visualizar características deste tipo.



(a) Terminações e bifurcações (nível 2)

(b) Largura, forma e poros (nível 3)

Figura 2.8: Constituição das impressões digitais em cinco classes: (a) terminações (*ridge ending*) e bifurcações (*bifurcation*), (b) largura, forma e poros.

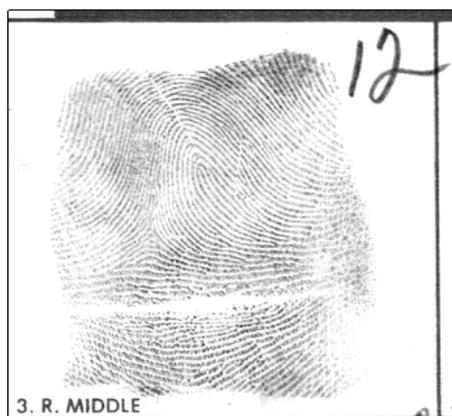
Para além destas características, as impressões digitais podem ainda ser constituídas por cortes ou cicatrizes temporárias ou permanentes. Na situação em que são temporárias, estas podem contribuir para a existência de erros, ao realizar o reconhecimento da impressão digital. Quando são permanentes, uma vez que não são consideradas características universais, não são aproveitadas no processo de reconhecimento. Por outro lado, se na criação do registo padrão já existirem essas características permanentes, estas não degradam o desempenho do sistema.

2.6.2 Aquisição de impressões digitais

Tal como referido na secção 2.3.1, um sistema biométrico pode ter componentes a operar em modo *online* ou *offline*, onde o componente responsável pela aquisição de impressões digitais pode funcionar nos dois modos. O modo *offline* geralmente encontra-se associado a um dos seguintes processos de aquisição:

- ***Rolled-ink*** - a ponta do dedo é preenchida com tinta, geralmente preta, sendo o dedo depois pressionado e deslizado sobre papel, assente numa base fixa e homogénea.
- ***Ink-on-paper*** - processo semelhante ao *rolled-ink*, no qual a tinta é colocada no papel *a priori*.
- ***Latent*** - recorre aos resíduos deixados pelos dedos em objetos para adquirir uma impressão digital. Para tal, recorre-se a produtos como o pó para polvilhar, à pulverização de ninidrina, defumação com iodo ou por imersão de nitrato de prata [27]. Este processo é bastante utilizado em aplicações forenses.

Nos processos de aquisição *offline* mencionados anteriormente, existe ainda a necessidade de digitalizar a amostra recolhida. As impressões digitais podem ainda ser recolhidas por *live-scan* através de sensores que dispensam a utilização de tinta e que realizam em tempo real a conversão da impressão digital num formato digital (modo *online*). Existem vários tipos de sensor *live-scan*, podendo estes ser divididos em três tipos de sensor:



(a) *Rolled-ink*

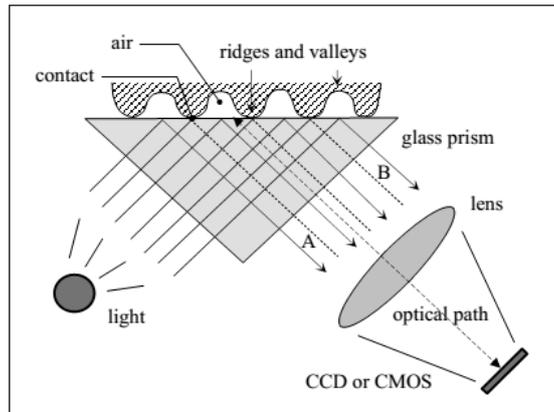


(b) Sensor ótico

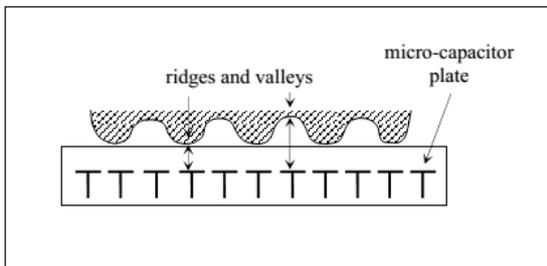
Figura 2.9: Impressões digitais adquiridas por: (a) processo de *rolled-ink*, (b) sensor ótico [34].

- **Ótico** - existem vários tipos de sensores óticos sendo os mais comuns os *Frustrated Total Internal Reflection* (FTIR). Os sensores FTIR utilizam um prisma de vidro, uma fonte de luz (um *Light Emitting Diode* - LED, ou um díodo *Light Amplification by Stimulated Emission of Radiation* - LASER) e um sensor *Charge Coupled Device* (CCD) ou *Complementary Metal-Oxide-Semiconductor* (CMOS) para criar uma amostra da impressão digital, encontrando-se caracterizado na 2.10a. Ao colocar-se a ponta do dedo sobre o prisma, as riscas ficam em contacto com a superfície deste, ficando os vales a uma determinada distância. Um dos lados do prisma é iluminado com a fonte de luz, enquanto que o sensor CCD ou CMOS se encontra no lado que forma um ângulo reto com o lado onde se encontra a fonte de luz, recebendo a luz refletida pela superfície em contacto com o dedo. A ausência de reflexão indica a presença de riscas, enquanto que a existência de luz proveniente da reflexão indica a presença de vales. Os sensores FTIR não possuem uma estrutura muito compacta e podem ser sensíveis a dedos secos, no entanto produzem imagens de qualidade. Estes não são muito dispendiosos e devido ao seu mecanismo de captura, são robustos à falsificação através da colocação de uma fotografia ou de uma imagem impressa de uma impressão digital [27][34]. Existem ainda outros tipos de sensores óticos como os baseados em fibras óticas ou em imagens multiespetrais.
- **Estado sólido** - os sensores de estado sólido, a par com os óticos FTIR, são os mais utilizados atualmente. Estes sensores possuem como grande vantagem a sua dimensão reduzida aliada ao baixo custo, permitindo assim a sua fácil implementação em dispositivos móveis como telemóveis/*smartphones* ou computadores portáteis. Na figura 2.10b apresenta-se um sensor de estado sólido do tipo capacitivo. Este tipo de sensor é constituído por uma matriz de eléctrodos, onde o dedo ao ser colocado sobre esta matriz atua como outro eléctrodo, dando origem a pequenos condensadores. As cargas elétricas criadas variam com a distância da superfície do dedo e a matriz, permitindo assim determinar a presença de riscas ou vales. Para além de sensores capacitivos, existem ainda sensores térmicos e sensores baseados no campo elétrico.
- **Ultrassons** - estes sensores recorrem ao envio de ondas acústicas e receção do consequente eco das mesmas para determinar a profundidade da imagem da impressão digital. É a partir do sinal de eco que é determinada a estrutura das riscas. Os sensores de ultrassons obtêm imagens de boa qualidade sendo constituídos por partes mecânicas e ainda são algo dispendiosos [34]. Na figura 2.10c encontra-se a estrutura de um sensor baseado

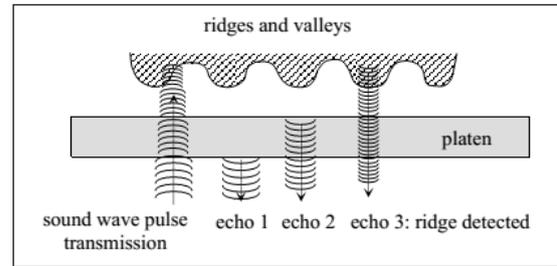
em ultrassons.



(a) Ótico



(b) Estado sólido



(c) Ultrassons

Figura 2.10: Tipos de sensor *live-scan*: (a) ótico, (b) estado sólido, (c) ultrassons [34].

Capítulo 3

Reconhecimento de impressões digitais: técnicas e ferramentas

Neste capítulo são apresentadas as técnicas e ferramentas de processamento de sinal, utilizadas neste projeto para realizar o reconhecimento de impressões digitais. Na secção 3.1 são abordadas as principais técnicas de extração e comparação de características, realizando uma breve descrição das mesmas. Na secção 3.2 é apresentado o algoritmo de extração de características, baseado no trabalho proposto em [42], enquanto que na secção 3.3 é descrito o processo de comparação de características de impressões digitais.

3.1 Extração e comparação de características baseado em minúcias

Existem diversos métodos para tirar partido das características apresentadas na secção 2.6, para assim criar uma representação da impressão digital. Os algoritmos de extração de características geralmente operam com características de nível 1 e 2, podendo-se considerar que existem quatro passos principais [27]:

- Estimação da orientação das riscas.
- Extração das riscas.
- Extração de singularidades.
- Extração de minúcias.

Na figura 3.1 encontra-se um diagrama de blocos representativo do processo de extração de características. Esta secção foca-se na apresentação de métodos, que embora sejam comuns em vários algoritmos, têm como objetivo de realizar a extração de minúcias para depois as utilizar no processo de comparação. A secção 3.2 é dedicada à apresentação de um algoritmo baseado em textura orientada, sendo este adotado neste trabalho.



Figura 3.1: Diagrama de blocos do processo de extração de características (minúcias) (adaptado de [27]).

3.1.1 Estimação da orientação das riscas

A orientação das riscas num determinado píxel é o ângulo θ que as riscas formam segundo o eixo dos x , como consequência, tem-se que θ assume valores entre $[0, 180[$. Geralmente a determinação da orientação das riscas é realizada em posições discretas em vez de píxel a píxel, permitindo assim diminuir a complexidade em termos computacionais, e estimando o valor dos restantes píxeis por interpolação [34].

Um dos métodos mais utilizados na determinação das riscas consiste no cálculo dos gradientes, $\partial_x(x, y)$ e $\partial_y(x, y)$, da imagem representativa da impressão digital ($I(x, y)$). O ∂_x e ∂_y são, respetivamente, as derivadas de $I(x, y)$ segundo o eixo dos x e dos y . A fase destas componentes, determinada através de

$$O(x, y) = \tan^{-1}(\partial_y(x, y)/\partial_x(x, y)), \quad (3.1)$$

representa a direção da variação máxima de intensidade, sendo a orientação das riscas ortogonal a essa direção. O cálculo da orientação das riscas através da equação (3.1) possui dois problemas: ambiguidade existente no cálculo da tangente e a sensibilidade ao ruído existente na impressão digital. O primeiro problema existe porque um vetor a duas dimensões com um ângulo θ possui a mesma orientação que outro vetor com ângulo $\theta + \pi$. Uma solução para este problema passa por multiplicar os ângulos por dois, ou seja, utilizar 2θ em vez de θ . O segundo problema faz com que a orientação detetada apresente erros em zonas em que a impressão digital possua ruído, sendo resolvido com a aplicação de um filtro passa-baixo de modo a suavizar a orientação e assim corrigir o padrão de riscas. Para suavizar a orientação das riscas deve-se considerar os seguintes passos:

- Remover a ambiguidade determinando $\Phi_x(x, y) = \cos(2O(x, y))$ e $\Phi_y(x, y) = \sin(2O(x, y))$.
- Aplicar um filtro passa-baixo às componentes em x e em y , $\Phi_x(x, y)$ e $\Phi_y(x, y)$, dando origem a $\Phi'_x(x, y)$ e $\Phi'_y(x, y)$.
- Determinar a orientação das riscas após a filtragem, $O' = 0,5 \times \tan^{-1}(\Phi'_y(x, y)/\Phi'_x(x, y))$.

Nas figuras 3.8a e 3.8b encontram-se, respetivamente, o resultado do cálculo da orientação das riscas antes e após a suavização.

3.1.2 Extração de singularidades

Através da orientação das riscas é possível determinar a localização das singularidades. As singularidades podem ser utilizadas num algoritmo de extração de características para encontrar o ponto de referência ou para determinar a que classe pertence a impressão digital. O *Poincaré index* (PI), consiste num método clássico de deteção de singularidades e pode ser definido por [27]:

$$PI = \frac{1}{\pi} \sum_{i=0}^7 \delta(O'[(i+1) \bmod 8] - O'[i]), \quad (3.2)$$

onde i indica qual o píxel vizinho ao que se efetua o cálculo e o $\delta(\theta)$ assume os valores dados por:

$$\delta(\theta) = \begin{cases} \theta - \pi, & \text{se } \theta > \pi/2 \\ \theta, & \text{se } -\pi/2 \leq \theta \leq \pi/2 \\ \theta + \pi, & \text{se } \theta < -\pi/2. \end{cases} \quad (3.3)$$

O PI pode assumir um de quatro valores: 0 indicando que não existe qualquer singularidade, 1 refere a presença de um *loop*, -1 indica a existência de um *delta* e o 2 indica a presença de um *whorl*. Um exemplo de cálculo do PI para os quatro casos apresentados encontra-se na figura 3.2.

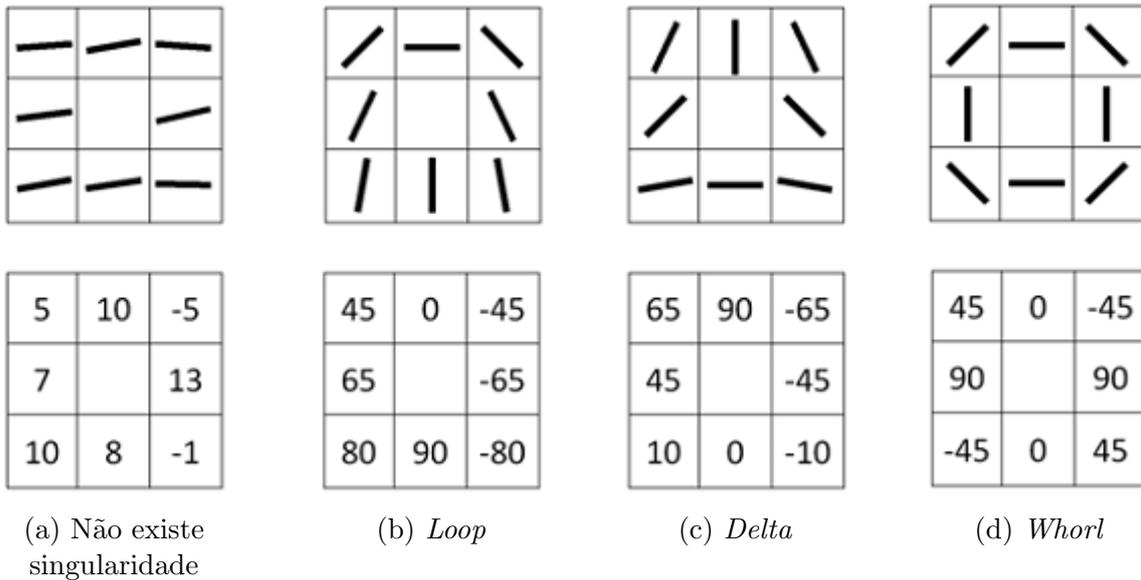


Figura 3.2: Exemplos de *Poincaré index* para as situações: (a) não existe uma singularidade, (b) existe um *loop*, (c) existe um *delta*, (d) existe um *whorl* (adaptado de [27]).

3.1.3 Extração das riscas

A partir da amostra de uma impressão digital é possível proceder logo à binarização da imagem (I_{bin}), considerando que os píxeis que ultrapassam um determinado nível de decisão são considerados vales (valor 1) e que os píxeis que se encontram abaixo do valor definido são riscas (valor 0), e assim sobre I_{bin} realizar a pesquisa de minúcias. No entanto, é necessário ter em conta que as riscas presentes na amostra podem apresentar poros e ter descontinuidades resultantes de cortes/feridas. Pode até mesmo ser detetada apenas uma risca, quando uma das riscas adjacentes não é detetada devido à presença de sujidade no dedo aquando da captura da amostra. Para minimizar estes problemas, é comum efetuar melhoramento da imagem através de filtros de Gabor [21], em que a orientação e a frequência destes têm valores de acordo com as das riscas.

Na figura 3.3 encontra-se um exemplo de melhoramento de imagem através de um conjunto de filtros de Gabor [21], enquanto que na secção 3.2.4, é abordado em mais detalhe este tipo de filtragem. Com a imagem melhorada pode-se então proceder à binarização da mesma, tal como



Figura 3.3: Melhoramento de impressões digitais através de filtros de Gabor: (a) impressão digital original, (b) impressão digital após melhoramento [27].

descrito anteriormente, seguindo-se o processo de *thinning* que representa as riscas detetadas com largura de somente um píxel (I_{thin}).

3.1.4 Extração de minúcias

Após a realização da operação de *thinning* é realizada a deteção das minúcias. Para tal, considera-se novamente uma vizinhança de oito píxeis na imagem I_{thin} e efetua-se a pesquisa através de [3]:

$$cn(p) = \frac{1}{2} \sum_{i=1..8} |\text{val}(p_{i \bmod 8}) - \text{val}(p_{i-1})|, \quad (3.4)$$

onde p representa o píxel, enquanto que $\text{val}(p) \in \{0, 1\}$ é o respetivo valor. Através desta equação pode-se concluir que existem as seguintes possibilidades:

- $cn(p) = 2$, trata-se de um ponto intermédio de uma risca;
- $cn(p) = 1$, corresponde a uma terminação;
- $cn(p) = 3$, corresponde a uma bifurcação;
- $cn(p) > 3$, identifica uma minúcia mais complexa, como por exemplo, o resultado da intersecção de várias riscas.

Na figura 3.4 exemplificam-se os três primeiros casos descritos. Este método pode ser falível e como tal retornar falsas minúcias devido a artefactos resultantes do processamento da imagem

ou devido ao ruído presente na imagem. Para minimizar o número de falsos positivos pode-se descartar as minúcias que [27]: não possuem uma risca adjacente num dos lados, indicando que esta se encontra num dos limites da imagem; se encontram próximas e a orientação das riscas são opostas, originando ilhas; existam em grande quantidade numa zona pequena. Após a remoção das espúrias, o conjunto de minúcias, ou *set* de minúcias, que representa a impressão digital pode ser formado.

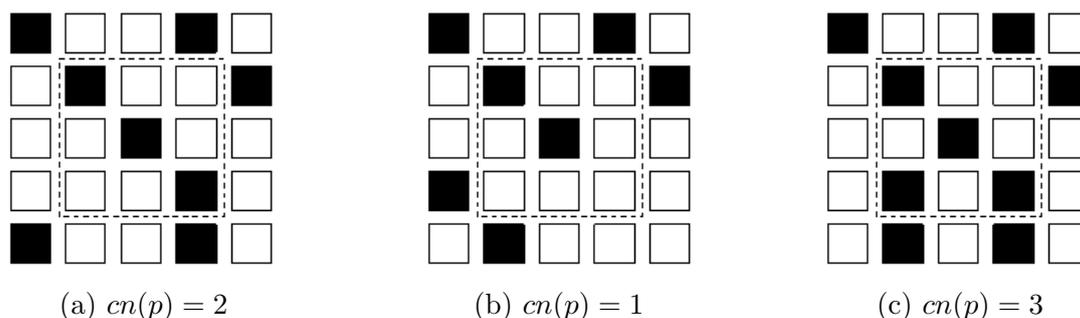


Figura 3.4: Detecção de minúcias: (a) não existe terminação ou bifurcação (ponto intermédio), (b) terminação, (c) bifurcação [34].

3.1.5 Comparação de características

Após o processo de extração de características e criação de um *set* de minúcias é então possível realizar a comparação de dois *sets* de características num processo com três passos [27]:

- **Alinhamento** - realiza a transformação dos dois conjuntos para o mesmo sistema de coordenadas fazendo com que os dois fiquem alinhados geometricamente. Um dos métodos para realizar o alinhamento passa por recorrer ao ponto de referência, centrando as minúcias em torno deste e realizando a rotação necessária de modo a alinhar os dois *sets*. Outro método envolve a utilização da transformada de Hough [8][45][48] que converte o problema de comparação de um padrão de pontos noutra de de deteção de máximos no espaço de parâmetros da transformada.
- **Correspondência** - é realizada a correspondência entre pares de minúcias dos dois *sets*, determinando a distância e o ângulo entre elas. Com base no nível de decisão definido para a distância e para o ângulo é decidido se a correspondência é válida ou não.
- **Geração do resultado** - após o alinhamento e a correspondência entre minúcias pode-se gerar o resultado com base em dois critérios. O primeiro recorre ao número de pares de

minúcias válidos, uma vez que a probabilidade de efetuar uma correspondência correta entre duas impressões digitais, é diretamente proporcional ao número de pares de minúcias encontrados. O segundo utiliza a percentagem de minúcias correspondidas dentro da área considerada entre os dois *sets*.

Na figura 3.5 encontra-se um diagrama de blocos representativo do processo de comparação de características por minúcias entre dois *sets*.

Embora a comparação por minúcias seja o método mais comum, existem outras abordagens como [34]:

- utilização da dimensão e forma dos contornos da impressão digital;
- recorrendo ao número, tipo e posição das singularidades;
- recorrendo à informação local ou global da textura;
- atributos geométricos e relação espacial entre riscas;
- utilização de características de nível 3.

Estas abordagens podem ser uma alternativa ou até um complemento a uma solução exclusivamente baseada em minúcias. A deteção de minúcias em algumas situações pode revelar-se problemática, como na presença de ruído ou quando o sensor utilizado possui uma pequena área e não é possível extrair minúcias em número suficiente.

3.1.6 Avaliação de desempenho

Para efetuar a avaliação de desempenho de um algoritmo de extração de características, e consequente avaliação do sistema (secção 2.5), é necessário recorrer a uma base de dados existente ou criar uma especificamente para o cenário de teste para depois analisar métricas como o FMR, FNMR e EER. Na comunidade científica é frequente a utilização de bases de dados disponibilizadas com o único propósito de comparar o desempenho de algoritmos de reconhecimento diferentes, como por exemplo as base de dados da *National Institute of Standards and Technology* (NIST) [37] ou as bases de dados criadas aquando da realização de competições de reconhecimento de impressões digitais, *Fingerprint Verification Competition* (FVC), em 2000 [32], 2002 [33], 2004 [31] e 2006 [53]. Note-se que os resultados obtidos num determinado

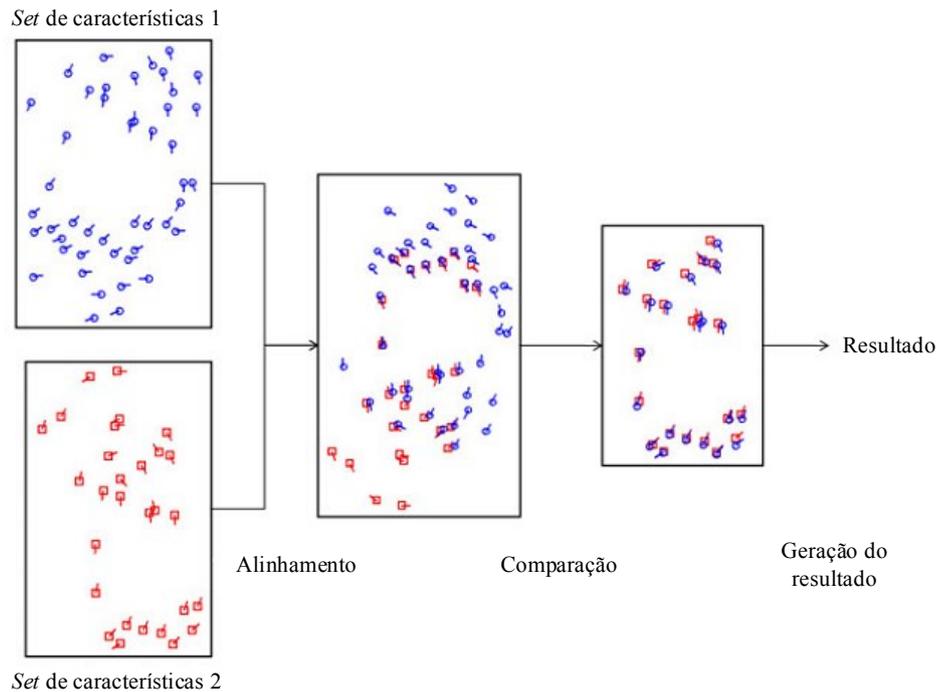


Figura 3.5: Diagrama de blocos do processo de comparação de características (minúcias) (adaptado de [27]).

cenário não são comparáveis com outro, ou seja, não é possível comparar diretamente os resultados apresentados na FVC2000 com os da FVC2006. Isto acontece porque as amostras das bases de dados criadas variam com: as características do sensor, a distribuição demográfica dos participantes que contribuíram, o número de amostras recolhidas, o controlo de qualidade realizado e com fatores ambientais (tais como temperatura, humidade e iluminação).

3.1.7 Desafios no reconhecimento de impressões digitais

Embora o reconhecimento através de impressões digitais seja um dos métodos biométricos mais estudados, a comunidade científica enfrenta ainda alguns desafios, tais como:

- **Falsificações** - o aumento do número de sistemas de reconhecimento de impressões digitais motivou alguns utilizadores a procurar soluções para contornar o sistema, existindo casos de quem tenha conseguido utilizar um dedo falso para contornar um acesso não autorizado ou até quem tenha realizado alterações na própria impressão digital de modo a invalidar o reconhecimento positivo [27].
- **Qualidade das amostras** - atualmente ainda é difícil digitalizar impressões digitais de ele-

vada qualidade de dedos em situações não ideais ou de utilizadores não treinados. Outro problema reside no reconhecimento através de impressões digitais do tipo *latent* devido à baixa qualidade das amostras resultantes, trazendo complicações quando se pretende ter um sistema automático de reconhecimento e muitas vezes o processo de comparação necessita de ser realizado manualmente.

- **Fusão de sistemas biométricos** - uma das soluções para aumentar a fiabilidade dos sistemas biométricos passa por conjugar diferentes indicadores biométricos. Um sistema de reconhecimento que realize a fusão de sistemas descorrelacionados, como por exemplo combinar impressões digitais com íris, terá melhor desempenho do que um sistema que combine duas impressões digitais do mesmo utilizador mas de dedos diferentes, ou seja, que faça a fusão de sistemas correlacionados.

3.2 Algoritmo de extração de características baseado em textura orientada

Tal como apresentado na secção 2.6, existem vários tipos de características que se podem extrair de uma impressão digital. A situação mais comum consiste em utilizar as características de nível 1 para classificar a impressão digital e para obter o ponto de referência, enquanto que as características de nível 2 são utilizadas para retirar a informação que permite associar a impressão digital a determinado utilizador. O algoritmo estudado neste trabalho, e que se encontra resumido na figura 3.6, recorre a características de nível 2 para representar a impressão digital. Contudo, ao contrário de outras implementações que recolhem a posição e direção das minúcias, este algoritmo realiza o reconhecimento por textura orientada [42]. Este tipo de reconhecimento consiste em dividir a zona de interesse de uma amostra em setores, com uma dimensão tal que seja possível detetar as perturbações provocadas pela presença de minúcias. Através da aplicação de um banco de filtros obtém-se um vetor de características, que depende da orientação e espaçamento entre riscas. Assim, podem-se considerar os seguintes passos no algoritmo de extração de características:

- **Determinação do ponto de referência** - para calcular o ponto de maior curvatura é necessário determinar em primeiro lugar a orientação de campo, i.e., a orientação das riscas. Segue-se a aplicação de um filtro de média que minimiza variações na orientação do

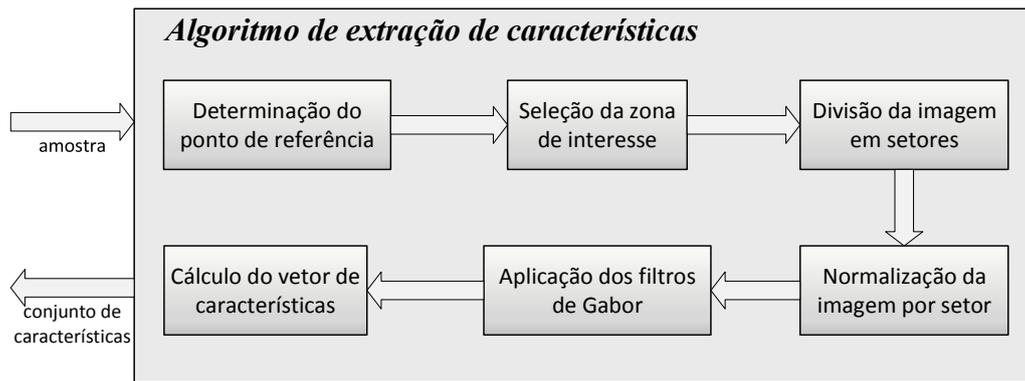


Figura 3.6: Algoritmo de extração de características.

campo, originadas pelo ruído presente na amostra. Ao resultado desta filtragem é retirada a componente em seno da orientação do campo, aplicando de seguida um filtro de variância. Este filtro tem como objetivo encontrar o ponto onde existe uma maior variação na orientação das riscas, encontrando-se o ponto de referência nas coordenadas do píxel com o valor máximo de variância.

- **Seleção da zona de interesse** - a zona de interesse é definida pela quantidade e dimensão dos setores escolhidos, sendo estes parâmetros dependentes do sensor utilizado. Esta zona encontra-se centrada no ponto de referência.
- **Divisão da imagem em setores** - a divisão da zona de interesse por setores é também definida pela quantidade e dimensão dos setores utilizados. Depois, é aplicado a cada setor uma normalização com base na sua média e variância.
- **Aplicação dos filtros de Gabor e cálculo do vetor de características** - para criar o vetor com um conjunto de características recorre-se à aplicação de um banco de filtros de Gabor [21], em que cada filtro se encontra ajustado para medir a orientação das riscas.

3.2.1 Determinação do ponto de referência

A determinação do ponto de referência assume um papel fundamental num algoritmo de reconhecimento de impressões digitais, uma vez que este ponto é essencial para retirar as características da impressão digital. No caso de não ser realizada uma deteção correta do ponto de referência, no processo de captura para a base de dados, ou nos processos de identificação ou autenticação, poderá comprometer a utilização do sistema para o utilizador.

No método de cálculo do ponto de referência proposto em [42], pretende-se determinar o ponto de maior curvatura recorrendo à orientação das riscas segundo os seguintes passos:

1. Cálculo da orientação de campo (O).
2. Filtragem passa-baixo da orientação de campo (O').
3. Aplicação do filtro de variância sobre a componente em seno de O' .
4. Determinação das coordenadas do ponto de referência.

3.2.1.1 Cálculo da orientação de campo

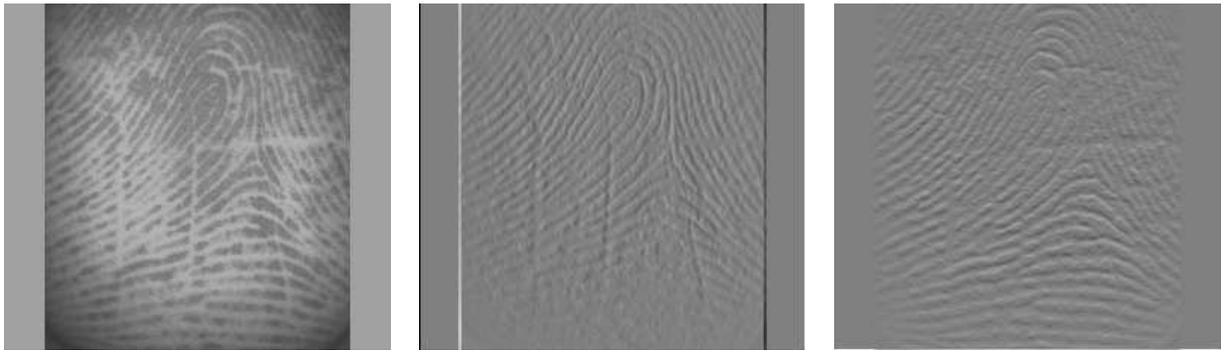
Para o cálculo da orientação de campo recorre-se a um algoritmo baseado na minimização do erro quadrático médio, ou *Least Mean Square* (LMS), proposto em [25]. Considerando uma imagem I como sendo a representação de uma impressão digital com resolução $M \times N$, a sua orientação de campo, O , é definida como sendo uma imagem de dimensão $P \times Q$. A orientação de campo é definida por blocos de píxeis não sobrepostos e não por um único píxel. A imagem O possui resolução inferior em relação à imagem I . Os blocos possuem resolução de $w \times w$, onde w define a resolução da imagem O : $P = \frac{M}{w}$ e $Q = \frac{N}{w}$. Assim, a orientação de campo, i.e., a orientação das riscas, pode ser calculada através das seguintes equações:

$$V_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} \partial_x^2(u, v) - \partial_y^2(u, v), \quad (3.5)$$

$$V_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v) \cdot \partial_y^2(u, v), \quad (3.6)$$

$$O(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(i, j)}{V_x(i, j)} \right), \quad (3.7)$$

onde ∂_x e ∂_y são os gradientes obtidos através da aplicação do operador de Sobel para deteção de contornos, $i \in \{0; P - 1\}$ e $j \in \{0; Q - 1\}$. Na figura 3.7, encontra-se a amostra de uma impressão digital (figura 3.7a), juntamente com o resultado do cálculo dos gradientes ao longo do eixo dos x (figura 3.7b) e ao longo do eixo dos y (figura 3.7c). A orientação de campo, sobreposta com a amostra original e representada como distribuição de intensidade, encontram-se, respetivamente, nas figuras 3.8a e 3.8c.



(a) Amostra original.

(b) Gradiente ∂_x .

(c) Gradiente ∂_y .

Figura 3.7: Exemplo de cálculo do gradiente: (a) amostra original, (b) gradiente da amostra ao longo do eixo dos x (∂_x), (c) gradiente da amostra ao longo do eixo dos y (∂_y).

3.2.1.2 Filtragem da orientação de campo

Devido à presença de ruído e de falhas nas riscas e vales na imagem I , a imagem de orientação de campo O nem sempre pode ser considerada uma representação correta da orientação das riscas. Uma vez que a orientação das riscas tem uma variação lenta numa vizinhança onde não existam singularidades, pode-se recorrer a um filtro passa-baixo para corrigir os possíveis erros presentes na imagem O .

Para realizar a filtragem passa-baixo é necessário converter a imagem O num vetor de campo contínuo definido como:

$$\Phi_x(i, j) = \cos(2O(i, j)), \quad (3.8)$$

e

$$\Phi_y(i, j) = \sin(2O(i, j)), \quad (3.9)$$

onde Φ_x e Φ_y são respetivamente a componentes em x e y do vetor de campo. A estes vetores é então aplicada a filtragem passa-baixo através de:

$$\Phi'_x(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_x(i - u.w, j - v.w), \quad (3.10)$$

e

$$\Phi'_y(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_y(i - u.w, j - v.w), \quad (3.11)$$

onde W é um filtro a duas dimensões, $w_\Phi \times w_\Phi$, de integral unitário (a soma de todos os elementos de W é 1). Segundo [42], uma vez que a filtragem é realizada ao bloco, utiliza-se um

filtro de média com dimensão 5×5 . O campo de orientação após a filtragem é dado por:

$$O'(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\Phi'_y(i, j)}{\Phi'_x(i, j)} \right). \quad (3.12)$$

Nas figuras 3.8b e 3.8d encontram-se, respectivamente, a orientação de campo sobreposta com a amostra original e a orientação de campo representada como distribuição de intensidade após a filtragem.

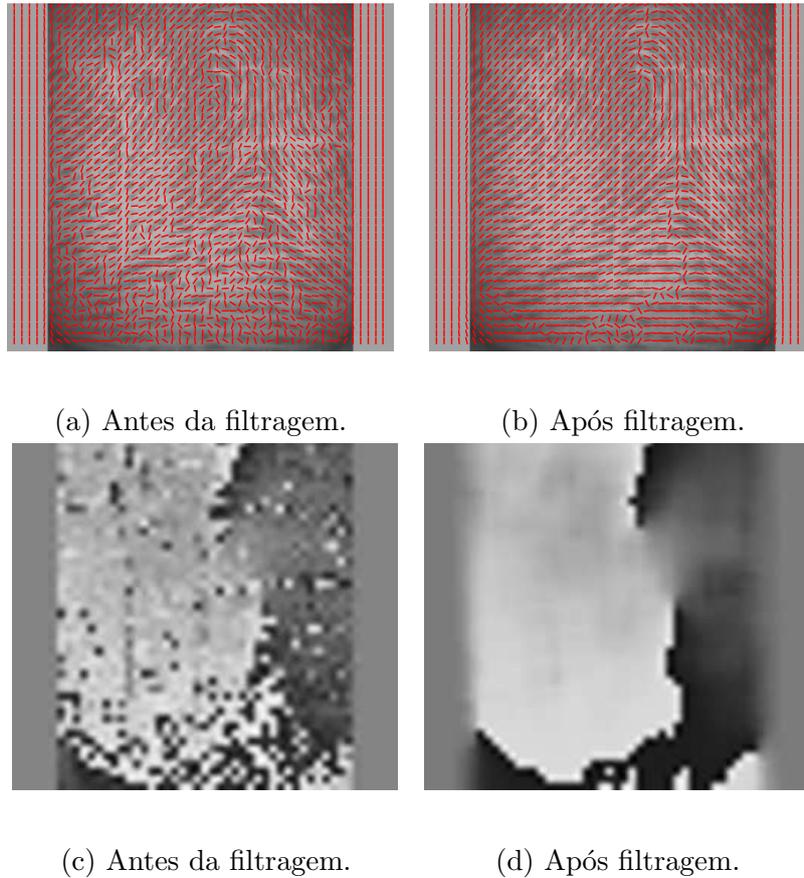


Figura 3.8: Cálculo da orientação de campo: orientação de campo sobreposta com a amostra original antes (a) e após (b) a filtragem; orientação de campo representada como distribuição de intensidade antes (c) e após (d) a filtragem.

3.2.1.3 Aplicação do filtro de variância sobre a componente em seno de O'

A componente em seno de O' é dada por:

$$\varepsilon(i, j) = \text{sen}(O'(i, j)), \quad (3.13)$$

sendo calculada a variância em cada píxel da imagem $\varepsilon(i, j)$ através de:

$$V(i, j) = EQ(i, j) - m_E^2(i, j), \quad (3.14)$$

com

$$EQ(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \varepsilon^2(i - u, j - v), \quad (3.15)$$

e

$$m_E(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \varepsilon(i - u, j - v), \quad (3.16)$$

onde W corresponde ao filtro descrito nas equações (3.10) e (3.11).

3.2.1.4 Determinação das coordenadas do ponto de referência

Com a variância calculada, resta determinar a localização do ponto de maior curvatura. Para tal define-se a máscara

$$K = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}, \quad (3.17)$$

de dimensão $w_k \times w_k$ que, ao realizar-se a correlação entre esta e V através de (3.18), faz sobressair no máximo de $Z(i, j)$ o local mais provável para a existência do ponto de referência,

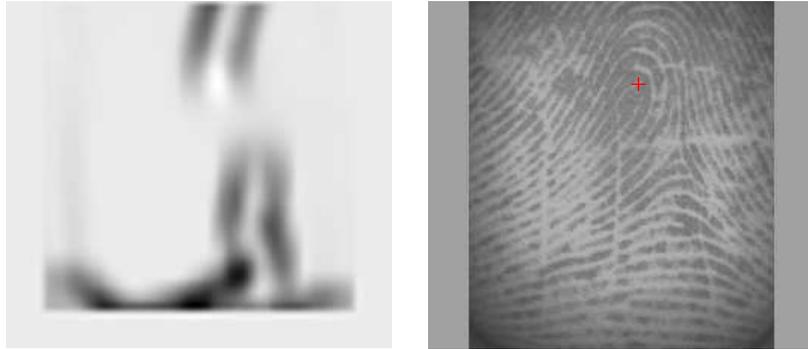
$$Z(i, j) = \sum_{u=-w_k/2}^{w_k/2} \sum_{v=-w_k/2}^{w_k/2} K(u, v) \cdot V(i - u, j - v). \quad (3.18)$$

As coordenadas do valor máximo presente na imagem Z são então as coordenadas do ponto de referência:

$$(x_{ref}, y_{ref}) = \arg_{(i,j)} \max\{Z(i, j) \cdot w\}, \quad (3.19)$$

onde a multiplicação por w permite obter as coordenadas na imagem I . Nas figuras 3.9a e 3.9b encontram-se, respetivamente, o resultado da aplicação do filtro de variância e a localização do

ponto de referência na amostra original. Note-se que a zona da primeira imagem onde os píxeis possuem maior intensidade, corresponde à localização do ponto de referência.



(a) Filtro de variância.

(b) Ponto de referência.

Figura 3.9: Determinação do ponto de referência: (a) resultado da aplicação do filtro de variância, (b) localização do ponto de referência na amostra original.

3.2.2 Seleção da zona de interesse e divisão em setores

Com o ponto de referência determinado, pode-se então definir a zona de interesse em torno do mesmo. A zona de interesse de uma impressão digital, $I(x, y)$, é definida por um conjunto de setores S_i [42]:

$$S_i = \{(x, y) | b(T_i + 1) \leq r < b(T_i + 1), \theta_i \leq \theta < \theta_{i+1}, 1 \leq x \leq M, 1 \leq y \leq N\}, \quad (3.20)$$

onde

$$T_i = i \operatorname{div} k, \quad (3.21)$$

$$\theta_i = (i \operatorname{mod} k) \times \left(\frac{2\pi}{k} \right), \quad (3.22)$$

$$r = \sqrt{(x - x_{ref})^2 + (y - y_{ref})^2}, \quad (3.23)$$

$$\theta = \tan^{-1} \left(\frac{y - y_{ref}}{x - x_{ref}} \right), \quad (3.24)$$

com $i = 0, \dots, (B \times k - 1)$, B indica o número de bandas centradas no ponto de referência, k é o número de setores por banda e b representa a largura de cada banda. É escolhida uma divisão circular em setores porque a rotação de uma amostra corresponde à rotação destes mesmos setores, sendo assim mais fácil realizar a comparação entre amostras diferentes. Os parâmetros

apresentados são definidos consoante o tipo de amostra usada, que por sua vez dependem diretamente do sensor utilizado. O parâmetro B depende da dimensão da amostra e da resolução em Pontos Por Polegada (PPP), ou *Dots Per Inch* (DPI), do sensor, i.e., este tem de ter um valor onde é maximizada a informação recolhida das riscas e vales sem que as amostras sejam rejeitadas por não ser possível obter a zona de interesse. O valor de b depende essencialmente da resolução em DPI, devendo-se definir o valor deste parâmetro de modo a que na largura de uma banda se encontre em média uma risca e um vale. O valor do k define se a informação em cada banda é local ou global. Os valores B , b e k variam com o tipo de aplicação e geralmente são determinados de forma empírica.

Na figura 3.10a encontra-se uma amostra de uma impressão digital com a zona de interesse centrada no ponto de referência. Sendo esta uma amostra recolhida num sensor de 500 DPI e resolução da imagem de 508×480 píxeis [42], são definidos os parâmetros $b = 20$, $B = 5$ e $k = 16$, obtendo-se 80 setores. Outro exemplo sobre a constituição da zona de interesse é representado na figura 3.10b. Para além dos parâmetros descritos anteriormente, evidencia-se o parâmetro *gap*, que define a distância entre o ponto de referência e a primeira banda.

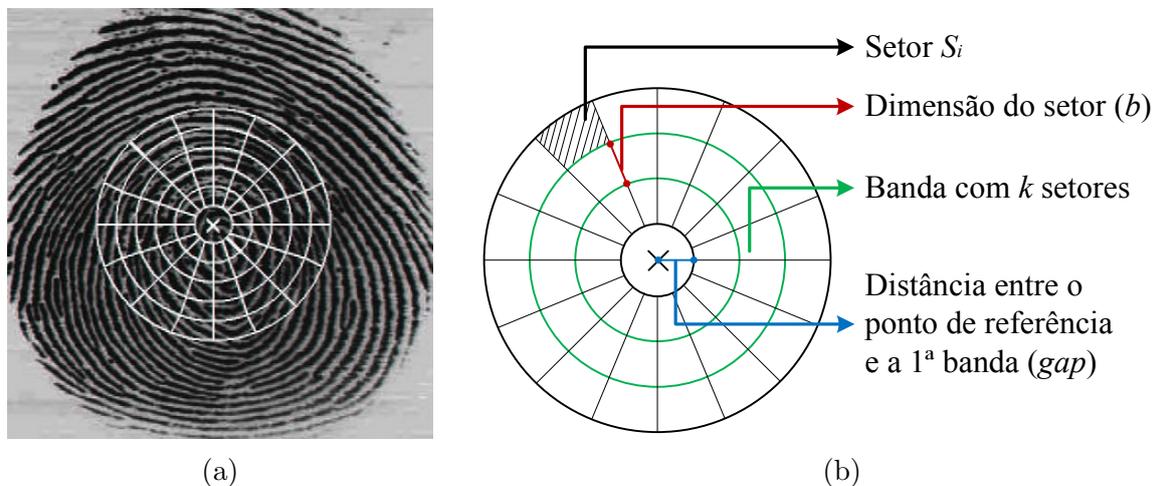


Figura 3.10: Zona de interesse: (a) impressão digital com a zona de interesse centrada no ponto de referência (\times) com os parâmetros $b = 20$, $B = 5$ e $k = 16$ [42], (b) constituição da zona de interesse.

3.2.3 Normalização da imagem por setores

Com a zona de interesse dividida em setores é necessário normalizar os mesmos de modo a: remover o ruído introduzido pelo sensor; corrigir a diferença nos níveis de cinzento que podem surgir devido a níveis de pressão diferentes no dedo, aquando da recolha da amostra.

Caso fosse realizada a normalização na zona de interesse completa, não seria possível efetuar a compensação dos diferentes níveis de cinzento. Considerando uma imagem $I_i(x, y)$, que representa os píxeis do setor S_i , realiza-se a normalização da imagem segundo a expressão:

$$I_{N_i} = \begin{cases} M_0 + \frac{\sqrt{V_0 \times (I_i(x, y) - M_i)^2}}{V_i}, & I_i(x, y) > M_i \\ M_0 - \frac{\sqrt{V_0 \times (I_i(x, y) - M_i)^2}}{V_i}, & I_i(x, y) \leq M_i \end{cases} \quad (3.25)$$

onde M_0 e V_0 representam respetivamente a média e a variância pretendidas na normalização, enquanto que M_i e V_i são respetivamente a média e a variância do setor S_i a normalizar. Na figura 3.11 apresenta-se o exemplo de uma zona de interesse antes e após a normalização por setores.

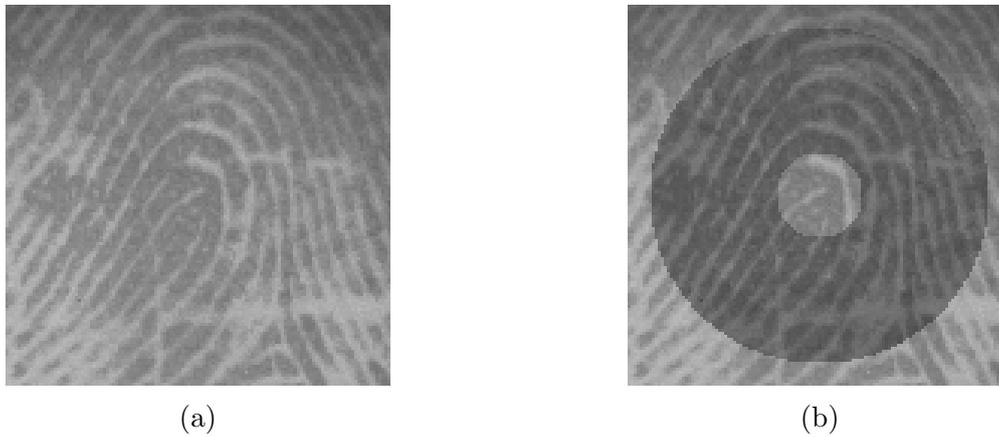


Figura 3.11: Exemplo de zona de interesse: (a) antes da normalização, (b) depois da normalização.

3.2.4 Aplicação dos filtros de Gabor

Em termos de análise local, as minúcias são consideradas perturbações existentes nas riscas. Através dos filtros de Gabor pretende-se capturar essas mesmas perturbações, dividindo a informação recolhida das mesmas segundo um conjunto de orientações, ao mesmo tempo que se remove o ruído existente. Um filtro de Gabor [21], do tipo passa-banda com orientação espacial, a duas dimensões é definido por:

$$G_{(x,y,f_0)} = \exp \left\{ -\frac{1}{2} \left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2} \right) \right\} \cos(2\pi f_0 x_\theta) \quad (3.26)$$

e

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix} = \begin{bmatrix} \sin(\theta) & \cos(\theta) \\ -\cos(\theta) & \sin(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.27)$$

onde x_θ e y_θ são as coordenadas após rotação dos eixos por um ângulo de $90^\circ - \theta$, f_0 representa a frequência das riscas e θ é a orientação do filtro. O valor do parâmetro f_0 , que indica a frequência da sinusóide segundo a direção θ , é obtido através da distância entre as riscas (d_r). Considerando que a distância entre riscas é aproximadamente $b/2$, o parâmetro f_0 é dado por $1/d_r$. As oito direções necessárias [42] para capturar as características locais são definidas segundo a expressão $i \times (180^\circ/8)$, $i = 0, \dots, 7$, dando origem às orientações $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$, estando estas presentes na figura 3.12. σ_x e σ_y são, res-

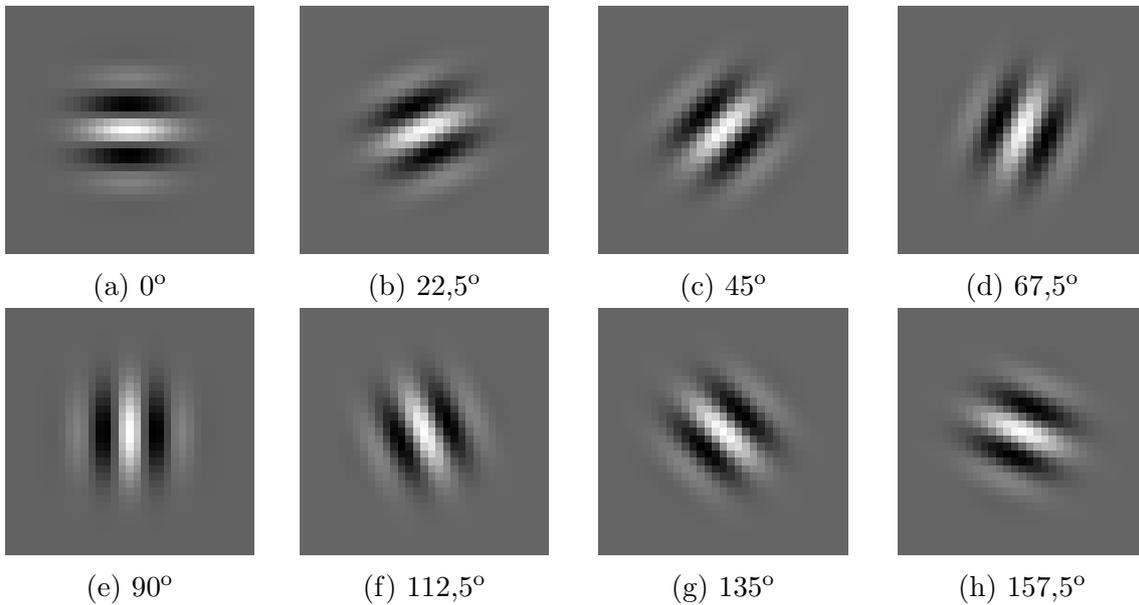


Figura 3.12: Máscaras de Gabor para as orientações $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$.

petivamente, as constantes espaciais da envolvente gaussiana ao longo eixo dos x e dos y . A escolha dos valores σ_x e σ_y envolve um compromisso entre robustez do filtro ao ruído (valores elevados) e sensibilidade aos detalhes das riscas e vales (valores baixos) [25]. No entanto, os valores destas duas constantes são determinados empiricamente, para adaptar o algoritmo ao tipo de imagem típica produzida pelo sensor. Os valores atribuídos aos parâmetros descritos anteriormente definem o banco de oito filtros de Gabor utilizados neste algoritmo. Na figura 3.13 pode-se observar um exemplo de filtragem para as 8 orientações.

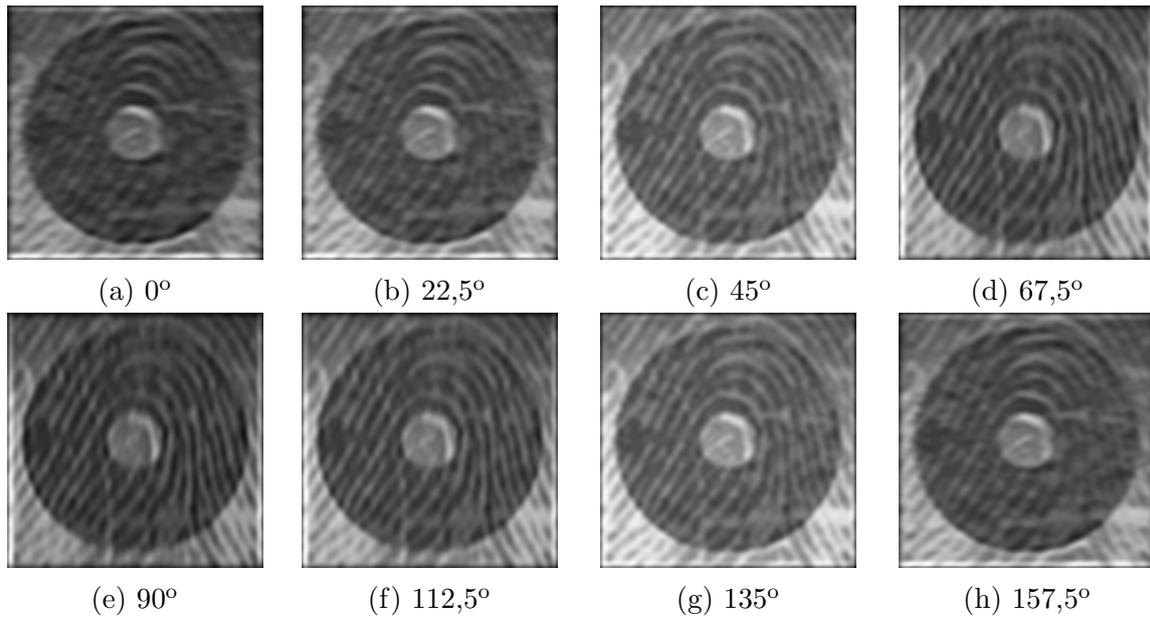


Figura 3.13: Exemplo de filtragem de Gabor para as orientações $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$.

3.2.5 Cálculo do vetor de características

O conjunto de vetores de características é uma representação compacta da amostra associada a uma impressão digital. É através destes vetores que é realizado o reconhecimento de impressões digitais. Sali Prabhakar [42] afirma que para realizar a classificação ou verificação de impressões digitais se pode utilizar cálculos estatísticos, propondo um método que recorre ao desvio médio absoluto dos píxeis de uma imagem, em relação ao seu valor médio. Ao efetuar este cálculo para cada um dos setores, e encontrando-se cada um desses setores separado em oito imagens resultantes da filtragem de Gabor, é então possível construir um vetor composto pelas características que representam um setor de uma amostra:

$$V_{i\theta} = \frac{1}{n_i} \left(\sum_{n_i} |F_{i\theta}(x, y) - P_{i\theta}| \right), \quad (3.28)$$

onde $F_{i\theta}$ representa a imagem filtrada segundo uma direção θ do setor S_i , n_i indica o número de píxeis existentes no setor S_i , enquanto que $P_{i\theta}$ é o valor médio dos píxeis de $F_{i\theta}$ no setor S_i . Ao criar-se um conjunto de características $V_{i\theta}$ para todos os setores de uma amostra obtém-se um *FingerCode*. A constituição do *FingerCode* encontra-se na figura 3.14, onde se pode verificar que para cada um dos setores existem oito valores associados.

	0°	22,5°	45°	67,5°	90°	112,5°	135°	157,5°
S_1 :	$V_{1 0^\circ}$	$V_{1 22,5^\circ}$	$V_{1 45^\circ}$	$V_{1 67,5^\circ}$	$V_{1 90^\circ}$	$V_{1 112,5^\circ}$	$V_{1 135^\circ}$	$V_{1 157,5^\circ}$
S_2 :	$V_{2 0^\circ}$	$V_{2 22,5^\circ}$	$V_{2 45^\circ}$	$V_{2 67,5^\circ}$	$V_{2 90^\circ}$	$V_{2 112,5^\circ}$	$V_{2 135^\circ}$	$V_{2 157,5^\circ}$
S_3 :	$V_{3 0^\circ}$	$V_{3 22,5^\circ}$	$V_{3 45^\circ}$	$V_{3 67,5^\circ}$	$V_{3 90^\circ}$	$V_{3 112,5^\circ}$	$V_{3 135^\circ}$	$V_{3 157,5^\circ}$
S_4 :	$V_{4 0^\circ}$	$V_{4 22,5^\circ}$	$V_{4 45^\circ}$	$V_{4 67,5^\circ}$	$V_{4 90^\circ}$	$V_{4 112,5^\circ}$	$V_{4 135^\circ}$	$V_{4 157,5^\circ}$
	• • •							
S_N :	$V_{N 0^\circ}$	$V_{N 22,5^\circ}$	$V_{N 45^\circ}$	$V_{N 67,5^\circ}$	$V_{N 90^\circ}$	$V_{N 112,5^\circ}$	$V_{N 135^\circ}$	$V_{N 157,5^\circ}$

Figura 3.14: Constituição do *FingerCode*.

3.3 Comparação de características de impressões digitais

Com a criação do conjunto de características para representar uma impressão digital (*FingerCode*), pode-se definir o método de comparação entre as mesmas a ser utilizado em sistemas de reconhecimento de impressões digitais. Para realizar a comparação entre dois *FingerCode* considera-se a seguinte expressão que determina a distância euclidiana entre eles:

$$d = \sqrt{\sum_i (F_{Code1}(i) - F_{Code2}(i))^2}, \quad (3.29)$$

onde F_{Code1} e F_{Code2} correspondem respectivamente ao *FingerCode* da amostra 1 e ao *FingerCode* da amostra 2, e o i representa o setor S_i associado ao *FingerCode* correspondente. Considerando um determinado nível de decisão pode-se decidir se a distância obtida permite uma validação positiva ou negativa.

O processo de comparação de *FingerCodes*, tal como foi descrito anteriormente, é sensível à rotação das amostras. Por exemplo, numa situação em que está a ser calculada a distância euclidiana entre dois *FingerCodes* com uma rotação α entre eles, quanto maior for o valor do α , maior será a distância entre os *FingerCodes*. Como consequência, as métricas FMR e FNMR (secção 2.4.1) poderão degradar-se, contribuindo para um pior desempenho do sistema biométrico. Para minimizar este problema, são definidos dois tipos de compensação [42]: efetuar rotação na amostra à qual se realiza a extração de características e aplicar a rotação dos *FingerCodes*.

A rotação da amostra é realizada em dois sentidos, $\pm\alpha$, em que α deve ser adaptado ao tipo de amostra utilizada. Pelos testes realizados em [42], uma vez que as características presentes

nos vetores que compõem os *FingerCodes* estão distanciadas em múltiplos de $22,5^\circ$ ($360^\circ/k$ em que $k = 16$), os *FingerCodes* são invariantes a rotações até $\pm 11,25^\circ$, tendo sido este o valor considerado em [42] para o parâmetro α . Tirando partido da distância entre os valores presentes nos vetores $V_{i\theta}$ pode-se realizar rotações em passos de $R \times 360^\circ/k$:

$$V_{i\theta}^R = V_{i'\theta'}, \quad (3.30)$$

$$i' = ((i + k - R) \bmod k) + (i \operatorname{div} k) \times k, \quad (3.31)$$

$$\theta' = (\theta + 180^\circ + (360^\circ/k) \times (-R)) \bmod 180^\circ, \quad (3.32)$$

onde $V_{i\theta}^R$ é o vetor rodado do *FingerCode*, $V_{i'\theta'}$ é o vetor original; R assume valores inteiros para definir qual o passo e o sentido do mesmo; k representa o número de setores por banda; i indica o setor do vetor sobre o qual se procede à rotação; θ assume as oito direções definidas nos filtros de Gabor. Considerando três valores para $R \in \{-1, 0, 1\}$ obtém-se o mesmo número de vetores: $V_{i\theta}^{-1}$, $V_{i\theta}^0$ e $V_{i\theta}^1$. As rotações dos nove *FingerCodes* resultantes, assumindo que $\alpha < \frac{360^\circ/k}{2}$, encontram-se representadas na figura 3.15.

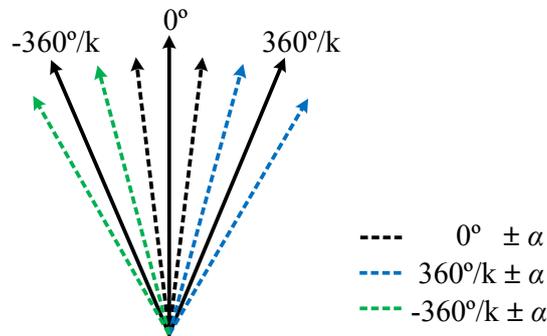


Figura 3.15: Ângulos de rotação dos *FingerCodes* para $R \in \{-1, 0, 1\}$ com $\alpha < \frac{360^\circ/k}{2}$.

Capítulo 4

Estudo e análise do sistema biométrico em *software*

No presente capítulo descreve-se a implementação do sistema biométrico proposto em *software*. Assim, na secção 4.1 é definida a constituição do sistema biométrico, descrevendo-se o sensor utilizado e caracterizando-se a base de dados criada. Na secção 4.2 o sistema é analisado recorrendo ao MATLAB (MATrix LABoratory)[50], sendo apresentados os parâmetros utilizados no algoritmo de reconhecimento de impressões digitais e avaliado o desempenho do mesmo. Na secção 4.3 é analisada a implementação em linguagem C do mesmo algoritmo de reconhecimento de impressões digitais, sendo esta executada no processador *MicroBlaze* [66] implementado na placa de desenvolvimento Xilinx Virtex-7 FPGA VC707 Evaluation Kit [67]. Esta análise permite a determinação do processo candidato a uma implementação em *hardware*, efetuando-se na secção 4.4 um estudo mais detalhado do mesmo.

4.1 Caracterização do sistema biométrico

Tal como é referido no capítulo 2, uma das opções para avaliar o desempenho de um sistema biométrico passa por desenvolver um cenário de testes. Tendo em conta este modelo de avaliação, é necessário definir os elementos que podem influenciar o comportamento do sistema. Pelo estudo do algoritmo de extração de características realizado no capítulo 3, pode-se concluir que o sensor a utilizar desempenha um papel fundamental na definição dos parâmetros do algoritmo, sendo também este sensor o elemento do sistema que determina o tipo de imagens que constitui a base de dados de teste. Na figura 4.1 apresenta-se um diagrama de blocos repre-

sentativo do cenário criado para efetuar a avaliação do sistema biométrico proposto, podendo este ser dividido em duas fases. Na primeira fase é realizada a recolha de amostras do sensor, com imagens de resolução 256×256 a 8 bits por píxel, com o objetivo de constituir uma base de dados própria, enquanto que na segunda fase é utilizada essa mesma base de dados para avaliar o desempenho do sistema através das métricas definidas no capítulo 2. Nesta última fase recorre-se diretamente à base de dados construída para criar conjuntos de características compostos por 3 *FingerCode*, resultantes da introdução da invariância à rotação, a fim de serem utilizados na comparação de características e serem armazenados em registos.

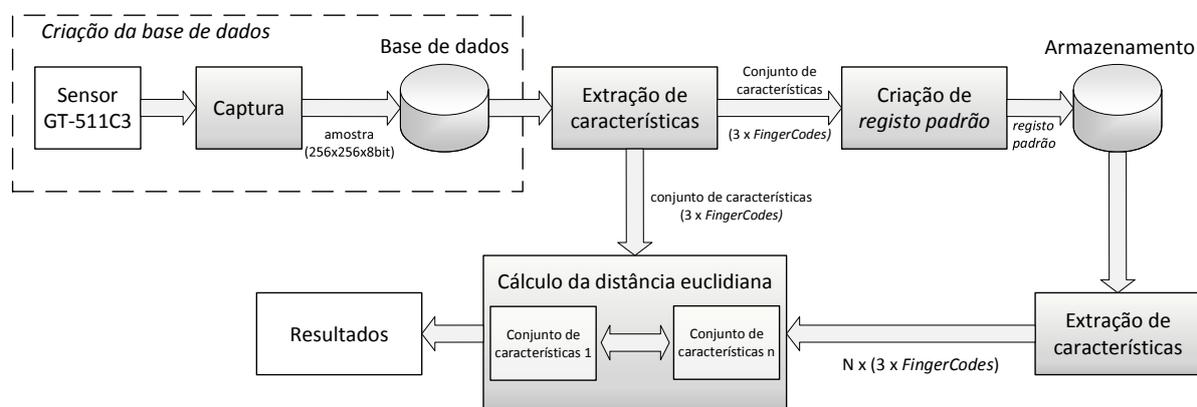


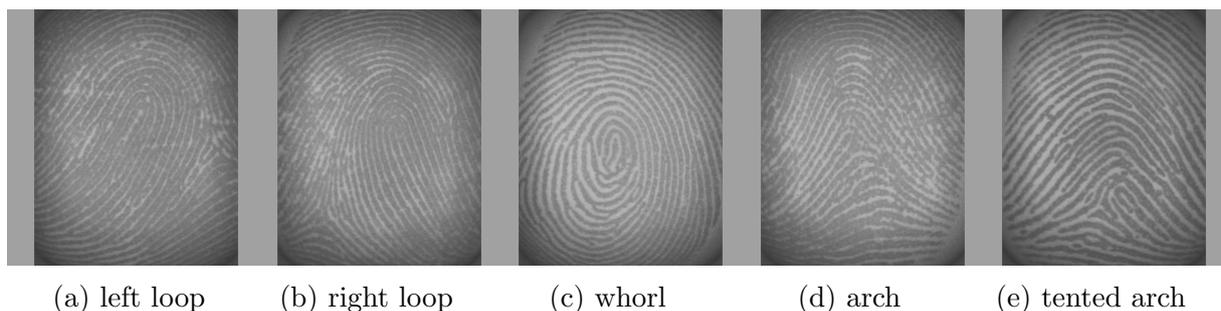
Figura 4.1: Diagrama de blocos do sistema biométrico utilizado na avaliação de desempenho.

4.1.1 Caracterização do sensor

Dentro das três categorias principais de sensores apresentadas na secção 2.6.2 (ópticos, estado sólido e ultrassons), pode-se ainda subdividir as mesmas em: sensores de contacto, em que o dedo permanece imóvel sobre a área de captura; e sensores de varrimento, onde existe uma área de contacto inferior à do dedo a deslocar aquando da captura. Neste projeto recorre-se a um sensor da ADH Technology, o GT-511C3 [1], tratando-se este de um sensor ótico de contacto. Na tabela 4.1 encontra-se um resumo das principais características, evidenciando que dos dois tipos de imagem em tons de cinzento disponível ("*template*" e "*RAW*") se opta pelo primeiro, por este já possuir um pré-processamento da imagem, que deixa apenas a área de interesse da impressão digital. Na figura 4.2, apresentam-se cinco impressões digitais das classes *left loop*, *right loop*, *whorl*, *arch* e *tented arch* que foram obtidas com este sensor.

Característica	Valor
Tipo de sensor	Ótico / contacto
Tipo de comunicação	UART ou USB 1.1
Área efetiva	14×12,5 (mm)
Densidade de pontos	450 DPI
Resolução da imagem RAW	320×240
Resolução da imagem <i>template</i>	256×256

Tabela 4.1: Especificações do sensor GT-511C3 [1].

Figura 4.2: Exemplo de impressões digitais recolhidas através do sensor GT-511C3: (a) *left loop*, (b) *right loop*, (c) *whorl*, (d) *arch* e (e) *tented arch*.

4.1.2 Caracterização da base de dados

Por forma a utilizar as métricas apresentadas no capítulo 2, FMR, FNMR e EER, é necessário dispor-se de uma base de dados de impressões digitais, podendo-se considerar a existência de dois tipos de base de dados. Uma primeira mais genérica, disponibilizada com o objetivo de avaliar o desempenho de algoritmos de reconhecimento, como por exemplo as base de dados da NIST [37] ou as bases de dados criadas durante as competições de reconhecimento de impressões digitais FVC [31][32][33][53]. A outra trata-se de uma base de dados com impressões digitais adquiridas pelo sensor a utilizar no sistema.

Atendendo a que este projeto consiste no desenvolvimento de um sistema biométrico dedicado para um sensor e que os parâmetros do algoritmo de extração de características se encontram otimizados para o mesmo, opta-se por utilizar uma base de dados própria. Na tabela 4.2 é apresentado um resumo da caracterização desta base de dados, enquanto que se apresenta uma descrição mais detalhada no anexo A. Inicialmente, a base de dados teve origem em 72 dedos únicos e em 3 amostras por cada um deles ($N_{amostras\ mesmo\ dedo}$), dando um total de 216 impressões digitais. No entanto, devido à dificuldade em encontrar um ponto de referência nalgumas impressões digitais do tipo *arch*, não foram considerados 7 dedos, resultando numa base

Classe	Número de amostras	Porcentagem
<i>Left loop</i>	72	36,92 %
<i>Right loop</i>	60	30,77 %
<i>Whorl</i>	54	27,69 %
<i>Arch</i>	6	3,08 %
<i>Tented Arch</i>	3	1,54 %
Total	195	100 %

Tabela 4.2: Caracterização da base de dados criada.

de dados com 195 amostras ($N_{amostras}$). Após inspeção visual, verifica-se que apenas no caso de um dedo, que deu origem às três amostras representadas na figura 4.3, é possível visualizar um pequeno desvio na detecção do ponto de referência, sendo este erro introduzido pela baixa qualidade das amostras em questão. No entanto, aquando da execução do algoritmo não é detetada uma influência significativa deste erro, indicando que o algoritmo é robusto a este tipo de variações. As amostras presentes na figura 4.3, permitem ainda observar a rotação existente entre amostras do mesmo dedo, sendo estas resultantes da liberdade dada aos utilizadores no processo de captura.

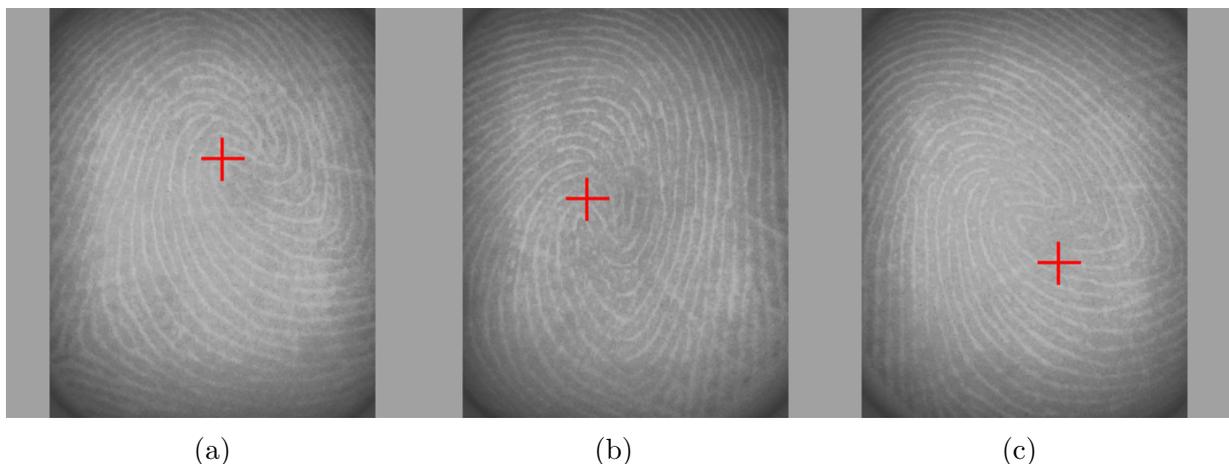


Figura 4.3: Exemplo de três impressões digitais do mesmo dedo, recolhidas através do sensor GT-511C3.

4.2 Estudo e análise em MATLAB

Com os componentes do sistema definidos, pode-se proceder à análise do algoritmo de reconhecimento de impressões digitais, nomeadamente a definição dos parâmetros do mesmo. Assim, tendo em conta a descrição realizada no capítulo 3 e os valores apresentados em [42] e

em [40][47] como ponto de partida, procede-se à variação dos parâmetros de forma individual de modo a analisar a influência dos mesmos nas métricas FMR, FNMR e EER. A medição do desempenho do algoritmo em MATLAB [50], através da base de dados criada, inicia-se com a extração das características das 195 amostras. Para o cálculo do FMR e do FNMR recorre-se, respetivamente, à determinação da distância euclidiana entre amostras de dedos diferentes e entre amostras de dedos iguais, sendo o número de determinações dado pelas equações (4.1) e (4.2), onde C se refere a combinação sem repetição.

$$N_{\text{dedos diferentes}} = C_2^{N_{\text{amostras}}} - N_{\text{amostras}} \quad (4.1)$$

$$N_{\text{dedos iguais}} = C_2^{N_{\text{amostras mesmo dedo}}} \times \frac{N_{\text{amostras}}}{N_{\text{amostras mesmo dedo}}} \quad (4.2)$$

Na tabela 4.3 encontram-se os valores indicados em [42] e em [40][47], juntamente com os valores obtidos empiricamente. Realizando uma comparação entre os valores utilizados como referência e os valores propostos, verifica-se pelos parâmetros B , b , k e f_0 que estes dependem diretamente da resolução das imagens do sensor utilizado. O número de bandas, B , que em [42] toma o valor 5 (para imagens de 508×480 a 500 DPI) ou 7 (para imagens de 832×768 a 500 DPI) dependendo da base de dados utilizada, e a dimensão do setor, b , que utiliza 16 píxeis, são agora ajustados para valores inferiores uma vez que a base de dados é composta por imagens de 256×256 píxeis com uma resolução de 450 DPI. A frequência das riscas também sofre um ajuste devido à menor resolução em DPI das imagens do sensor. O valor da distância entre o ponto de referência e a primeira banda, que não se encontra definido (*n.d.*) em [42], mantém o valor usado em [40][47] uma vez que o aumento do mesmo resulta numa diminuição da área útil para obter as características das impressões digitais; por sua vez, uma diminuição faz com que

Parâmetro	Salil [42]	Carlos[47] e César[40]	Valores propostos
N.º de bandas (B)	5 ou 7	3	3
Dimensão do setor (b)	20	15	14
N.º de setores por banda (k)	16	16	11
Distância entre ponto de referência e a 1.ª banda (gap)	<i>n.d.</i>	1,5	1,5
Ângulo de rotação (ang)	11,25°	7°	27°
Frequência das riscas (f_0)	0,1	0,1	0,135

Tabela 4.3: Parâmetros do algoritmo de reconhecimento de impressões digitais.

os dados presentes na primeira banda não sejam significativos. O ângulo de rotação proposto apresenta uma variação elevada em comparação com os restantes, devido à rotação existente entre amostras do mesmo dedo na base de dados criada. Os 27° de rotação fazem com que os *FingerCode* possuam as rotações: $\{-59,7^\circ; -32,7^\circ; -27^\circ; -5,7^\circ; 0^\circ; 5,7^\circ; 27^\circ; 32,7^\circ; 59,7^\circ\}$.

É também possível observar no anexo C o efeito da variação dos vários parâmetros nas métricas que conduziram aos valores escolhidos. Note-se que numa primeira versão do sistema, apenas no conjunto de características que dava origem ao registo armazenado era introduzida a invariância à rotação, resultando numa degradação das métricas utilizadas na avaliação.

4.2.1 Resultados experimentais

Assumindo a gama de valores de distância euclidiana obtidos como sendo os valores possíveis para o nível de decisão, pode-se então construir o gráfico apresentado na figura 4.4. Neste gráfico apresentam-se o FMR e o FNMR em função do nível de decisão, bem como o EER resultante da interseção das duas curvas. Assumindo uma aplicação que possua como requisito um FMR inferior a 1%, será necessário definir o nível de decisão para 161,5 o que dá origem a um FNMR de 22,05%. Pelos resultados obtidos, pode-se concluir que o algoritmo estudado neste trabalho permite a implementação de um sistema biométrico de reconhecimento de impressões digitais, com valores aceitáveis para os diferentes indicadores de desempenho.

4.3 Estudo e análise em linguagem C

O sistema de reconhecimento de impressões digitais tem como alvo a implementação do algoritmo proposto neste trabalho numa placa de desenvolvimento da Xilinx, a Virtex-7 FPGA VC707 Evaluation Kit. Para tal, recorre-se ao processador *MicroBlaze* [66] e à linguagem C para realizar a primeira implementação do algoritmo de extração de características em *software*. Esta implementação tem como objetivo principal a análise do tempo de execução do algoritmo, por forma a determinar quais as componentes que podem ser melhoradas de forma significativa recorrendo à FPGA, i.e., com a realização de um acelerador em *hardware* dedicado. Na secção 5.1 é realizada uma caracterização mais detalhada dos elementos que constituem a FPGA presente na placa de desenvolvimento.

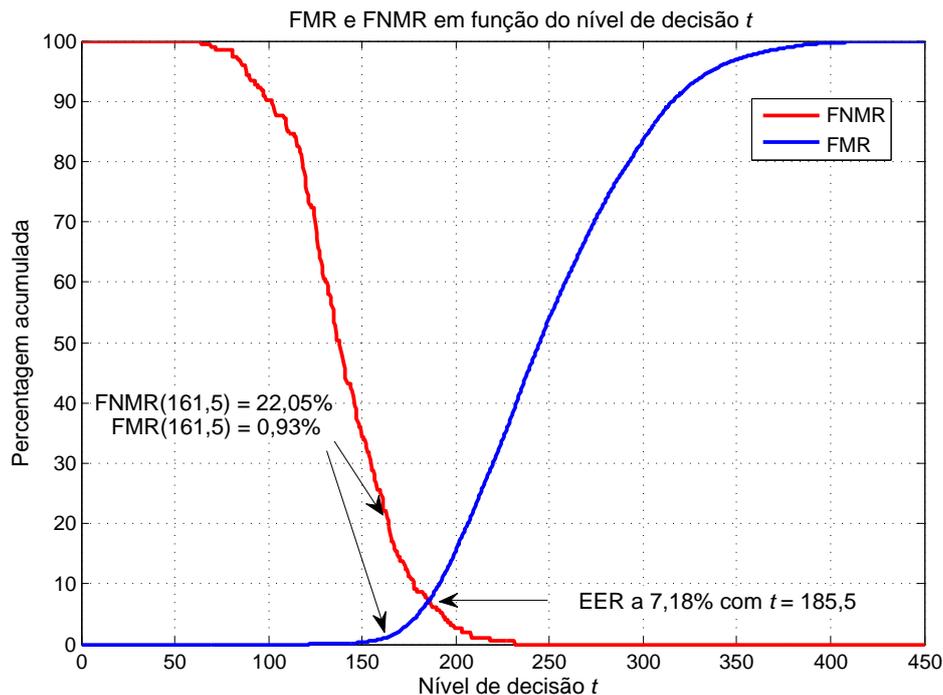


Figura 4.4: FMR e FNMR em função do nível de decisão t .

4.3.1 Arquitetura de teste

Na figura 4.1 encontra-se a arquitetura utilizada para realizar a avaliação do algoritmo em linguagem C, sendo esta composta por um processador *MicroBlaze*, um controlador do protocolo de comunicação série (RS232), um temporizador (*timer*), um controlador de interrupções e um controlador de memória *Double Data Rate* (DDR). O processador *MicroBlaze* possui uma unidade de vírgula flutuante (*Float-Point Unit* - FPU) e duas memórias *cache* (uma de dados e outra de instruções). No dimensionamento da memória *cache* de dados não foi considerada a filtragem de Gabor, devido à dimensão original das janelas (33×33) necessitarem de uma memória de 32 kB para armazenar temporariamente as linhas processadas. Assim, para disponibilizar espaço suficiente para o armazenamento temporário das linhas nos restantes processos de filtragem, foi definida uma *cache* de 8 kB.

4.3.2 Resultados experimentais

A execução do algoritmo no *MicroBlaze* recorrendo à ferramenta *Xilinx Software Development Kit* (XSDK)[63] permite a opção *profile* que, recorrendo a um contador e a um controlador de interrupções, faz com que seja retornado o número de vezes que o processador executou de-

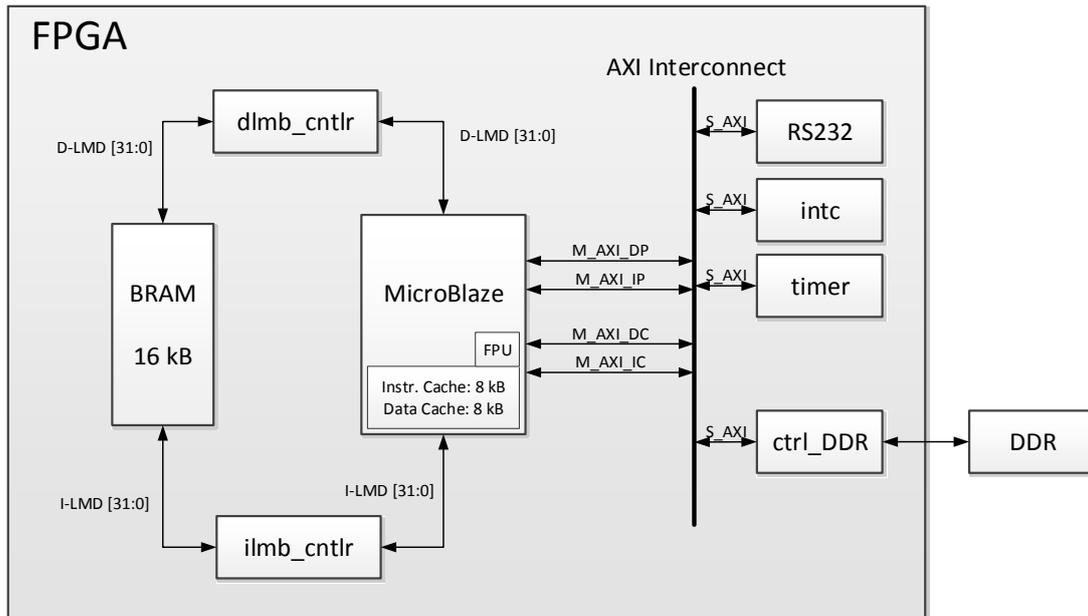


Figura 4.5: Arquitetura utilizada na avaliação do sistema biométrico.

terminada operação. Para tal, é definida uma frequência de amostragem e uma posição de memória onde são armazenadas as amostras, permitindo à ferramenta criar um relatório com a percentagem de utilização das funções presentes no código.

Na figura 4.6 encontra-se um resumo do relatório gerado pelo XSDK, sendo este composto pela percentagem de execução das principais operações do algoritmo. Pelos resultados obtidos pode-se verificar que o processador despende 67,11% do tempo de execução na aplicação dos filtros de Gabor. Note-se que esta percentagem resulta da aplicação de 8 filtros a duas dimensões sobre 3 imagens (a original e duas rodadas), para cada execução do algoritmo de extração de características. Destacam-se ainda as operações com vírgula flutuante que ocupam mais de 20% do tempo de execução. Pode-se então concluir que, atendendo ao seu peso computacional no algoritmo, a filtragem de Gabor é a componente candidata a ser implementada em hardware.

4.4 Otimização dos filtros de Gabor

A aplicação de um filtro a duas dimensões tem por base o varrimento de uma janela ao longo de uma imagem, começando-se por centrar uma janela com dimensão $L_x \times L_y$ no primeiro píxel da primeira linha e por efetuar o somatório da multiplicação ponto a ponto da sobreposição entre a janela e os píxeis da imagem. Após o cálculo do primeiro píxel, desloca-se a janela na horizontal, de píxel em píxel, e efetuam-se os mesmos cálculos até se atingir o fim da linha. O

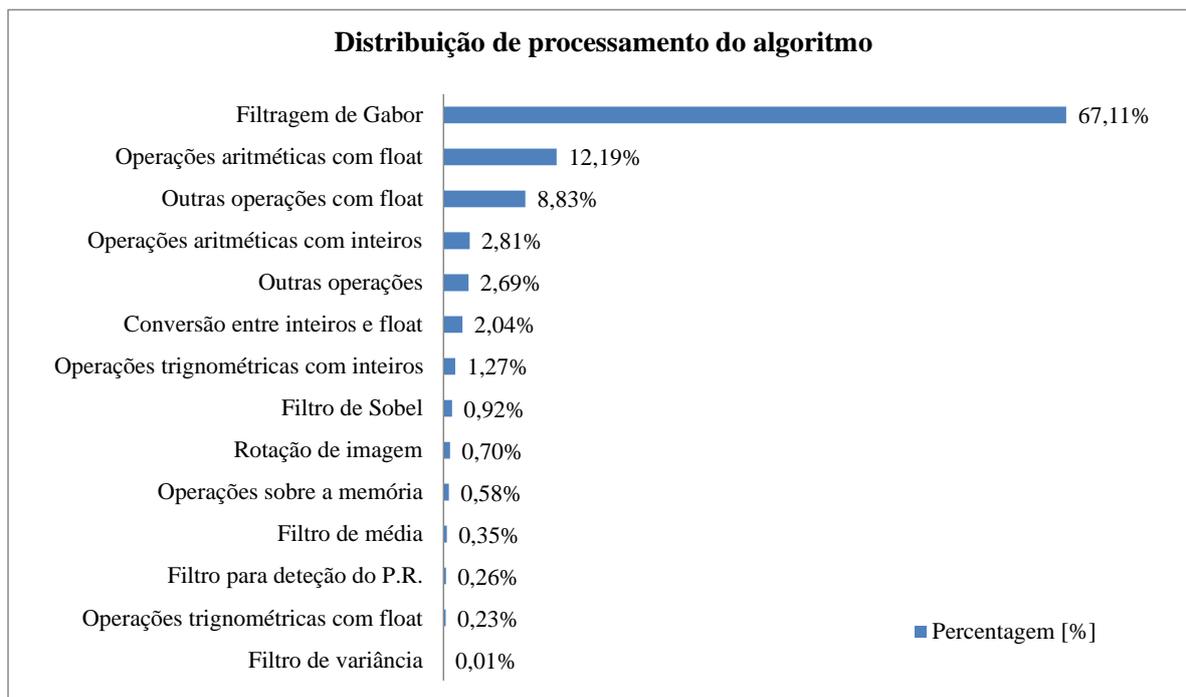
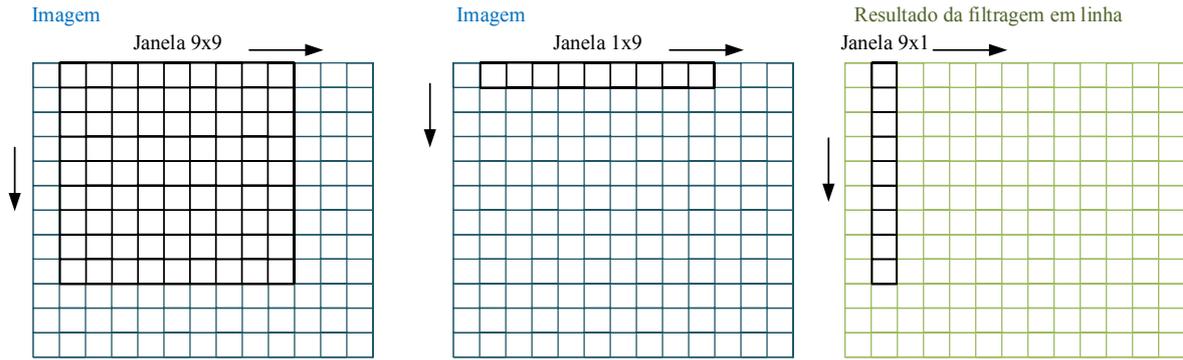


Figura 4.6: Distribuição de processamento do algoritmo, obtido através da opção *profile* na ferramenta XDSK.

processo repete-se nas linhas seguintes e até chegar ao último píxel da última linha da imagem. Esta abordagem resulta em $L_x \times L_y$ multiplicações por cada píxel da imagem. Outra opção para a aplicação de um filtro a duas dimensões passa por separar a janela do mesmo numa linha (com dimensão L_x) e uma coluna (com dimensão L_y), aplicando primeiro o filtro em linha à imagem e sobre o resultado é aplicado o filtro em coluna. Nesta abordagem reduz-se o número de multiplicações de $L_x \times L_y$ para $L_x + L_y$ multiplicações (por cada píxel da imagem).

Na figura 4.7 apresenta-se um exemplo de aplicação para cada uma das duas abordagens considerando uma janela de 9x9, ou seja, com a mesma dimensão das janelas presentes no banco de filtros de Gabor. No primeiro caso (figura 4.7a), significa que sobre cada píxel são realizadas 81 multiplicações, enquanto que no segundo caso (figura 4.7b) são apenas realizadas 18 multiplicações.

Em [5], é proposto um método de separação de filtro de Gabor a duas dimensões em dois filtros, aplicando o método descrito anteriormente para as orientações de 0° e 90° , enquanto que para as restantes orientações é realizado um ajuste nos ângulos de modo a que seja possível percorrer a imagem em diagonais. Na figura 4.8 ilustra-se este processo, em que os ângulos definidos $\{0^\circ; 22,5^\circ; 45^\circ; 67,5^\circ; 90^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$ são aproximados, respetivamente, para $\{0^\circ; 26,6^\circ; 45^\circ; 63,4^\circ; 90^\circ; 116,6^\circ; 135^\circ; 153,4^\circ\}$. Os ângulos resultam do método utilizado



(a) Com janela 9x9.

(b) Com janela separada.

Figura 4.7: Filtragem de Gabor com: (a) janela 9x9, (b) janela separada em uma linha e uma coluna.

na seleção dos píxeis a processar, dando origem a uma variação nos ângulos definidos. Este método implica uma degradação das métricas [5]. Como tal, foi proposta em [4] uma solução que permite aplicar os filtros segundo qualquer direção realizando reamostragem (decimação e interpolação) da imagem. O processo de reamostragem da imagem para as orientações $\{22,5^\circ; 67,5^\circ; 112,5^\circ; 157,5^\circ\}$ implica processamento extra, não sendo adequado para implementação em *hardware*, porque implica gasto excessivo de recursos sem vantagens apreciáveis. Tanto em [5] como em [4] são propostas soluções específicas para os filtros de Gabor através de um ajuste dos seus parâmetros. No entanto, existe a possibilidade de se utilizar uma solução mais eficiente que funciona para qualquer conjunto de parâmetros, através da decomposição em valores singulares (*Singular Value Decomposition* - SVD)[7][49] da máscara dos filtros.

4.4.1 Decomposição em valores singulares

A janela de um filtro a duas dimensões diz-se separável se for possível exprimir a mesma a partir do produto de dois vetores a uma dimensão. Segundo o método de decomposição em valores singulares, dada uma matriz $A \in \mathfrak{R}$ com dimensão $P \times Q$ é possível decompô-la no produto de três matrizes $A = USV^T$, onde U e V são matrizes unitárias e S é uma matriz retangular diagonal com valores diagonais não negativos. A matriz U tem dimensão $P \times P$ e as colunas da mesma são os vetores próprios de AA^T , a matriz V tem dimensão $Q \times Q$ e as suas colunas são os vetores ortogonais de $A^T A$. A matriz S , com dimensão $P \times Q$, possui os valores singulares de A .

Para realizar a separação dos filtros de Gabor pela decomposição em valores singulares

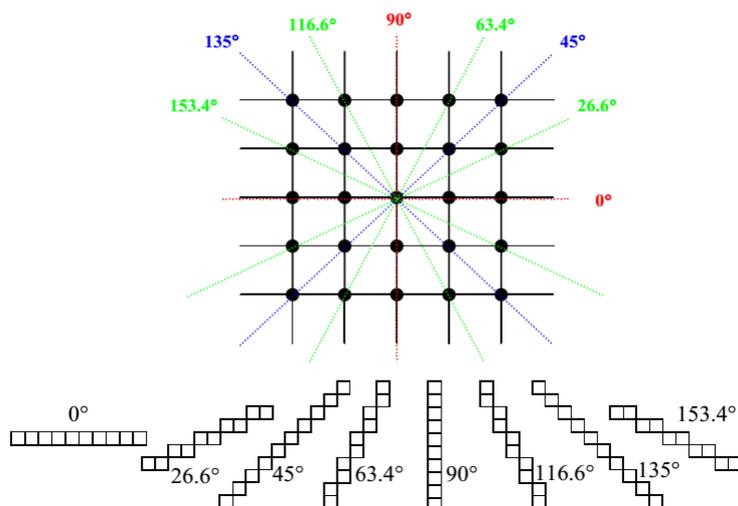


Figura 4.8: Aplicação dos filtros de Gabor separáveis com o conjunto de ângulos proposto em [5].

consideram-se somente os valores mais significativos da diagonal da matriz S segundo a condição $\text{diag}(S(> 1e^{-2}))$, ou seja, os valores singulares acima de 0,01. Neste caso em particular resulta na utilização de um valor significativo para as orientações de 0° e 90° , e dois valores significativos para as restantes orientações. Assim a decomposição dos filtros de Gabor é dada por:

$$h_{1(ns \times Q)} = \left(V_{(Q \times ns)} \times \sqrt{\text{diag}(S_{(ns \times ns)})} \right)^T, \quad (4.3)$$

e

$$h_{2(P \times ns)} = U_{(P \times ns)} \times \sqrt{\text{diag}(S_{(ns \times ns)})}, \quad (4.4)$$

onde ns indica o número de valores significativos da matriz S e h_1 e h_2 são, respetivamente, as respostas impulsionais dos filtros a uma dimensão em linha e em coluna. Considerando que ns é igual a 1 ou a 2, dependendo da orientação escolhida, e que a aplicação em linha e em coluna pode ser definida pela ordem de processamento dos píxeis da imagem. Na figura 4.9 encontra-se um diagrama de blocos representativo do processo de filtragem. As respostas impulsionais h_1 e h_2 , podem ser vistas como filtros do tipo *Finite Impulse Response* (FIR) [39]. No anexo B encontra-se uma descrição mais detalhada da decomposição SVD dos filtros de Gabor.

4.4.2 Resultados experimentais

Em [42] é proposta a utilização de filtros com janelas de 33×33 . No entanto, para o sistema biométrico estudado neste trabalho verifica-se pela figura 4.10 que a janela pode ser reduzida

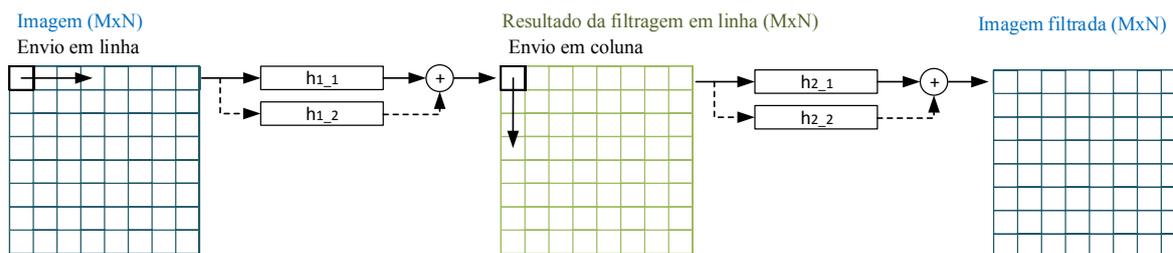


Figura 4.9: Aplicação dos filtros de Gabor separáveis por SVD.

para 9×9 sem alterar significativamente o desempenho do mesmo. Desta forma, é possível reduzir o número de multiplicações da operação de filtragem de 66 para 18 (ou de 1089 para 81, sem filtros separáveis). Note-se que para as orientações $\{22,5^\circ; 45^\circ; 67,5^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$, com $ns = 2$, duplica-se o número de multiplicações.

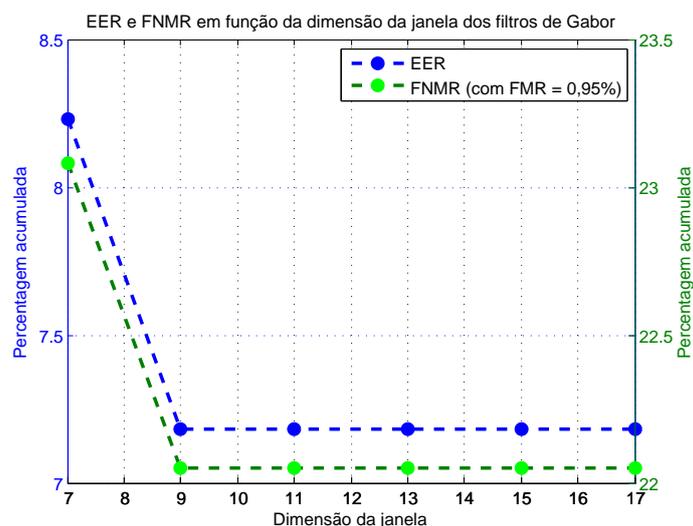


Figura 4.10: EER e FNMR em função da dimensão da janela dos filtros de Gabor.

Em termos de aplicação em *hardware* existe a necessidade de verificar se é necessário recorrer a vírgula flutuante ou se é possível utilizar uma representação com vírgula fixa, uma vez que a primeira solução geralmente resulta numa maior ocupação de recursos. Analisando os resultados da filtragem de Gabor para a base de dados de teste, pode-se afirmar que pelos valores obtidos são apenas necessários 12 bits (com sinal) para representar a parte inteira. Quanto à parte fraccionária, pelo gráfico presente na figura 4.11, verifica-se que o EER e o FNMR se mantêm com truncatura até 10 bits, aumentando o FNMR com a diminuição da precisão.

Com a redução da dimensão das janelas dos filtros de Gabor definida para 9×9 , a criação de filtros separáveis e a utilização de 22 bits em vírgula fixa para realizar a filtragem, pode-se

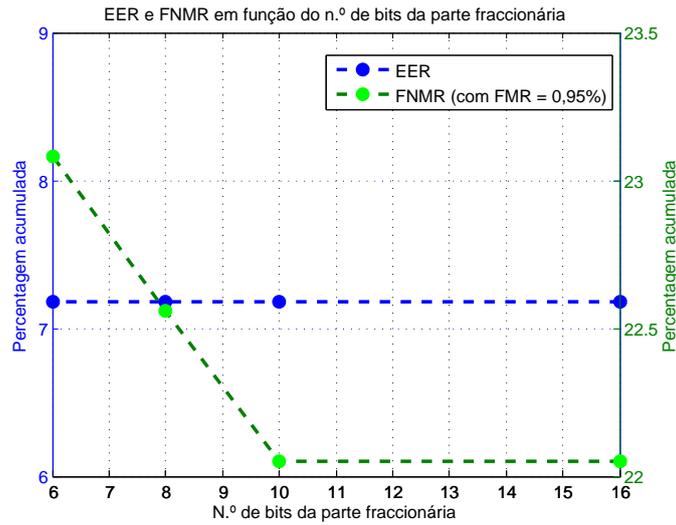


Figura 4.11: EER e FNMR em função do número de bits da parte fracionária.

visualizar na figura 4.12 o efeito destas alterações nos indicadores FMR e FNMR. Considerando os valores de referência definidos na figura 4.4, EER a 7,18% e FNMR a 22,05% quando FMR é 0,93%, verifica-se que estes se mantêm com a exceção do FMR que passou a ser considerado o valor 0,95%. As curvas de FMR e FNMR sofrem uma deslocação, resultando numa diminuição do nível de decisão, para o mesmo EER.

Analisando aos resultados obtidos, pode-se então considerar uma dimensão de 9×9 para as janelas dos filtros de Gabor, bem como a conversão dos dados para 22 bits em vírgula fixa, na implementação da filtragem de Gabor em *hardware*.

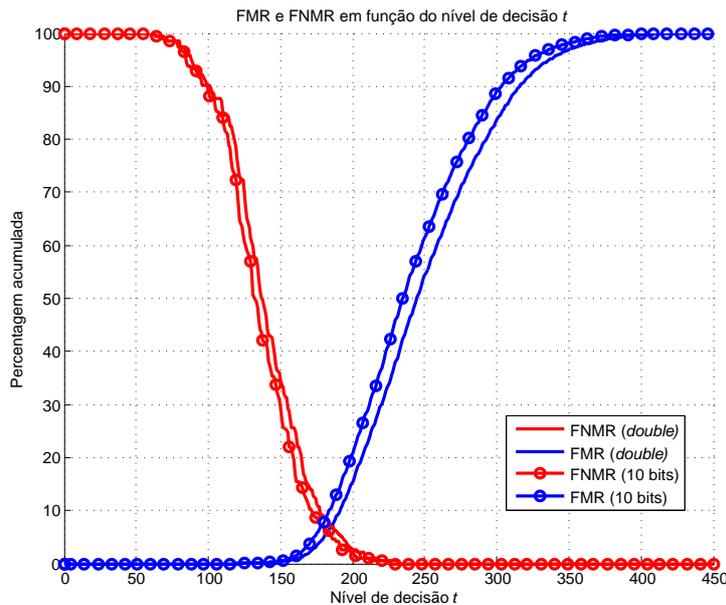


Figura 4.12: FMR e FNMR em função do nível de decisão t com precisão *double* e com 10 bits na parte fracionária.

Capítulo 5

Implementação do sistema biométrico em *hardware*

O estudo realizado no capítulo 4 possibilita o desenvolvimento de um periférico que permita melhorar o desempenho da implementação realizada sobre o processador *MicroBlaze* para a realização da filtragem de Gabor. Para tal, neste capítulo, é definido um acelerador de *hardware* para a realização desta operação com uma interface adequada para utilização conjunta com o processador *MicroBlaze*.

Este capítulo inicia-se na secção 5.1 com a caracterização da FPGA presente na placa de desenvolvimento Virtex-7 FPGA VC707 Evaluation Kit, a Virtex-7 XC7VX485T-2FFG1761C [60], que foi utilizada para realizar um protótipo do sistema biométrico proposto. Na secção 5.2 é realizada a descrição da arquitetura do sistema biométrico com aceleração por *hardware*. Para além da descrição do diagrama de blocos que representa este sistema, na secção 5.3 são ainda apresentados a arquitetura do acelerador proposto e os vários componentes que o constituem. Na secção 5.4 é realizada uma análise ao desempenho do sistema, comparando as soluções com e sem recurso à aceleração por *hardware*.

5.1 Caracterização da FPGA

As FPGAs desenvolvidas pela Xilinx apresentam características diferentes consoante a sua série, encontrando-se a Virtex-7 XC7VX485T-2FFG1761C associada à série 7 [60]. Os blocos de lógica reconfigurável presentes nas FPGAs da Xilinx são denominados por *Configurable Logic Block* (CLB) [59], sendo estes constituídos por *slices*. Cada FPGA da série 7 inclui duas

slices por CLB, em que cada uma delas tem quatro *Lookup Tables* (LUTs) de seis entradas e oito *flip-flops*. Algumas destas LUTs podem ainda ser utilizadas como: *route-thrus* para aceder a componentes internos; memória distribuída ou como *Shift Register Lookup table* (SRL).

Para além de CLBs, as FPGAs da série 7 incluem outros blocos, tais como Block RAMs (BRAMs) e blocos *Digital Signal Processing* (DSP). As BRAMs, RAMB36E1 [61], são memórias de duplo porto com 36 kb de capacidade, podendo ser utilizadas como dois blocos independentes de 18 kb [60]. Cada DSP, com a designação DSP48E1 [58], inclui um multiplicador de 25×18 bits, um acumulador e um pré-somador. Os DSPs podem ser configurados em cascata para realizar funções matemáticas, filtros e operações aritméticas complexas sem recorrer a blocos de lógica reconfigurável.

Na tabela 5.1 encontra-se um resumo das principais características da FPGA Virtex-7 XC7VX485T-2FFG1761C.

Recursos CLB					BRAMs (36 kb)	DSP48E1
<i>Slices</i>	LUTs de 6 entradas	Memória distribuída [kb]	SRLs [kb]	<i>Flip-Flops</i>		
75900	303600	8175	4088	607200	1030	2800

Tabela 5.1: Principais características da Virtex-7 XC7VX485T-2FFG1761C [60].

5.2 Arquitetura do sistema biométrico com aceleração por hardware

Na figura 5.1 apresenta-se a arquitetura do sistema biométrico desenvolvido, sendo este constituído pelos seguintes periféricos: um processador *MicroBlaze*, responsável pela resolução do algoritmo de reconhecimento de impressões digitais e pelo controlo do sistema; um controlador do protocolo de comunicação série (RS232); um temporizador (*timer*), um controlador de interrupções; o acelerador proposto (*filter_Gabor*) e dois controladores de memória (*bram_ctrl0* e *bram_ctrl1*), em que cada controlador se encontra associado a uma memória de 64 kB. Estas memórias são de duplo porto, permitindo assim o acesso tanto por parte do processador como do periférico *filter_Gabor*.

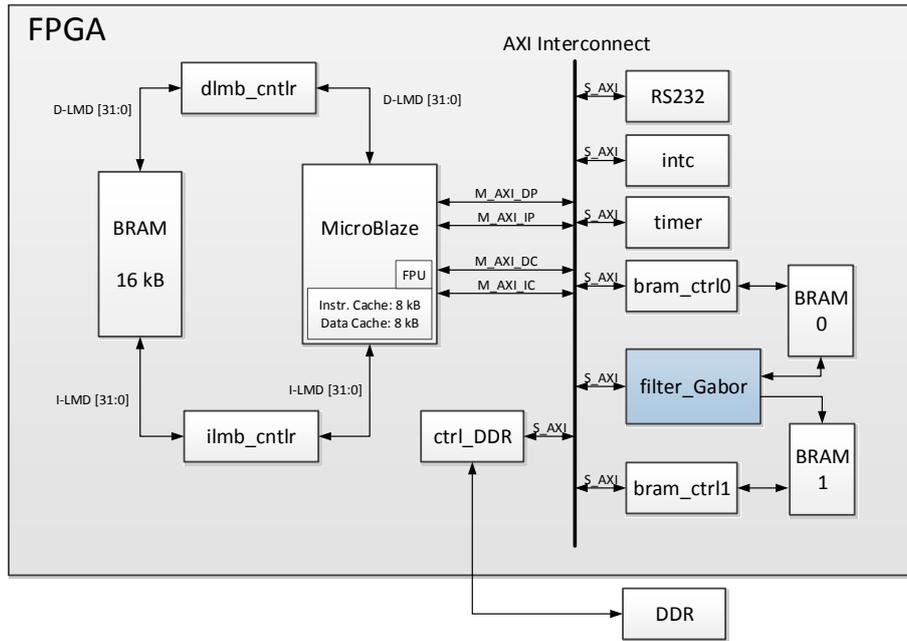


Figura 5.1: Arquitetura do sistema biométrico.

5.3 Arquitetura do periférico *filter_Gabor*

O periférico *filter_Gabor*, cujo diagrama de blocos é apresentado na figura 5.2, é composto por três componentes principais: *AXI Slave IPIF*, *User Logic* e *main*.

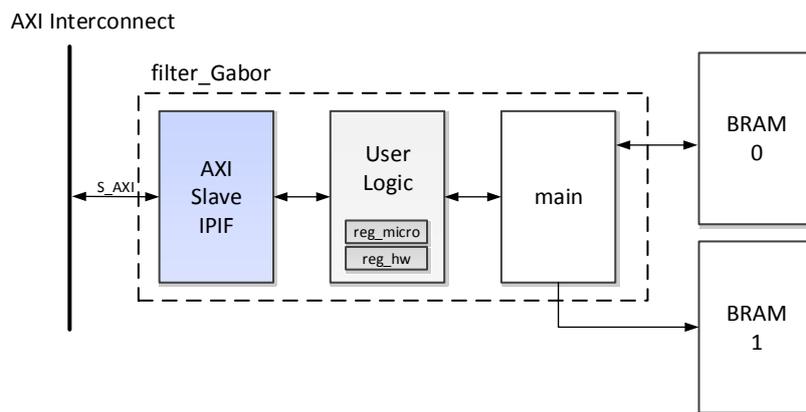


Figura 5.2: Arquitetura do periférico *filter_Gabor*.

5.3.1 *AXI Slave IPIF* e *User Logic*

O componente *AXI Slave IPIF*, gerado através do *software* Xilinx EDK [63], efetua a conexão do periférico à interface *Advanced eXtensible Interface (AXI) Interconnect* do processador

MicroBlaze usando uma ligação AXI Slave, permitindo desta forma o acesso do *MicroBlaze* ao periférico através de dois registos partilhados.

O componente *User Logic* realiza a gestão das leituras e escritas nos registos *reg_micro* e *reg_hw*, que servem para permitir que tanto o periférico como o *MicroBlaze* possam efetuar leituras dos registos, sem deixar de assegurar que apenas o *reg_micro* pode ser escrito pelo *MicroBlaze* e que só o periférico pode escrever no *reg_hw*.

A estrutura dos registos *reg_micro* e *reg_hw*, encontra-se descrita na figura 5.3. O registo *reg_micro*, serve para controlar a operação do periférico e é comandado pelo *MicroBlaze*, que o afeta com o valor lógico '1' no bit de menor peso quando se pretende dar início à filtragem (*start*). O segundo bit é utilizado para reiniciar o sistema (*reset*), enquanto que os três bits seguintes são utilizados para indicar ao periférico qual dos oito filtros de Gabor deve ser utilizado na filtragem (*SEL_Gabor_Filter*).

O registo *reg_hw* contém o registo de estado do periférico e serve para informar o *MicroBlaze* do seu estado atual de operação. Através do bit de menor peso indica ao *MicroBlaze* que terminou a filtragem (*done*), quando este tiver o valor lógico '1', e pelos quatro bits seguintes indica em que estado da máquina de estados interna (ver figura 5.6) se encontra o componente responsável pela filtragem (*reg_pstate*).



Figura 5.3: Configuração dos registos partilhados.

5.3.2 *Main*

O componente *main* é responsável por executar a filtragem de Gabor, realizando a gestão entre os componentes que o constituem e os registos *reg_micro* e *reg_hw*. É composto pelo componente *core*, que realiza a filtragem de Gabor, e pelas memórias *mem_aux1* e *mem_aux2*, que são utilizadas para guardar os resultados intermédios do processo de filtragem.

O diagrama *Register-Transfer Level* (RTL) do componente *main* é apresentado na figura 5.4. Este componente recebe como sinais de entrada *DO_B_mem_in*, com origem na memória BRAM 0, os dados armazenados em *reg_micro* e o sinal *CLK*, que irá definir a frequência de operação dos componentes de *main*. À saída do componente *main*, os sinais *data_out_f*, *addr_out_f* e *data_ready_f* são utilizados, respetivamente, para enviar os resultados da filtragem, os endereços associados a esses resultados e a indicação para ativar a memória BRAM 1.

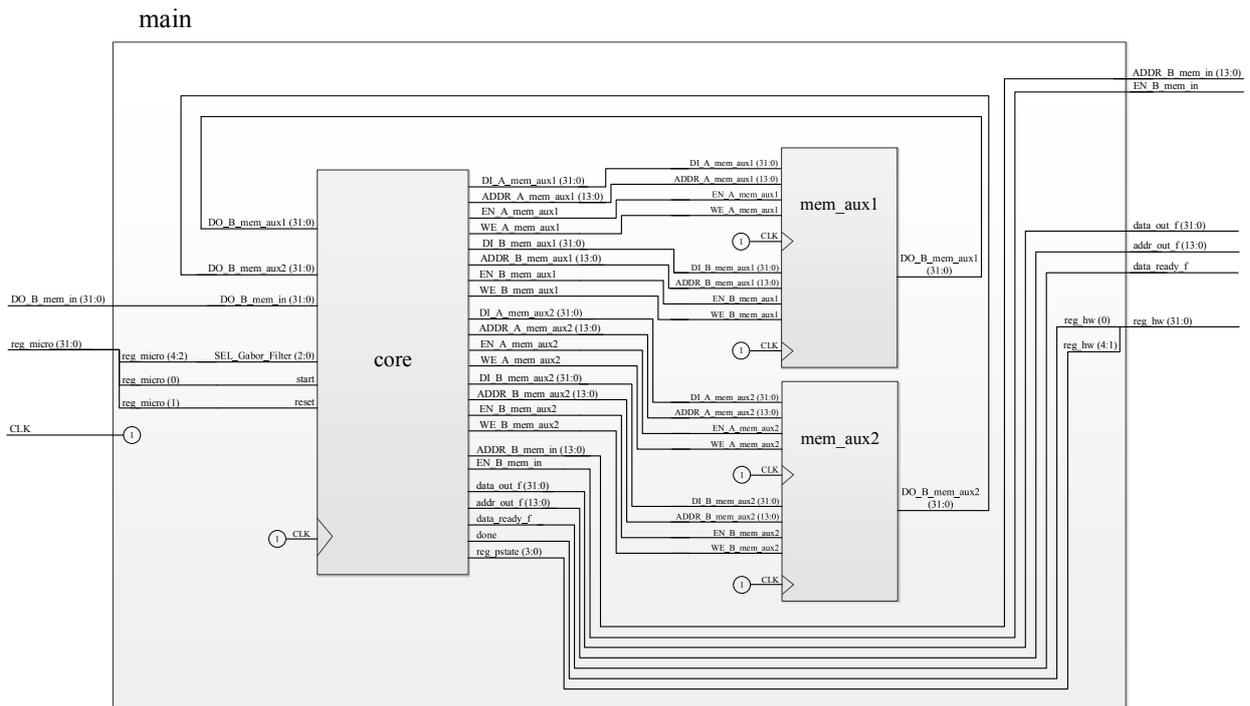


Figura 5.4: Diagrama RTL do componente *main*.

A figura 5.5 mostra o diagrama RTL parcial do componente *core*, sendo este responsável por recolher os dados da memória BRAM 0 (ver figura 5.1), convertê-los de vírgula flutuante para vírgula fixa (*Float2Fixed*), processá-los e enviar os resultados para a BRAM 1, já convertidos em vírgula flutuante (*Fixed2Float*). O processamento sobre os dados consiste em realizar uma filtragem em linha, guardando os resultados nas memórias *mem_aux1* e *mem_aux2*, para depois realizar a filtragem em coluna, a fim de se obter a filtragem em duas dimensões. No final do processo de filtragem em coluna, e antes de se efetuar a conversão para vírgula flutuante, é realizada a combinação dos resultados no bloco *Res_sel*, consoante o tipo de filtragem de Gabor seleccionada.

Na figura 5.6 apresenta-se o diagrama da máquina de estados presente no componente *core*,

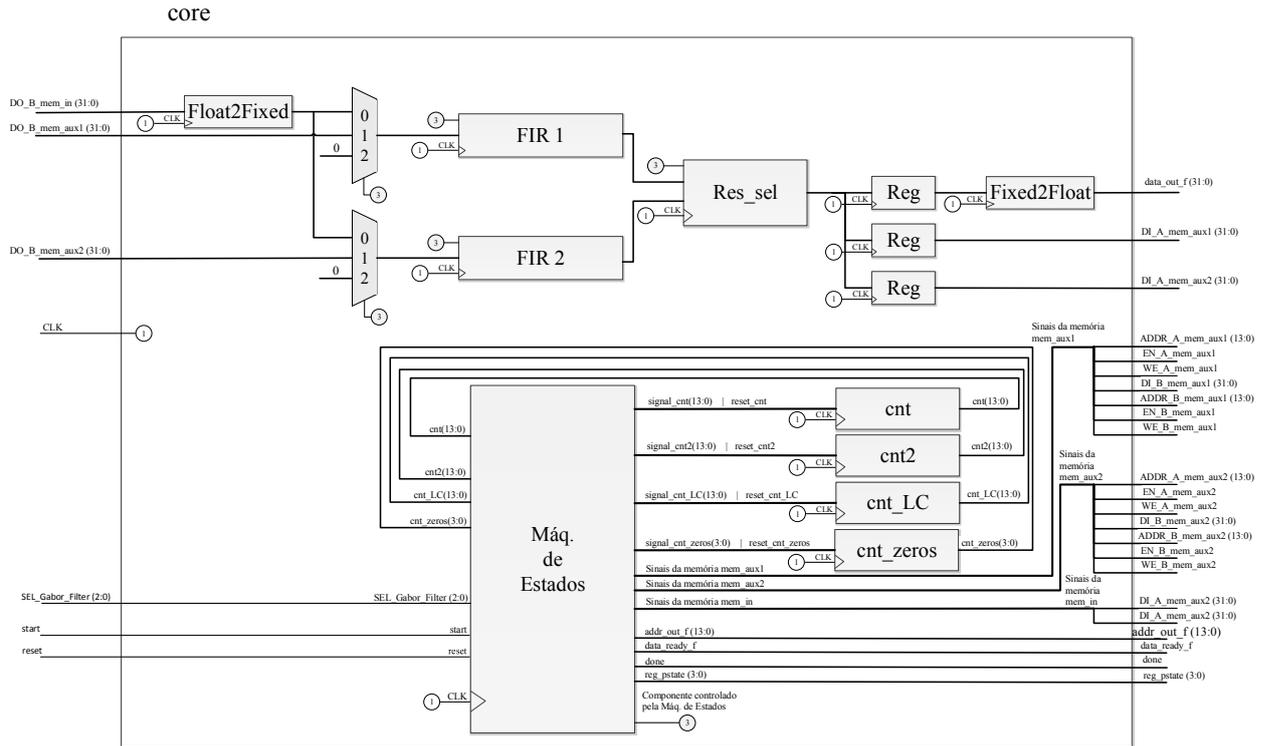


Figura 5.5: Diagrama RTL do componente *core*.

podendo-se dividir a mesma em duas fases: filtragem em linha e filtragem em coluna. No primeiro estado (estado 0) o sistema encontra-se à espera do sinal *start* para dar início ao processo de filtragem, mudando para o segundo estado (estado 1) quando o sinal *start* toma o valor lógico '1'. O estado 1 é utilizado para ativar os filtros FIR e o contador *cnt_LC* que regista o número de elementos que foram enviados para os filtros FIR, de modo a identificar-se quando terminam os dados de uma linha e de uma coluna. No estado 2 é dado início à filtragem em linha, enviando-se os dados para os módulos FIR. Os contadores *cnt* e *cnt_LC* são atrasados, para identificar o momento em que terminam os dados ou quando termina uma linha. Quando este último caso ocorre, existe uma transição para o estado 3 onde são enviados valores a zero para os filtros, de modo a remover os valores existentes nos módulos FIR e com isso iniciar-se uma nova filtragem. O número de valores a zero enviados é determinado pela ordem do filtro. Admitindo um filtro de ordem N , é necessário enviar $N-1$ valores a zero. Neste caso, para filtros de ordem 9 é necessário enviar 8 valores a zero. O estado 4 existe somente para garantir o envio do último valor a filtrar, transitando a máquina de estados de imediato para o estado seguinte. O estado 5 avalia o sinal *cnt_delay* para garantir que os resultados da filtragem em linha se encontram armazenados. Este sinal resulta da introdução de um atraso no contador *cnt*, de modo a que este esteja em sincronismo com os resultados à entrada das memórias *mem_aux1*

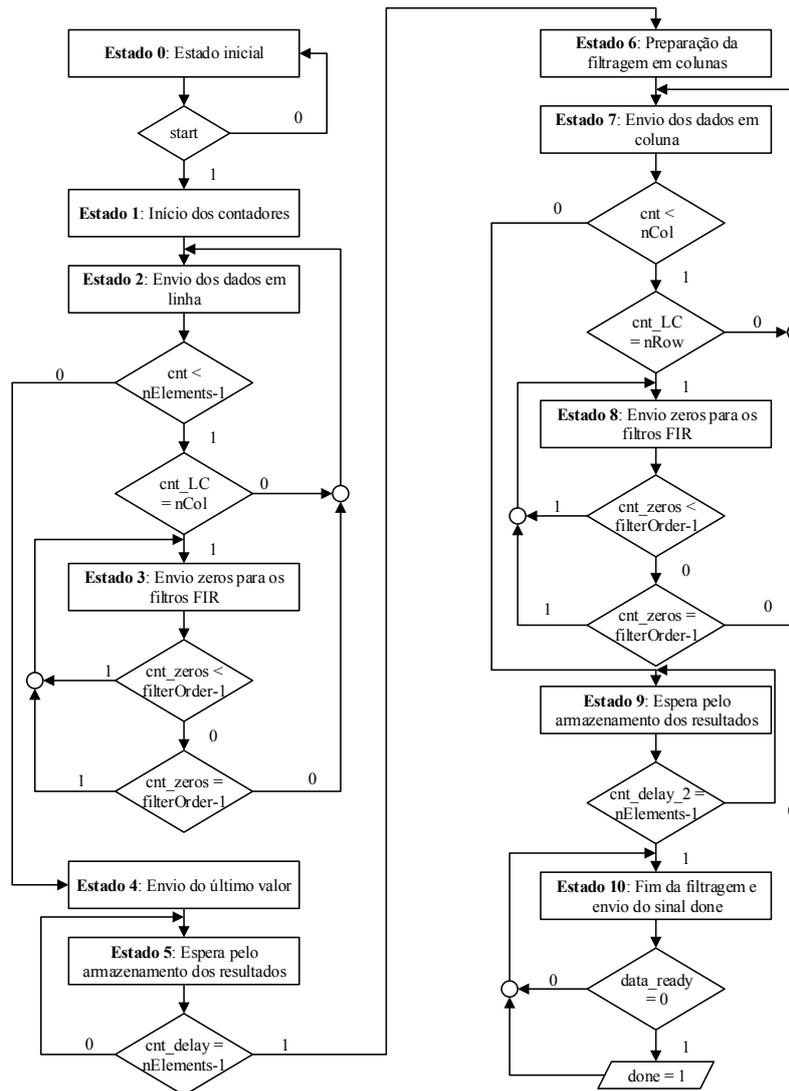


Figura 5.6: Máquina de estados do componente *core*.

e *mem_aux2*.

A filtragem em coluna inicia-se no estado 6 com a preparação dos contadores e dos restantes sinais de controlo, seguindo-se o envio dos dados em coluna para os filtros FIR no estado 7. Tal como na fase de filtragem em linha, é necessário avaliar quando termina uma coluna para se colocar a zero as amostras anteriores dos filtros FIR (estado 8). No estado 9, é novamente avaliado um sinal que tem origem no contador usado para enviar os dados para os filtros FIR, sendo este atrasado de modo a garantir o sincronismo com os dados à entrada da memória BRAM 1 para ser possível avaliar a conclusão do processo de filtragem. No estado 10, é ativado o sinal de *done* para indicar ao processador o fim da filtragem, ficando o sistema neste estado até que o processador envie um sinal de *reset* para voltar ao estado inicial.

5.3.2.1 IP LogiCOREs

No capítulo 4, ao realizar-se o estudo da filtragem de Gabor, verificou-se que seria possível recorrer a filtros FIR para realizar este processo. Estudou-se ainda o efeito da redução dos dados no desempenho do sistema biométrico, chegando-se assim a 12 bits para a parte inteira e 10 bits para a parte fracionária.

Deste modo, os componentes que realizam a conversão entre vírgula fixa e vírgula flutuante, bem como os que aplicam a filtragem do tipo FIR, são criados, respetivamente, através dos IP LogiCOREs da Xilinx: *Floating-Point Operator v5.0* [65] e *FIR Compiler v5.0* [64]. Na tabela 5.2 encontra-se um resumo da configuração dos componentes *Float2Fixed* e *Fixed2Float* apresentados anteriormente. Note-se que a latência utilizada é o valor máximo permitido, e o valor que permite uma frequência de operação mais elevada [65].

Componentes <i>Float2Fixed</i> e <i>Fixed2Float</i>	
N.º de bits em vírgula fixa	22
N.º de bits em vírgula flutuante	32
Latência (n.º de ciclos de relógio)	6
Ciclos por operação	1

Tabela 5.2: Resumo da configuração dos IP LogiCOREs *Float2Fixed* e *Fixed2Float*.

Na tabela 5.3 pode-se observar a configuração definida no componente FIR. Este componente encontra-se configurado com 8 conjuntos diferentes de coeficientes, cada um deles com 9 elementos. O conjunto de coeficientes utilizado é definido com base na filtragem de Gabor selecionada. Os 25 bits de saída são definidos pelo processo de configuração quando é imposto 10 bits de parte fracionária, descartando-se os 3 bits adicionais após a aquisição dos resultados.

Componente FIR	
N.º de bits à entrada	22 (10 de parte fracionária)
N.º de bits à saída (truncatura dos bits menos significativos)	25 (10 de parte fracionária)
N.º de conjuntos de coeficientes	8
N.º de coeficientes	9 (não simétricos)
Precisão dos coeficientes	18 (16 de parte fracionária)
Latência (n.º de ciclos de relógio)	15
Ciclos por amostra	1

Tabela 5.3: Resumo da configuração do IP LogiCORE FIR.

5.4 Resultados experimentais

5.4.1 Ocupação de recursos

Na tabela 5.4 apresenta-se a ocupação de recursos da arquitetura desenvolvida, obtida através da ferramenta *Integrated Software Environment (ISE) Project Navigator* [63]. Analisando a tabela, verifica-se que são utilizadas 64 memórias BRAM de 36 kb, em que: i) metade das mesmas são utilizadas na implementação dos módulos de memória BRAM 0 e BRAM 1; ii) o periférico *filter_Gabor* necessita de 22 memórias; iii) as restantes 10 memórias são utilizadas pelo processador *MicroBlaze*. Em termos de DSPs, o periférico *filter_Gabor* utiliza 18 blocos para realizar as operações de multiplicação, enquanto que o *MicroBlaze* utiliza 5 blocos.

Recursos	Ocupados	Disponíveis	Percentagem de ocupação
N.º de <i>slices</i>	4576	75900	6%
N.º de LUTs de 6 entradas	10798		
como lógica	9386	303600	3%
como memória	1065	130800	1%
de duplo porto	476	-	-
SRL	589	-	-
como <i>route-thrus</i>	347	-	-
N.º de <i>Flip-Flops</i>	9377	607200	1%
N.º de BRAMs de 36 kb	64		
<i>filter_Gabor</i>	22	1030	2%
BRAM 0	16	1030	2%
BRAM 1	16	1030	2%
<i>MicroBlaze</i>	10	1030	1%
N.º de DSP48E1	23		
<i>filter_Gabor</i>	18	2800	1%
<i>MicroBlaze</i>	5	2800	<1%

Tabela 5.4: Ocupação de recursos da arquitetura.

Na tabela 5.5 apresenta-se a ocupação de recursos do periférico *filter_Gabor*, para um processo de síntese com um perfil equilibrado e no estado de implementação *Placed and Routed*, onde se destacam o número de BRAMs e DSPs. As 22 BRAMs resultam da utilização de duas memórias auxiliares com $127 \times 127 \times 22$ bits. Uma vez que os coeficientes utilizados não são simétricos, cada filtro FIR de ordem 9 necessita de 9 blocos DSP48E1.

O caminho crítico do periférico *filter_Gabor* encontra-se no acesso às memórias auxiliares,

limitando a frequência máxima de operação a 265,9 MHz. O processo de filtragem, desde o sinal *start* até ao sinal *done*, demora 34336 ciclos de relógio. Admitindo a frequência máxima de operação, este periférico é capaz de processar aproximadamente 7746 imagens por segundo. Sabendo que no processo de extração de características é aplicada a filtragem de Gabor 24 vezes por amostra, isto significa que este processo pode ser realizado em aproximadamente 3 ms, não comprometendo o desempenho do sistema numa aplicação em tempo real.

Recursos	Ocupados	Disponíveis	Percentagem de ocupação
N.º de <i>slices</i>	477	75900	1%
N.º de LUTs de 6 entradas	1158		
como lógica	1007	303600	1%
como memória (SRL)	129	130800	1%
como <i>route-thrus</i>	22	-	-
N.º de <i>Flip-Flops</i>	1231	607200	1%
N.º de BRAMs de 36 kb	22	1030	2%
N.º de DSP48E1	18	2800	1%

Tabela 5.5: Ocupação de recursos do periférico *filter_Gabor*.

Os valores apresentados, principalmente em termos de lógica reconfigurável, permitem a realização do sistema proposto em FPGAs de gamas inferiores.

5.4.2 Avaliação de desempenho

Na avaliação de desempenho do sistema biométrico, para além das métricas consideradas no capítulo 4 (EER, FMR e FNMR), é ainda analisado o erro médio quadrático (*Mean Squared Error* - MSE) e a relação sinal ruído (*Signal-to-Noise Ratio* - SNR) entre as imagens resultantes da filtragem de Gabor, com e sem aceleração por *hardware*. Considerando duas imagens I_1 e I_2 com resolução $M \times N$, o MSE entre elas é dado por:

$$\text{MSE}(I_1, I_2) = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (I_1(x, y) - I_2(x, y))^2, \quad (5.1)$$

enquanto que a SNR pode ser determinada por:

$$\text{SNR}(I_1, I_2) = 10 \times \log_{10} \left(\sum_{x=1}^M \sum_{y=1}^N \left(\frac{I_1(x, y)^2}{(I_1(x, y) - I_2(x, y))^2} \right) \right). \quad (5.2)$$

Estas métricas permitem avaliar o erro introduzido pela redução de dados, de vírgula flutuante (32 bits) para vírgula fixa (22 bits), realizada durante o processo de filtragem de Gabor.

Assim, para avaliar o desempenho do sistema biométrico, foi criado um cenário no qual as 195 imagens que constituem a base de dados são inicialmente carregadas na memória DDR do sistema. Para cada uma das imagens foram criados os registos que as representam, armazenando-se num ficheiro de texto os *FingerCodes* que constituem os registos. Com os registos gerados, recorreu-se ao MATLAB para recriar os gráficos de FMR e FNMR em função do nível de decisão, encontrando-se estes representados na figura 5.7. Neste gráfico também se replicam os resultados apurados na figura 4.12, considerando-se os 10 bits de precisão na parte fracionária.

Comparando os resultados obtidos em MATLAB (efetuando a truncatura dos valores das imagens antes da aplicação dos filtros de Gabor) com os valores resultantes da execução no processador *MicroBlaze*, verifica-se que o EER e o FNMR (para FMR igual a 0,95%) são iguais. Pode-se então concluir que as diferenças verificadas, bem como o erro introduzido pelos blocos responsáveis pela conversão dos dados entre vírgula flutuante e vírgula fixa, não são suficientes para degradar significativamente o desempenho do sistema.

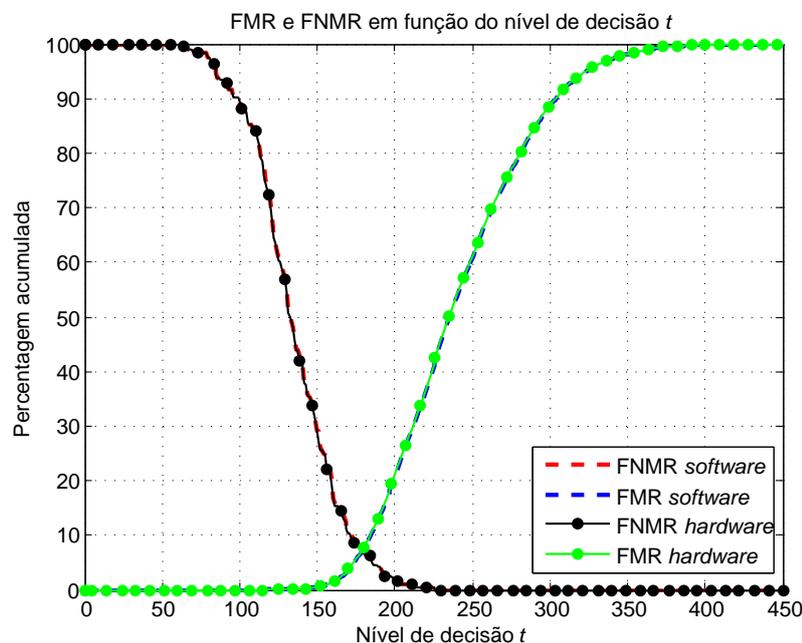


Figura 5.7: Comparação entre as curvas de FMR e FNMR em função do nível de decisão t , obtidas com e sem aceleração por *hardware*.

Na figura 5.8 encontram-se representados os valores MSE e SNR resultantes da comparação entre as imagens obtidas após a filtragem de Gabor com e sem aceleração por *hardware*. Os

resultados obtidos para uma amostra da base de dados, considerando as 8 orientações definidas, permitem concluir que para 0° e 90° , a utilização de apenas um filtro FIR dá origem a um SNR mais elevado e a um MSE mais baixo do que para as restantes orientações. Independentemente da orientação, os valores de SNR e MSE rondam, respetivamente, os 100 dB e 1×10^{-4} , justificando assim os resultados obtidos na figura 5.7.

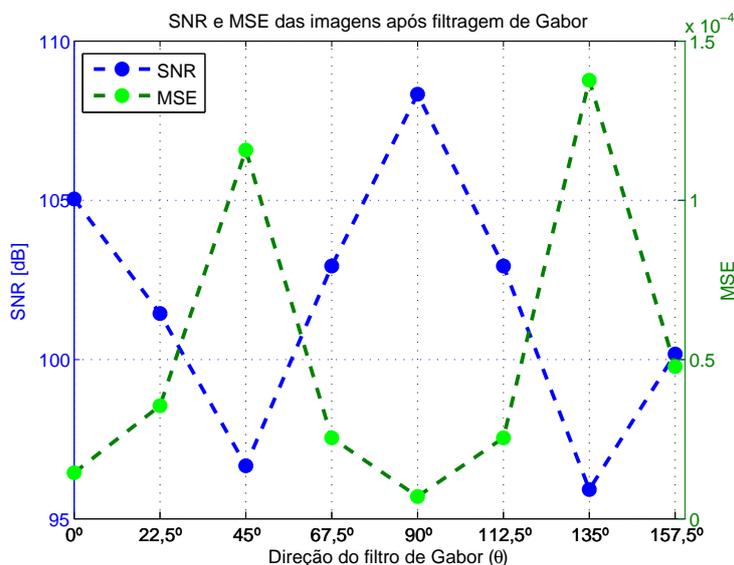


Figura 5.8: SNR e MSE das imagens após filtragem de Gabor.

Relativamente à capacidade de processamento do sistema desenvolvido, na tabela 5.6 apresentam-se os tempos de execução do algoritmo de extração de características com e sem aceleração por *hardware*. Nesta tabela também se pode verificar o tempo de execução da filtragem de Gabor.

	Tempos de execução		
	Sem aceleração por <i>hardware</i>	Com aceleração por <i>hardware</i>	Ganho
Extração de características	9758 ms	3584 ms	2,72
Filtragem de Gabor (por cada direção)	254 ms	0,347 ms	731,99

Tabela 5.6: Tempos de execução do algoritmo de extração de características com o *MicroBlaze* e o componente *filter_Gabor* a 100 MHz.

Considerando a mesma frequência de operação para o barramento AXI e para o processador *MicroBlaze*, 100 MHz, constata-se que o processo de extração de características é 2,72 vezes mais rápido quando se recorre à aceleração por *hardware*. Analisando apenas o componente *filter_Gabor*, verifica-se que a implementação em *hardware* permite resolver a filtragem de

Gabor num tempo 731,99 vezes inferior ao da implementação em *software*. Os resultados obtidos permitem ainda concluir que a filtragem de Gabor, que antes ocupava mais de 60% do tempo de processamento, passa a ter um peso computacional de apenas 2,3%, justificando-se assim o acelerador por *hardware* desenvolvimento.

Capítulo 6

Conclusões

No presente trabalho é proposta a implementação de um sistema de reconhecimento de impressões digitais baseado em FPGA. O sistema desenvolvido recorre a uma placa de desenvolvimento da Xilinx, a Virtex-7 FPGA VC707 Evaluation Kit, para implementar um algoritmo de extração de características baseado em textura orientada. Para tal, recorreu-se ao *softcore* *MicroBlaze* para implementar parte do algoritmo em linguagem C, e à linguagem de descrição de *hardware* VHDL para implementar um co-processador dedicado com vista à aceleração da realização da operação de filtragem de Gabor.

Após um estudo teórico do algoritmo de extração de características, foi apresentada a constituição do sistema biométrico (SB) implementado, com destaque para o sensor ótico que providencia as amostras de impressões digitais que constituem a base de dados utilizada. Por forma a otimizar o algoritmo de extração de características ao tipo de imagens presentes na base de dados, foi realizado um estudo individual aos parâmetros do algoritmo em MATLAB. No estudo são analisadas as métricas *Equal-Error Rate* (EER), *False Match Rate* (FMR) e *False Non-Match Rate* (FNMR), o que permitiu realizar um SB com EER de 7,18% e um FNMR de 22,05% quando FMR é 0,93%.

Na sequência do estudo realizado em MATLAB, procedeu-se à implementação e análise do sistema proposto usando linguagem C e o processador *MicroBlaze*. Recorrendo às ferramentas disponibilizadas pela Xilinx, foi criado um relatório que permitiu verificar quais os componentes do algoritmo com maior percentagem de utilização. Os resultados obtidos permitiram identificar a filtragem de Gabor como sendo um processo que ocupa cerca de 67% do tempo total de utilização do processador, e portanto candidato a ser acelerado recorrendo à utilização de *hardware* dedicado.

Para a implementação em *hardware* do periférico responsável pela filtragem de Gabor (*filter_Gabor*), foi alterada a dimensão original das janelas de 33×33 para 9×9 píxeis, uma vez que o erro introduzido não provocava uma variação significativa das métricas consideradas. Para cada uma das janelas, foi ainda realizada a separação por *Singular Value Decomposition* (SVD) em dois filtros a uma dimensão. Desta forma, foi possível realizar uma implementação em *hardware* menos complexa e recorrendo a filtros *Finite Impulse Response* (FIR). O resultado é obtido realizando numa primeira iteração a filtragem em linha, sobre o qual é então aplicada a filtragem em coluna. Antes de realizar a filtragem dos dados, estes são convertidos de vírgula flutuante (32 bit) para vírgula fixa (22 bit), repondo-se o formato inicial quando o processo termina. Esta solução permitiu uma poupança nos recursos utilizados no periférico, tanto em ocupação de memória como em número de blocos de *Digital Signal Processing* (DSP). O circuito obtido apresenta como frequência máxima de operação 265,9 MHz. Em termos de recursos ocupados pelo periférico *filter_Gabor*, destacam-se 18 blocos de DSP (DSP48E1), 22 memórias BRAM de 36 kb (RAMB36E1) e 477 *slices*. Já na arquitetura completa do sistema, o destaque vai não só para as 32 memórias RAMB36E1 necessárias para instanciar as duas memórias partilhadas entre o MicroBlaze e o *filter_Gabor*, mas também para as 4576 *slices* necessárias à implementação de todo o sistema.

Comparando o desempenho do sistema, entre as soluções com e sem aceleração por *hardware*, verificou-se que o EER se mantém em 7,18%. Ao calcular o SNR entre as imagens filtradas por *software* e *hardware*, obtiveram-se valores próximos de 100 dB. Para o mesmo cenário, o MSE é aproximadamente 1×10^{-4} . Este valores justificam o facto de o EER se manter, mesmo com o erro resultante da separação das janelas e da conversão dos dados entre vírgula flutuante e vírgula fixa. Verifica-se ainda que se obteve um ganho no tempo de processamento no algoritmo de 2,72 vezes e que a filtragem de Gabor, individualmente, possui um ganho de 731,99 vezes. Estes resultados levaram a uma redução do peso computacional da filtragem de Gabor para aproximadamente 2,3%, indicando que um aumento da frequência de operação do periférico *filter_Gabor* não traria um ganho significativo ao desempenho do sistema.

6.1 Trabalho futuro

Como trabalho futuro seria interessante adicionar ao sistema desenvolvido, o sensor adotado no dimensionamento dos parâmetros que constituem o algoritmo de extração de características.

Ao existir um módulo que realize a comunicação entre o sensor e o processador, daria a possibilidade de criar um sistema biométrico completo.

Também seria interessante o estudo de um método de filtragem que misture o conceito de filtragem por janela deslizante com a separação da janela em filtros a uma dimensão. Na figura 6.1 apresenta-se o exemplo de uma arquitetura baseada em janela deslizante com dimensão 5×5 píxeis. Esta técnica recorre a um conjunto de *buffers* de linha para armazenar as linhas da imagem a filtrar, removendo a necessidade de voltar a ler píxeis anteriores. Na figura 6.2 apresenta-se o diagrama de blocos de uma solução possível para a decomposição da janela de 5×5 píxeis em dois filtros FIR. Tal como foi estudado neste trabalho, para realizar a filtragem de Gabor com direções diferentes de 0° e 90° , é necessário utilizar dois blocos de filtragem.

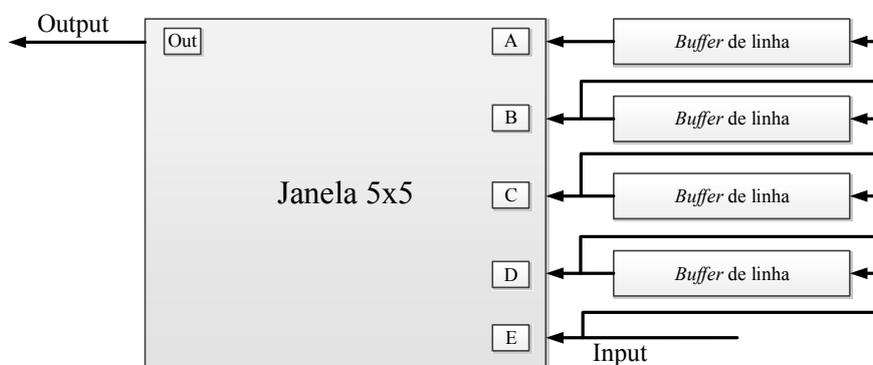


Figura 6.1: Diagrama de blocos da arquitetura proposta com recurso a janela deslizante.

Esta abordagem permitiria substituir as duas memórias utilizadas para guardar os resultados intermédios, por um conjunto de *buffers*. Enquanto que as duas memórias têm uma dimensão igual à da imagem a filtrar, cada um dos *buffers* teria como dimensão a resolução da imagem na horizontal. Assim, ao realizar-se a filtragem em apenas uma iteração, seria possível reduzir o tempo de processamento para metade, à custa de um maior número de blocos DSP. No entanto, devido utilização de *buffers*, esta abordagem permitiria escalar mais facilmente a arquitetura para realizar a filtragem de imagens com maior resolução vertical.

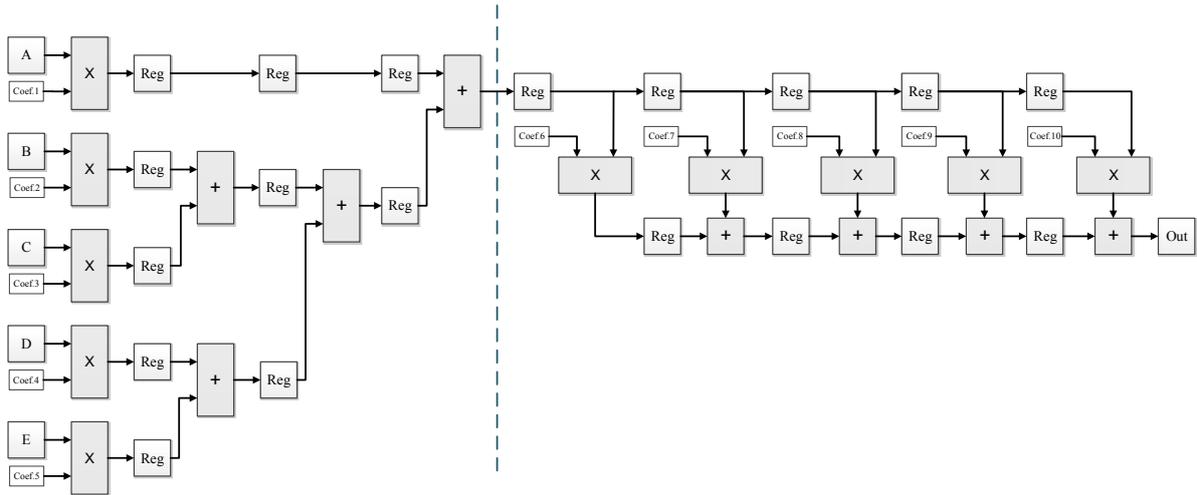


Figura 6.2: Diagrama de blocos da arquitetura proposta.

Referências

- [1] ADH Technology Co. Ltd. Fingerprint Scanner GT-511C3, datasheet v1.1. 26 de Março, 2013, ADH Technology Co. Ltd.
- [2] Alilla, A., Faccio, M., Vali, T., Marotta, G., e DeSantis, L. (2013). A new low cost fingerprint recognition system on FPGA. Em *2013 IEEE International Conference on Industrial Technology (ICIT)*, páginas 988–993. IEEE.
- [3] Arcelli, C. e Baja, G. (1985). A Width-Independent Fast Thinning Algorithm. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):463–474.
- [4] Areekul, V., Watchareeruetai, U., Suppasriwasuseth, K., e Tantaratana, S. (2005). Separable Gabor filter realization for fast fingerprint enhancement. Em *IEEE International Conference on Image Processing 2005*, volume 3, páginas III–253. IEEE.
- [5] Areekul, V., Watchareeruetai, U., e Tantaratana, S. (2004). Fast Separable Gabor Filter for Fingerprint Enhancement. Em *Biometric Authentication*, páginas 403–409. Springer Berlin Heidelberg.
- [6] Ashbourn, J. (2011). Guide to Biometrics for Large-Scale Systems: Technological, Operational, and User-Related Factors. Springer.
- [7] Baker, K. (2005). Singular value decomposition tutorial. Technical report, Ohio State.
- [8] Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. Em *Pattern Recognition*, 13(2) páginas 111–122.
- [9] Barrenechea, M., Altuna, J., Mendicute, M., e Ser, J. (2007). A Low-Cost FPGA-based Embedded Fingerprint Verification and Matching System. Em *2007 Fifth Workshop on Intelligent Solutions in Embedded Systems*, páginas 250–261. IEEE.

- [10] C. Chen e J. Dai (2005). An embedded fingerprint authentication system with reduced hardware resources requirement. Em *Proceedings of the Ninth International Symposium on Consumer Electronics, 2005. (ISCE 2005)*., páginas 145–150. IEEE.
- [11] Carter, W., Duong, K., Freeman, R., Hsieh, H., Ja, J., Mahoney, J., Ngo, L., e Sze, S. (1986). A user programmable reconfigurable logic array. Em *Proceedings of the Custom Integrated Circuits Conference*, páginas 233–235.
- [12] Chao, G., Lee, S., Lai, H., e Horng, S. (2005). Embedded Fingerprint Verification System. Em *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, volume 2, páginas 52–57. IEEE.
- [13] Cooley, J. e Tukey, J. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90) páginas 297–297.
- [14] Danese, G., Giachero, M., Leporati, F., Matrone, G., e Nazzicari, N. (2009). An FPGA-Based Embedded System for Fingerprint Matching Using Phase-Only Correlation Algorithm. Em *2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*, páginas 672–679.
- [15] Fons, F., Fons, M., Canto, E., e Lopez, M. (2007). Flexible hardware for fingerprint Image Processing. Em *2007 Ph.D Research in Microelectronics and Electronics Conference*, páginas 169–172. IEEE.
- [16] Fons, M., Fons, F., e Canto, E. (2006a). Design of an Embedded Fingerprint Matcher System. Em *2006 IEEE International Symposium on Consumer Electronics*, páginas 1–6. IEEE.
- [17] Fons, M., Fons, F., e Canto, E. (2006b). Design of FPGA-based Hardware Accelerators for On-line Fingerprint Matcher Systems. Em *2006 Ph.D. Research in Microelectronics and Electronics*, páginas 333–336. IEEE.
- [18] Fons, M., Fons, F., Canto, E., e Cantó, E. (2010). Fingerprint Image Processing Acceleration Through Run-Time Reconfigurable Hardware. Em *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(12) páginas 991–995.

- [19] Fons, M. e López, M. (2006). Hardware-Software Co-design of a Fingerprint Image Enhancement Algorithm. Em *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, páginas 3496–3501. IEEE.
- [20] Fu, Q., Wu, A., e Li, Y. (2011). Fingerprint Identification System Based on SOPC. Em *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, páginas 1–4. IEEE.
- [21] Gabor, D. (1947). Theory of communication. Em *Journal of the Institution of Electrical Engineers - Part I: General*, 94(73) páginas 58–58.
- [22] Gupta, P., Ravi, S., Raghunathan, A., e Jha, N. (2005). Efficient fingerprint-based user authentication for embedded systems. Em *Proceedings. 42nd Design Automation Conference, 2005.*, páginas 244–247. IEEE.
- [23] Hauck, S. e DeHon, A. (2007). Reconfigurable Computing: The Theory and Practice of FPGA-based Computation.
- [24] Henry, E. (1900). Classification and Uses of Fingerprints. George Routledge & Sons Ltd.
- [25] Jain, A. (1998). Fingerprint image enhancement: algorithm and performance evaluation. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8) páginas 777–789.
- [26] Jain, A., Bolle, R., e Pankanti, S. (2005). Biometrics: Personal Identification in Networked Society. Springer.
- [27] Jain, A., Ross, A., e Nandakumar, K. (2011). Introduction to Biometrics. Springer.
- [28] Lindoso, A., Entrena, L., e Izquierdo, J. (2007). FPGA-Based Acceleration of Fingerprint Minutiae Matching. Em *2007 3rd Southern Conference on Programmable Logic*, páginas 81–86. IEEE.
- [29] Lindoso, A., Entrena, L., Lopez-Ongil, C., e Liu, J. (2005). Correlation-based fingerprint matching using FPGAs. Em *2005 IEEE International Conference on Field-Programmable Technology, 2005.*, páginas 87–94. IEEE.

- [30] Lopez, M., Canto, E., e Balaguer, V. (2008). FPGA implementation of a minutiae extraction fingerprint algorithm. Em *2008 IEEE International Symposium on Industrial Electronics*, páginas 1920–1925. IEEE.
- [31] Maio, D. e Maltoni, D. (2004). FVC2004: Third Fingerprint Verification Competition. Em *Proceedings of the First International Conference on Biometric Authentication*, páginas 1–7.
- [32] Maio, D., Maltoni, D., Cappelli, R., Wayman, J., e Jain, A. (2002a). FVC2000: fingerprint verification competition. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3) páginas 402–412.
- [33] Maio, D., Maltoni, D., Cappelli, R., Wayman, J., e Jain, A. (2002b). FVC2002: Second Fingerprint Verification Competition. Em *Object recognition supported by user interaction for service robots*, volume 3, páginas 811–814. IEEE Comput. Soc.
- [34] Maltoni, D., Maio, D., Jain, A., e Prabhakar, S. (2009). *Handbook of Fingerprint Recognition*. Springer, London.
- [35] Militello, C., Conti, V., Sorbello, F., e Vitabile, S. (2008). A Novel Embedded Fingerprints Authentication System Based on Singularity Points. Em *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, páginas 72–78. IEEE.
- [36] Neji, N., Boudabous, A., Kharrat, W., e Masmoudi, N. (2011). Architecture and FPGA implementation of the CORDIC algorithm for fingerprints recognition systems. Em *Eighth International Multi-Conference on Systems, Signals & Devices*, páginas 1–5. IEEE.
- [37] NIST. National Institute of Standards and Technology. <http://www.nist.gov>. último acesso a: 12/04/2014.
- [38] NIST. NIST Biometric Image Software (NBIS), Release 4.2.0, National Institute of Standards and Technology. <http://www.nist.gov/itl/iad/ig/nbis.cfm>. último acesso a: 22/09/2014.
- [39] Oppenheim, A. (1975). *Digital Signal Processing*. Prentice Hall.
- [40] Pacheco, C. (2003). Autenticação com Impressão Digital, Relatório de projeto final de curso. Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores, Instituto Superior de Engenharia de Lisboa.

- [41] Phillips, P., Martin, A., Wilson, C., e Przybocki, M. (2000). An introduction evaluating biometric systems. Em *Computer*, 33(2) páginas 56–63.
- [42] Prabhakar, S. (2001). Fingerprint classification and matching using a filterbank. Tese de doutoramento, University Michigan State.
- [43] Proença, H. (2006). Non-Cooperative Biometric Iris Recognition. Tese de doutoramento, Universidade da Beira Interior.
- [44] Ratha, N., Jain, A., e Rover, D. (1995). An FPGA-based point pattern matching processor with application to fingerprint matching. Em *Proceedings of Conference on Computer Architectures for Machine Perception*, páginas 394–401. IEEE Comput. Soc. Press.
- [45] Ratha, N., Karu, K., e Jain, A. (1996). A real-time matching system for large fingerprint databases. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8) páginas 799–813.
- [46] Razak, A. e Taharim, R. (2009). Implementing Gabor Filter for fingerprint recognition using Verilog HDL. Em *2009 5th International Colloquium on Signal Processing & Its Applications*, páginas 423–427. IEEE.
- [47] Reis, C. (2003). Autenticação com Impressão Digital, Relatório de projeto final de curso. Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores, Instituto Superior de Engenharia de Lisboa.
- [48] Stockman, G., Kopstein, S., e Benett, S. (1982). Matching Images to Models for Registration and Object Detection via Clustering. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(3) páginas 229–241.
- [49] Strang, G. e Borre, K. (1997). Linear algebra, geodesy, and GPS. Wellesley-Cambridge Press.
- [50] The MathWorks, inc. MATrix LABoratory (MATLAB). <http://www.mathworks.com/>. The MathWorks, Inc., Natick, Estados Unidos.
- [51] Trauring, M. (1963). Automatic Comparison of Finger-Ridge Patterns. Em *Nature*, 197(4871) páginas 938–940.

- [52] Universidade de Bolonha. FVC-onGoing - on-line evaluation of fingerprint recognition algorithms. <https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx>. último acesso a: 26/08/2014.
- [53] Universidade de Bolonha. FVC2006 - Fourth International Fingerprint Verification Competition. <http://bias.csr.unibo.it/fvc2006/>. último acesso a: 12/11/2014.
- [54] Volder, J. (1959). The CORDIC Trigonometric Computing Technique. Em *IEEE Transactions on Electronic Computers*, EC-8(3) páginas 330–334.
- [55] Wayman, J. (2001). Fundamentals of biometric authentication technologies. Em *International Journal of Image and Graphics*, 01(01) páginas 93–113.
- [56] Wegstein, J. (1970). Automated fingerprint identification, NBS technical note 538. Em *National Bureau of Standards*, página 31.
- [57] Wilson, C., Candela, G., e Watson, C. (1994). Neural network fingerprint classification. Em *Journal of Artificial Neural Networks*, 1(2) páginas 203–228.
- [58] Xilinx, Inc. 7 Series DSP48E1 Slice - User Guide, UG479 (v1.7). 10 de Maio, 2014, Xilinx, Inc., São José, Estados Unidos.
- [59] Xilinx, Inc. 7 Series FPGAs Configurable Logic Block - User Guide, UG474 (v1.6). 11 de Agosto, 2014, Xilinx, Inc., São José, Estados Unidos.
- [60] Xilinx, Inc. 7 Series FPGAs Overview - Datasheet, DS180 (v1.15). 18 de Fevereiro, 2014, Xilinx, Inc., São José, Estados Unidos.
- [61] Xilinx, Inc. 7 Series Memory Resources - User Guide, UG473 (v1.10.1). 9 de Maio, 2014, Xilinx, Inc., São José, Estados Unidos.
- [62] Xilinx, Inc. Applications. <http://www.xilinx.com/applications/index.htm>. último acesso a: 12/11/2014.
- [63] Xilinx, Inc. Integrated Software Environment (ISE) Design Suite. <http://www.xilinx.com/products/design-tools/ise-design-suite.html>. Xilinx, Inc., Natick, São José, Estados Unidos.

- [64] Xilinx, Inc. IP LogiCORE FIR Compiler v5.0 - Datasheet, DS534. 1 de Março, 2011, Xilinx, Inc., São José, Estados Unidos.
- [65] Xilinx, Inc. LogiCORE IP Floating-Point Operator v5.0 - Datasheet, DS335. 1 de Março, 2011, Xilinx, Inc., São José, Estados Unidos.
- [66] Xilinx, Inc. MicroBlaze Processor Reference Guide - Embedded Development Kit EDK 14.7, UG081 (v14.7). 2 de Outubro, 2013, Xilinx, Inc., São José, Estados Unidos.
- [67] Xilinx, Inc. VC707 Evaluation Board for the Virtex-7 FPGA - User Guide, UG885 (v1.3). 22 de Agosto, 2013, Xilinx, Inc., São José, Estados Unidos.
- [68] Zhou, R., Sin, S., Li, D., Isshiki, T., e Kunieda, H. (2010). Ultra fast fingerprint indexing for embedded system. Em *2010 International Conference of Soft Computing and Pattern Recognition*, páginas 87–92. IEEE.

Anexo A

Constituição da base de dados

Apresenta-se tabela A.1 a constituição da base de dados, sendo esta composta por 195 amostras. As amostras encontram-se agrupadas em conjuntos de 3, significando que têm origem no mesmo dedo, e são caracterizadas consoante a classe e a deteção correta do ponto de referência. Note-se que as percentagens obtidas, para cada uma das classes, estão de acordo com a distribuição na população mundial [57].

Tabela A.1: Constituição da base de dados.

Amostras	<i>Left loop</i>	<i>Right loop</i>	<i>Whorl</i>	<i>Arch</i>	<i>Tended arch</i>	Ponto de referência
1 a 3	✓					✓
4 a 6	✓					✓
7 a 9		✓				✓
10 a 12		✓				✓
13 a 15		✓				✓
16 a 18		✓				✓
19 a 21	✓					✓
22 a 24			✓			✓
25 a 27			✓			✓
28 a 30			✓			✓
31 a 33		✓				✓
34 a 36	✓					✓

Tabela A.1: Constituição da base de dados (continuação).

Amostras	<i>Left loop</i>	<i>Right loop</i>	<i>Whorl</i>	<i>Arch</i>	<i>Tended arch</i>	Ponto de referência
37 a 39		✓				✓
40 a 42			✓			✓
43 a 45			✓			✓
46 a 48			✓			✓
49 a 51	✓					✓
52 a 54	✓					✓
55 a 57			✓			✓
58 a 60			✓			✓
61 a 63	✓					✓
64 a 66		✓				✓
67 a 69	✓					✓
70 a 72					✓	✓
73 a 75	✓					✓
76 a 78	✓					✓
79 a 81		✓				✓
82 a 84		✓				✓
85 a 87		✓				✓
88 a 90			✓			✓
91 a 93			✓			✓
94 a 96			✓			✓
97 a 99		✓				✓
100 a 102		✓				✓
103 a 105		✓				✓
106 a 108			✓			✓
109 a 111			✓			✓
112 a 114	✓					✓

Tabela A.1: Constituição da base de dados (continuação).

Amostras	<i>Left loop</i>	<i>Right loop</i>	<i>Whorl</i>	<i>Arch</i>	<i>Tended arch</i>	Ponto de referência
115 a 117	✓					✓
118 a 120			✓			✓
121 a 123			✓			✓ ¹
124 a 126			✓			✓
127 a 129	✓					✓
130 a 132	✓					✓
133 a 135	✓					✓
136 a 138	✓					✓
139 a 141		✓				✓
142 a 144				✓		✓
145 a 147				✓		✓
148 a 150		✓				✓
151 a 153	✓					✓
154 a 156	✓					✓
157 a 159	✓					✓
160 a 162		✓				✓
163 a 165		✓				✓
166 a 168		✓				✓
169 a 171	✓					✓
172 a 174			✓			✓
175 a 177	✓					✓
178 a 180			✓			✓
181 a 183	✓					✓
184 a 186	✓					✓
187 a 189		✓				✓
190 a 192	✓					✓

¹Desvio de aproximadamente 3 píxeis numa das amostras, que não influencia o desempenho do sistema.

Tabela A.1: Constituição da base de dados (continuação).

Amostras	<i>Left loop</i>	<i>Right loop</i>	<i>Whorl</i>	<i>Arch</i>	<i>Tended arch</i>	Ponto de referência
193 a 195		✓				✓
Total	72	60	54	6	3	-
Percentagem	36,92%	30,77%	27,69%	3,08%	1,54%	-
P. Mundial	33,8%	31,7%	27,9%	3,7%	2,9%	-

Anexo B

Decomposição SVD dos filtros de Gabor

A Singular Value Decomposition (SVD) [7][49], apresentada no capítulo 4, segue a equação

$$A_{(P \times Q)} = U_{(P \times P)} S_{(P \times Q)} V_{(Q \times Q)}^T. \quad (\text{B.1})$$

Considerando que a matriz A representa a janela de um filtro bi-dimensional de Gabor com dimensões 9×9 , pode-se decompor a mesma nas respostas impulsionais:

$$h_{1(n_s \times Q)} = \left(V_{(Q \times n_s)} \times \sqrt{\text{diag}(S_{(n_s \times n_s)})} \right)^T, \quad (\text{B.2})$$

e

$$h_{2(P \times n_s)} = U_{(P \times n_s)} \times \sqrt{\text{diag}(S_{(n_s \times n_s)})}, \quad (\text{B.3})$$

onde são apenas considerados os valores mais significativos da matriz S segundo a condição $\text{diag}(S(> 1e^{-2}))$. Esta condição é suficiente para se obter a separabilidade dos filtros de Gabor através das respostas impulsionais h_1 e h_2 .

Para o banco de filtros de Gabor considerado neste trabalho, com as orientações $\{0^\circ; 90^\circ\}$ obtém-se $n_s = 1$, enquanto que para as orientações $\{22,5^\circ; 45^\circ; 67,5^\circ; 112,5^\circ; 135^\circ; 157,5^\circ\}$ obtém-se $n_s = 2$. Segue-se um exemplo para cada uma destas situações.

Janela com orientação 0°

Considerando a janela de 9×9 para a orientação de 0° :

$$A_{0^\circ} = \begin{bmatrix} -0,0177 & -0,0426 & -0,0795 & -0,1157 & -0,1311 & -0,1157 & -0,0795 & -0,0426 & -0,0177 \\ -0,0363 & -0,0872 & -0,1629 & -0,2370 & -0,2685 & -0,2370 & -0,1629 & -0,0872 & -0,0363 \\ -0,0103 & -0,0247 & -0,0461 & -0,0671 & -0,0760 & -0,0671 & -0,0461 & -0,0247 & -0,0103 \\ 0,0790 & 0,1895 & 0,3540 & 0,5150 & 0,5836 & 0,5150 & 0,3540 & 0,1895 & 0,0790 \\ 0,1353 & 0,3247 & 0,6065 & 0,8825 & 1,0000 & 0,8825 & 0,6065 & 0,3247 & 0,1353 \\ 0,0790 & 0,1895 & 0,3540 & 0,5150 & 0,5836 & 0,5150 & 0,3540 & 0,1895 & 0,0790 \\ -0,0103 & -0,0247 & -0,0461 & -0,0671 & -0,0760 & -0,0671 & -0,0461 & -0,0247 & -0,0103 \\ -0,0363 & -0,0872 & -0,1629 & -0,2370 & -0,2685 & -0,2370 & -0,1629 & -0,0872 & -0,0363 \\ -0,0177 & -0,0426 & -0,0795 & -0,1157 & -0,1311 & -0,1157 & -0,0795 & -0,0426 & -0,0177 \end{bmatrix}_{(9 \times 9)},$$

obtem-se a matriz

$$S_{0^\circ} = \begin{bmatrix} 2,5741 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{(9 \times 9)},$$

dando origem às seguintes respostas impulsiais:

$$h_{1(1 \times 9)} = \begin{pmatrix} \begin{bmatrix} 0,0719 \\ 0,1725 \\ 0,3223 \\ 0,4690 \\ 0,5314 \\ 0,4690 \\ 0,3223 \\ 0,1725 \\ 0,0719 \end{bmatrix}_{(9 \times 1)} \times \sqrt{2,5741} \end{pmatrix}^T = \begin{bmatrix} 0,1154 \\ 0,2768 \\ 0,5171 \\ 0,7524 \\ 0,8526 \\ 0,7524 \\ 0,5171 \\ 0,2768 \\ 0,1154 \end{bmatrix}_{(9 \times 1)}^T,$$

e

$$h_{2(9 \times 1)} = \begin{bmatrix} -0,0958 \\ -0,1963 \\ -0,0556 \\ 0,4266 \\ 0,7310 \\ 0,4266 \\ -0,0556 \\ -0,1963 \\ -0,0958 \end{bmatrix}_{(9 \times 1)} \times \sqrt{2,5741} = \begin{bmatrix} -0,1537 \\ -0,3149 \\ -0,0892 \\ 0,6845 \\ 1,1728 \\ 0,6845 \\ -0,0892 \\ -0,3149 \\ -0,1537 \end{bmatrix}_{(9 \times 1)}.$$

Janela com orientação 22,5°

Considerando agora a janela de 9×9 para a orientação de 22,5°:

$$A_{22,5^\circ} = \begin{bmatrix} -0,0050 & -0,0250 & -0,0657 & -0,1135 & -0,1353 & -0,1129 & -0,0650 & -0,0244 & -0,0048 \\ -0,0384 & -0,1036 & -0,1949 & -0,2560 & -0,2284 & -0,1261 & -0,0257 & 0,0203 & 0,0218 \\ -0,0790 & -0,1625 & -0,2214 & -0,1690 & 0,0021 & 0,1725 & 0,2234 & 0,1632 & 0,0791 \\ -0,0584 & -0,0532 & 0,0736 & 0,3475 & 0,6251 & 0,6982 & 0,5304 & 0,2813 & 0,1040 \\ 0,0364 & 0,1825 & 0,4831 & 0,8364 & 1,0000 & 0,8364 & 0,4831 & 0,1825 & 0,0364 \\ 0,1040 & 0,2813 & 0,5304 & 0,6982 & 0,6251 & 0,3475 & 0,0736 & -0,0532 & -0,0584 \\ 0,0791 & 0,1632 & 0,2234 & 0,1725 & 0,0021 & -0,1690 & -0,2214 & -0,1625 & -0,0790 \\ 0,0218 & 0,0203 & -0,0257 & -0,1261 & -0,2284 & -0,2560 & -0,1949 & -0,1036 & -0,0384 \\ -0,0048 & -0,0244 & -0,0650 & -0,1129 & -0,1353 & -0,1135 & -0,0657 & -0,0250 & -0,0050 \end{bmatrix} \quad (9 \times 9)$$

obtem-se a matriz

$$S_{22,5^\circ} = \begin{bmatrix} 2,3755 & 0 & 0 & \dots & 0 \\ 0 & 0,9899 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (9 \times 9)$$

dando origem às seguintes respostas impulsiais:

$$h_{1(2 \times 9)} = \begin{pmatrix} \begin{bmatrix} 0,0213 & -0,1675 \\ 0,1065 & -0,3450 \\ 0,2820 & -0,4712 \\ 0,4882 & -0,3617 \\ 0,5837 & 0 \\ 0,4882 & 0,3617 \\ 0,2820 & 0,4712 \\ 0,1065 & 0,3450 \\ 0,0213 & 0,1675 \end{bmatrix}_{(9 \times 2)} \times \sqrt{\begin{bmatrix} 2,3755 & 0 \\ 0 & 0,9899 \end{bmatrix}_{(2 \times 2)}} \begin{bmatrix} 0,0328 & -0,1667 \\ 0,1642 & -0,3433 \\ 0,4346 & -0,4688 \\ 0,7524 & -0,3599 \\ 0,8996 & 0,0000 \\ 0,7524 & 0,3599 \\ 0,4346 & 0,4688 \\ 0,1642 & 0,3433 \\ 0,0328 & 0,1667 \end{bmatrix}_{(9 \times 2)}^T \quad ,$$

e

$$h_{2(9 \times 2)} = \begin{bmatrix} -0,0976 & 0,0007 \\ -0,1647 & 0,1814 \\ 0,0015 & 0,4768 \\ 0,4508 & 0,4897 \\ 0,7212 & 0 \\ 0,4508 & -0,4897 \\ 0,0015 & -0,4768 \\ -0,1647 & -0,1814 \\ -0,0976 & -0,0007 \end{bmatrix}_{(9 \times 2)} \times \sqrt{\begin{bmatrix} 2,3755 & 0 \\ 0 & 0,9899 \end{bmatrix}_{(2 \times 2)}} = \begin{bmatrix} -0,1504 & 0,0007 \\ -0,2539 & 0,1805 \\ 0,0023 & 0,4744 \\ 0,6949 & 0,4872 \\ 1,1116 & 0 \\ 0,6949 & -0,4872 \\ 0,0023 & -0,4744 \\ -0,2539 & -0,1805 \\ -0,1504 & -0,0007 \end{bmatrix}_{(9 \times 2)} \quad .$$

Ao realizar a decomposição SVD, verifica-se que as respostas impulsiais resultantes para as orientações $\{0^\circ; 22,5^\circ; 45^\circ\}$ podem ser utilizadas para obter as respostas impulsiais das restantes orientações. Assim, na definição dos coeficientes dos filtros FIR, são consideradas apenas as seguintes respostas impulsiais:

$$\begin{aligned}h_1 &= [0,1154 \quad 0,2768 \quad 0,5171 \quad 0,7524 \quad 0,8526 \quad 0,7524 \quad 0,5171 \quad 0,2768 \quad 0,1154], \\h_2 &= [-0,1537 \quad -0,3149 \quad -0,0892 \quad 0,6845 \quad 1,1728 \quad 0,6845 \quad -0,0892 \quad -0,3149 \quad -0,1537], \\h_3 &= [0,0328 \quad 0,1642 \quad 0,4346 \quad 0,7524 \quad 0,8996 \quad 0,7524 \quad 0,4346 \quad 0,1642 \quad 0,0328], \\h_4 &= [-0,1667 \quad -0,3433 \quad -0,4688 \quad -0,3599 \quad 0 \quad 0,3599 \quad 0,4688 \quad 0,3433 \quad 0,1667], \\h_5 &= [-0,1504 \quad -0,2539 \quad 0,0023 \quad 0,6949 \quad 1,1116 \quad 0,6949 \quad 0,0023 \quad -0,2539 \quad -0,1504], \\h_6 &= [0,0007 \quad 0,1805 \quad 0,4744 \quad 0,4872 \quad 0 \quad -0,4872 \quad -0,4744 \quad -0,1805 \quad -0,0007], \\h_7 &= [-0,0997 \quad -0,0736 \quad 0,2200 \quad 0,7285 \quad 1 \quad 0,7285 \quad 0,2200 \quad -0,0736 \quad -0,0997], \\h_8 &= [-0,0915 \quad -0,3162 \quad -0,5652 \quad -0,4981 \quad 0 \quad 0,4981 \quad 0,5652 \quad 0,3162 \quad 0,0915].\end{aligned}$$

Anexo C

Análise de sensibilidade aos parâmetros do algoritmo de reconhecimento de impressões digitais

Esta análise tem como objetivo avaliar, através das métricas EER e FNMR, a sensibilidade do algoritmo de reconhecimento de impressões digitais, à variação dos seus parâmetros. A análise aos resultados obtidos dá origem aos valores apresentados na tabela 4.3. Pelos resultados presentes nas figuras C.1, C.2, C.3 e C.4, são definidos os valores dos parâmetros b , k e B . Nas figuras C.5, C.6 e C.7, é definido o parâmetro gap em função do valor do parâmetro b . Enquanto que nas figuras C.8 e C.9 são determinados, respetivamente, os valores dos parâmetros ang e f .

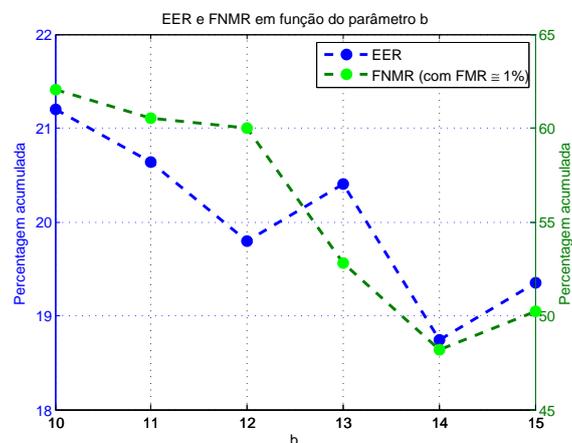


Figura C.1: EER e FNMR em função do parâmetro b com: $B = 3$, $k = 16$, $gap = 1, 5$, $ang = 7^\circ$ e $f = 0, 1$.

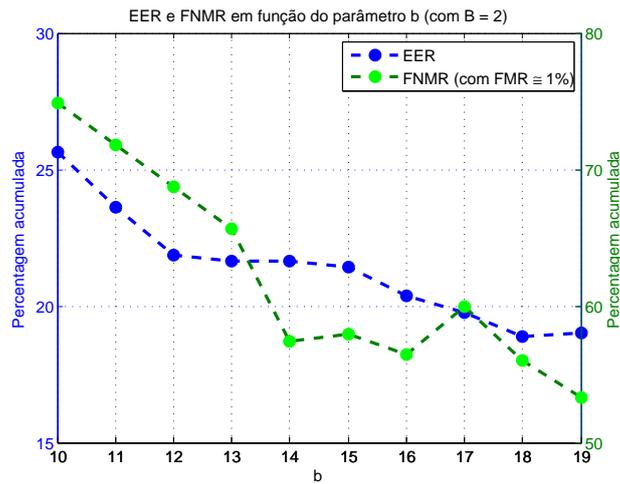


Figura C.2: EER e FNMR em função do parâmetro b com: $B = 2$, $k = 16$, $gap = 1,5$, $ang = 7^\circ$ e $f = 0,1$.

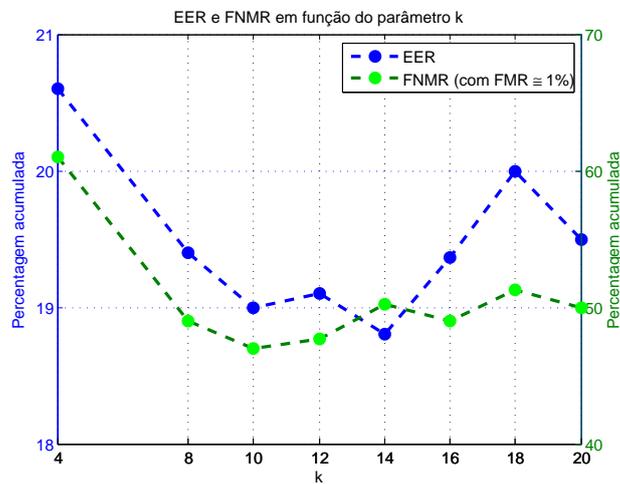


Figura C.3: EER e FNMR em função do parâmetro k com: $B = 3$, $b = 15$, $gap = 1,5$, $ang = 7^\circ$ e $f = 0,1$.

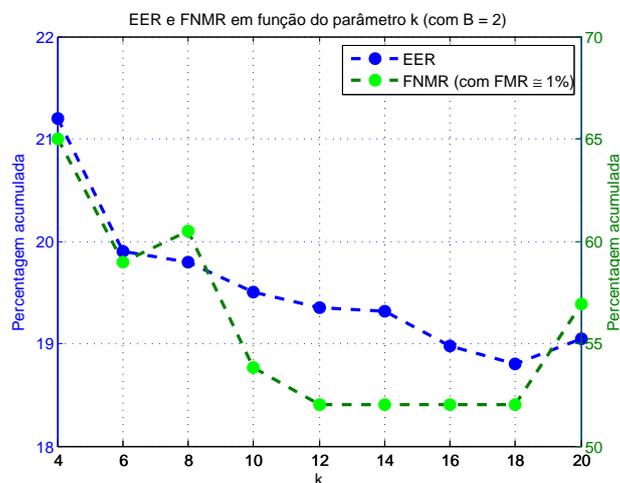


Figura C.4: EER e FNMR em função do parâmetro k com: $B = 2$, $b = 15$, $gap = 1,5$, $ang = 7^\circ$ e $f = 0,1$.

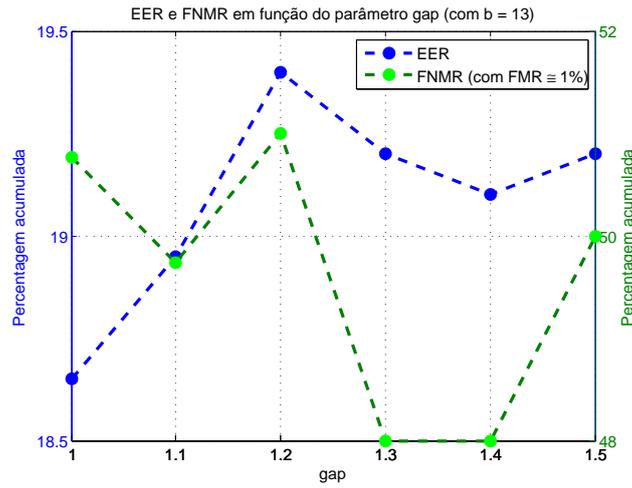


Figura C.5: EER e FNMR em função do parâmetro gap com: $B = 3$, $b = 13$, $k = 12$, $ang = 7^\circ$ e $f = 0, 1$.

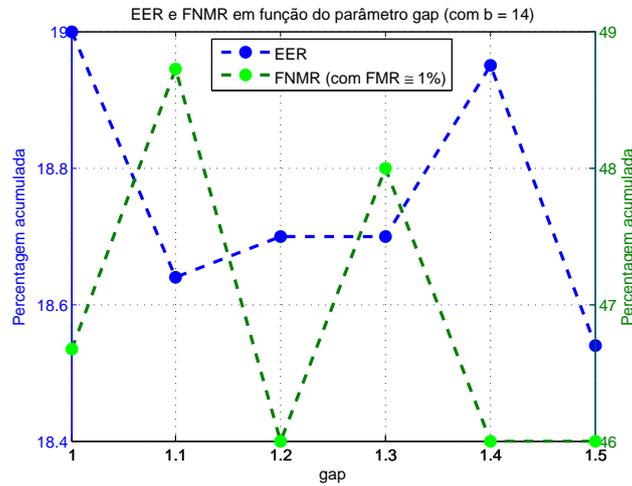


Figura C.6: EER e FNMR em função do parâmetro gap com: $B = 3$, $b = 14$, $k = 12$, $ang = 7^\circ$ e $f = 0, 1$.

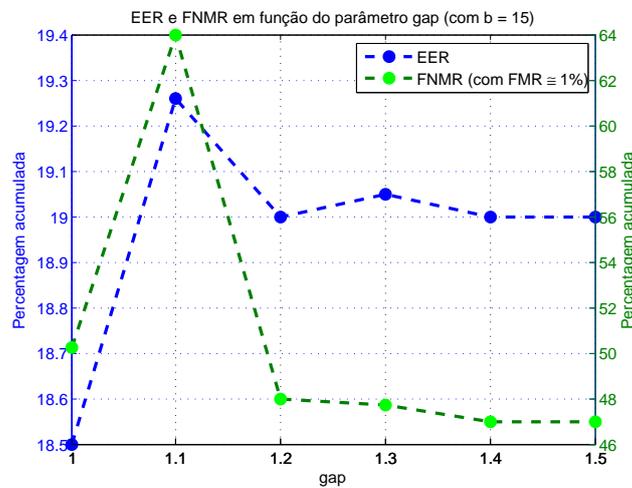


Figura C.7: EER e FNMR em função do parâmetro gap com: $B = 3$, $b = 15$, $k = 12$, $ang = 7^\circ$ e $f = 0, 1$.

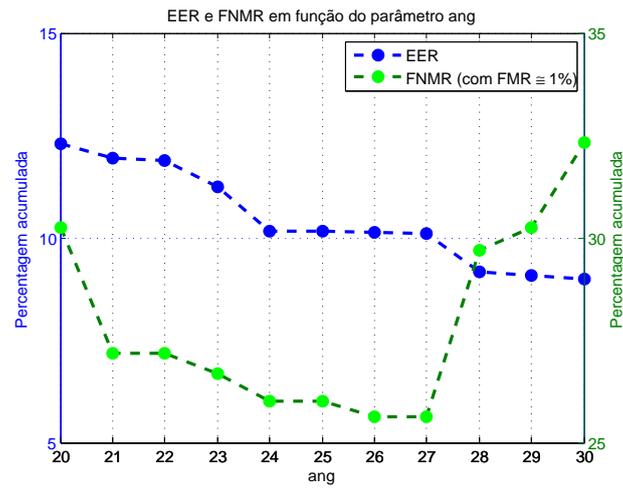


Figura C.8: EER e FNMR em função do parâmetro ang com: $B = 3$, $b = 14$, $k = 11$, $gap = 1,5$ e $f = 0,1$.

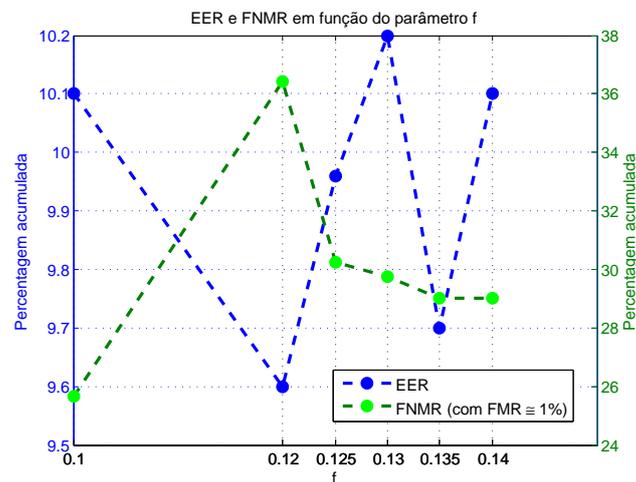


Figura C.9: EER e FNMR em função do parâmetro f com: $B = 3$, $b = 14$, $k = 11$, $gap = 1,5$ e $ang = 27^\circ$.